UNIVERSITY OF SOUTHAMPTON

FACULTY OF PHYSICAL SCIENCES AND ENGINEERING

SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Quantum-Assisted Multi-Objective Optimization of Heterogeneous Networks

by

Dimitrios Alanis

MEng, MSc

A doctoral thesis submitted in partial fulfillment of the requirements for the award of Doctor of Philosophy at the University of Southampton

January 2017

SUPERVISORS:

Prof. Lajos Hanzo

FREng, FIEEE, FIEE, DSc, RS Wolfson Fellow

Chair of Southampton Wirelss Group

and

Dr. Soon Xin Ng (Michael)

PhD, BEng, CEng, MIET, SMIEEE, FHEA

Associate Professor

School of Electronics and Computer Science

University of Southampton

Southampton SO17 1BJ

United Kingdom

Dedicated to my parents, George and Vasiliki,
and to my sister, Kelly.

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL SCIENCES AND ENGINEERING
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

Quantum-Assisted Multi-Objective Optimization of Heterogeneous Networks

by Dimitrios Alanis

Some of the Heterogeneous Network (HetNet) components may act autonomously for the sake of achieving the best possible performance. The attainable routing performance depends on a delicate balance of diverse and often conflicting Quality-of-Service (QoS) requirements. Finding the optimal solution typically becomes an NP-hard problem, as the network size increases in terms of the number of nodes. Moreover, the employment of user-defined utility functions for the aggregation of the different objective functions often leads to suboptimal solutions. On the other hand, Pareto Optimality is capable of amalgamating the different design objectives by relying on an element of elitism.

Although there is a plethora of bio-inspired algorithms that attempt to address the associated multi-component optimization problem, they often fail to generate all the routes constituting the Optimal Pareto Front (OPF). As a remedy, we initially propose an optimal multi-objective quantum-assisted algorithm, namely the Non-dominated Quantum Optimization (NDQO) algorithm, which evaluates the legitimate routes using the concept of Pareto Optimality at a reduced complexity. We then compare the performance of the NDQO algorithm to the state-of-the-art evolutionary algorithms, demonstrating that the NDQO algorithm achieves a near-optimal performance. Furthermore, we analytically derive the upper and lower bounds of the NDQO's algorithmic complexity, which is of the order of $O(N)$ and $O(N\sqrt{N})$ in the best- and worst-case scenario, respectively. This corresponds to a substantial complexity reduction of the NDQO from the order of $O(N^2)$ imposed by the brute-force (BF) method.

However again, as the number of nodes increases, the total number of routes increases exponentially, making its employment infeasible despite the complexity reduction offered. Therefore, we propose a novel optimal quantum-assisted algorithm, namely the Non-Dominated Quantum Iterative Optimization (NDQIO) algorithm, which exploits the synergy between the hardware parallelism and the quantum parallelism for the sake of achieving a further complexity reduction, which is on the order of $O(\sqrt{N})$ and $O(N\sqrt{N})$ in the best- and worst-case scenarios, respectively. Additionally, we provide simulation results for demonstrating that our NDQIO algorithm achieves an average complexity reduction of almost an order of magnitude compared to the near-optimal NDQO algorithm, while activating the same order of comparison operators.

Apart from the traditional QoS requirements, the network design also has to consider the nodes' user-centric social behavior. Hence, the employment of socially-aware load balancing becomes imperative for avoiding the potential formation of bottlenecks in the network's packet-flow. Therefore, we also propose a novel algorithm, referred to as the Multi-Objective Decomposition Quantum Optimization (MODQO) algorithm, which exploits the quantum parallelism to its full potential by exploiting the database correlations for performing multi-objective routing optimization, while at the same time balancing the tele-traffic load among the nodes without imposing a substantial degradation on the network's delay and power consumption. Furthermore, we introduce a novel socially-aware load balancing metric, namely the normalized entropy of the normalized composite betweenness of the associated socially-aware network, for striking a better trade-off between the network's delay and power consumption. We analytically prove that the MODQO algorithm achieves the full-search based accuracy at a significantly reduced complexity, which is several orders of magnitude lower than that of the full-search. Finally, we compare the MODQO algorithm to the classic NSGA-II evolutionary algorithm and demonstrate that the MODQO succeeds in halving the network's average delay, whilst simultaneously reducing the network's average power consumption by 6 dB without increasing the computational complexity.

Declaration of Authorship

I, **Dimitrios Alanis**, declare that the thesis entitled **Quantum-Assisted Multi-Objective Optimization of Heterogeneous Networks** and the work presented in it are my own and has been generated by me as the result of my own original research. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University;
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- Where I have consulted the published work of others, this is always clearly attributed;
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- Parts of this work have been published as: [1,2,3].

Signed: .... Date: ....

Acknowledgements

First of all, I would like to wholeheartedly thank my supervisors Professor Lajos Hanzo and Dr Soon Xin Ng (Michael) for their guidance, patience, stimulation and assistance upon the work to be presented. Additionally, I am grateful for the lectures offered by Professor Lie-Liang Yang, Professor Sheng Chen Dr Rob Maunder, Dr Rong Zhang and Dr Mohammed El-Hajjar during the "MSc in Wireless Communications" that provided all the necessary background for the fruition of this thesis. Moreover, I wish to thank Dr Zunaira Babar, Dr Panagiotis Botsinis, Dr Hung Viet Nguyen and Mr Daryus Chandra for their assistance in the development of the quantum-assisted routing algorithms as well as for the fruitful discussions which led to proposed algorithms' implementation. I also acknowledge the use of the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this work. Finally, I wish to thank my parents, Georgios and Vasiliki, as well as my sister, Kelly, for their moral and financial support, which made the completion of this thesis possible.

List of Publications

Lead-Author Publications:

1. **D. Alanis**, P. Botsinis, S. X. Ng, and L. Hanzo, "Quantum-Assisted Routing Optimization for Self-Organizing Networks," *IEEE Access*, vol. 2, pp. 451–472, 2014. DOI: 10.1109/ACCESS.2014.2322013.

2. **D. Alanis**, P. Botsinis, Z. Babar, S. X. Ng, and L. Hanzo, "Non-Dominated Quantum Iterative Routing Optimization for Wireless Multihop Networks," *IEEE Access*, vol. 3, pp. 1704–1728, 2015. DOI: 10.1109/ACCESS.2015.2478793.

3. **D. Alanis**, J. Hu, P. Botsinis, Z. Babar, S. X. Ng, and L. Hanzo, "Quantum-Assisted Joint Multi-Objective Routing and Load Balancing for Socially-Aware Networks," *IEEE Access*, vol. 4, pp. 9993–10028, 2016. DOI: 10.1109/ACCESS.2016.2629671.

Additional Publications:

1. P. Botsinis, **D. Alanis**, S. X. Ng, and L. Hanzo, "Low-Complexity Soft-Output Quantum-Assisted Multiuser Detection for Direct-Sequence Spreading and Slow Subcarrier-Hopping Aided SDMA-OFDM Systems," *IEEE Access* , vol. 2, pp. 451–472, 2014. DOI: 10.1109/ACCESS.2014.2322013.

2. Z. Babar, P. Botsinis, **D. Alanis**, S. X. Ng, and L. Hanzo, "The Road From Classical to Quantum Codes: A Hashing Bound Approaching Design Procedure," *IEEE Access*,vol. 3, pp. 146–176, 2015. DOI: 10.1109/ACCESS.2015.2405533.

3. P. Botsinis, **D. Alanis**, Z. Babar, S. X. Ng, and L. Hanzo, "Iterative Quantum-Assisted Multi-User Detection for Multi-Carrier Interleave Division Multiple Access Systems ", *IEEE Transactions on Communications*,vol. 63, no. 10, pp. 3713-3727, Oct. 2015. DOI: 10.1109/TCOMM.2015.2458857.

4. Z. Babar, P. Botsinis, **D. Alanis**, S. X. Ng, and L. Hanzo, "Fifteen Years of Quantum LDPC Coding and Improved Decoding Strategies," *IEEE Access*, vol. 3, pp. 2492–2519, 2015. DOI: 10.1109/ACCESS.2015.2503267.

5. P. Botsinis, **D. Alanis**, Z. Babar, S. X. Ng, and L. Hanzo, "Non-Coherent Quantum Multiple Symbol Differential Detection for Wireless Systems", *IEEE Access*, vol. 3, pp. 569–598, 2015. DOI: 10.1109/ACCESS.2015.2432015.

6. Z. Babar, P. Botsinis, **D. Alanis**, S. X. Ng and L. Hanzo, "Construction of Quantum LDPC Codes From Classical Row-Circulant QC-LDPCs," *IEEE Communications Letters*, vol. 20, no. 1, pp. 9–12, Jan. 2016. DOI: 10.1109/LCOMM.2015.2494020.

7. S. Yang, X. Xu, **D. Alanis**, and L. Hanzo, 'Is the Low-Complexity Mobile Relay Aided FFR-DAS Capable of Outperforming the High-Complexity CoMP?," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 2154–2169, 2016. DOI: 10.1109/TVT.2015.2416333.

8. Z. Babar, H. Nguyen, P. Botsinis, **D. Alanis**, D. Chandra, S. X. Ng, and L. Hanzo, "Serially-Concatenated Unity-Rate Codes Improve Quantum Codes Without Coding-Rate Reduction," *IEEE Communications Letters*, vol. 20, no. 10, pp. 1916–1919, 2016. DOI: 10.1109/LCOMM.2016.2593874.

9. P. Botsinis, **D. Alanis**, Z. Babar, H. Nguyen, D. Chandra, S. X. Ng, and L. Hanzo, "Quantum-aided Multi-User Transmission in Non-Orthogonal Multiple Access Systems," *IEEE Access*, vol. 4, pp. 7402-7424, 2016.

10. Z. Babar, L. Hanzo, H. Nguyen, P. Botsinis, **D. Alanis**, D. Chandra, S. X. Ng, R. G. Maunder, "A Fully-Parallel Quantum Turbo Decoder," *IEEE Access*, vol. 4, pp. 6073–6085, 2016. DOI: 10.1109/ACCESS.2016.2591904.

11. P. Botsinis, **D. Alanis**, Z. Babar, H. Nguyen, D. Chandra, S. X. Ng, and L. Hanzo, 'Joint Quantum-Assisted Channel Estimation and Data Detection," *IEEE Access*, vol. 4, pp. 7658–7681, 2016.

12. P. Botsinis , Z. Babar , **D. Alanis** , D. Chandra, H. Nguyen, S. X. Ng, and L. Hanzo, "Quantum Error Correction Protects Quantum Search Algorithms Against Decoherence," *Scientific Reports*, vol. 6, 2016. DOI:10.1038/srep38095.

13. H. Nguyen, Z. Babar, **D. Alanis**, P. Botsinis, D. Chandra, L. Hanzo, and S. X. Ng, "EXIT-chart Aided Quantum Code Design Improves the Normalised Throughput of Realistic Quantum Devices,“ *IEEE Access*, vol. PP, no. 99, pp. 1-1, 2016. DOI: 10.1109/ACCESS.2016.2591910.

14. P. Botsinis, **D. Alanis**, Z. Babar, S. X. Ng, and L. Hanzo, "Coherent Versus Non-Coherent Quantum-Assisted Solutions in Wireless Systems," *IEEE Wireless Communications Magazine*, 2016. In press.

15. Z. Babar, L. Hanzo, H. Nguyen, P. Botsinis, **D. Alanis**, D. Chandra, S. X. Ng, R. G. Maunder, 'Unity-Rate Codes Maximize the Normalized Throughput of On-Off Keying Visible Light Communication," *IEEE Photonics Technology Letters*, vol. PP, no. PP, pp. 1–1, 2016. DOI: 10.1109/LPT.2016.2625808.

Contents

List of Symbols

B_{sin}	Singular Betweenness
B_{com}	Composite Betweenness
$\bar{B}_{com}$	Normalized Composite Betweenness
C	Pareto Completion Ratio
CL	Overall Route Power Dissipation
CD	Overall Route Delay
$D(x)$	x-th Route Delay
$\bar{D}(S)$	Average Network Delay of the Route-Combination S
$\mathbf{f}(x)$	Utility Vector of the x-th route
$f_k(x,i)$	Lower Comparison Function Between the x-th and the i-th Routes
$f_k^{\bullet}(x,i)$	Comparison Function Between the x-th and the i-th Routes based on the $\bullet$ operator
$\mathcal{F}_{\text{MC}}$	MC Friendship Matrix
$\mathcal{F}_{\text{MR}}$	MR Friendship Matrix
$\mathcal{G}$	Grover's QSA Operator
$g(x,i)$	Dominance Operator Function Between the x-th and the i-th Routes
H	Quantum Hadamard Gate
$\bar{H}$	Normalized Entropy
$L_{BBHT}^{QD,\max}$	BBHT-QSA time-out in $\mathcal{G}$ Applications
$L_{\text{CM}}^{\text{outer}}$	Complexity of the CM method for the MODQO outer step

$L_{\mathrm{CM}}^{\mathrm{ref}}$	Reference Complexity of the CM Method for the MODQO Outer Step
L_{dB}	Path Loss per Individual Link in dB
$L_{DHA}^{QD,\max}$	DHA time-out in $\mathcal{G}$ Applications
L_{ES}	Complexity of the Exhaustive Search
$L_{NDQIO,tot}^{P\,\max}$	Upper Parallel Complexity Bound of the NDQIO Algorithm
$L_{NDQIO,tot}^{P,\min}$	Lower Parallel Complexity Bound of the NDQIO Algorithm
$L_{NDQIO,tot}^{S,\max}$	Upper Sequential Complexity Bound of the NDQIO Algorithm
$L_{NDQIO,tot}^{S,\min}$	Lower Sequential Complexity Bound of the NDQIO Algorithm
$L_{NDQO}^{tot,\max}$	Upper Complexity Bound of the NDQO Algorithm
$L_{NDQO}^{tot,\min}$	Lower Complexity Bound of the NDQO Algorithm
L_{qr}	Length of a Quantum Register in qubits
$L_{\mathrm{MODQO}}^{\mathrm{inner}}$	MODQO Algorithm Inner Step Complexity
$L_{\mathrm{MODQO}}^{\mathrm{outer}}$	MODQO Algorithm Outer Step Complexity
$L_{\mathrm{MODQO}}^{\mathrm{tot}}$	MODQO Algorithm Total Complexity
$L_{\mathrm{QM}}^{\mathrm{outer}}$	Complexity of a Single Iteration of the QM Method for the MODQO Outer Step
$L_{\mathrm{QM}}^{\mathrm{ref}}$	Reference Complexity of a Single Iteration of the QM Method for the MODQO Outer Step
λ_c	Carrier Frequency
$\mathcal{M}_m$	Measurement Operator of the State m
N	Total Number of Legitimate Routes
N_G	Number of generations in the NSGA-II
N_{MC}	Number of Mesh Clients
N_{MR}	Number of Mesh Routers
N_{OPF}	Number of Pareto-optimal Routes
N_{pop}	Number of individuals per generation in the NSGA-II
O	Grover's QSA Quantum Oracle Gate
N_r	Total Number of Active Routes
P_c	Crossover Probability in the NSGA-II

P_d	Pareto Distance
P_e	Bit Error Ratio
P_m	Mutation Probability in the NSGA-II
$P^{t,act}$	Transmission Power Matrix of all the Network Links
S	Set of Legitimate Routes
S^{OPF}	Set of Pareto-optimal Route-Combinations
$S^{\text{OPF}}_{(n)}$	Set of Pareto-optimal Route-Combinations after n Trellis Stages
$S^{\text{OPF}}_{\text{MC}}$	Set of Pareto-optimal MC Routes
$S^{\text{OPF}}_{\text{MR}}$	Set of Pareto-optimal MR Routes
t	Number of valid solutions
T_n	n-qubit Quantum Toffoli Gate
U_f	Generic Quantum Unitary Operator implementing the function $f(x)$
U_{f_k}	Quantum Unitary Operator implementing the Comparison Operator $f_k(x,i)$
U_g	Quantum Unitary Operator implementing the Dominance Operator $g(x,i)$
$U_{g'}$	Parallel Quantum Unitary Operator implementing the Dominance Operator $g(x,i)$
U_G	Quantum Unitary Operator for Parallel Activation of Multiple $U_{g'}$ Operators
x	Index of the Legitimate Route in the Route List
α	Pheromone Intensity in the MO-ACO
β	Intrinsic Affinity Weighting in the MO-ACO
η	Intrinsic Affinity Matrix in the MO-ACO
ζ	Number of individuals per generation in the MO-ACO
Ξ	Number of generations in the MO-ACO
τ	Multi-pheromone Structure in the MO-ACO
$\|\psi\rangle$	Quantum State ψ

List of Acronyms

ACO	Ant Colony Optimization
AWGN	Additive White Gaussian Noise
BBHT	Boyer, Brassard, Høyer and Tapp
BCO	Bee Colony Optimization
BER	Bit Error Ratio
BF	Brute Force
BSC	Binary Symmetric Channel
BW-BBHT	Backward BBHT-QSA
CD	Classical Domain
CDMA	Code-Division Multiple Access
CF(E)	Cost Function (Evaluation)
CLT	Central Limit Theorem
CM	Classical Merging
CNOT	Controlled-NOT quantum gate
DAF	Decode And Forward
DCCP	Dynamic Coverage and Connectivity Problem
DHA	Dürr-Høyer-Algorithm
DTN	Delay Tolerant Networks
DN	Destination Node
ES	Exhaustive Search
GQCR	Global Quantum Control Register

GQIR	Global Quantum Index Register
GOW	Global Oracle Workspace
HetNet	Heterogeneous Network
HGR	Hybrid Geographic Routing
HP	Hardware Parallelism
IoT	Internet of Things
HYMN	HYbrid Multihop Network
LQCR	Local Quantum Control Register
LQIR	Local Quantum Index Register
LTE	Long Term Evolution
LOW	Local Oracle Workspace
MC	Mesh Client
MO-ACO	Multi-Objective Ant Colony Optimization
MODE	Multi-Objective Differential Evolution
MODQO	Multi-Objective Decomposition Quantum Optimization
MR	Mesh Router
MUD	Multi-User Detection
MUT	Multi-User Transmission
NDQO	Non-dominated Quantum Optimization
NDQIO	Non-dominated Quantum Iterative Optimization
NOMA	Non-Orthogonal Multiple Access
NP	Non-deterministic Polynomial
NSGA-II	Non-dominated Sort Genetic Algorithm II
OF	Objective Function
OFDM	Orthogonal Frequency-Division Multiplexing
OPF	Optimal Pareto Front
OSN	Online Social Network
OW	Oracle Workspace
PF	Pareto Front

PSO	Particle Swarm Optimization
PLR	Packet Loss Ratio
RWBS	Repeated Weighted Boosting Search
QAE	Quantum Amplitude Estimation
QCR	(Global/Local)Quantum Control Register
QCA	Quantum Counting Algorithm
QD	Quantum Domain
QFT	Quantum Fourier Transformation
QGOA	Quantum Genetic Optimization Algorithm
QIR	(Global/Local) Quantum Index Register
QM	Quantum Merging
QMA	Quantum Mean Algorithm
QoS	Quality of Service
QP	Quantum Parallelism
QPA	Quantum Phase Algorithm
QR	Quantum Register
QRWBS	Quantum Repeated Weighted Boosting Search
QSA	Quantum Search Algorithm
QWSA	Quantum Weighted Sum Algorithm
RN	Relay Node
SN	Source Node
SNA	Social Network Analysis
SNR	Signal-to-Noise Ratio
SR	Self-Repair
UF	Utility Function
UV	Utility Vector
VoIP	Voice over Internet Protocol
VANET	Vehicular Ad-hoc Network
WMHN	Wireless Multihop Network

WMN	Wireless Mesh Network
WSN	Wireless Sensor Network
XOR	Exclusive OR gate

Chapter 1

Introduction

1.1 Motivation

Back in 1991, Weiser [4] unveiled his vision for ubiquitous computing, where most aspects of human life would be supported by portable computing units, which he termed as "pads". Twenty five years later, this vision has come to fruition, since indeed our daily lives rely on "pads", which are commonly known as *tablets* and *smart phones*. More specifically, for the latter, there is a prediction by *eMarketer*[1] that their market penetration will surpass the 2-billion mark in 2016, accounting for about 30% of the world's population. Explicitly, the proliferation of mobile networking devices has lead to an exponential increase in the bandwidth requirements of this ever increasing tele-traffic load [5]. For the sake of mitigating this spectral shortage, several solutions have been advocated, such as *millimeter Wave Communications* (mm-Wave) [6, 7] and *Optical Wireless Communications* [8, 9], which is also often referred to as *Light Fidelity* (LiFi) [10]. These solutions succeed in increasing the area spectral efficiency both by utilizing the hitherto unlicensed spectrum [5] of the mm-Wave bands [6] as well as of the visible light bands [10] and by mitigating the interference levels due the higher path-losses experienced by these bands compared to the classic *Radio Frequency* (RF) frequency-region [11]. Note that the latter objective can also be achieved in networks operating in the RF band by jointly deploying *Distributed Antenna Systems* (DAS) [12, 13] as well as optimal power control. Explicitly, this deployment minimizes the transmit power required by creating smaller virtual cells within a larger over-sailing cell, since the mobile users tend to be located closer to the distributed antennas, thus exhibiting a reduced path-loss.

However, the aforementioned technologies providing relatively small coverage areas have to be deployed in conjunction with the classic cellular infrastructure for the sake of improving the over-sailing cell's coverage, especially at the cell edges [15]. In this way, network consisting of different and diverse smaller underlay networks is formed, which is often referred to as *Hetergeneous Network* (HetNet) [14]. A conceptual representation of a HetNet

[1] `http://www.emarketer.com/Article/2-Billion-Consumers-Worldwide-Smartphones-by-2016/1011694`

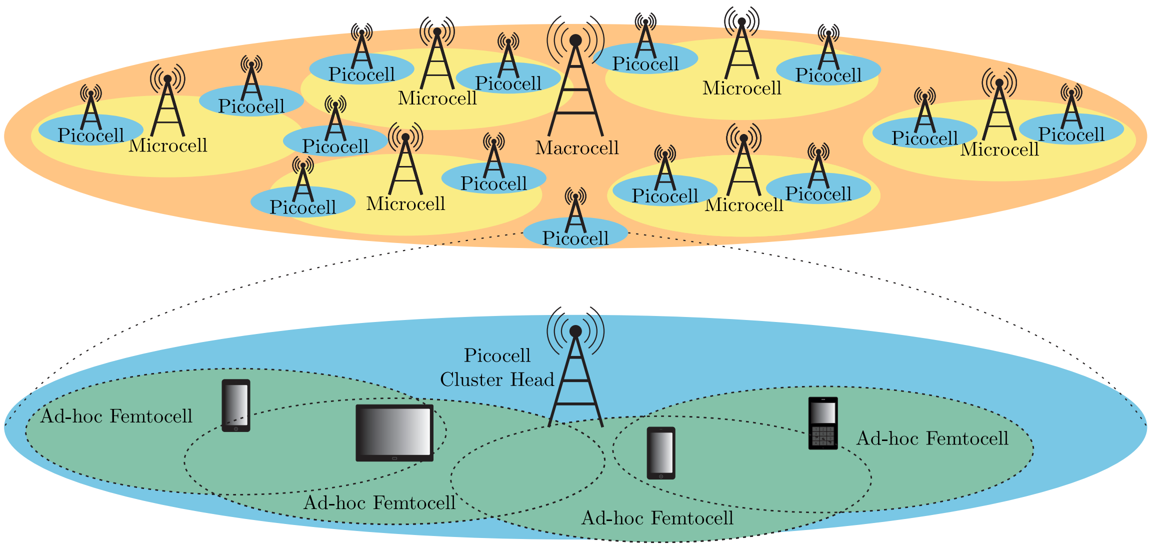


Figure 1.1: Overlay cells in a *Heterogeneous Network* (HetNet) [14]. The bottom figure zooms in a specific picocell of the top figure, where the mobile nodes form their own *ad-hoc femtocells* facilitating multihop cooperative communications.

is shown in Fig. 1.1, where the HetNet's coverage area is that of the large over-sailing cell, which is termed as a *macrocell*. This type of cell can be illuminated by a 4G *Long-Term Evolution* (LTE) [16] or even a 3G [17] base station. Hence, for the sake of improving both the coverage quality and the area spectral efficiency several smaller cells are deployed [18], which are termed as *microcells* in Fig. 1.1. These specific cells can be operating either under the *LTE Advanced* (LTE-A) standard [18] or using mm-Wave communications. Often, even smaller cells are deployed, which are termed as *picocells* in Fig. 1.1 and may be illuminated by using either *Wireless Fidelity* (WiFi) [17] or optical wireless access points.

Although the diverse mobile nodes located in the picocells' coverage area have their downlink served by the picocells' base stations as seen in the bottom sub-figure of Fig. 1.1, this may not be the case for their uplink. Naturally, since these specific nodes only have access to restricted sources of power, their maximum allowable transmit power is tuned accordingly for the sake of ensuring that the nodes remain active for as long as possible [19]. This may be prohibitive for establishing a direct link with the picocell base stations owing to their excessive power requirements for reliable transmission, which do not comply with the aforementioned power constraint. To mitigate this problem, each of the mobile nodes can create a *mobile hotspot* [20] forming an even smaller *ad-hoc* cell, which is referred to as an *ad-hoc femtocell* [15] in Fig. 1.1. Explicitly, these specific ad-hoc femtocells are capable of serving their neighboring nodes for the sake of relaying their messages either to another ad-hoc femtocell closer to the picocell base station or even to the base station [21], should the respective link be sufficiently reliable. For instance, observe that the far left hand-side node in the bottom figure of Fig. 1.1 is unable to establish a link with the picocell, since the latter lies outside its transmission range. Therefore, it has to utilize its neighboring node as a relay for the sake of reliably sending its message to the picocell base station, which acts as a gateway. Naturally, specific coordination is required for supporting the femtocells' forwarding operation [16] for the sake of maximizing the HetNet's performance. This can be realized by providing an intelligent central node, which is termed as *cluster head* in Fig. 1.1 and it is accommodated by the picocell base station for the sake of simplicity.

This specific multihop function [22], which facilitates both direct and indirect communication between the source and the destination nodes, is capable of potentially increasing the area spectral efficiency [23], as the ad-hoc femtocells proliferate. In fact, multihop user cooperation is already considered by the LTE-A standard [24]. Based on the aforementioned paradigm, the HetNets' cells can be viewed as clusters capable of supporting multihop transmission coordinated by the respective base station. Therefore, as benefit of this specific coordination capability, HetNets can also be treated as heterogeneous *Wireless Multihop Networks* (WMHNs) [25]. Explicitly, this concept can be readily applied to all networks ranging from *Wireless Sensor Networks* (WSNs) [26] and *wireless ad-hoc networks* [27] to *smart grid networks* [28], as long as they are comprised by nodes equipped by the required capabilities or interfaces.

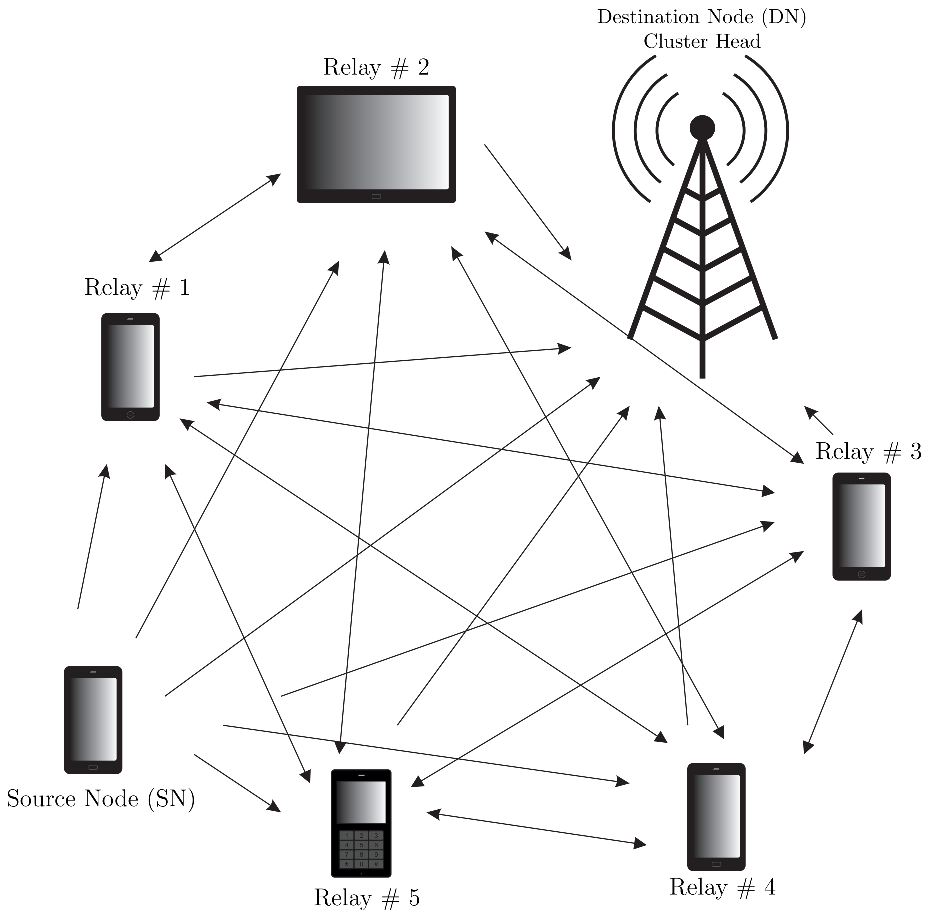


Figure 1.2: Routing in HetNets.

Based on this specific paradigm, the HetNets' cell coordination function can be viewed as routing through heterogeneous WMHNs, as portrayed in Fig. 1.2. Observe in this figure that the *Source Node* (SN) has to transmit its message to the *Destination Node* (DN), which acts as a gateway [26]. This transmission can be undertaken either by establishing a direct link between the SN and the DN or by using a cloud of *Relay Nodes* (RNs). Based on the specific network optimization criteria considered, the DN, which acts as cluster head as well, has to find the optimal route. Having defined our main concept, let us now proceed by presenting the diverse optimization criteria considered.

1.2 Classical Routing Approaches

Each of the WMHN nodes attempts to optimize its performance in terms of different and often conflicting Quality of Service (QoS) parameters, such as the Bit Error Ratio (BER), the Packet Loss Ratio (PLR) and the end-to-end delay, while having access to a restricted amount of power. Therefore, optimal routing is essential for satisfying the aforementioned QoS criteria. Nevertheless, as the number of WMHN nodes involved escalates, the total number of potential routes increases exponentially, turning the routing optimization problem into a *Non-deterministic Polynomial-time hard* (NP-hard) one [29], hence requiring sophisticated heuristic methods. Let us now proceed by presenting the related work carried out in the field of routing.

1.2.1 Single-Objective Routing Approaches

A plethora of single-objective studies exist in the literature [30, 31, 32, 33, 34, 19, 35, 36, 37, 38, 39, 40, 41, 42, 43], each addressing different routing aspects. In a nutshell, these specific studies consider the optimization objectives in a single-component aggregate function in an attempt to optimize the latter using either a heuristic or a formal systematic optimization method. To elaborate further, several of these studies [30, 31, 32, 33] utilize Dijkstra's algorithm [44] for the sake of identifying the optimal routes. Explicitly, this technique is capable of approaching the optimal routes at the cost of imposing a complexity on the order of $O(E^3)$, where E corresponds to the number of edges in the network's graph. For instance, Zuo *et al.* [31] employed this specific algorithm for the sake of optimizing the route's energy efficiency in the context of wireless ad-hoc networks. Hu *et al.* [30] utilized Dijkstra's algorithm for minimizing both the power consumption and the delay, quantified in terms of the number of hops, in socially-aware networks. Additionally, Dehghan *et al.* [33] adapted this specific algorithm to the problem of cooperative routing and attempted to maximize the route's energy efficiency.

Additionally, the beneficial properties of *convex optimization* [45] have also been exploited in the context of routing optimization. To elaborate further, Dall'Anese and Giannakis [34] transformed the non-convex routing problem of cognitive random access networks into a convex one using successive convex approximations for the sake of minimizing both the routes' *Packet Loss Ratio* (PLR) [17] and the resultant outage probability. Additionally, Yetgin *et al.* [19] maximized the network lifetime in the context of WSNs using a similar approach. In fact, this specific metric encapsulates several optimization objectives [35], such as the power consumption, the nodes' battery levels and the route's delay. Based on this specific metric, Abdulla *et al.* [36] have maximized the lifetime of WSNs by introducing a range of Hybrid Multihop Network (HYMN) parameters. The so-called *Network Utility* [37] also constitutes a meritorious single-component optimization. Apart from the

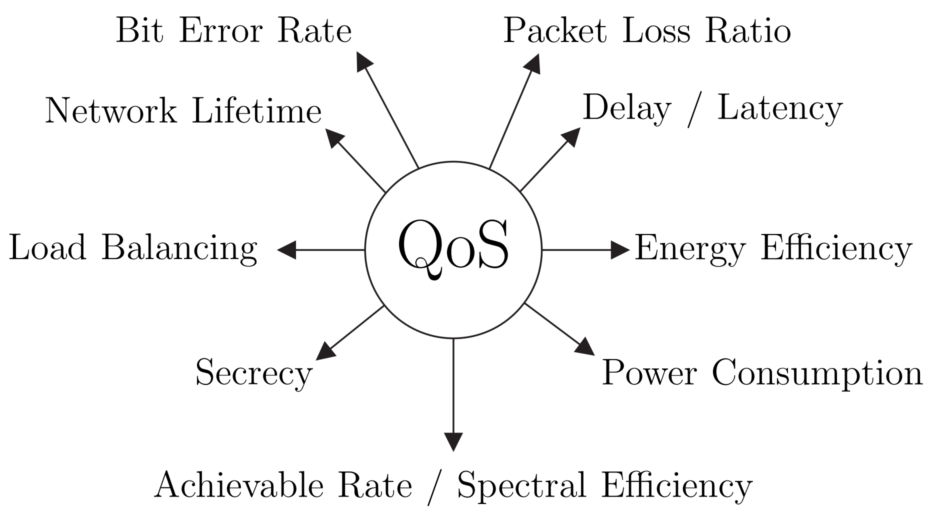


Figure 1.3: Some of the notable QoS criteria.

aforementioned objectives, Network Utility also takes into account the routes' achievable rate [38], while providing a more holistic view of the routing problem.

In contrast to the aforementioned methodology, there exist several single-component studies, which do not lie within the aforementioned trends. To elaborate further, Zhu *et al.* [39] have proposed a routing protocol that succeeds in minimizing the energy consumption of *Wireless Sensor Networks* (WSNs) by organizing the nodes using *Hausdorff clustering* [46]. Additionally, Chen *et al.* [40] have conceived a Hybrid Geographic Routing (HGR) scheme for minimizing the total energy dissipation, while satisfying the end-to-end delay constraints. Furthermore, Al-Rabayah and Malaney [41] have designed a hybrid routing protocol for minimizing the routing overhead incurred by broken links in *Vehicular Ad-hoc Networks* (VANETs). Huang *et al.* [42] minimized the control-channel setup cost, while balancing the tele-traffic load among the nodes of a *Software-Defined* HetNet. Additionally, Yao *et al.* [43] proposed a routing scheme for maximizing the established routes' secrecy, thus securing them from potential eavesdroppers.

Based on the aforementioned contributions, it is clear that the QoS optimization criteria selected for the sake of optimizing the quality of the routes are rather diverse. As a recap, in Fig. 1.3 we present the diverse QoS criteria considered in the aforementioned contributions, which often entail juxtaposed constraints. Having briefly elaborated on the single-component routing optimization techniques let us now proceed with a detailed description of the multi-component techniques.

1.2.2 Multi-Component Routing Approaches

Despite the numerous single-objective approaches advocated in the literature, focusing on a single requirement may unduly degrade the remaining metrics. This problem may be mitigated [47] by using a multi-objective approach. Likewise, all the requirements considered may be optimized jointly without the need for user-defined parameters in order to aggregate the different design objectives [48].

In fact, there are some comprehensive studies in the literature [29, 49, 50, 51, 52], which investigate diverse aspects of WSNs using the multi-objective approach relying on evolutionary algorithms. For example, Yetgin *et al.* [29] used both the Non-dominated Sorting Genetic Algorithm II (NSGA-II) and the Multiobjective Differential Evolution Algorithm (MODE) for optimizing the transmission routes in terms of their end-to-end delay and power dissipation. They used the concept of Pareto Optimality [53] for evaluating the fitness of multi-objective problems. While considering a similar context, Camelo *et al.* [49] employed the NSGA-II in order to satisfy the same QoS requirements for both the ubiquitous Voice over Internet Protocol (VoIP) and for file transfer. Moreover, Perez *et al.* [50] used a multi-objective model for optimizing both the number of sensor nodes used in a WSN and the total energy dissipation of the network, which allowed the minimization of the WSN's deployment cost. Martins *et al.* [51] employed a hybrid multi-objective evolutionary algorithm for solving the Dynamic Coverage and Connectivity Problem (DCCP) of WSNs subjected to node failures.

Among the evolutionary algorithms, the so-called *Ant Colony Optimization* (ACO) conceived by Dorigo and Di Caro [54, 55] has been extensively used for optimizing routing problems. In the so-called *Ant Network* (AntNet) [55], each ant representing a legitimate route travels from a source node (SN) to the destination node (DN), while traversing a different number of nodes. Each ant moving from one node to another deposits an amount of pheromone across its route depending on the heuristic value of the Objective Function (OF) that is being optimized. To elaborate further, as the value of the OF increases throughout the route-search, the intensity of the pheromone would increase as well. Based on the deposited pheromone, the ant in nature are capable of sensing in binary fashion, whether a specific route is leading up to the source of food or not. Meritorious routes attract more ants and as additional ants choose to follow a specific link between two nodes, the pheromone they deposit is cumulatively superimposed so that the optimal routes would have the highest pheromone intensity. An additional factor, namely the so-called *intrinsic affinity*, was introduced for avoiding premature convergence to local optima, which particularly corresponds to the *a priori* probability of each solution being the globally optimum solution.

Further extensions of the *AntNet* model have also been proposed. Both Golshahi *et al.* [56] and Chandra *et al.* [57] have implemented a routing protocol using the *AntNet* model; two types of agents have been used: a forward-oriented one which would seek to explore the network and a backward-oriented one, which would perform the ACO operations including the pheromone update and fitness-evaluation of the routes. Finally, Wang and Wu [58] developed an ACO routing algorithm for optimizing the performance of fault tolerant *Hypercube Networks* at reduced complexity by exploiting the regularities exhibited by these networks. Additionally, Pinto *et al.* introduced the concept of Pareto Optimality in [52] for conceiving a *Multiobjective Max-Min Ant System* (MMAS) for solving the multi-objective mutlicast routing problem. Finally, a quantum-inspired version of the ACO (QACO) algorithm has been proposed by Wang *et al.* [59]. Jiang *et al.* [60] invoked an enhanced version of the QACO design for increasing the lifetime of a WSN by minimizing its energy consumption.

Most of the aforementioned multi-component methods advocate only two optimization components [61]. However, involving more components results in increasing the complexity imposed by the aforementioned methods, owing to the ever-increasing number of optimal routes [47]. *Consequently, since we will opt for incorporating more than two optimization objectives in our studies for the sake of adopting a more holistic approach, we will need sophisticated optimization methods for mitigating the escalating complexity.* Fortunately, quantum search algorithms provide an attractive framework for tackling this complexity escalation, when the number of objectives is increased. We will assume that the cluster-head of Fig. 1.2 is in possession of a *gate-based quantum computer* [62], which is capable of employing these specific algorithms. Additionally, the cost of employing a quantum computer is justified for densely populated scenarios, such as an airport terminal or a stadium, where the classical methods may become overwhelmed.

Let us now proceed with presenting the recent advances in quantum computing.

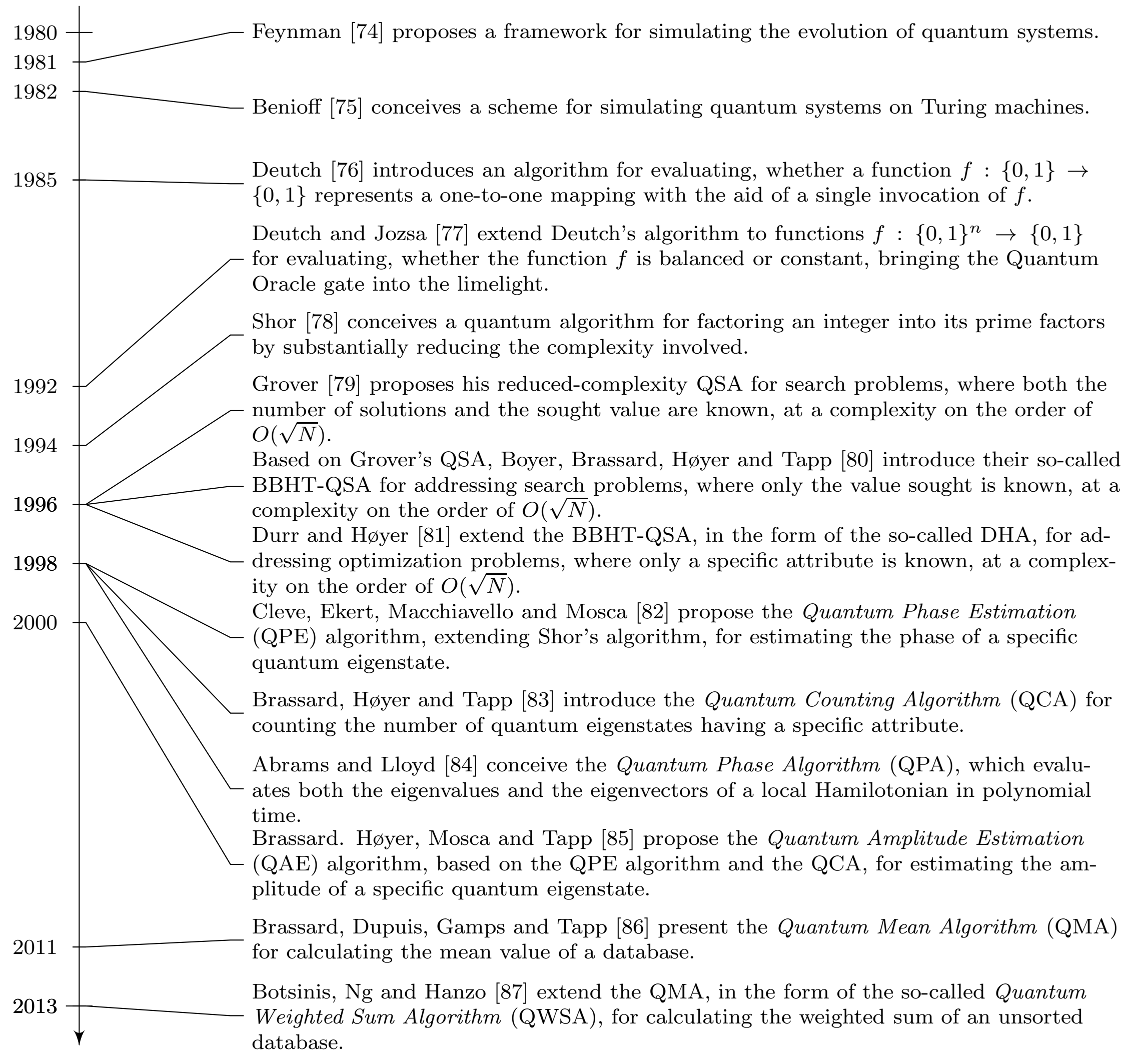
1.3 A Leap into the Quantum World

The ever-reducing transistor size following Moore's law is approaching the point, where the so-called *quantum effects* [62] become prevalent in the transistors' operation [63]. This specific trend implies that the quantum-effects become unavoidable, hence rendering the research of quantum computation systems an urgent necessity. In fact, a *quantum annealing chipset* [64] is already commercially available from *D-Wave*[2] [65, 66]. Apart from the quantum annealing architecture, the so-called *gate-based architecture* [67], which relies on building computational blocks using quantum gates in a similar fashion to classic gates, is attracting increasing attention due to the recent advances in *quantum stabilizer codes* [68, 69, 70, 71, 72, 73], which are capable of mitigating the *decoherence* effects encountered by quantum circuits [62]. Explicitly, the aforementioned advances provide us with a great potential in rendering NP-hard problems, such as the multiple-objective routing problem, tractable, since quantum systems are capable of addressing these specific problems at a near-full-search-based accuracy, while imposing a reduced complexity by exploiting the powerful concept of *Quantum Parallelism* (QP) [62]. Let us now provide a brief introduction to some of the most popular quantum search algorithms for the sake of highlighting their merits.

1.3.1 Introduction to Quantum Search Algorithms

The inspiration of quantum computation was provided by Feynman [74], who proposed in 1981 a novel framework for conveying information by the spin of an electron and for simulating the evolution of the quantum states. In the following year, Benioff [75] proposed a technique of simulating quantum systems on Turing machines. Several years later, the ef-

[2]`https://www.dwavesys.com/d-wave-two-system`



Feynman [74] proposes a framework for simulating the evolution of quantum systems.

Benioff [75] conceives a scheme for simulating quantum systems on Turing machines.

Deutch [76] introduces an algorithm for evaluating, whether a function $f : \{0,1\} \to \{0,1\}$ represents a one-to-one mapping with the aid of a single invocation of f.

Deutch and Jozsa [77] extend Deutch's algorithm to functions $f : \{0,1\}^n \to \{0,1\}$ for evaluating, whether the function f is balanced or constant, bringing the Quantum Oracle gate into the limelight.

Shor [78] conceives a quantum algorithm for factoring an integer into its prime factors by substantially reducing the complexity involved.

Grover [79] proposes his reduced-complexity QSA for search problems, where both the number of solutions and the sought value are known, at a complexity on the order of $O(\sqrt{N})$.

Based on Grover's QSA, Boyer, Brassard, Høyer and Tapp [80] introduce their so-called BBHT-QSA for addressing search problems, where only the value sought is known, at a complexity on the order of $O(\sqrt{N})$.

Durr and Høyer [81] extend the BBHT-QSA, in the form of the so-called DHA, for addressing optimization problems, where only a specific attribute is known, at a complexity on the order of $O(\sqrt{N})$.

Cleve, Ekert, Macchiavello and Mosca [82] propose the *Quantum Phase Estimation* (QPE) algorithm, extending Shor's algorithm, for estimating the phase of a specific quantum eigenstate.

Brassard, Høyer and Tapp [83] introduce the *Quantum Counting Algorithm* (QCA) for counting the number of quantum eigenstates having a specific attribute.

Abrams and Lloyd [84] conceive the *Quantum Phase Algorithm* (QPA), which evaluates both the eigenvalues and the eigenvectors of a local Hamilotonian in polynomial time.

Brassard. Høyer, Mosca and Tapp [85] propose the *Quantum Amplitude Estimation* (QAE) algorithm, based on the QPE algorithm and the QCA, for estimating the amplitude of a specific quantum eigenstate.

Brassard, Dupuis, Gamps and Tapp [86] present the *Quantum Mean Algorithm* (QMA) for calculating the mean value of a database.

Botsinis, Ng and Hanzo [87] extend the QMA, in the form of the so-called *Quantum Weighted Sum Algorithm* (QWSA), for calculating the weighted sum of an unsorted database.

Figure 1.4: Timeline of quantum computing milestones.

fect of QP has been exploited by Deutch [76], who conceived an algorithm, named after him as the *Deutch's Algorithm*, for determining whether a binary function $f : \{0,1\} \to \{0,1\}$ has or has not one-to-one mapping by only using a single call of the function. An extension of this algorithm, namely the so-called *Deutch-Jozsa Algorithm* [77], was conceived for determining whether a function $f : \{0,1\}^n \to \{0,1\}$ is balanced or constant. The *Deutch-Jozsa Algorithm* laid the foundations for the development of the so-called *Quantum Oracle* gates [62], which are quantum circuits implementing a generic mapping function $f : \{0,1\}^N \to \{0,1\}^M$ and they are capable of calculating all the pairs of possible inputs-outputs of f using a single call of f by exploiting the QP.

Based on these gates, Grover [79] has proposed a *Quantum Search Algorithm* (QSA), which has been shown to be optimal by Zalka [88]. This QSA is capable of finding a desired solution stored in an unsorted database by imposing a low complexity, which is on the order of $O(\sqrt{N})$, as long as both the number of valid solutions and the solution to be found are known to the optimization process. An even more powerful extension of Grover's QSA has been introduced by Boyer *et al.* [80] in the form of the so-called *Boyer-Brassard-Høyer-Tapp* QSA (BBHT-QSA), which is applicable in the specific scenario, where the actual number of valid solutions is unknown, whilst imposing the same order of complexity, namely $O(\sqrt{N})$. A further extension of the BBHT-QSA has been conceived by Dürr and Høyer [81], where the *Durr-Høyer Algorithm* (DHA) is employed for identifying the extreme values of an unsorted database, while imposing a low complexity, which is on the order of $O(\sqrt{N})$. Subsequently, Malossini *et al.* [89] proposed the so-called *Quantum Genetic Optimization Algorithm* (QGOA), which constitutes a steady-state GA [90, 91], in which the mating process is enhanced by the DHA.

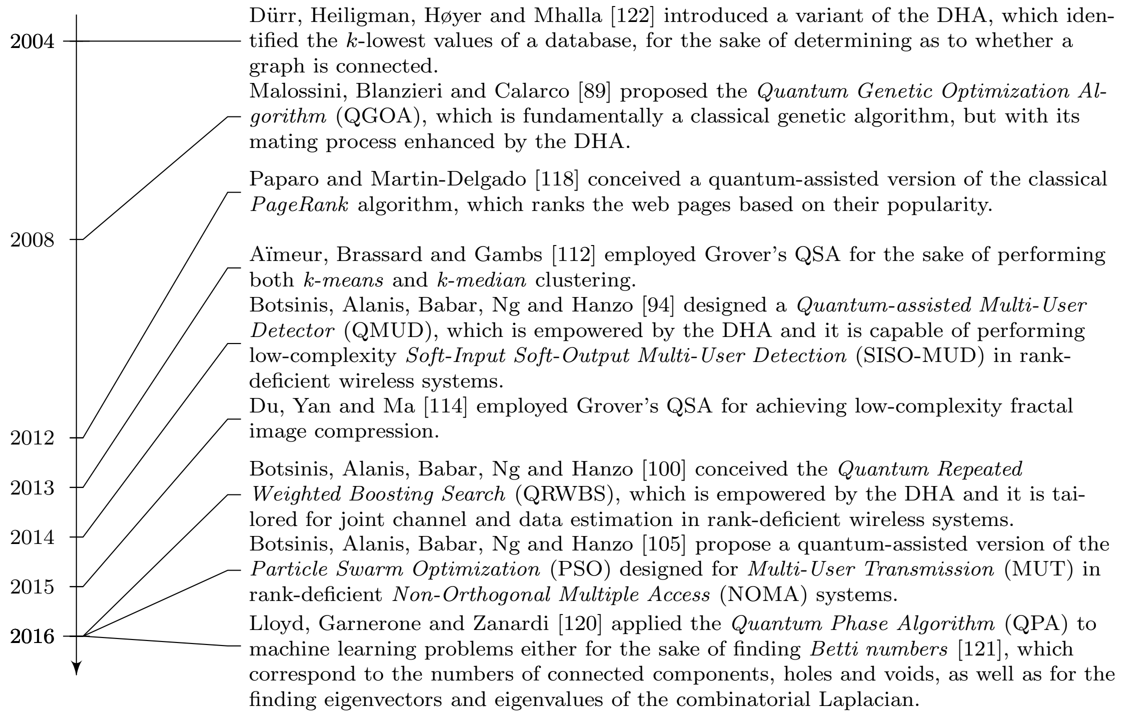
Furthermore, several contributions exist, which exploit the properties of the *Quantum Fourier Transformation* (QFT) [67, 92]. In particular, Shor [78] has proposed a quantum algorithm for addressing the prime integer factorization problem at a complexity on the order of $O\left[\log(N)^3\right]$. Shor's algorithm formed the basis for the concept of *Quantum Phase Estimation* (QPE), which was proposed by Cleve *et al.* [82] and allowed the estimation of the phase of a specific quantum eigenstate. This innovation led, in turn, to the concept of both *Quantum Counting Algorithm* (QCA) [83] as well as to that of *Quantum Amplitude Estimation* (QAE) [85]. Based on these concepts, Brassard *et al.* [86] proposed the so-called *Quantum Mean Algorithm* (QMA) for calculating the mean of values found in an unsorted database at a reduced complexity. Additionally, Abrams and Lloyd [84] conceived the so-called *Quantum Phase Algorithm* (QPA) for the sake of evaluating the eigenvalues and the eigenvectors of a local Hamiltonian, relying on the properties of the QFT. The QPA operated at a complexity, which exhibited a polynomially increasing complexity with the search-space-size, as opposed to the exponential trend of the full search. The milestones of quantum computing are summarized in the timeline of Fig. 1.4.

1.3.2 Applications of Quantum Search Algorithms

A variety of quantum-assisted solutions [87, 93, 94, 95, 96] have been advocated in the context of the classic Multi-User Detection (MUD) problem, which jointly determines the specific legitimate symbols transmitted by each of the users supported. More specifically, the detection only relies on the noise-contaminated received signal, on the knowledge of the symbol constellation of each user and on their channel states. In lightly loaded, or full-rank systems, where the sum of number of transmit antennas of all users is lower than or equal to the number of receive antennas at the base station, low-complexity linear MUDs exhibit an adequate performance. However, in rank-deficient systems, where the number of transmit antennas is higher than the number of receive antennas, sophisticated non-linear search techniques have to be employed. The optimal classic MUD is the maximum likelihood MUD, which performs a full search for finding the specific combination of legitimate symbols that minimizes the mean squared error criterion. As demonstrated in [87, 93] the specific variant of Grover's QSA proposed by Dürr and Høyer in [81] for finding the minimum in a database may be exploited in the MUD application considered. In [87], a particular variant of the quantum mean algorithm proposed by Brassard *et al.* [86] was employed for providing soft estimates of the transmitted symbols, which relies on outputting the probability of a specific symbol transmitted by a particular user, instead of making a hard decision. The resultant soft symbol estimates at the output of an MUD represent the confidence of the MUD that a particular symbol was transmitted, which helps to substantially improve the performance of the subsequent channel decoder, while allowing the base station to perform multiple iterations exchanging soft information between the channel decoder and the MUD, which again improves the overall performance [97]. Still considering the employment of quantum search algorithms in the MUD application area, in [94, 95] the presented solution outperformed the previously proposed soft output quantum-assisted MUD of [87] by using an algorithm relying on an amalgam of the Dürr-Høyer algorithm [81] with classical computing. The same methodologies were also shown to perform well in non-coherent wireless communications systems, where the channel states are not available at the base station [96].

Grover's QSA as well as its variants have also been combined with classical heuristic evolutionary algorithms apart from the GA [89], such as the *Repeated Weighted Boosting Search* (RWBS) [98] and the *Particle Swarm Optimization* (PSO) [99]. To elaborate further, the quantum-assisted version of the RWBS has been employed by Botsinis *et al.* [100] in the field of wireless communications for improving channel estimation [101,102] accuracy. The *Multi-User Transmission* (MUT) process [103,104] may also benefit from quantum-assisted evolutionary algorithms, relying on Grover's QSA, which may be employed in the downlink of a communications system, for preprocessing the transmitted signal for eliminating the multi-user interference at the transmitter. More explicitly, the goal of the multi-user transmission is to "pre-cancel" both the anticipated deleterious effects of the downlink channel and of the multi-user interference by transmitting a carefully preprocessed signal instead of each user's original symbols, so that eventually each user will flawlessly receive the intended legitimate symbol. This preprocessing is performed based on a perturbation vector applied to the information symbol vector, which is found by minimizing a predetermined criterion, such as the mean squared error anticipated at the output of each downlink receiver. In this context, Botsinis *et al.* [105] proposed a quantum-assisted PSO capable of evaluating the optimal complex values for this specific perturbation vector in the context of the downlink of rank-deficient *Non-Orthogonal Multiple Access* (NOMA) systems [106].

The cluster analysis technique [107] may also be improved by Grover's QSA, which may be used for the vector quantization of the channel in wireless communications, where these channel codebooks have to be shared between the base station and the mobile users, so that the latter can feed back to the base station the estimated and quantized channel states [108]. Clustering is also used in unsupervised machine learning used for data mining [109], as well as for diverse other applications, such as Gait Recognition [110, 111]. With the aid of Grover's QSA, a quantum-assisted *k-means* or *k-median* algorithm has been developed [112,113] for performing low-complexity clustering, while benefiting from the reduced complexity of Grover's QSA. Furthermore, Grover's QSA may also be used in the field of image compression, as proposed in [114] for achieving low-complexity fractal image compression, while removing the requirement of pre-processing tools, as well as for feature extraction from medical images [115]. Still in the context of feature extraction, a quantum-assisted version of the classical *PageRank* algorithm [116,117], which ranks the web pages based on their popularity, has been proposed by Paparo and Martin-Delgado [118]. This



Dürr, Heiligman, Høyer and Mhalla [122] introduced a variant of the DHA, which identified the k-lowest values of a database, for the sake of determining as to whether a graph is connected.

Malossini, Blanzieri and Calarco [89] proposed the *Quantum Genetic Optimization Algorithm* (QGOA), which is fundamentally a classical genetic algorithm, but with its mating process enhanced by the DHA.

Paparo and Martin-Delgado [118] conceived a quantum-assisted version of the classical *PageRank* algorithm, which ranks the web pages based on their popularity.

Aïmeur, Brassard and Gambs [112] employed Grover's QSA for the sake of performing both *k-means* and *k-median* clustering.

Botsinis, Alanis, Babar, Ng and Hanzo [94] designed a *Quantum-assisted Multi-User Detector* (QMUD), which is empowered by the DHA and it is capable of performing low-complexity *Soft-Input Soft-Output Multi-User Detection* (SISO-MUD) in rank-deficient wireless systems.

Du, Yan and Ma [114] employed Grover's QSA for achieving low-complexity fractal image compression.

Botsinis, Alanis, Babar, Ng and Hanzo [100] conceived the *Quantum Repeated Weighted Boosting Search* (QRWBS), which is empowered by the DHA and it is tailored for joint channel and data estimation in rank-deficient wireless systems.

Botsinis, Alanis, Babar, Ng and Hanzo [105] propose a quantum-assisted version of the *Particle Swarm Optimization* (PSO) designed for *Multi-User Transmission* (MUT) in rank-deficient *Non-Orthogonal Multiple Access* (NOMA) systems.

Lloyd, Garnerone and Zanardi [120] applied the *Quantum Phase Algorithm* (QPA) to machine learning problems either for the sake of finding *Betti numbers* [121], which correspond to the numbers of connected components, holes and voids, as well as for the finding eigenvectors and eigenvalues of the combinatorial Laplacian.

Figure 1.5: Timeline of the applications of quantum search algorithms.

specific algorithm has been then been also deployed by Paparo *et al.* [119] in the context of complex networks. It has been found that it takes into account the specific significance than its classical counterpart. Additionally, Lloyd *et al.* [120] applied the QPA to machine learning problems either for the sake of finding *Betti numbers* [121], which correspond to the numbers of connected components, holes and voids, as well as for the finding eigenvectors and eigenvalues of the combinatorial Laplacian.

Finally, Dürr *et al.* [122] proposed a variant of Grover's QSA, which is capable of finding the k lowest values of a function, which assumes N legitimate entries, by requiring a complexity as low as $O(\sqrt{kN})$. This quantum algorithm has a complexity $\sqrt{k}$ times lower than that exhibited by employing the DHA, which finds the minimum of a database, k different times. Quantitatively, this would result in a complexity of $O(k\sqrt{N})$. The authors of [122] suggested the employment of this particular algorithm for finding whether a graph of N vertices is connected for reducing the solution's complexity.

Some of the most important applications of quantum algorithms are summarized in the timeline of Fig. 1.5. Based on the above discussions, we may summarize that quantum computing offers a well-equipped parallel-processing toolbox for rendering NP-hard problems tractable, such as the multi-objective routing problem. *For the sake of benefiting from this complexity reduction offered by the quantum algorithms, we will stipulate the fundamental assumption that the cluster heads of Figs. 1.1 and 1.2 are in possession of a quantum computer.* However, the deployment of a quantum computer may be rather costly and hence such a deployment may be either justified in systems with an excessive number of heterogeneous nodes, such as airports, or even *aeronautical networks* [123].

1.4 Contributions

The classical computing methods mentioned in the previous section are suboptimal. To elaborate further, not only they fail to spot all the paths that belong to the *Optimal Pareto Front* (OPF) [124], but they also often mistakenly identify others which are not optimal, since they may converge to local minima. In Chapter 4 we propose a QSA, which addresses these problems and at the same time exploit the property of QP, resulting in a beneficial complexity reduction. In particular, our contributions in this specific chapter, which is based on our contribution in [1], may be summarized as follows:

1) *We have proposed a novel quantum-assisted algorithm, namely the Non-dominated Quantum Optimization (NDQO) algorithm, which optimizes the multi-objective routing problem using the exponential search algorithm of [80]. We have also improved the latter algorithm and derived the NDQO algorithm's complexity upper- and lower-bounds by taking into consideration the number of classical cost function evaluations (CFE) invoked by this algorithm.*

2) *We have addressed the OPF partial generation, by ensuring that all the optimal legitimate routes will be identified, despite imposing a reduced complexity.*

3) *We have characterized the performance versus complexity of the NDQO algorithm and have demonstrated that it achieves the optimal performance at a complexity, which is on the order of $O(N^{3/2})$ in the worst-case scenario.*

The NDQO algorithm presented in Chapter 4 - albeit near-optimal, when compared to the full-search-based method - offers fruitful ground for improvement, since it is classified as a "Hybrid Quantum-Serial Processing" algorithm. This suggests that a further reduction of its complexity is possible by exploiting the potential synergies between the QP and the HP. Furthermore, one of its features is that its complexity is on the order of $O(N)$ [1]. Whilst this complexity increase is indeed much more moderate than the exponentially escalating *Maximum Likelihood* (ML) complexity, this linear complexity increase may become prohibitive in the context of the NDQO algorithm for high dimensionality problems, where the total number of routes is excessively high [29]. Therefore, with the goal of achieving a further complexity reduction, we propose in Chapter 5 a novel quantum-assisted algorithm, namely the *Non-Dominated Quantum Iterative Optimization* (NDQIO) algorithm. Our contributions in this specific chapter, which is based on our contribution in [2], related to the latter algorithm may be summarized as follows:

1) *We have developed a novel framework both for combining quantum unitary operators and for activating them in parallel, with the goal of achieving a further reduction in the complexity by exploiting the synergies between QP and HP.*

2) *The proposed NDQIO algorithm exploits this parallelism with the aid of the algorithms of [80] and [81] for finding the optimum of a multi-objective routing problem in WMHNs. We have also derived the algorithm's upper and lower complexity bounds.*

3) *We have further reduced the complexity of the NDQO algorithm by introducing the novel element of elitism, which allows the NDQIO algorithm to be terminated once it concludes that the entire OPF has been identified.*

4) *We have characterized the performance versus complexity of the NDQIO algorithm and have demonstrated that it achieves the full-search-based optimal performance at a normalized complexity, which is several orders of magnitude lower than that of the NDQO algorithm.*

In Chapter 6, we will extend our network model for the sake of addressing the joint multi-objective routing and load balancing problem of socially-aware networks. In this specific problem, the NDQIO algorithm employed in socially-oblivious networks provides us with some clear design guidelines. Explicitly, the hybrid framework exploiting the synergy between the QP and the HP provides a substantial complexity reduction, namely by a factor of $O(K\sqrt{N})$ [2], where the factor K corresponds to the number of parallel independent quantum processes stemming from the HP, while N is the database size. Nevertheless, as Zalka [88] pointed out, Grover's QSA and inherently all the Grover-based QSAs, such as the BBHT-QSA, the DHA and the NDQIO algorithm, are optimal in terms of their complexity reduction, as long as the database entries are uncorrelated. Naturally, Zalka's proof of Grover's QSA optimality provides us with a further design consideration, namely the *database correlation exploitation*, as portrayed in Fig. 6.1. We note that the actual complexity reduction offered by the database correlation exploitation strictly depends on the optimization problem, thus its achievable complexity reduction order cannot be quantified using a closed-form expression. Nevertheless, we can view this method as a means of extracting a series of shorter uncorrelated databases from the original database, thus, effectively reducing the database length N. Explicitly, this transfomation pushes the complexity reduction offered by the hybrid HP and QP framework to its full potential. Based on these design considerations, our contributions in this specific chapter, which is based on our contribution in [3], related to the MODQO algorithm may be summarized as follows:

1) *We propose a quantum-assisted multi-objective optimization algorithm, namely the Mutli-Objective Decomposition-based Quantum Optimization (MODQO) algorithm, which relies on a novel optimal decomposition framework for jointly optimizing the routing and performing load balancing in socially-aware twin-layered networks.*

2) *We develop a novel framework for decomposing the joint multi-objective routing and load balancing problem into a series of low-complexity sub-problems and we prove*

that the Optimal Pareto Front of the decomposed problem is identical to the Optimal Pareto Front of its composite counterpart.

3) *We propose a new metric for characterizing the distribution of the tele-traffic load, namely the normalized entropy $\bar{H}$ of the respective distribution, which circumvents the biasing towards the minimum delay solution imposed by relying on the use of the standard deviation of the respective distribution.*

4) *We analytically characterize the complexity imposed by the MODQO algorithm, which is on the order of $O(\sqrt{N})$ and $O(N_{MR}^{2N_{MC}^2})$ for networks having N_{MR} routers and N_{MC} users in the best-case and the worst-case scenarios, respectively, down from $O(N^{2N_{MC}^2})$ imposed by the exhaustive search, with $N \gg N_{MR}$ being the total number of Hamiltonian routes between two specific users. Additionally, we demonstrate that the average complexity of the MODQO algorithm is multiple orders of magnitude lower than that of the exhaustive search, when considering realistic network sizes.*

5) *Finally, we compare the MODQO accuracy to that of the* Non-dominated Sort Genetic Algorithm II *(NSGA-II) [124] operating at an identical computational complexity and demonstrate that the MODQO algorithm is capable of improving both the delay and power consumption by at least that of two-hop durations and at least 4 dB, respectively, for networks having 10 mesh routers.*

1.5 Report Structure

The rest of this report is structured as follows:

(a) In **Chapter 2** we will first introduce all the assumptions made about our single-source, single-destination network model. In particular, firstly we will discuss the WMHN specifications leading to the specific choice of our optimization objectives in Section 2.2. Then, in Section 2.3 we will provide a brief introduction to the concept of Pareto optimality. Subsequently, we will present the state-of-the-art evolutionary algorithms, namely the NSGA-II [124, 29] and the MO-ACO algorithm [54, 52] in Sections 2.4 and 2.5, respectively. We note that these specific algorithms will be used to benchmark our proposed algorithms.

(b) In **Chapter 3** we will first provide an introduction to the quantum computing postulates presented in Section 3.2. Subsequently, three popular quantum algorithms, namely Grover's QSA, the exponential search algorithm of [80] and a quantum algorithm proposed in [81] for identifying the minimum of an unsorted database, will be discussed in Sections 3.3, 3.4 and 3.5, respectively. Note that these specific algorithms constitute the cornerstones of the algorithms advocated in the next chapters.

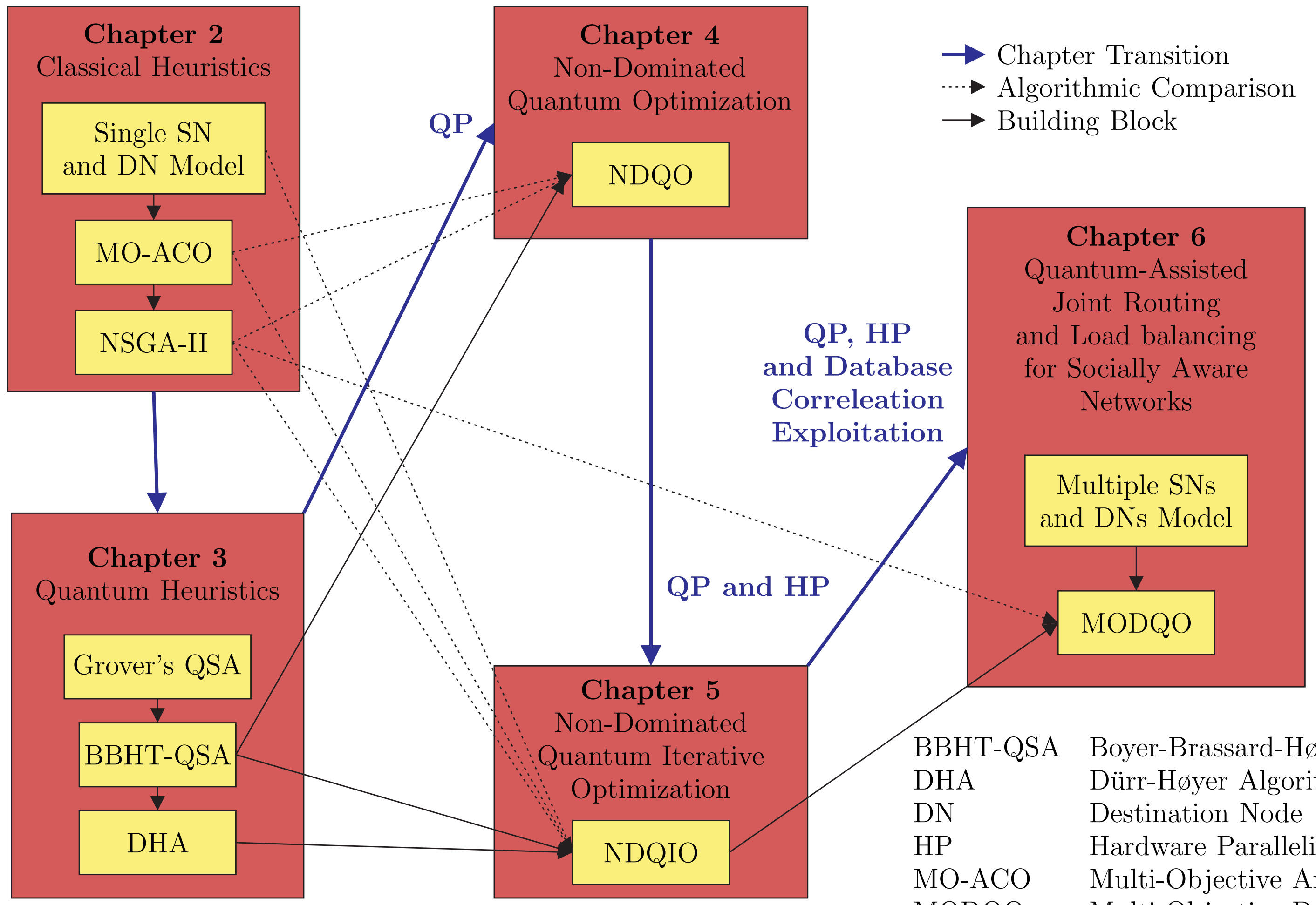
(c) In **Chapter 4**, we will present our first proposed solution, namely the so-called *Non-Dominated Quantum Optimization* (NDQO) algorithm [1], for addressing the multi-objective routing problem, when considering the system model of Section 2.2. For this specific reason we will utilize the BBHT-QSA presented in Section 3.4 for the sake of conceiving a quantum-assisted process searching for “more beneficial” routes, thus finding the Pareto-optimal routes, whilst benefiting from the complexity reduction offered by the QP. This specific process is presented in Section 4.3 and it is referred to as the *BBHT-QSA chain*. This process constitutes the cornerstone of the NDQO algorithm. Subsequently, the NDQO algorithm will be evaluated in terms of its accuracy versus complexity trade-off. For the accuracy evaluation, we will utilize the NSGA-II and the MO-ACO algorithm, presented in Sections 2.4 and 2.5, respectively, as benchmarkers.

(d) In **Chapter 5**, we will propose in Section 5.2 a novel hybrid framework amalgamating both QP and the so-called *Hardware Parallelism* (HP) for attaining a further complexity reduction. Subsequently, we will apply this framework to the BBHT-QSA chain processes of Section 4.3 and propose the so-called *Non-Dominated Iterative Quantum Optimization* (NDQIO) algorithm [2] in Section 5.3 for addressing the multi-objective routing problem, when considering the system model of Section 2.2. Explicitly, the NDQIO algorithm benefits both from the aforementioned hybrid framework as well as from a process, which allows the algorithm to conclude as to whether it has or has not identified all of the optimal routes, as detailed in Section 5.3.2. Subsequently, the NDQIO algorithm will be characterized in terms of its accuracy versus complexity. For the accuracy evaluation, we will utilize the NDQO algorithm, the NSGA-II and the MO-ACO algorithm, presented in Sections 4.4, 2.4 and 2.5, respectively, as our benchmarkers. Note that the complexity imposed by the NDQIO algorithm will be compared to that of the NDQO as well.

(e) In **Chapter 6**, we address joint multi-objective routing and load balancing problem in the context of socially-aware networks. Since the system model of Section 2.2 is not directly applicable in this specific scenario, we will appropriately adapt it in Section 6.2 in order to account both for multiple SNs as well as DNs and introduce a socially-aware load balancing metric as a secondary objective, namely the *normalized entropy of the normalized composite betweeness.* in Section 6.3.2 we will then propose our novel algorithm, namely the *Multi-Objective Decomposition Quantum Optimization* (MODQO) algorithm [3], which exploits both the NDQIO’s hybrid framework as well as the database correlation exhibited by this scenario, as detailed in Section 6.3.4. The complexity imposed by the MODQO algorithm will be evaluated in Section 6.4.1, while its heuristic accuracy will be compared to that of the NSGA-II of Section 2.4.

(f) Finally, in **Chapter 7** our conclusions and future research will be presented.

Finally, the structure of this report is portrayed in Fig. 1.6.


Chapter 2
Classical Heuristics
Single SN and DN Model
MO-ACO
NSGA-II
QP
Chapter 4
Non-Dominated Quantum Optimization
NDQO
Chapter Transition
Algorithmic Comparison
Building Block
Chapter 6
Quantum-Assisted Joint Routing and Load balancing for Socially Aware Networks
Multiple SNs and DNs Model
MODQO
QP, HP and Database Correleation Exploitation
QP and HP
Chapter 3
Quantum Heuristics
Grover's QSA
BBHT-QSA
DHA
Chapter 5
Non-Dominated Quantum Iterative Optimization
NDQIO

BBHT-QSA	Boyer-Brassard-Høyer-Tapp QSA
DHA	Dürr-Høyer Algorithm
DN	Destination Node
HP	Hardware Parallelism
MO-ACO	Multi-Objective Ant Colony Optimization
MODQO	Multi-Objective Decomposition Quantum Optimization
NDQIO	Non-Dominanted Quantum Iterative Optimization
NDQO	Non-Dominanted Quantum Optimization
NSGA-II	Non-dominated Sort Genetic Algorithm II
QP	Quantum Parallelism
QSA	Quantum Search Algorithm
SN	Source Node

Figure 1.6: The structure of this report.

Chapter 2

Classical Heuristics for Multi-Objective Routing

2.1 Introduction

As mentioned in the introductory chapter, in Section 2.2 we will first introduce the network model considered for multi-objective routing optimization in WMHNs. In this specific network model, we are assuming the presence of a single Source Node (SN) and single Destination Node (DN) for the sake of assessing both the accuracy and the complexity of the quantum-assisted solutions advocated in Chapters 4 and 5. By contrast, in Chapter 6 we will extend this specific model for the sake of employing it in the context of multi-objective routing in socially-aware networks. Subsequently, in Section 2.3, we will elaborate on the concept of Pareto optimality [53], because all of our advocated quantum-assisted solutions rely on it for the joint optimization of all the QoS criteria considered.

As it has been made plausible in the introductory chapter, the vast majority of classical computing contributions in the field of multi-objective routing relied on a pair of popular particular algorithms, namely the *Non-dominated Sort Genetic Algorithm - II* (NSGA-II) [124,29] and the *Multi-Objective Ant Colony Optimization* (MO-ACO) algorithm [54,55,52]. Therefore, we will describe in detail both the aforementioned algorithms in Sections 2.4 and 2.5, respectively, and quantify their complexity for the sake of using them as benchmarkers of our novel quantum-assisted algorithms. We note that the specific version of the NSGA-II used is identical to the one proposed by Yetgin *et al.* in [29]. By contrast, the MO-ACO had to be appropriately adapted for its employment in the multi-objective routing problem inspired by [52] and [125]. Finally, we note that in this specific chapter we do not claim any new contributions, since the aforementioned algorithms already exist in the literature. Let us now proceed with the detailed portrayal of the system model considered for the single-SN and single-DN multi-objective routing in WMHNs.

2.2 Network Specifications

2.2.1 Model Assumptions

In this subsection we will elaborate on the underlying assumption of the WMHN model considered. Let us now list these particular assumptions as follows:

- As far as the network architecture is concerned, a fully inter-connected network has been assumed. The coverage area was assumed to be a (100×100) m^2 square block.
- The relay node (RN) locations were generated using a uniform random distribution within this area, whereas the SN and the DN were located at the two opposite corners of this square block.
- Moreover, each route is assumed to traverse through each node at most once. Therefore, unnecessary loops, which would result in potentially excessive delay and power consumption, may be avoided.
- As for the interference level experienced by each node, owing to the *Central Limit Therorem* (CLT) [126] the interference caused by multiple users accessing the same channel may be treated as *Additive White Gaussian Noise* (AWGN). More specifically, each node's interference level obeys the normal distribution. The mean of this was set to -90 dBm with a standard deviation of 10 dB.
- The DN acts as an intelligent central node collecting information concerning both the nodes' geo-locations and the interference levels experienced by them. For the sake of simplicity, we have assumed that the DN node has perfect knowledge of the aforementioned parameters. For the sake of employing the quantum algorithms of Chapters 4 and 5, we assume furthermore that this node has access to a *gate-based* quantum computer [62]. This could be realized either by having this specific type of quantum computer installed at the DN side or by the DN having a seamless link to this quantum computer.
- As for the optimization process, the DN collects the network's information at a frequency depending on the RNs' velocity as well as on how fast the interference changes. It then performs multi-objective routing optimization based on this updated information. Let us emphasize that in the context of this treatise we have not considered the frequency of invoking this tracking process. By contrast, we only considered random snapshots of this network.

The rest of the system parameters are summarized in Table 2.1. An illustrative example of the topology considered is shown in Fig. 2.1, where the formation of all the legitimate routes is shown for an 8-node WMHN. We note that although the SN is capable of transmitting its message towards all nodes, it is not possible for the other nodes to send their message back to the SN, due to the previous assumptions. Therefore, the opposite of this principle would apply for the DN.

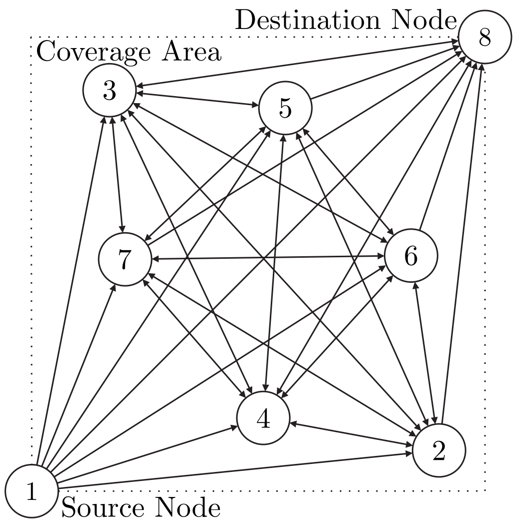


Figure 2.1: Exemplified network topology for an 8-node WMHN.

Table 2.1: WMHN Network Parameters

Network Coverage Area	(100×100) m^2 Square Block
Modulation	QPSK
Mean Node Interference, μ_I	-90 dBm
Node Interference Std, σ_I	10 dB
Power Consumption Model	Log-Distance Path-Loss
	Model with $a = 3$
Transmission Power	20 dBm
Carrier Frequency	1.8 GHz
Delay Unit	Single Hop

2.2.2 Optimization Criteria

According to the physical layer model considered, each message transmission uses QPSK modulation over an uncorrelated Rayleigh fading environment [11], where the *Bit Error Ratio* (BER), P_e, versus the *Bit-to-Noise power Ratio*, E_b/N_0, relationship is given by [126]:

$$P_e = \frac{1}{2}\left(1 - \sqrt{\frac{E_b/N_0}{E_b/N_0 + 1}}\right). \tag{2.1}$$

In the multiple access layer, each RN is capable of retransmitting the received messages using the classic *decode-and-forward* (DAF) scheme [127]. More specifically, each node decodes the received messages and then performs encoding and modulation in order to forward it to the next node. As the received message may be corrupted by errors due to erroneous detection at the previous nodes and also since QPSK is used, the channel can be modelled by a two-stage *Binary Symmetric Channel* (BSC) [11] as presented in Fig.

2.2. The RN corresponds to the intermediate node of Fig. 2.2 and the route has two BER values, one for each of the two links established. It is possible to transform this channel into a single-stage one, which would be described by a single overall BER $P_{e,12}$, given by:

$$P_{e,12} = P_{e,1} + P_{e,2} - 2P_{e,1}P_{e,2}. \tag{2.2}$$

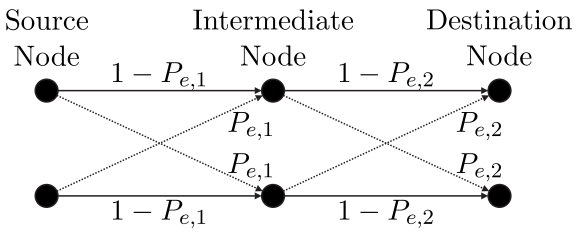


Figure 2.2: Two stage Binary Symmetric Channel for the case of an intermediate node.

The last term in Eq. (2.2) corresponds to the propagated errors, i.e. to the errors that were introduced by the first link and have been luckily "corrected" by the introduction of another error within the second link. Additionally, Eq. (2.2) may be used for recursively calculating the overall BER of a particular route.

Furthermore, another factor to be considered is the network's delay. In the proposed system the delays are introduced by the DAF scheme, since a finite time-duration would be required for a RN to perform all the necessary operations before forwarding a message. For simplicity, the service queue is assumed to have zero length, hence the messages would be forwarded almost instantly with a short delay equal to the DAF signal processing operation duration. Hence, the total delay of the route would be proportional to its number of established hops.

Moreover, as far as the power consumption is concerned, only the free-space path-loss of each link has been considered. In particular, the *path loss exponent* was set to $\alpha = 3$. Hence, the path-loss L of single link may be formulated as [11]:

$$L = P_{\text{Tx,dB}} - P_{\text{Rx,dB}} = 10\alpha \log_{10}\left(\frac{4\pi d}{\lambda_c}\right) \text{ [dB]}, \tag{2.3}$$

where $P_{\text{Tx,dB}}$, $P_{\text{Rx,dB}}$, d and λ_c stand for the transmitted power, the received power, the distance between the nodes of a link and the carrier wavelength respectively. The transmission power was set to 20 dBm for each link, while the carrier frequency was set to 1.8 GHz, which would result in a wavelength of 0.1667 m.

Last but not least, bearing in mind all the above assumptions, an N-node network may be modeled as a graph $G(E, V)$ having E edges and V vertices, which is formulated as:

$$v_{i,j} = \begin{cases} [\mathrm{SNR}_{i,j}, L_{i,j}, D_{i.j}] & \forall i,j \in V : i \neq j, \\ & j \neq 1, i \neq N, \\ \varnothing & \text{otherwise}, \end{cases} \tag{2.4}$$

where $v_{i.j}$ stands for the transition weight, $\mathrm{SNR}_{i,j}$ is the received SNR, $L_{i,j}$ denotes the power losses due to path-loss and $D_{i.j}$ represents the delay of forwarding a message from node i to node j. Additionally, the symbol $\varnothing$ in Eq. (2.4) corresponds to a transition that is not legitimate. Moving on to the route fitness evaluation, assuming a legitimate route x, which belongs to the set of all the possible legitimate routes S, its *Utility Vector* (UV) $\mathbf{f}(x)$ is described by a vector formulated as:

$$\mathbf{f}(x) = [P_{e,x},\ CL_x,\ CD_x]\,, \tag{2.5}$$

where $P_{e,x}$, CL_x, CD_x stand for the overall BER, the cumulative linear-domain sum of the path-losses and the cumulative sum of the delays for route x respectively. We note that each of the single components of the UV for Eq. (2.5) is often referred to as *Utility Function* (UF). Finally, we remind that the system parameters considered are summarized in Table 2.1.

2.3 Pareto Optimality

Since the proposed approach is a multi-objective one based on Eq. (2.5), the optimality of the route-solution vectors should be defined. Explicitly, the evaluation of the UV used for quantifying the performance of the WMHN network considered in Eq. (2.5) can be undertaken with the aid of the *Pareto Dominance* concept [47], which is encapsulated in Definitions 1 and 2, while the corresponding *Pareto-optimality* conditions are given in Definitions 3 and 4. We note that Definitions 1 and 2 are tailored for jointly minimizing the UFs, since our design objective is to establish the active routes with the minimum possible average delay and power consumption. Nevertheless, they can be readily invoked for maximization problems by substituting the “less (than)” operators by “greater (than)”.

Definition 1. Weak Pareto Dominance *[47]: A particular solution x_1, associated with the UV $\mathbf{f}(x_1) = [f_1(x_1), ..., f_K(x_1)]$, where K corresponds to the number of optimization objectives, is said to weakly dominate another solution x_2, associated with the UV $\mathbf{f}(x_2) = [f_1(x_2), ..., f_n(x_2)]$, iff $\mathbf{f}(x_1) \succeq \mathbf{f}(x_2)$, i.e. we have $f_i(x_1) \leq f_i(x_2)\ \forall i \in \{1, ..., K\}$ and $\exists j \in \{1, ..., K\}$ such that $f_j(x_1) < f_j(x_2)$.*

Definition 2. Strong Pareto Dominance *[47]: A particular solution x_1, associated with the UV $\mathbf{f}(x_1) = [f_1(x_1), ..., f_K(x_1)]$, where K corresponds to the number of optimization objectives, is said to strongly dominate another solution x_2, associated with the UV $\mathbf{f}(x_2) = [f_1(x_2), ..., f_K(x_2)]$, iff $\mathbf{f}(x_1) \succ \mathbf{f}(x_2)$, i.e. we have $f_i(x_1) < f_i(x_2)\ \forall i \in \{1, ..., K\}$.*

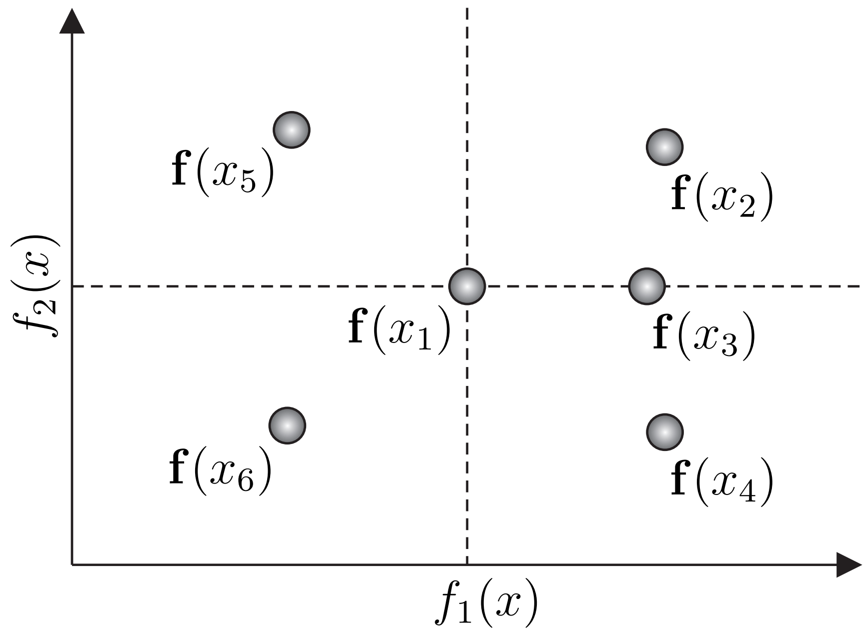


Figure 2.3: A graphical example portraying the dominance relationship of the route-solutions $\{x_i\}_{i=2}^{6}$ with respect to the route-solution x_1 for a generic Pareto-optimality problem with $K = 2$ UFs.

Let us now describe the use of the weak and strong Pareto dominance, encapsulated in Defs. 1 and 2, respectively, with the aid of a graphical example, which is shown in Fig. 2.3. We note that the example presented in Fig. 2.3 portrays a Pareto-optimality problem associated with $K = 2$ UFs. As a reference, we use the route-solution x_1, which is associated with the UV $\mathbf{f}(x_1)$, and we will assess its dominance relationship with respect to the route-solutions $\{x_i\}_{i=2}^{6}$, which are associated with the UVs $\{\mathbf{f}(x_i)\}_{i=2}^{6}$. It is clear that the route-solution x_2 is both strongly and weakly dominated by the route-solution x_1, since we have $f_i(x_1) < f_i(x_2)$, $\forall i \in \{1, 2\}$. The difference between strong and weak Pareto dominance becomes visible, when we assess the dominance relationship between the route-solution x_3 and that of x_1. Since we have $f_1(x_1) = f_1(x_3)$, there is no strong dominance relationship between those two route-solutions; however, x_1 weakly dominates x_3, since we have $f_2(x_1) < f_2(x_3)$. Explicitly, the solution x_1 weakly dominates all the potential route-solutions that lie in the plane defined by the boundaries $[f_1(x_1), +\infty]$ and $[f_2(x_1), +\infty]$ including the boundaries, while it does not strongly the boundaries. As for the rest of the route-solutions shown in Fig. 2.3, there is no dominance relationship among x_1, x_4 and x_5, since we have $f_1(x_5) < f_1(x_1) < f_1(x_4)$, while $f_2(x_5) > f_2(x_1) > f_2(x_4)$. Finally, the route-solution x_6 dominates, both weakly and strongly, the route-solution x_1, since we jointly have $f_1(x_6) < f_1(x_1)$ and $f_2(x_6) < f_2(x_1)$.

Definition 3. Weak Pareto-optimality *[47]: A particular solution x_i, associated with the UV $\mathbf{f}(x_i) = [f_1(x_i), ..., f_N(x_i)]$, where N corresponds to the number of optimization objectives, is considered as weakly Pareto-optimal iff there exists no solution that strongly dominates x_i, i.e. iff $\nexists x_j$ such that $\mathbf{f}(x_j) \succ \mathbf{f}(x_i)$.*

Definition 4. Strong Pareto-optimality *[47]: A particular solution x_i, associated with* *the UV* $\mathbf{f}(x_i) = [f_1(x_i), ..., f_N(x_i)]$*, where* N *corresponds to the number of optimization objectives, is considered as strongly Pareto-optimal iff there exists no solution that weakly dominates* x_i*, i.e. iff* $\nexists x_j$ *such that* $\mathbf{f}(x_j) \succeq \mathbf{f}(x_i)$*.*

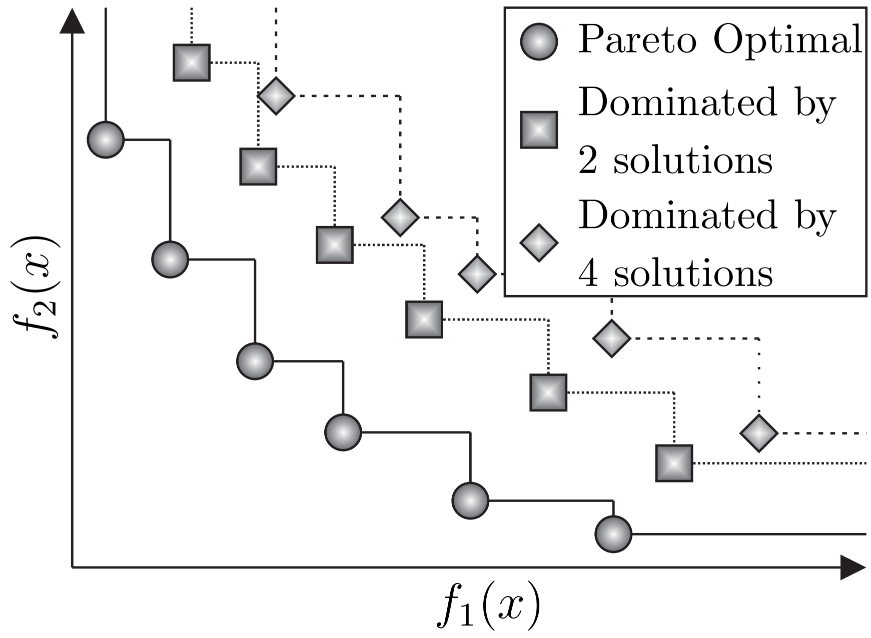


Figure 2.4: Distribution of the route-solutions into fronts based on the number of route-solutions that dominate a specific route-solution for a generic Pareto-optimality problem with $K = 2$ UFs.

Based on Definitions 1 and 2, it is possible to group the route-solutions based on the number of route-solutions that dominate them. Such groups of route-solutions form fronts in the solution space, which are often referred to as *Pareto Fronts* (PF). Naturally, the entire set of Pareto-optimal route-solutions will form a single PF, since all of these route-solutions share the characteristic of being either strongly or weakly dominated by no route-solution, based on Definitions 3 and 4,respectively. We note that this specific front is often referred to as the *Optimal Pareto Front* (OPF). A graphical example of the formation of various PFs is presented in Fig. 2.4 for a generic minimization Pareto-optimality problem associated with $K = 2$ UFs. Still referring to the same figure, observe that the curves formed by the route-solutions of a specific PF dominate the respective curves associated with higher rank PFs, i.e. PFs formed by route-solutions that are dominated by a higher number of route-solutions.

Explicitly, the OPF is composed by route-solutions having UFs, which cannot be further optimized individually without degrading the fitness of the rest of the UFs, as it can be observed in Fig. 2.4. As far as our specific application is concerned, when considering weak OPFs, there exist route-solutions classified as Pareto-optimal, which may have the same metric, say in terms of their delay CD, yet exhibiting a worse performance in terms both of their power consumption CL and of their BER P_e. Nevertheless, the route-solution that jointly exhibits a lower power consumption and a lower BER seems to outperform the other one, since it jointly minimizes all of the UFs considered in Eq. (2.5). This specific caveat is rectified by the employment of strong Pareto-optimality, since the specific route-solution associated with a lower power consumption would dominate the other route-solution and, hence, the latter will not be included in the respective OPF. In the context of this treatise, we will consider weak Pareto-optimality for the design of our quantum-assisted solutions in Chapters 4 and 5, since it constitutes a more generic and, thus, a more challenging case in terms of its complexity owing to the increased number of Pareto optimal routes. By contrast, in Chapter 6 we will utilize the strong Pareto-optimality, since our design objective is to identify the specific solutions, where it is not possible to improve any of the single UFs without degrading the rest of the UFs considered.

Based on Definitions 3 and 4 related to Pareto-optimality, it is possible to quantify the fitness of a specific route with respect to its Pareto-optimality. Explicitly, this specific fitness is quantified using the concept of *Pareto Distance*, which is encapsulated in Definition 5.

Definition 5. Pareto Distance. *Given a set of route-solutions S and a particular route-solution x_i, belonging to the set $x_i \in S$, its distance from the OPF may be defined as the probability P_d of being dominated by the other solutions of S. This is formally expressed as:*

$$P_d(x_i) = \frac{\#\{\mathbf{f}(x_j) \Diamond \mathbf{f}(x_i), \forall j,\ i \in \{0, 1, ..., |S|-1\}\}}{|S|}, \tag{2.6}$$

where the operator $\#\{\cdot\}$ quantifies the number of times that the condition in the curly brackets is satisfied, while the operator $|\cdot|$ represents the total number of elements of a set and $\mathbf{f}(\cdot)$ is the UV defined in Eq. (2.5). Additionally, the operator $\Diamond$ represents the generic Pareto dominance comparison operator corresponding to the operators $\succeq$ and $\succ$ for strong and weak Pareto-optimality problems, respectively.

Based on Definition 5, the Pareto Distance function $P_d(\ )$ is limited to the range $[0, 1]$. Naturally, the routes belonging to the OPF will have the minimum possible distance, which is equal to 0, whereas the sub-optimal routes would exhibit higher distances. Therefore, the optimization problem takes the form of:

$$\begin{aligned} \text{find} \quad & x \\ \text{s. t.} \quad & x \in S,\ P_d(x) = 0. \end{aligned} \tag{2.7}$$

Finally, let us define the complexity metric invoked for the quantification of the computational complexity. Since the calculation of the Pareto Distance in Eq. (2.6) requires the evaluation of the dominance operator ($\succeq$), we will define the *Cost Function* (CF) of our optimization problem as a single application of this operator between two solution vectors. The calculation of each route-solution vector requires a single UF evaluation based on Eq. (2.5).

Let us now apply the principles of *Pareto Optimality* to our WMHN model. Based on the definition of the route-solution vector in Eq. (2.5), we attempt to jointly optimize the performance of our WMHN in terms of the time delay, the BER and the energy dissipation. Since these performance metrics conflict with each other, the OPF consists of each performance metric's global minimum and the specific routes corresponding to route solution vectors, which lie in the space defined by the global minima and they are not dominated by any other route-solution vector. Our main interest lies in determining the latter solutions, rather than the three global minima. For instance, the optimal route in terms of the time delay would be the direct route from the SN to the DN without traversing through any RNs. However, this link would potentially suffer from an excessive power dissipation, since the distance between the SN and the DN may be long, hence leading to a low E_b/N_0 as well. Similar disadvantages may apply in the general case for all the global minima of each parameter.

Furthermore, as mentioned in the previous section, determining all the OPF routes provides us with useful information about the trade-offs amongst the diverse parameters considered [47], hence resulting in a more beneficial design in terms of the various QoS requirements. For this reason, all the legitimate routes have to be examined in terms of their *Pareto Distance* for the sake of identifying those that have a *Pareto Distance* of zero. Assuming that $|S| = N$, where N corresponds to the total number of legitimate routes, the examination of a single route would invoke the dominance operator $N/2$ times on average and N times in the worst-case scenario. The total number of legitimate routes increases exponentially with the number of nodes N_{nodes}, and it is equal to [29]:

$$N = \sum_{i=0}^{N_{\text{nodes}}-2} \frac{(N_{\text{nodes}}-2)!}{(N_{\text{nodes}}-2-i)!}. \tag{2.8}$$

Since this operation is carried out for every legitimate route, the resultant average and maximum brute force (BF) complexity, $L_{\text{BF}}^{\text{avg}}$ and $L_{\text{BF}}^{\text{max}}$ respectively, are equal to :

$$\begin{aligned} L_{\text{BF}}^{\text{avg}} &= N^2/2 &&= O(N^2), \\ L_{\text{BF}}^{\text{max}} &= N^2 &&= O(N^2). \end{aligned} \tag{2.9}$$

Hence, sophisticated search methods are required for determining the OPF in polynomial time. As already mentioned in Section I, this ambitious goal may be achieved with the aid of QSAs.

2.4 The Non-Dominated Sort Genetic Algorithm II

The *Non-Dominated Sort Genetic Algorithm II* (NSGA-II) [124] has been designed for addressing multi-objective problems and it belongs to the family of *Evolutionary Algorithms* (EAs). Naturally, there is an initial population of random candidates, which are also often referred to as *individuals*. They represent distinct route-solutions, which are in turn referred to as *chromosomes.* A *fitness function* is introduced also for multi-objective problems in the same fashion as in single-objective optimization. This operation is carried out by sorting the route-solutions or chromosomes in the PFs using the Pareto optimality principle. Furthermore, for the route-solutions, which belong to the same PF, a fitness function has been introduced, namely the so-called *crowding distance.* Explicitly, the crowding distance quantifies the distance of a specific route-solution from its neighbors and its detailed description will be presented in Section 2.4.1.

Algorithm 2.1 NSGA-II [124]

```
1:  Initialize the population to $P_1$ with $N_{pop}$ random individuals.
2:  for $g = 1$ to $N_G$ do
3:      Perform a non-dominated sort on $P_g$ and output the Pareto Fronts $F$.
4:      Calculate the crowding distance $d$ of each individual.
5:      Identify the $N$ strongest individuals, which form the set $R_g$.
6:      For each offspring individual perform a binary tournament selection from the set $R_g$
        twice, one for each parent, and store the pairs to the set $S_g$
7:      Perform crossover with probability $P_c$ between each pair stored in $S_g$ forming the
        offspring population $C_g$.
8:      Perform mutation with probability $P_m$ for each individual of $C_g$ and repair them,
        storing the repaired mutated individuals in the set $C_g^m$ .
9:      Set $P_{g+1} = P_g \cup C_g^m$.
10: end for
```

Initially, a random population consisting of N_{pop} individuals is sorted based on both its Pareto Front rank as well as its crowding distance and the fittest $N_{pop}/2$ individuals are selected. To elaborate further, an individual is deemed fitter than another if the first has a lower Pareto distance value. If they both exhibit the same value, i.e. they belong to the same PF, the fitter individual is the one that has a higher crowding distance. Subsequently, a pair of independent binary tournament selections [128] are activated for the sake of constructing the *mating pool*; each of these procedures selects exactly $N_{pop}/2$ individuals and includes them into the mating pool, hence creating $N_{pop}/2$ pairs of individuals or chromosomes. The *crossover* genetic operator is then imposed on each of the $N_{pop}/2$ pairs producing $N_{pop}/2$ individuals, which are often referred to as *offspring.* Then the *mutation* operator is applied to each of the individuals comprising the offspring. Note that we will elaborate on the crossover and mutation operators in Section 2.4.3. For the sake of visiting each node at most once, the potential loops are removed from the offspring population. More specifically, should a particular offspring route be found to have visited a node twice or more, this node is removed, hence complying with the Hamiltonian route constraint. Finally, the offspring pool is combined with the parent population and the same process is repeated for a number of generations. The overall process of NSGA-II is formally presented in Alg. 2.1. The processes of crowding distance calculation, binary tournament selection and genetic operators are presented in the next sections. Let us now proceed with analyzing each of the sub-processes of the NSGA-II.

2.4.1 Crowding Distance

The concept of *crowding distance* has been introduced in the NSGA-II as a measure of elitism against its predecessor, namely the NSGA, where the selection between members belonging to the same PF was carried out with the aid of user-defined variables. This metric relies upon the principle that more remotely located route-solutions in the hyperplane defined by the UVs of the all the legitimate route-solutions, i.e. route-solutions that exhibit a higher crowding distance value, have a tendency to produce offspring individuals exhibiting a higher degree of diversity [124].

Algorithm 2.2 Crowding Distance Evaluation Method for a Single PF [124]

```
1: Initialize the vector $I$ containing the crowding distance of each individual in a specific
   PF to a zero vector.
2: for $k \leftarrow 1$ to $K$ do
3:    Sort the individuals according to their $m$-th OF and store their sorted indices to
      vector $i$ and the $m$-th OF sorted values to the vector $D^c_m$.
4:    Set $f_{\max} \leftarrow \max\{D^c_k\}$.
5:    Set $f_{\min} \leftarrow \min\{D^c_k\}$.
6:    Set $I(i_1) \leftarrow \infty$ and $I(i_{|i|}) \leftarrow \infty$, where the operation $|i|$ denotes the length of the vector
      $i$.
7:    for $j \leftarrow 2$ to $|i| - 1$ do
8:       Set $I(i_j) \leftarrow I(i_j) + \frac{D^c_{k,j+1} - D^c_{k,j-1}}{f^{\max}_k - f^{\min}_k}$
9:    end for
10: end for
11: Export the vector $I$ and exit.
```

The process of evaluating the crowding distances of the route-solutions of a specific PF is summarized in Alg. 2.2. Based on Alg. 2.2, for a specific PF the vector I of crowding distances is initialized to zero. Then, for each of the UFs, which are assumed to be K in total, the route-solutions are sorted in ascending order based on the specific UF. We note that in Alg. 2.2 the vector D_k represents the sorted values of the k-th UF, the vector i corresponds to the relative sorting indices, while $f^{\max}_k$ and $f^{\min}_k$ are the maximum as well as minimum observations of the k-th UF, respectively. The method initializes to infinity the crowding distances associated with route-solutions having the minimum and maximum values in terms of the k-th UF. Subsequently, the crowding distance $I_k(i_j)$ of the i_j-th route of the PF in terms of the k^{th} UF is formulated as follows:

$$I_k(i_j) = \frac{D^c_{k,j+1} - D^c_{k,j-1}}{f^{\max}_k - f^{\min}_k}. \tag{2.10}$$

Finally, the overall crowding distance of a specific route-solution is derived by the sum of the crowding distances $\{I_k(i_j)\}_{k=1}^{K}$ of the entire set of UFs. The variables of Eq. (2.10) associated with the crowding distance evaluation are portrayed in Fig. 2.5. Last but not least, we note that dividing the distances by the range of each UF at a specific PF results in the normalization of the crowding distances associated with each of the UFs, thus allowing their summation. Explicitly, Deb *et al.* [124] demonstrated that solutions with higher crowding distance, which may be interpreted as solutions being more distant to their neighbors belonging to the same PF, are capable of producing better solutions using the genetic operators. This could be justified by the fact that the solution diversity is enhanced by those having higher crowding distance.

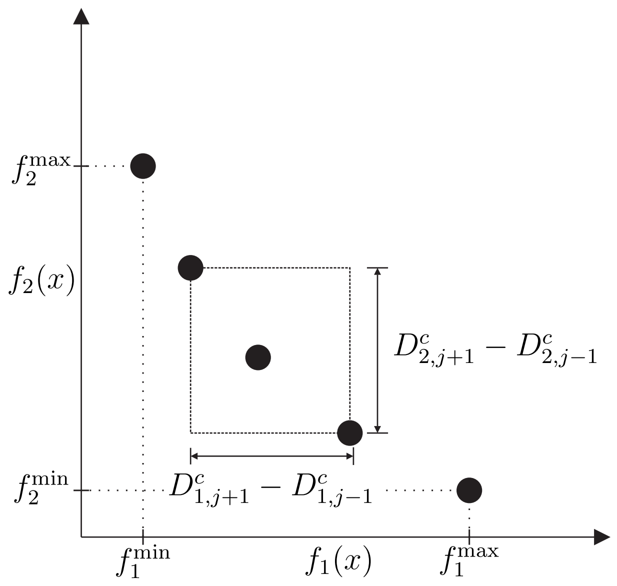


Figure 2.5: Visual representation of the crowding distance evaluation process of Alg. 2.2.

2.4.2 Binary Tournament Selection

The binary tournament selection has been introduced by Chakraborty [128] and it is utilized for the sake of providing the genetic algorithm with immunity against premature convergence. The aim of this procedure is to select $N_{pop}/4$ surviving route-solutions from a population of N_{pop} parent individuals. To elaborate further, two solutions are selected randomly from the entire set of N_{pop} parent solutions and are first compared in terms of their Pareto distance P_d, defined in Eq. (2.6); the surviving route-solution is the one exhibiting the lowest P_d value. If they belong to the same PF, their crowding distances are compared and hence the one associated with the highest crowding distance value is selected as the surviving one. In the extreme case, where they are found to have the same crowding distance value, the surviving solution is chosen randomly. The surviving solution is appended to the parents set S_g. This procedure is then repeated $N_{pop}/4 - 1$ times in order to fill the set of surviving solutions.

In the NSGA-II (Alg. 2.1) this procedure is invoked twice, since the resultant *mating pool* has to have the same size as the initial population, consisting of $N_{pop}/4$ pairs. This *mating pool* will be used as the input to perform the *Genetic Operation Crossover*, which is presented in the next subsection.

2.4.3 Genetic Operators

The *genetic operators* are responsible for creating the offspring at each generation and they are classified into two basic types, namely *crossover* and *mutation* [29]. The first operator creates the offspring individuals from the parent population S_g. Explicitly, the *mating pool*'s population that has been created from the two binary tournament selection processes are used is the parents. Each of the parent chromosomes is then segmented into two parts, as shown in Fig. 2.6(a). The first child would be created by serially concatenating the first part of the first parent and the second part of the second parent. By contrast, the second child is created by the first part of the second parent and the second part of the first parent.

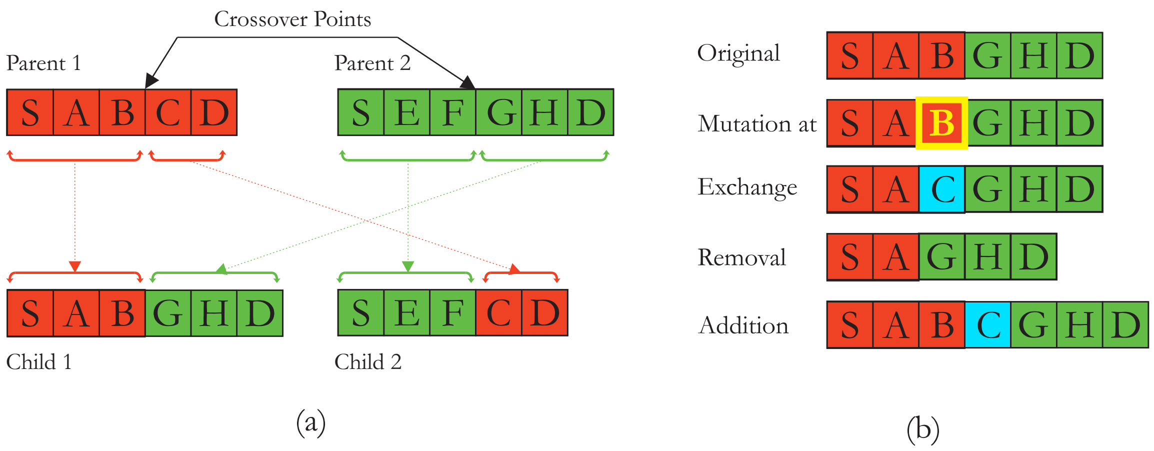


Figure 2.6: The genetic operators (a) *crossover* and (b) *mutation*. Note that the nodes "S" and "D" correspond to the SN and the DN, respectively.

As soon as the crossover operation is completed, the mutation operation is applied to the offspring population created. This operation modifies the created population with the aid of three equiprobable operations: *exchange*, *removal* and *addition*. In all three cases, a random node is selected and one of the aforementioned operations is randomly applied to each of the offspring individuals. To elaborate further, in the exchange operation the selected node is substituted by another one, whereas in the removal operation it is removed from the route. Finally, in the addition operation a new node is inserted right after the selected node. All three processes are shown in Fig. 2.6(b). Since the offspring individual may exhibit having loops after the genetic operators, a repair process, which removes the potential loops, is activated for satisfying the optimization constraints, ensuring that the routes visit the available node at most once.

2.5 Multi-Objective Ant Colony Optimization

The *Ant Colony Optimization* (ACO) is a bio-inspired algorithm, which was conceived by Dorigo *et al.* [54] in 1996, who used a multi-agent approach for simulating the behavior of an ant colony for solving for the *Traveling Salesman Problem* (TSP). To elaborate further, single ants may be viewed as agents, since they exhibit the following characteristics [54,55]:

- Ants have poor-to-mediocre eyesight and some tribes may be considered completely blind.
- They are able to communicate with each other using specific hormones, referred to as "pheromones". These are detected by chemical sensors, which are located on the ants' heads.
- Each ant, in order to find the best route from its nest to the source of food, senses the pheromones and follows the route associated with the highest pheromone intensity.
- Since an open environment is assumed, there is a probability that a perturbation from the environment may distract the ant, which hence does not follow the route having the highest pheromone intensity.
- The choice of route is considered independent for each ant and the pheromone trails they leave are considered for the next generation of ants.

The algorithm simulates the behavior of the individual ants of an ant colony for locating the optimal route from their nest to food. At each stage, which is often referred as *generation*, a certain number of ants sets out from their nest. Initially, in the absence of pheromone, the ants choose their routes randomly, as shown in Fig. 2.7(a); the routes are considered equiprobable. As they move along the routes they deposit pheromone to be utilized by the ants of the next generations. The longer the route the lower the pheromone intensity would be, as it tends to evaporate over time. In the next generations, the ants choose the routes to follow randomly again; however, the probabilities of the routes are no longer equiprobable, they depend on the pheromone intensity. Moreover, perturbations by the environment are possible; this would mean that even though a certain route is chosen, actually a "neighboring" one is followed. Repeating the experiment for a number of generations, the ants converge over time to the optimal route, since the pheromone intensity increases over the generations, as seen in Fig. 2.7(b,c,d).

Dorigo's original algorithm conceived in 1996 was then finalized by Dorigo and Di Caro [55] in 1999. They used the same principles for adapting the algorithm to the problem of single-objective networks' routing optimization, introducing the concept of *intrinsic affinity*, an additional heuristic factor. In the so called *AntNet*, each ant travels from the SN to the DN, while visiting a different number of intermediate nodes. Each ant moving from one node to another leaves an amount of pheromone across the trail, depending on the value of the UF considered for optimization. As for the heuristic factor, the average delay per node has been used; the node transition probability depends both on the pheromone intensity and on the intrinsic affinity.

Some further extensions of the *AntNet* model have also been proposed. Both Golshahi *et al.* [56] and Chandra *et al.* [57] implemented a routing protocol using the *AntNet* model; two types of agents have been used: a forward agent, which searches for the routes, and a

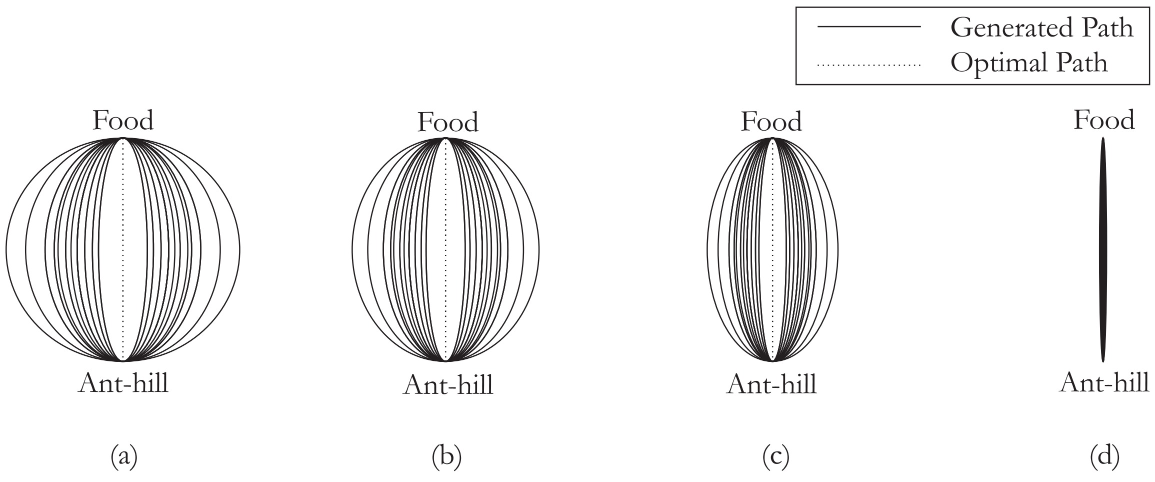


Figure 2.7: The process of converging to the optimal route.

backward agent, which performs the ACO operations. Finally, Wang and Wu [58] developed an ACO routing algorithm for optimizing the performance of fault tolerant *Hypercube Networks* at a reduced complexity by exploiting the regularities that these networks exhibit.

The above versions of the *Ant Colony optimization* were used for single-objective optimization. However, the optimization approach adopted throughout this report is a multi-objective one. Several variations of the Multi-Objective (MO) ACO exist [129]. Some use a single pheromone [130,131,132] or intrinsic affinity structure [133,130,134]; the resultant pheromone intensity would be calculated using an aggregation of the multiple objective functions or heuristic factors, respectively. Others use multiple pheromones [133,132,134] or heuristic factors [135], each for every objective function. The main difference between these two approaches would be mainly for the pheromone case, as the pheromone update process would differ.

2.5.1 Detailed Algorithmic Steps

Before proceeding with the detailed description of the algorithmic steps, the assumptions made concerning the pheromone structure and the intrinsic affinity should be mentioned. As far as the pheromone structure τ is concerned, multiple pheromone intensities have been used, each for every UF in a similar fashion to [52]. However, during the calculation of the route probability these values are jointly evaluated using the Pareto optimality principle for the sake of avoiding the need for aggregate user-defined weights. As for the intrinsic affinity, a single value is used for each route and it is related to the Pareto dominance of the solution over the set of solutions generated. A more detailed discussion on the intrinsic affinity and on the pheromone intensity models is made in Sections 2.5.2 and 2.5.3, respectively. Moreover, the extraction of the Pareto Fronts of each generation is undertaken by a “master” process, which collects the unique solutions generated from the previous and current generations at each stage and which sorts them into Pareto Fronts.

Furthermore, we have opted for limiting the pheromone levels within a specific range as in [136], since this choice provides the algorithm with immunity to premature convergence to local minima. Explicitly, it forces each pheromone intensity value to reside within a limited range, which in turn results in hitherto "unexplored" solutions being generated as well as ensuring that the pheromones representing the local minima do not reach excessive values, which drive the algorithm towards premature convergence. We note that for all the MO-ACO's results of Chapters 4 and 5 a limited range of $[0.1, 1]$ has been used for the pheromone values, since we observed through extensive simulations that the MO-ACO operates efficiently.

Algorithm 2.3 Multi-Objective Ant Colony Optimization Algorithm

```
 1: Initialize the multi-level pheromone intensities $\tau$, the combined pheromone $T$ and the
    intrinsic affinity $\eta$.
 2: Set $OPF_1 = \varnothing$.
 3: for  $g = 1$ to $\Xi$ do
 4:     Calculate the node-transition probabilities $P$.
 5:     Generate $\zeta$ ants according to $P$.
 6:     Evaluate the routes $R_g$ the ants followed.
 7:     Perform non-dominated sort on the routes $R_g \cup OPF_g$ and store the Pareto optimal
        routes to $OPF_{g+1}$.
 8:     Update the pheromone intensities $\tau$ and then the combined pheromone $T$.
 9: end for
10: Export the vector $OPF_{\Xi+1}$ and exit.
```

The MO-ACO algorithm is formally presented in Alg. 2.3. Firstly, the initialization procedure is invoked. During this procedure the pheromone matrix τ_{init} is initialized to the lower bound of the range. A similar procedure is also carried out for the intrinsic affinity η matrix. Subsequently, the transition probability matrix P is evaluated by jointly taking into account both the intrinsic affinity and the pheromone levels. Then, using the transition matrix P, the routes are generated. We note that their UVs are only calculated, when the ant has reached the DN. After the evaluation of the routes followed by the ants, both the pheromone matrix and the intrinsic affinity vector are updated. If the termination condition is satisfied, the algorithm terminates and exports the OPF; otherwise, it continues to the next generation. In our specific application, the maximum affordable number of generations has been set as the termination condition, since the algorithm has not been designed to be "decision-directed", thus it has no knowledge of the true OPF. An analysis of the intrinsic affinity, the pheromone intensity and the calculation of the transition probability matrix is discussed in Sections 2.5.2, 2.5.3 and 2.5.4, respectively.

2.5.2 Intrinsic Affinity

The heuristic factor of intrinsic affinity provides the ACO family algorithms with robustness against converging to local minima [54]. In our specific application, we have followed a similar approach to that of [125], where Xu *et al.* populated the intrinsic affinity matrix with the distance of each bit from the bits of the *Minimum Mean Square Error* (MMSE) solution. Based on Eq. (2.4), each node-transition $v_{i,j}$ is characterized by a specific received SNR, power dissipation and delay. Therefore, we are able to derive a quasi-MMSE solution if we perform a *Non-Dominated Sort* (NDS) operation for each transition from a specific node to all the possible ones, i.e. a NDS for each row of the vertices matrix V of Eq. (2.4). After this operation, the number $N_{i,j}^{dom}$ of the vertices that dominate a specific vertex $v_{i,j}$ is derived. Note that when a vertex $v_{i,j}$ is equal to $\varnothing$, we assume having $v_{i,j} = [\infty, \infty, \infty]$ in the NDS process. Following a similar approach to that of [125], the intrinsic affinity matrix η is formally written as:

$$\eta = \begin{bmatrix} e^{-N_{1,1}^{dom}} & e^{-N_{1,2}^{dom}} & \cdots & e^{-N_{1,N_{nodes}}^{dom}} \\ e^{-N_{2,1}^{dom}} & e^{-N_{2,2}^{dom}} & \cdots & e^{-N_{2,N_{nodes}}^{dom}} \\ \vdots & \vdots & \ddots & \vdots \\ e^{-N_{N_{nodes},1}^{dom}} & e^{-N_{N_{nodes},2}^{dom}} & \cdots & e^{-N_{N_{nodes},N_{nodes}}^{dom}} \end{bmatrix}, \tag{2.11}$$

where each element $\eta_{i,j}$ of the intrinsic affinity matrix corresponds to the transition form the i-th node to the j-th one. This initialization concept is based on the fact that Pareto optimal transitions are "rewarded", whereas the sub-optimal transitions are "penalized", despite the fact that the optimal transitions do not exclusively form optimal routes. The concept of "reward versus penalty" will be clarified in Section 2.5.4, where the calculation of probabilities is presented. Let us now proceed by defining the pheromone intensity initialization and update processes.

2.5.3 Pheromone Intensity

As mentioned in Section 2.5.1, we opt for using a multi-level pheromone model, so that each of the optimization objectives corresponds to a single pheromone intensity value. The approach followed in [52] involves a so-called aggregation function of the different pheromones used for combining the distinct pheromone intensities into a combined one with user-defined weighting. By contrast, we will utilize the concept of Pareto-optimality in a similar manner as in the intrinsic affinity matrix calculation.

To elaborate further, the resultant multi-pheromone structure can be modeled as a $(N_{nodes} \times N_{nodes})$-element matrix τ corresponding to all the possible node-transitions. The matrix τ is formally expressed as follows:

$$\tau = \begin{bmatrix} \tau_{1,1} & \tau_{1,2} & \cdots & \tau_{1,N_{nodes}} \\ \tau_{1,1} & \tau_{2,2} & \cdots & \tau_{2,N_{nodes}} \\ \vdots & \vdots & \ddots & \vdots \\ \tau_{N_{nodes},1} & \tau_{N_{nodes},2} & \cdots & \tau_{N_{nodes},N_{nodes}} \end{bmatrix}, \tag{2.12}$$

where each element $\tau_{i,j}$ is a vector having a number of elements that is equal to the number of UFs considered, namely 3 in our scenario based on Eq. (2.5). This is formally stated as follows:

$$\tau_{i,j} = \left[\tau_{i,j}^{(1)}, \tau_{i,j}^{(2)}, \tau_{i,j}^{(3)}\right], \tag{2.13}$$

where each element of the multi-level pheromone intensity vector depends on the BER, on the power dissipation and on the delay of a specific route, respectively.

During the initialization process, which takes place in Step 1 of Alg. 2.3, the pheromone intensity values $\tau_{i,j}^{(k)}$ are set to the lower bound of the selected pheromone range, which is formally expressed as:

$$\tau_{i,j}^{k} = \tau_{\min}, \; \forall \; i, \; j, \; k. \tag{2.14}$$

Let us proceed with the pheromone update process, which is activated in Step 8 of Alg. 2.3. During each generation, the pheromone update process is invoked as soon as the routes are generated and their UVs are calculated. Initially, the amount of additional pheromone $\Delta\tau_{i,j}^{(k)}$ is determined. Explicitly, the range of each specific UF value is monitored across the generations. This is necessary for an accurate evaluation of the pheromone levels, since the algorithm sorts the solutions generated within the observed range. Assuming that R_g is the set containing the solutions of the g-th generation, the calculation of the amount of additional pheromone $\Delta\tau_{i,j}^{(k)}$ assigned to the k-th element of the pheromone intensity vector is equal to:

$$\Delta\tau_{i,j}^{(k)} = \sum_{\substack{n=0 \\ v_{i,j} \in R_{g,n}}}^{\zeta-1} \frac{f_{\max}^{k} - f_k(R_{g,n})}{\left(f_{\max}^{k} - f_{\min}^{k}\right)|R_{g,n}|}, \tag{2.15}$$

where $f_{\max}^{k}$ and $f_{\min}^{k}$ correspond to the maximum and minimum observed values of the k-th UF, respectively, when taking into account all the hitherto encountered generations. Additionally, $R_{g,n}$ denotes the n-th route followed during the g-th generation and it represents a vector containing all the vertices of the specific route followed. Additionally, the function $f_k(R_{g,n})$ denotes the UF k-th objective for the route $R_{g,n}$. Moreover, the operation $|R_{g,n}|$ denotes the length of the route $R_{g,n}$ in terms of vertices and its presence in the denominator of the sum of Eq. (2.15) is justified by the fact that an equal reward quantified in terms of pheromone intensity is given to all the vetrices $v_{i,j}$ forming the specific route.

Furthermore, the effect of the pheromone evaporation has to be taken into account, when updating the pheromone levels. This is expressed by the *evaporation rate* ρ, which represents the specific portion of the single-level pheromone that is being evaporated, in other words lost across the consecutive generations. The pheromone update is carried out using the following formula [54]:

$$\tau_{i,j}^{(k)} = (1-\rho)\,\tau_{i,j}^{(k)} + \Delta\tau_{i,j}^{(k)}, \; \forall \; i, \; j, \; k. \tag{2.16}$$

Additionally, the concept of minimum and maximum pheromone level [136] is introduced in Eq. (2.16), since its benefit to the MO-ACO is twofold. On the one hand, it enables the algorithm to escape from local extremities due to the saturation caused by the maximum value of pheromones, while on the other hand it allows the generation of routes having low the pheromone levels. This is formally expressed as follows [136]:

$$\tau_{i,j}^{(k)} = \begin{cases} \tau_{\min} & , \quad \tau_{i,j}^{(k)} < \tau_{\min} \\ \tau_{i,j}^{(k)} & , \quad \tau_{\min} \leq \tau_{i,j}^{(k)} \leq \tau_{\max} \\ \tau_{\max} & , \quad \tau_{\max} < \tau_{i,j}^{(k)} \end{cases} . \tag{2.17}$$

Finally, since no aggregate function of the pheromone intensities having appropriate weighting has been utilized, an appropriate method of combining the multi-level pheromones should be defined. To elaborate further, the principle of Pareto optimality is employed. Each vertex pheromone intensity vector is sorted first using the NDS, similar to the intrinsic affinity case and the order of dominance $N_{i,j}^{dom,\tau}$ is evaluated for each vertex of a specific row in the matrix V. Note that this parameter is equal to the number of vetrices, which dominate a specific vertex. Explicitly, this parameter would be equal to zero for the lowest-rank PF vertices, since no solutions will dominate them. Then, the *combined pheromone intensity* T is defined as follows [125]:

$$T_{i,j} = \exp\left(-N_{i,j}^{dom,\tau}\right), \tag{2.18}$$

where the index of the combined pheromone intensity $T_{i,j}$ corresponds to the specific transition from the i-th node to the j-th one. The exponential function was chosen for separating more effectively the *combined pheromone intensity* of different PFs.

2.5.4 Calculation of Probability and Ant Generation

Having defined the matrices of the intrinsic affinity η and the combined pheromone intensity T, using Eq. (2.11) and Eqs. (2.12)-(2.18) respectively, the corresponding node-transition probability matrix P is defined as follows [55]:

$$P_{i,j} = \frac{(T_{i,j})^{\alpha} (\eta_{i,j})^{\beta}}{\sum\limits_{j=1}^{N_{nodes}} (T_{i,j})^{\alpha} (\eta_{i,j})^{\beta}}, \tag{2.19}$$

where α is the weight of the combined pheromone T and β is the weight of the intrinsic affinity η. We note that the denominator of Eq. (2.19) acts as a normalization factor, which is necessary for complying with the condition, where we have $0 \leq P_{i,j} \leq 1$, $\forall j$. This condition must be satisfied for each of the elements in the P matrix, so that each element $P_{i,j}$ represents the transition probability from the i-th node to the j-th.

Finally, the ant generation process is activated after the calculation of the transition matrix P in Step 5 of Alg. 2.3. The process initially appends the SN to the route vector $R_{g,n}$. Subsequently, the elements of the transition-probability matrix P are observed. The second node of the route will be chosen based on the node-transition probabilities located at the first row of P. A pair of random numbers are cast, namely c_1 and c_2. The first random number c_1 controls whether the next node will be generated using the density function imposed by the respective row of P or by random search, where all nodes' selection obeys a uniform distribution. To elaborate further, a *random search probability* P_s is defined; if $c_1 > P_s$, then the second node of the route vector $R_{g,n}$ is cast using the rule:

$$R_{g,n,2} \;=\; i \quad , \quad \sum_{n=2}^{i-1} p_{n,2} < c_2 \leq \sum_{n=2}^{i} p_{n,2} \quad . \tag{2.20}$$

Otherwise, the symbols are equiprobable, thus Eq. (2.20) is modified as:

$$R_{g,n,2} = \; i \quad , \quad \sum_{n=2}^{i-1} \frac{1}{N_{nodes}-1} < c_2 \leq \sum_{n=2}^{i} \frac{1}{N_{nodes}-1} \quad . \tag{2.21}$$

If the DN is reached, the route is finalized and its UFs are calculated; otherwise, the node generation process continues by casting a new node. However, the constraint of not forming loops has to be satisfied as well. This is guaranteed by excluding the already visited nodes from the generation set and by re-normalizing the probabilities of the remaining nodes. For instance, for an 8-node network, if the temporary route vector $R_{g,n}$ is $(1, 2, 5)$, then the eligible nodes for the fourth place would belong to the set $\{3, 4, 6, 7, 8\}$. The re-normalization factor $f_{norm,i}$ would be equal to:

$$f_{norm,i} = \sum_{\substack{j\,=\,2 \\ i\,\in\,E}}^{N_{nodes}} p_{i,j}, \tag{2.22}$$

where E is the set containing all the eligible destinations, Eqs. (2.20, 2.21) will be transformed into:

$$R_{g,n,k} = \begin{cases} i, \quad \sum\limits_{\substack{n\,=\,2 \\ n\,\in\,E}}^{i-1} \frac{p_{j,n}}{f_{norm,j}} < c_2 \leq \sum\limits_{\substack{n\,=\,2 \\ n\,\in\,E}}^{i} \frac{p_{j,n}}{f_{norm,j}} \quad \text{and} \quad c_1 < P_s \\ i, \quad \sum\limits_{\substack{n\,=\,2 \\ n\,\in\,E}}^{i-1} \frac{1}{|E|} < c_2 \leq \sum\limits_{\substack{n\,=\,2 \\ n\,\in\,E}}^{i} \frac{1}{|E|} \quad \text{and} \quad c_1 \geq P_s \end{cases} , \tag{2.23}$$

where the operation $|\centerdot|$ represents the number of elements in a set, $R_{g,n,k}$ is the k-th element of the route vector $R_{g,n}$ and the index j corresponds to the previously generated node, i.e. we have $j = R_{g,n,k-1}$. This process is repeated until the route is finalized by reaching the DN. We note that the concept of *random search probability* has been conceived for the quantum version of this algorithm by Wang *et al.* [59], where it is referred to as *exploitation probability* P_e. It acts in the same way as the exchange operation of the mutation process of the NSGA-II and assists the algorithm in escaping from local extremities. The steps followed to generate the routes are summarized in the flowchart of Fig. 2.8.

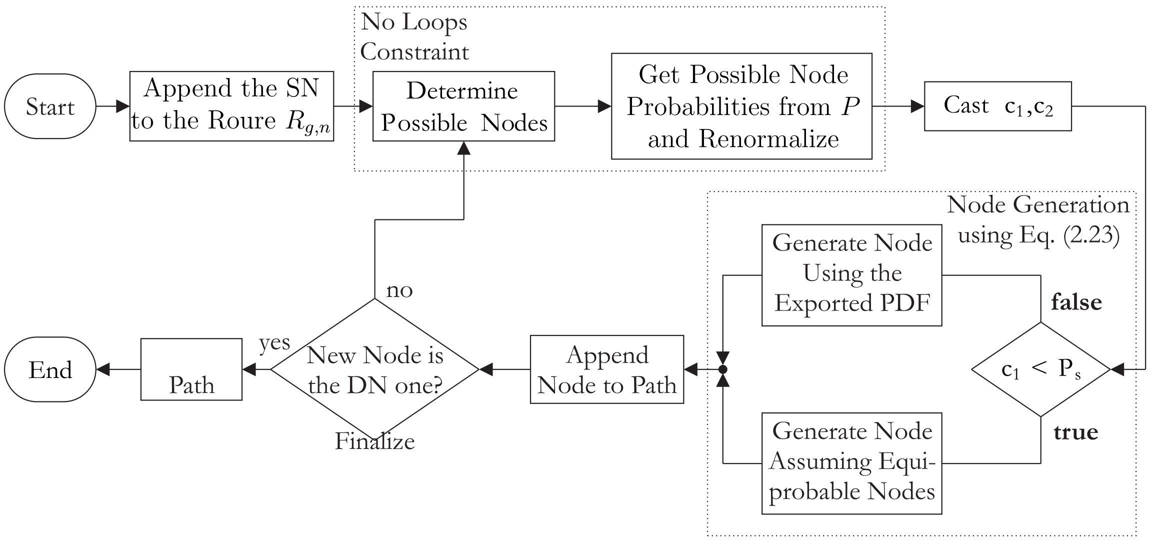

No Loops Constraint
Start
Append the SN to the Roure $R_{g,n}$
Determine Possible Nodes
Get Possible Node Probabilities from P and Renormalize
Cast c_1,c_2
Node Generation using Eq. (2.23)
Generate Node Using the Exported PDF
false
$c_1 < P_s$
true
Generate Node Assuming Equi-probable Nodes
Append Node to Path
no
New Node is the DN one?
yes
Finalize
Path
End

Figure 2.8: Flowchart of the process of creating routes at each MO-ACO generation.

Having elaborated on the routes' generation process, let us now provide a tutorial example for the sake of highlighting the specifics of this process. Hence, let us consider the employment of the MO-ACO Alg. in a 5-node WMHN with its random search probability set to $P_s = 0.1$. In this specific scenario we will assume that the nodes 1 and 5 are the SN and DN nodes, respectively. Let us furthermore assume that the MO-ACO Alg. has reached a state determined by the following transition matrix P:

$$P = \begin{bmatrix} 0 & 0.093 & 0.150 & 0.135 & 0.623 \\ 0 & 0 & 0.680 & 0.295 & 0.025 \\ 0 & 0.237 & 0 & 0.396 & 0.367 \\ 0 & 0.403 & 0.204 & 0 & 0.393 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{2.24}$$

where we note that the transition probabilities that are equal to zero are justified by the fact that a route cannot revisit the current node or the SN. Additionally, there is no further transition after visiting the DN. Therefore, the respective edges have been initialized to zero in the intrinsic affinity matrix η.

The SN is initially appended in the route as portrayed in Fig. 2.8. Consequently, the route x is initialized as follows:

$$x = \{1\}. \tag{2.25}$$

Subsequently, the possible transitions are determined, yielding:

$$E = [e_{1,2}, e_{1,3}, e_{1,4}, e_{1,5}], \tag{2.26}$$

$$p = [0.093, 0.150, 0.135, 0.623], \tag{2.27}$$

$$p_{cum} = [0.093, 0.242, 0.377, 1], \tag{2.28}$$

where p stands for the transition probability distribution of the next node and p_{cum} corresponds to its CDF. Recall that E denotes the set of the available edges or links based on the route x. At this stage the two random numbers, namely c_1 and c_2, are cast; let us assume that we have $c_1 = 0.09$ and $c_2 = 0.67$. Since we have $c_1 < P_s$ the uniform transition probability distribution will be assumed for the selection of the next node will be utilized. Hence, Eqs. (2.27) and (2.28) will be modified as follows:

$$p' = [0.25, 0.25, 0.25, 0.25], \tag{2.29}$$

$$p'_{cum} = [0.25, 0.50, 0.75, 1]. \tag{2.30}$$

Therefore, the 3-rd edge of E will be chosen, since we have $p'_{cum,2} < c_2 < p'_{cum,3}$ in Eq. (2.23). This edge corresponds to the 4-th node, based on Eq. (2.26), yielding the route:

$$x = \{1 \rightarrow 4\}. \tag{2.31}$$

Since the route has not reached the DN, namely the node associated with the index 5, a new node has to be appended. Based on the nodes comprising x in Eq. (2.31), the possible transitions are evaluated as follows:

$$E = [e_{4,2}, e_{4,3}, e_{4,5}], \tag{2.32}$$

$$p = [0.403, 0.204, 0.393], \tag{2.33}$$

$$p_{cum} = [0.403, 0.607, 1]. \tag{2.34}$$

Subsequently, the two random numbers, namely c_1 and c_2, are cast once again; let us now assume that we have $c_1 = 0.62$ and $c_2 = 0.84$. Since we have $c_1 > P_s$, the PDF of Eq. (2.33) will be utilized. Therefore, the 3-rd edge of E will be chosen, since we have $p'_{cum,2} < c_2 < p'_{cum,3}$ in Eq. (2.23). This edge corresponds to the 5-th node, based on Eq. (2.32), yielding the route:

$$x = \{1 \rightarrow 4 \rightarrow 5\}. \tag{2.35}$$

Since the route has reached the DN, namely the node associated with the index 5, the route is finalized and a new route will be generated as long as the maximum number ζ of ants has not been reached.

2.6 Benchmark Algorithms' Complexities

In the previous sections we provided detailed discussions on all the sub-processes of the NSGA-II and the MO-ACO algorithms. Let us now consider their computational complexities in terms of the number of *Cost Function Evaluations* (CFEs), where a single CFE represents a single Pareto dominance comparison. In this section, we will derive the associated computational complexities for the sake of using them as benchmarkers for the proposed quantum-assisted algorithms presented in Chapters 4, 5 and 6.

As far as the NSGA-II is concerned, the computational complexity is solely contributed by the Step 3 of Alg. 2.1 at each generation. To elaborate further, let us assume that the NSGA-II iterates for N_G generations having N_{pop} individuals in its initial population; the NDS process would involve N_{pop}^2 CFEs, since the number of routes which dominate a specific route within P_g has to be calculated for every route of the P_g set. We note that as the unit of calculation we use the number of dominance operator employments. As for the calculation of the crowding distance of Step 4 in Alg. 2.1, assuming we have K UFs, a sorting algorithm is used K times, once per UF. Assuming that the sorting algorithm imposes a complexity, which is on the order of $O(M\log(M))$ [124], the resultant complexity quantified in terms of the number of dominance operator employments would be $O[M\log(M/K)] = O[M\log(M)]$, where M corresponds to the total number of routes belonging to a specific PF. Nevertheless, since we are using the NSGA-II as a benchmarker for our quantum-assisted algorithms, we may assume that the sorting algorithm is capable of sorting the routes belonging to a specific PF imposing a complexity equal to M CFEs, which corresponds to the lower bound of the sorting algorithms' complexity. Therefore, at each generation the algorithm would impose an extra N_{pop} CFEs, hence resulting in a total complexity of:

$$L_{NSGA-II} = N_G\ N_{pop}(N_{pop}+1) > N_G N_{pop}^2, \tag{2.36}$$

across all the N_G generations. Hence, we will use the lower bound for comparison with our quantum-assisted algorithm, i.e. we have $L_{NSGA-II} = N_G N_{pop}^2$.

As far as the MO-ACO is concerned, a certain amount of computational complexity is imposed by the intrinsic affinity calculation, by the pheromone update and by the evaluation of the routes followed at each generation. At this stage, let us assume that the MO-ACO is initialized to have Ξ generations and ζ ants per generation. The intrinsic affinity initialization process of Step 1 in Alg.2.3 invokes the NDS process once per row, yielding a total complexity of N_{nodes}^3 CFEs according to Eq. (2.11). Then, in the pheromone update process, the calculation of the combined pheromone intensity T in Step 8 of Alg. 2.3 imposes the same amount of complexity per generation, i.e. we have N_{nodes}^3 CFEs per generation. Finally, the NDS process of Step 7 in Alg. 2.3 results in ζ^2 CFEs per generation, yielding a total complexity of:

$$L_{MO-ACO} = \Xi\left(\zeta^2 + N_{nodes}^3\right) + N_{nodes}^3 > \Xi\zeta^2, \tag{2.37}$$

across all Ξ generations. Hence, we will use the lower bound for comparison with our quantum-assisted algorithm, i.e. we have $L_{MO-ACO} = \Xi\zeta^2$.

For the sake of simplicity, we will set the number of NSGA-II individuals evaluated by the end of each generation equal to the number of generations, i.e. we rely on $N_G = N_{pop}$ and $\Xi = \zeta$. Assuming now that the maximum complexity of the NDQO algorithm is $\max\left\{L^{tot}_{CFE}\right\}$, in order to match the complexities of the quantum-assisted algorithms, the MO-ACO and the NSGA-II will rely on:

$$N_G = \Xi = \sqrt[3]{\max\left\{L^{tot}_{CFE}\right\}}. \tag{2.38}$$

Finally, the input parameters of the NSGA-II and the ACO used in our comparative case study are shown in Table 2.2.

Table 2.2: Input Parameters of the Benchmarking Algorithms

NSGA-II [29, 124]		MO-ACO [55, 52]	
Number of Generations, N_G	$\sqrt[3]{\max\left\{L^{tot}_{CFE}\right\}}$	Number of Generations, Ξ	$\sqrt[3]{\max\left\{L^{tot}_{CFE}\right\}}$
Initial Population N_{pop}	$\sqrt[3]{\max\left\{L^{tot}_{CFE}\right\}}$	Number of Ants, ζ	$\sqrt[3]{\max\left\{L^{tot}_{CFE}\right\}}$
Mutation Probability, P_m	0.5	Pheromone weighting, α	1.2
Crossover Probability, P_c	0.9	Intrinsic Affinity weighting, β	0.6

2.7 Chapter Summary

In this chapter, we have discussed all the necessary assumptions needed in order to define the optimization problem, namely the network topology and the concept of Pareto-optimality. In the next chapter, the family of quantum heuristic algorithms will be discussed in order to complete the system's portrayal. Additionally, we have provided detailed discussions on the classical evolutionary algorithms, namely on the NSGA-II and on the MO-ACO algorithm. In the next chapters, we will use them as benchmarks for our quantum-assisted algorithms. Furthermore, we have derived the lower bounds of the complexity imposed by the aforementioned algorithms. In particular, the resultant lower bounds are:

$$\begin{aligned} L_{NSGA-II} &= N_G\, N_{pop}^2, \\ L_{MO-ACO} &= \Xi\, \zeta^2, \end{aligned}$$

where N_G and Ξ correspond to the number of the generations in the NSGA-II and MO-ACO algorithm respectively, while N_{pop} and ζ denote initial population of the NSGA-II and the number of ants in the MO-ACO algorithm, respectively. In the next chapters, we will rely on the above complexities for benchmarking the quantum-assisted algorithms against the respective classical solutions by matching the computational complexities of all the algorithms involved. Finally, in the following chapter we will present some QSAs, which constitute the fundamental sub-processes of our proposed algorithms.

Chapter 3

Quantum-Based Heuristics

3.1 Introduction

In this chapter we will lay the foundations, based on which our quantum-assisted algorithms will operate. We commence with a brief introduction to *Quantum Computing Postulates* in Section 3.2, which form the framework for the operation of quantum-mechanical systems. As we will demonstrate in Section 3.2, the beneficial complexity reduction offered by quantum algorithms relies upon the *Quantum Parallelism* (QP), which is a result of the specific property of *quantum bits* (qubits), namely that they can be in the superposition of their one and zero states. This is in stark contrast to the classical bits, which can only be in a either one or zero. This specific property is exploited by the quantum search algorithms conceived for finding certain entries in a database. Naturally, we have to transform our routing problem, where the database is comprised by all the legitimate routes of Section 2.2, into a binary combinatorial search problem for the sake of facilitating the employment of *Quantum Search Algorithms* (QSAs). For this reason, we will map every route to a specific binary index of the database using *Lehmer Encoding/Decoding* [137], which offers a one-to-one mapping with the aid of factorial decomposition. The specifics of this transformation are described in Appendix A.

Subsequently, we will elaborate on three popular QSAs, which have to be carefully ameliorated for employment in our multi-objective routing problem. More specifically, in Section 3.3 we will present *Grover's Quantum Search Algorithm* (QSA) [79]. This specific QSA is of utmost importance, since it features the so-called *Grover operator* $\mathcal{G}$, which is the cornerstone for a broad family of QSAs referred to as *Quantum Amplitude Amplification Algorithms* (QAAA). However, Grover's QSA is readily applicable in specific search problems, where both the value sought and the number of solutions are known beforehand. Given its importance, we will provide a low-paced 4-node WMHN tutorial example in Section 3.3.3 for the sake of outlining its merits. Following Grover's QSA, we will detail the so-called *Boyer-Brassard-Høyer-Tapp QSA* (BBHT-QSA) [80] and the *Durr-Høyer Algorithm* (DHA) [81] in Sections 3.4 and 3.5, respectively. Both these algorithms employ the

Grover operator $\mathcal{G}$ and we will investigate them in detail in this chapter, since they will be used as sub-processes in our novel QSAs. Let us now proceed by presenting the quantum computing postulates.

3.2 Quantum Computing Postulates

In this section we will provide detailed discussions regarding the four Quantum Mechanics postulates [62]. More specifically, the first postulate defines the state of a quantum system; the second describes the system's evolution over time; the third postulate formulates the process of observing or measuring its state; the fourth postulate describes the process of forming a composite quantum system from. individual qubits.

(a) **State Space**: the first postulate defines the state of a quantum system.

(b) **Time Evolution**: the second describes the system's evolution over time.

(c) **Measurement**: the third postulate formulates the process of observing or measuring its state.

(d) **Composite Systems**: the fourth postulate describes the process of forming a composite quantum system from individual qubits.

Note that we will define the state of a qubit in the first postulate discussions Let us now proceed by analyzing the specifics of each postulate.

3.2.1 State Space

Based on the **first postulate** [67], a quantum system's state is formulated as follows:

$$|\phi\rangle = \sum_{i=0}^{M-1} \varphi_i |\phi_i\rangle = (\varphi_0, \varphi_1, ..., \varphi_{M-1})^T , \tag{3.1}$$

where the complex valued φ_i represents the amplitude of the *basis state* $|\phi_i\rangle$ and there are $M = 2^m$ basis states in total. The symbol $|\cdot\rangle$ represents a quantum-domain basis state, which is simply referred to as "ket". The squared modulus $|\varphi_i|^2$ of the amplitude corresponds to the probability of the quantum system residing in the state $|\phi_i\rangle$, namely the probability of observing the state $|\phi_i\rangle$. Naturally, the sum of theses probabilities should be equal to unity, as encapsulated in:

$$\sum_{i=0}^{M-1} |\varphi_i|^2 = 1. \tag{3.2}$$

Similar to the classical bit, the quantum systems' smallest unit of information is the qubit [62]. Explicitly, the qubit has two basis states, namely the state $|\phi_0\rangle \equiv |0\rangle$ and the state

$|\phi_1\rangle \equiv |1\rangle$. Therefore, Eq. (3.1) is reduced to [67]:

$$|\phi\rangle = \varphi_0 |0\rangle + \varphi_1 |1\rangle , \tag{3.3}$$

where the same normalization constraint has to be satisfied, i.e. we have $|\varphi_0|^2 + |\varphi_1|^2$. Moreover, for the complex conjugate transpose of $|\phi\rangle$ the notation used is $\langle\phi|$, which is equal to:

$$\langle\phi| \quad = \quad |\phi\rangle^\dagger \quad = \quad (\varphi_0^*, \varphi_1^*, ..., \varphi_{M-1}^*) \ \cdot \tag{3.4}$$

Since the amplitudes assume complex values, the system state's argument would lie within the M-dimensional Hilbert space. Furthermore, a beneficial property of quantum systems observed from Eq. (3.1) is that given a quantum register (QR) storing m qubits, the QR may assume all M states simultaneously, which is often termed as being in the superposition of basis states. It is exactly this property, which the quantum algorithms exploit for the sake of carrying out operations in parallel. Finally, note that we will elaborate on the process of forming QRs out of the individual qubits in Section 3.2.4.

3.2.2 Time Evolution

Based on the **second postulate** [62], the evolution of a physical system's state versus time is characterized by a set of unitary transformations, which is formulated as follows:

$$|\psi\rangle = U |\phi\rangle , \tag{3.5}$$

where U is a unitary matrix. In other words, we have $U^{-1} = U^\dagger$, where $U^\dagger$ is the complex conjugate transpose of U. Equation (3.5) stems from the Schrödinger equation [62]. Unitary operators are represented by linear matrices and the assumption of linearity assists in preventing the occurrence of some "strange" phenomena such as the *time travel* [62] stemming from non-linearities. Naturally, this property allows us to break down the operations encapsulated in Eq. (3.5) into simpler ones using *quantum gates* [62], hence assisting us in reducing the hardware complexity of quantum algorithms. In fact, there is a suite of components having a unitary response. Some of the most common single-qubit quantum gates, which we will utilize in our quantum-assisted algorithms in the next chapters are the *Hadamard* gate H and the *Rotation* gate R_θ. Explicitly, their single-qubit transfer matrices are [62]:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} , \ R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} . \tag{3.6}$$

The *Hadamard* gate is primarily used for mapping the ground state $|0\rangle$ to the equal superposition of the states $|0\rangle$ and $|1\rangle$, while the *Rotation* gate rotates the qubit state by an angle of θ within the area defined by the eigenstates $|0\rangle$ and $|1\rangle$. These operations are formally expressed as follows:

$$|0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) \equiv |+\rangle\,, \tag{3.7}$$

$$|1\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right) \equiv |-\rangle\,, \tag{3.8}$$

$$|0\rangle \xrightarrow{R_\theta} \cos\theta\,|0\rangle - \sin\theta\,|1\rangle\,. \tag{3.9}$$

Apart from these simple operations, it is possible to carry out controlled operations. A commonly used gate belonging to this family of components is the *Controlled-NOT* (CNOT) gate [67]. More specifically, this specific gate has two input qubits or QRs and it performs the *Exclusive OR* (XOR) operation of its two inputs storing the output in the second qubit or QR. Thus, it is equivalent to the classic XOR gate and its function is formulated as follows:

$$|c\rangle\,|t\rangle \xrightarrow{\text{CNOT}} |c\rangle\,|c \oplus t\rangle\,, \tag{3.10}$$

where the state $|c\rangle$ is often referred to as the *control* register, while $|t\rangle$ is the *target* register. In fact, the CNOT gate represents a special case of a family of quantum gates, which are commonly known unitary operators U_f [62]. They are capable of implementing a binary function $f : \{0, 1, \ldots, N-1\} \to \{0, 1\}$ in the quantum domain. Their quantum circuit is shown in Fig. 3.1; due to the superposition of states of the QR $|x\rangle_1$ it is possible to carry out the function's calculations in parallel, which is the main advantage of quantum computing. Their operation may be formulated as follows:

$$|x\rangle_1\,|0\rangle_2 \xrightarrow{U_f} |x\rangle_1\,|0 \oplus f(x)\rangle_2 \equiv |x\rangle_1\,|f(x)\rangle_2\,. \tag{3.11}$$

Note that the subscripts of the "kets" are used for distinguishing the two inputs of the QRs. These unitary operators are the main component for the construction of *Quantum Oracle gates* [62]. Therefore, the QR $|x\rangle_1$ is often referred to as a *Quantum Index Register* (QIR), since it points to the indices of the input states, while the second input is commonly known as the *Oracle Workspace* (OW), since all the Oracle operations are carried out in this specific QR. The unitary operators U_f play an integral role both in QSAs and in quantum optimization algorithms, since they are capable of outputting a state constituting by the superposition of all the possible outputs of a function f. More explicitly, its output state is in the superposition of all the possible inputs, as it becomes plausible based on Eq. (3.11). Therefore, these operators constitute the fundamental manipulations of QP and, which result in a search complexity reduction.

3.2.3 Measurement Operation

Based on the **third postulate** [67], the quantum measurement or observation operation is described by a set of measurement operators $\{\mathcal{M}_m\}_{m=0}^{M-1}$, which are applied to the state space, when the system is subjected to a measurement or observation. The index m indi-

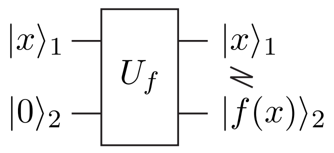


Figure 3.1: Quantum circuit implementing a specific function $f(x)$; the subscripts of the "kets" are used in order to distinguish the two input QRs and the jagged line represents the *entanglement* between the two output QRs. The input QR $|x\rangle_1$ can be either at one of the N specific basis states, i.e. we have $|x\rangle_1 \in \{0,1\}^{\otimes \log_2 N}$, or at the superposition of the N basis states, i.e. we have $|x\rangle_1 = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle_1$; in the latter case, the operation of U_f is encapsulated by Eq. (3.23).

cates that the measurement's outcome will be equal to m and the probability $p(m)$ of this outcome is given upon assuming a general initial state of $|\psi\rangle$ as follows:

$$p(m) = \langle\psi| \mathcal{M}_m^\dagger \mathcal{M}_m |\psi\rangle , \tag{3.12}$$

Explicitly, the post-measurement state of the related QR or qubit becomes [62]:

$$|\psi'\rangle = \frac{\mathcal{M}_m |\psi\rangle}{\sqrt{p(m)}}. \tag{3.13}$$

In the special case, where there are two basis states, namely $\mathcal{M}_0 = |0\rangle\langle 0|$ and $\mathcal{M}_1 = |1\rangle\langle 1|$ and an arbitrary state of the qubit $|\psi\rangle = a|0\rangle + b|1\rangle$, we may arrive at:

$$p(0) = |\psi\rangle \mathcal{M}_0^\dagger \mathcal{M}_0 \langle\psi| = |a|^2 , \tag{3.14}$$

$$p(1) = |\psi\rangle \mathcal{M}_1^\dagger \mathcal{M}_1 \langle\psi| = |b|^2 , \tag{3.15}$$

where the post-measurement state will respectively become equal to:

$$|\psi'\rangle_{m=0} = \frac{\mathcal{M}_0 |\psi\rangle}{|a|} = \frac{a|0\rangle}{|a|}, \tag{3.16}$$

$$|\psi'\rangle_{m=1} = \frac{\mathcal{M}_1 |\psi\rangle}{|b|} = \frac{b|1\rangle}{|b|}. \tag{3.17}$$

Observe from Eqs. (3.16) and (3.17) that the qubit collapses into a classical bit state after the measurement operation on the computational basis of $\{|0\rangle, |1\rangle\}$. Naturally, the same phenomenon occurs, when measuring a QR storing multiple qubits, based on Eq. (3.13). Consequently, the measurement operation has a deleterious effect on the QP and thus thwarts to the quantum systems' capability of offering complexity reduction. Hence, we want to circumvent the invocation of the measurement or observation operation by sophisticated manipulation of the superimposed state of the QR. Explicitly, this specific manipulation is carried out by the Grover operator $\mathcal{G}$ [79], as we will demonstrate in Section 3.3.

3.2.4 Composite Systems

The **fourth postulate** of quantum mechanics describes the state of a length K QR, which is composed by individual QRs of length 1 or simply qubits. The resultant state can be expressed as the tensor product of the individual register states. For instance, in case of a QR comprised by 2 qubits $|\psi_1\rangle$ and $|\psi_2\rangle$, the resultant state will become [62]:

$$
\begin{aligned}
|\psi\rangle &= |\psi_1\rangle_1 \otimes |\psi_2\rangle_2 && (3.18)\\
&= (\alpha\,|0\rangle_1 + \beta\,|1\rangle_1) \otimes (\gamma\,|0\rangle_2 + \delta\,|1\rangle_2) && (3.19)\\
&= \underbrace{\alpha\gamma}_{a_{00}}|00\rangle + \underbrace{\alpha\delta}_{a_{01}}|01\rangle + \underbrace{\beta\gamma}_{a_{10}}|10\rangle + \underbrace{\beta\delta}_{a_{11}}|11\rangle\,, && (3.20)
\end{aligned}
$$

where the operator $\otimes$ corresponds to the tensor product. Naturally, the modulus squared of the individual amplitudes a_{ij} have to comply with the following constraint:

$$\sum_{\forall i,j} |a_{ij}|^2 = 1, \qquad (3.21)$$

for the sake of ensuring that the probability of observing the states $|ij\rangle$ sums to unity.

A notable consequence constituted by the above postulates is the "spooky" phenomenon of *quantum entaglement* [67], which expresses a linkage between the single qubits of a multiple-qubit state, when the multiple-qubit state is an entangled one. This linkage occurs even when these single qubits are delivered to two arbitrary remote locations. Explicitly, a composite state is referred to as entangled if it is not possible to describe the composite state by the individual qubits, i.e. if it is not possible for Eq. (3.20) to be written in the from of Eq. (3.18). This specific linkage among the states of the composite system has a direct impact on the measurement operation. To elaborate further, if we assume that the QIR of the U_f operator portrayed in Fig. 3.1 is in the superposition of all the possible N states, i.e. we have:

$$|x\rangle_1 = \sum_i |i\rangle/\sqrt{N}, \qquad (3.22)$$

then the output of the operator U_f, considering both the QIR and the OW, will be in the superposition of composite states. Hence, we have:

$$|x\rangle_1\,|f(x)\rangle_2 = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1} |i\rangle_1\,|f(i)\rangle_2. \qquad (3.23)$$

Explicitly, Eq. (3.23) suggests that if a *partial measurement* [87] is carried out concerning the QIR, then the state of the OW will also collapse to the state $|f(i)\rangle_2$, assuming the observable state $|i\rangle_1$ in the QIR. A direct consequence of the *quantum entanglement* is the so-called *no-cloning theorem* [62], which dictates that it is physically infeasible to clone the state of a quantum system. Therefore, it imposes the constraint that only qubits having known and/or orthogonal states may be copied. This theorem provides security in quantum communications and it is the main facilitator behind perfect-secrecy offered by *Quantum*

Key Distribution (QKD) [138,139,140,141].

3.3 Grover's Quantum Search Algorithm

This specific QSA has been proposed by Grover [79] in the mid '90s. It was initially developed for finding a single solution stored in a database having uncorrelated entries, such as a telephone book. It has been shown to be optimal by Zalka [88] in terms of the number of database queries by matching the lower bound of database queries [142], while guaranteeing ~100% search success. Its optimality has been investigated analytically by Boyer *et al.* in [80], where they have succeeded in deriving the upper bound of the number of the so-called *Grover iterations* [142] needed for maximizing the heuristic success probability. Moreover, they have extended its employment to the case, where multiple solutions exist; however, in both cases the actual number t of solutions has to be known beforehand for the sake of determining the required number of Grover iterations.

Let us now proceed with a more analytic description of Grover;s QSA. As mentioned in the previous paragraph, Grover's QSA relies upon carrying out multiple Grover iterations. Explicitly, this specific iteration is equivalent to a single application of the so-called *Grover Operator* $\mathcal{G}$, the transfer which has a transfer matrix formulated as follows [79]:

$$\mathcal{G} = \underbrace{HP_0H}_{D} \cdot O, \tag{3.24}$$

where H is the *Hadamard Gate* defined in Eqs. (3.7) and (3.8), while P_0 represents a quantum gate which flips the sign of all the states apart from the *zero state*, i.e. we have $|x\rangle \overset{P_0}{\to} -|x\rangle$ if $|x\rangle \neq |0\rangle$. Additionally, the operator O is the *quantum oracle gate*, which is capable of recognizing and "marking" the desired solutions, which are the specific solutions that satisfy the generic constraint $f(x) = \delta$ with δ being the value sought in the search problem to be solved. Finally, the operator $D = HP_0H$ is commonly referred to as the *Diffusion Matrix* [79], which employment performs an inversion on the the state amplitudes about average [79]. Note that the specifics of this operator will be discussed in Section 3.3.1. Let us now proceed by analyzing the specifics of the quantum oracle gate O.

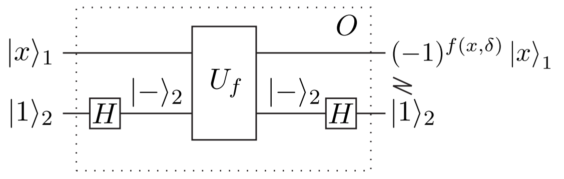


Figure 3.2: Quantum circuit of a quantum oracle O; three quantum gates are used: two *Hadamard* gates H and a unitary quantum oracle U_f. This quantum circuit is also known as *phase kick-back* [112] as it succeeds in altering the phase by π of the states for which we have $f(x,\delta) = 1$ or equivalently $f(x) = \delta$. Note that the value δ is considered to be predefined and, hence, it is not portrayed in the figure.

As far as the quantum oracle gate O is concerned, it utilizes the unitary operator U_f, which is similar but not identical to that portrayed in Fig. 3.1. This operator implements the binary function $f(x,\delta)$, relying on an arbitrary function $f(x):\{0,\dots,N-1\}\rightarrow\{0,1\}^n$, where n corresponds to the representation accuracy of $f(x)$ in bits, and compares the value $f(x)$ corresponding to the database index x against the value δ being sought, which has the same representation accuracy in bits. The entire operation can be formulated using the binary function $f(x,\delta)$, which has the input constituted by a particular index x of the database and outputs 1, as and when the examined index stores the corresponding value $f(x)$, which is equal to the value δ being sought; otherwise, it returns 0. This can be formally expressed as follows:

$$f(x,\delta)=\begin{cases}1 & f(x)=\delta,\\ 0 & \text{otherwise.}\end{cases} \tag{3.25}$$

The quantum circuit of the oracle gate O is shown in Figure 3.2, where we can observe that apart from the unitary operator U_f implementing the function $f(x,\delta)$ of Eq. (3.25), two *Hadamard gates* H are used, one at the input and one at the output of the U_f operator's OW. The first *Hadamard gate* H acts only on the second QR, the state of which is set to $|1\rangle_2$ and it will map this state to the state $|-\rangle_2=\left(|0\rangle_2-|1\rangle_2\right)/\sqrt{2}$. Since no controlled operations take place, no *entanglement* relationship exists between the two QR cells. Viewing the system as the composite of having the initial state $|x\rangle_1|1\rangle_2$, the outcome of applying the Hadamard gate H becomes:

$$|x\rangle_1|1\rangle_2\xrightarrow{H}|x\rangle_1|-\rangle_2\,. \tag{3.26}$$

The system's QRs are then fed into the unitary operator U_f. Since U_f involves a controlled operation, which is a modulo-2 addition ($\oplus$) of the outcome of the binary function $f(x)$ to the state of the second register, the two QRs will become entangled and using Eq. (3.11) the following system state is arrived at:

$$|x\rangle_1|-\rangle_2=\frac{1}{\sqrt{2}}|x\rangle_1\left(|0\rangle_2-|1\rangle_2\right)\xrightarrow{U_f}\frac{1}{\sqrt{2}}|x\rangle_1\left(|0\oplus f(x)\rangle_2-|1\oplus f(x)\rangle_2\right). \tag{3.27}$$

At this point, let us examine the resultant state of the second QR for the two possible values that $f(x,\delta)$ may assume. If we have $f(x)=\delta$, then we get $f(x,\delta)=1$ and the state of the second register becomes:

$$|-\oplus f(x,\delta)\rangle_2|_{f(x,\delta)=1}=\frac{1}{\sqrt{2}}\left(|0\oplus 1\rangle_2-|1\oplus 1\rangle_2\right)=\frac{1}{\sqrt{2}}\left(|1\rangle_2-|0\rangle_2\right)=-|-\rangle_2\,. \tag{3.28}$$

Otherwise, if we have $f(x)\neq\delta$, then we get $f(x)=0$ and the state of the second QR becomes:

$$|-\oplus f(x,\delta)\rangle_2|_{f(x,\delta)=0}=\frac{1}{\sqrt{2}}\left(|0\oplus 0\rangle_2-|1\oplus 0\rangle_2\right)=\frac{1}{\sqrt{2}}\left(|0\rangle_2-|1\rangle_2\right)=|-\rangle_2\,. \tag{3.29}$$

Hence, Eqs. (3.28) and (3.28) can be combined into a single one in terms of their corresponding value $f(x)$:

$$|-\oplus f(x,\delta)\rangle_2 = (-1)^{f(x,\delta)}\,|-\rangle_2\,, \tag{3.30}$$

where it is clear that the operator U_f modifies the phase of the second QR by π, if $f(x)=\delta$. Again, this kind of quantum circuits are commonly referred to in parlance as *Phase Kick-back* quantum circuits [112]. Due to the entanglement of the two QRs, it is legitimate to consider that the phase of the first QR is indeed altered, while the second one remains intact. This is formally stated as:

$$|x\rangle_1\,[(-1)^{f(x,\delta)}\,|-\rangle_2] \equiv [(-1)^{f(x,\delta)}\,|x\rangle_1]\,|-\rangle_2\,. \tag{3.31}$$

Hence, the system state at the output of U_f may be derived by substituting Eqs (3.30) and (3.31) into Eq. (3.27), yielding:

$$|x\rangle_1\,|-\rangle_2 \xrightarrow{U_f} [(-1)^{f(x,\delta)}\,|x\rangle_1]\,|-\rangle_2\,. \tag{3.32}$$

At the output oracle U_f, a second *Hadamard* gate H is used at the second QR in order to map its state from $|-\rangle_2$ to $|1\rangle_2$; however, since the inverse of U_f, namely $U_f^\dagger$, has not been applied, the entanglement relationship between the two QRs will still hold and thus the phase flip imposed on the second QR may still be regarded as if it was applied to the first one. Hence, the oracle gate O would "mark" the intended states by flipping their phase by π and this operation may be formulated as [79]:

$$|x\rangle_1\,|1\rangle_2 \xrightarrow{O} [(-1)^{f(x,\delta)}\,|x\rangle_1]\,|1\rangle_2\,. \tag{3.33}$$

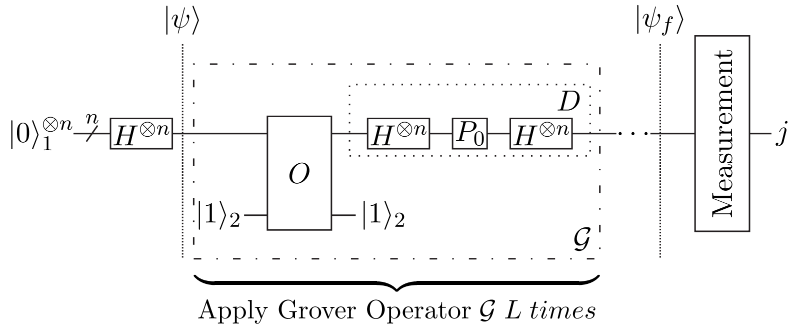


Figure 3.3: Quantum circuit of Grover's QSA; the *Grover Operator* $\mathcal{G}$ is applied to the initial state $|\psi\rangle$ L times resulting in the final state $|\psi_f\rangle$ which is then measured resulting, in turn, in the exported classic state j.

Having discussed the operation of the quantum oracle O, let us now examine Grover's QSA operation a little further. Its quantum circuit is shown in Fig. 3.3. First, the system state $|\psi\rangle$ is initialized to the superposition of all the possible states; this is arranged by using a QR in the pure state $|0\rangle_1^{\otimes n}$, where $N=2^n$ is the total number of the states considered and an appropriate-size Hadamard gate $H^{\otimes n}$. Therefore, the initial state formation of $|\psi\rangle$ is represented as:

$$|0\rangle_1^{\otimes n} \xrightarrow{H^{\otimes n}} |+\rangle_1^{\otimes n} \equiv |\psi\rangle \,, \tag{3.34}$$

and the initial state $|\psi\rangle$ becomes equal to:

$$|\psi\rangle = |+\rangle_1^{\otimes n} = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle_1. \tag{3.35}$$

Then, the *Grover Operator* $\mathcal{G}$ is applied L times to the initial state $|\psi\rangle$, which yield the final state $|\psi_f\rangle$ formulated as:

$$|\psi_f\rangle = \mathcal{G}^L |\psi\rangle = \sum_{i=0}^{N-1} \psi_{f,i} |i\rangle_1, \tag{3.36}$$

where $\psi_{f,i}$ denotes the amplitude of the state $|i\rangle_1$ and its respective probability of occurrence upon measurement would be $|\psi_{f,i}|^2$. We note that the number L of $\mathcal{G}$ applications will be examined in Section 3.3.2. Moreover, the steps of the QSA are summarized in Alg. 3.1. Finally, the final state is measured using a *Quantum Detector* (QD) [62] and the classic measurement outcome j is exported, which obeys the the *Probability Density Function* (PDF) of:

$$P(j = i) = |\psi_{f,i}|^2 \,, \text{ with } \sum_{i=0}^{N-1} |\psi_{f,i}|^2 = 1. \tag{3.37}$$

Since the quantum oracle O calls U_f only once, the complexity of this operator quantified in terms of the U_f queries would be equal to 1 [142]. Therefore, the complexity of QSA is quantified in terms of the number L of $\mathcal{G}$ applications, since this operation would call the oracle gate O L times. We note that the complexity of the observation or measurement operation considered is assumed to be negligible.

Algorithm 3.1 Grover's Quantum Search Algorithm

```
1: Define the Grover Operator $\mathcal{G}$ using (3.24).
2: Calculate the number of Grover Iterations $L = \lfloor \frac{\pi}{4}\sqrt{N/t} \rfloor$.
3: Apply the $\mathcal{G}$ operator $L$ times starting from the initial state $|\psi\rangle$ of Eq. (3.36), resulting
   in the final state $|\psi_f\rangle = \mathcal{G}^L |\psi\rangle$.
4: Observe $|\psi_f\rangle$ in the QD and obtain $|j\rangle$.
5: Set $x_s \leftarrow j$, output $x_s$ and exit.
```

3.3.1 Diffusion Matrix

In this section, we will derive an analytic formula for the elements of the *Diffusion Matrix*. Assuming that the list contains $N = 2^n$ elements the *Hadamard Gate* takes the form $H^{\otimes n}$, where H is the transfer matrix defined in Eq. (3.6) and the operation $\otimes$ denotes the *Kronecker Tensor Product* [62]. Therefore, each element H_{ij} of the $H^{\otimes n}$ matrix would be equal to [79]:

$$H_{ij} = \frac{1}{\sqrt{2^n}}(-1)^{\bar{i}\cdot\bar{j}}, \tag{3.38}$$

where $\overline{i}$ and $\overline{j}$ denote the binary representation of the decimal numbers i and j in binary strings of length equal to n and the operation $\overline{i} \cdot \overline{j}$ denotes the bitwise dot product of the two binary strings. As far as the quantum oracle P_0 is concerned, it will flip phases of all the other states by π of all the states apart from the *zero state* $|0\rangle^{\otimes n}$, leading to a diagonal transfer matrix P_0 equal to [79]:

$$P_{0,ij} = \begin{cases} 0 & i \neq j, \\ 1 & i = j = 0, \\ -1 & i = j \neq 0. \end{cases} \tag{3.39}$$

Furthermore, the operator P_0 can be decomposed into two matrices [79]:

$$P_0 = -I_N + R = \underbrace{2\,|0\rangle^{\otimes n}\,\langle 0|^{\otimes n}}_{R} - I_N, \tag{3.40}$$

where I_N is the $(N \times N)$ identity matrix and $R = 2\,|0\rangle^{\otimes n}\,\langle 0|^{\otimes n}$ is defined as [79]:

$$R_{ij} = \begin{cases} 2 & i = j = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{3.41}$$

Therefore, the *Diffusion Matrix* D is decomposed into two separate matrices, namely D_1 and D_2, as follows [79]:

$$D = H(R + I_N)H = \underbrace{HRH}_{D_1} - \underbrace{HI_NH}_{D_2}. \tag{3.42}$$

Explicitly, the matrix D_2 is equal to:

$$D_2 = HI_NH = HH = HH^T = I_N. \tag{3.43}$$

As for the matrix D_1, each of its elements $D_{1,ij}$ is equal to:

$$D_{1,ij} = \sum_{k=0}^{2^n-1} \sum_{l=0}^{2^n-1} H_{il} R_{lk} H_{kj}. \tag{3.44}$$

However, since R has only a single non-zero element, which is the first element of its diagonal, Eq. (3.44) is reduced to:

$$D_{1,ij} = H_{i0} R_{00} H_{0j} = 2H_{i0}H_{0j} = \frac{2}{2^n}(-1)^{\overline{i}\cdot\overline{0}+\overline{0}\cdot\overline{j}} = \frac{2}{N}. \tag{3.45}$$

Finally, it is possible to derive each element D_{ij} of the *Diffusion Matrix* applying Eqs. (3.45) and (3.43) into Eq. (3.42) [79]:

$$D_{ij} = \begin{cases} -1 + \frac{2}{N} & i = j, \\ \frac{2}{N} & \text{otherwise.} \end{cases} \tag{3.46}$$

Let us now define an $(N \times N)$-element matrix P, whose elements are all equal to $1/N$. Naturally, this matrix performs an averaging operation if applied to a vector $\overline{\nu}$. To elaborate further, the product $\overline{a} = P\overline{\nu}$ is a vector, the elements of which are all equal to the average of the elements of $\overline{\nu}$. Using the matrix P, D is expressed as follows:

$$D = -I_N + 2P. \tag{3.47}$$

Finally, should D be applied to a vector $\overline{\nu}$, the outcome $\overline{k} = D\overline{\nu}$ is equal to [79]:

$$\overline{k} = D\overline{\nu} = -\overline{\nu} + 2\overline{a} = [\overline{a} + (\overline{a} - \overline{\nu})]. \tag{3.48}$$

Explicitly, Eq. (3.48) indicates that the operator D performs an *inversion about the average value* [79]. Assuming that the input $\overline{\nu}$ of D is initially in the equal superposition of all the possible N states and that the quantum oracle O “marks” the specific states satisfying the condition $f(x) = \delta$, the vector $\overline{\nu}$ becomes:

$$\overline{\nu}_i = \begin{cases} 1/\sqrt{N} & f(i) \neq \delta, \\ -1/\sqrt{N} & f(i) = \delta. \end{cases} \tag{3.49}$$

Based on Eq. (3.49), each element of the averaging vector $\overline{a}_i$ becomes equal to:

$$\overline{a}_i = \frac{N-2}{N\sqrt{N}}, \ \forall i \in \{0, 1, ..., N-1\}, \tag{3.50}$$

while after the application of D, each element of $\overline{k}$ becomes:

$$\overline{k}_i = \begin{cases} \frac{N-4}{N}\frac{1}{\sqrt{N}} & f(i) \neq \delta, \\ \frac{3N-4}{N}\frac{1}{\sqrt{N}} & f(i) = \delta. \end{cases} \tag{3.51}$$

Consequently, the amplitude of the marked elements will be amplified by a factor $(3N - 4)/N$, while the amplitudes of the other elements would be attenuated by a factor of $(N-4)/N$, since we have:

$$\frac{3N-4}{N} > 1 > \frac{N-4}{N}, \ \forall N > 2. \tag{3.52}$$

Let us now consider two tutorial examples. We assume that the total number of solutions for the first one is $N = 4$, while for the second we have $N = 8$. Additionally, we have $f(x) = x$ for the operator U_f and the intended solution is set to $\delta = 1$. Note that we have expressed δ in the decimal basis. Applying Eqs. (3.49)-(3.51) in our exemplified cases we get:

$$\overline{\nu}_{1,i} = \begin{cases} 1/2 & i \neq 1, \\ -1/2 & i = 1. \end{cases}, \ \overline{a}_{1,i} = \frac{1}{4}, \ \overline{k}_{1,i} = \begin{cases} 0 & i \neq 1, \\ 1 & i = 1. \end{cases} \tag{3.53}$$

$$\overline{\nu}_{2,i} = \begin{cases} \frac{1}{2\sqrt{2}} & i \neq 1, \\ -\frac{1}{2\sqrt{2}} & i = 1. \end{cases}, \ \overline{a}_{2,i} = \frac{3}{8\sqrt{2}}, \ \overline{k}_{2,i} = \begin{cases} \frac{1}{4\sqrt{2}} & i \neq 1, \\ \frac{5}{4\sqrt{2}} & i = 1. \end{cases} \tag{3.54}$$

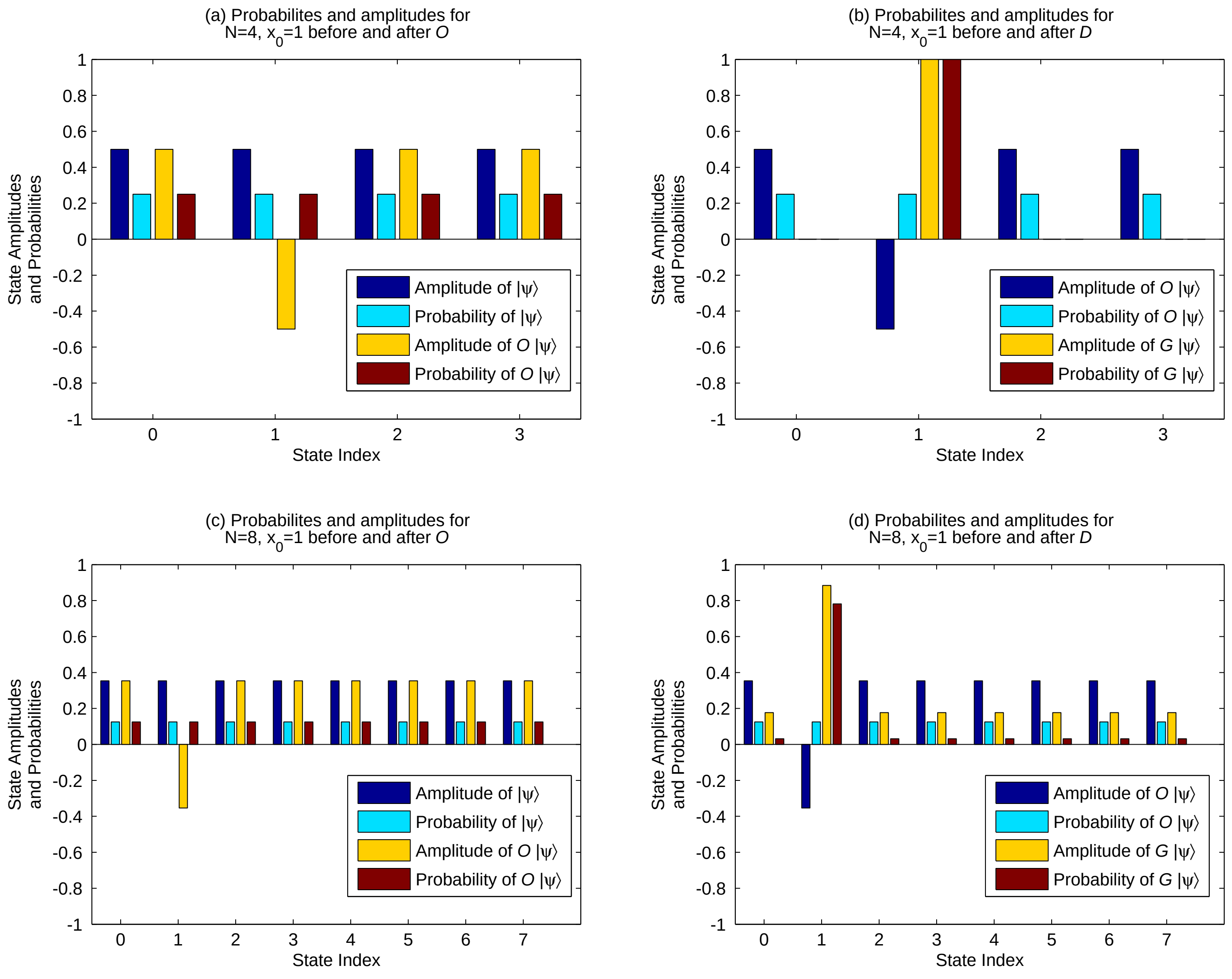


Figure 3.4: Amplitudes and probabilities of the states for one Grover Iteration for the cases of $N = 4$ (a,b) and $N = 8$ (c,d); the case study involves the evolution of the amplitudes and the respective probabilities of the states before and after the application of the quantum oracle O (a,c) and before and after the application of the Diffusion Matrix D (b,d).

The theoretical results extracted from Eqs. (3.53) and (3.54) corresponding to $N_1 = 4$ and to $N_2 = 8$, respectively, are confirmed by the simulation results of the Fig. 3.4. It may be readily observed from Figs. 3.4(a,c) that the application of the oracle gate O only alters the phase of the intended state $|1\rangle$, but not the respective probabilities of the states. The latter is carried out by the application of the Diffusion Matrix D, as it is portrayed in Figs. 3.4(b,d), where the amplitude and, consequently, the probability of $|1\rangle$ is also amplified, whereas the amplitudes and the respective probabilities of the other states are reduced, as expected. Last but not least, comparing the outcome of a single application of the *Grover Operator* $\mathcal{G}$ for the two cases, it seems that a single application is sufficient for $N_1 = 4$ for detecting the state $|1\rangle$ with a probability equal to unity; however, for $N_2 = 8$ a single application of $\mathcal{G}$ results in detecting $|1\rangle$ with a reduced probability around 0.78. Bearing this in mind, in the next subsection we will provide some discussions on the number of $\mathcal{G}$ applications that maximize the probability of successfully detecting the valid solutions.

3.3.2 Maximizing of the Probabilty of Successful Detection

In the previous section the effect of a single *Grover Iteration* in the quantum system state has been discussed and the impact of the number of $\mathcal{G}$ applications has been considered with the aid of examples. In fact, this issue is of utter importance, since it is directly related to the computational complexity of Grover's QSA. At this point, we remind that a single application of $\mathcal{G}$ involves a single CFE, which is quantified in terms of the number of U_f activations. The tight bounds on Grover's QSA complexity been derived by Boyer *et al.* [80], while aiming for maximizing its success probability. They have also extended the search problem to the case where multiple valid solutions exist in a database. Hence, in this chapter we will utilize their methodology so as to derive the optimal number of $\mathcal{G}$ applications, given the size of the database N and the number of valid solutions t.

Let us now define the set S containing all the possible solutions, the set S_1 containing the valid solutions and S_0 as the set containing the invalid solutions, i.e. solutions where we have $f(x) \neq \delta$. Naturally, the sets S_1 and S_0 are mutually exclusive, i.e. we have $S_1 \cap S_0 = \varnothing$, while their union is equal to set S. In other words, we have $S_1 \cup S_0 = S$. Hence, the binary oracle function $f(x, \delta)$ of Eq. (3.25) can be modified in order to describe the existence of multiple valid solutions as follows:

$$f(x,\delta) = \begin{cases} 1 & x \in S_1, \\ 0 & \text{otherwise.} \end{cases} \tag{3.55}$$

Let us define furthermore two eigenvectors $|\Psi_1\rangle$ and $|\Psi_1\rangle$; the first one consists of the convex sum of the valid solution states, while the latter exclusively contains the invalid states. The states of the aforementioned eigenvectors are formulated as follows:

$$|\Psi_0\rangle = \frac{1}{\sqrt{N-t}} \sum_{x \in S_0} |x\rangle, \tag{3.56}$$

$$|\Psi_1\rangle = \frac{1}{\sqrt{t}} \sum_{x \in S_1} |x\rangle, \tag{3.57}$$

where again, t corresponds to the number of valid solutions. Therefore, assuming a pair of real numbers k_j and l_j satisfying the condition $tk_j^2 + (N-t)l_j^2 = 1$ and using the eigenvectors of Eqs. (3.56) and (3.57) the system state after j applications of $\mathcal{G}$ can be formulated as follows [80]:

$$|\psi_j\rangle \equiv \mathcal{G}^j |\psi\rangle = \sqrt{t}k_j |\Psi_1\rangle + \sqrt{N-t}l_j |\Psi_0\rangle . \tag{3.58}$$

Naturally, the amplitude k_j corresponds to the amplitude of a valid solution, while l_j corresponds to the amplitude of an invalid solution. During the initialization of Grover's QSA , in other words for $j = 0$, the state ψ is initialized to the superposition of all the available states yielding:

$$k_0 = l_0 = \frac{1}{\sqrt{N}}. \tag{3.59}$$

After the application of the oracle O the phase of the states of the eigenvector $|\Psi_1\rangle$ is flipped by π; hence, the *averaging vector* $\overline{a}$ [80] right before applying the diffusion matrix D becomes equal to:

$$\overline{a}_i = -\frac{t}{N}k_j + \frac{N-t}{N}l_j. \tag{3.60}$$

Hence, the amplitudes k_{j+1} and l_{j+1} for the $(j+1)$-th iteration are derived using Eq. (3.47) as follows [80]:

$$k_{j+1} = 2\overline{a}_i - (-k_j) = \frac{N-2t}{N}k_j + \frac{2(N-t)}{N}l_j, \tag{3.61}$$

$$l_{j+1} = 2\overline{a}_i - l_j = -\frac{2t}{N}k_j + \frac{N-2t}{N}l_j. \tag{3.62}$$

Therefore, a recursive formula has been derived for the calculation of the amplitudes of both the valid and invalid solutions. At this point, let us define the angle θ as the angle that the argument of the Grover's QSA initial state $|\psi_0\rangle$ defined in Eq. (3.36) forms with the eigenvector $|\Psi_0\rangle$ in the $|\Psi_0\rangle |\Psi_1\rangle$ plane. Consequently, for the sake of quantifying this specific angle, the inner product of the vectors $|\psi_0\rangle$ and $|\Psi_1\rangle$ is utilized, yielding:

$$\sin\theta = \langle\psi_0|\Psi_1\rangle = \sqrt{\frac{t}{N}}. \tag{3.63}$$

Using standard algebraic techniques and Eqs. (3.61), (3.62) and (3.63), it is possible to derive closed form expressions for the amplitudes, hence circumventing the employment of recursive formulae [80]:

$$\left.\begin{array}{lcl} k_j & = & \frac{1}{\sqrt{t}} \sin\left((2j+1)\theta\right) \\ l_j & = & \frac{1}{\sqrt{N-t}} \cos\left((2j+1)\theta\right) \end{array}\right\}, \tag{3.64}$$

while the system state as a function of the number j of $\mathcal{G}$ applications is derived by substituting Eq. (3.64) into Eq. (3.58) as follows:

$$|\psi_j\rangle \equiv \mathcal{G}^j |\psi\rangle = \sin\left[(2j+1)\theta\right] |\Psi_1\rangle + \cos\left[(2j+1)\theta\right] |\Psi_0\rangle . \tag{3.65}$$

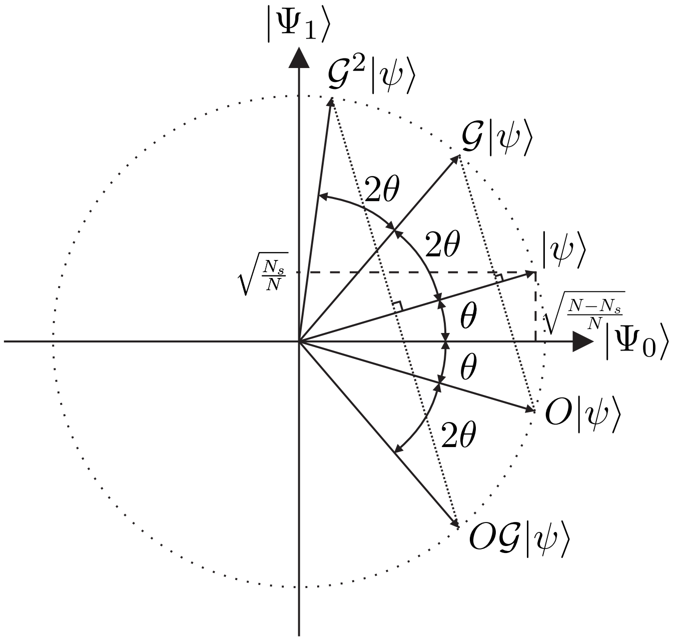


Figure 3.5: Geometric representation of Grover applications; the oracle gate O results in mirroring the system argument about the $|\Psi_0\rangle$ axis, while a single application of $\mathcal{G}$ would shift the phase of the system argument by 2θ counter-clockwise.

It becomes clear from Eq. (3.65) that a single $\mathcal{G}$ application results in shifting the system state's phase by 2θ in the $\Psi_0\Psi_1$ plane. The geometric representation of the QSA is shown in Fig. 3.5, where we can observe that the oracle gate O mirrors the system state about the axis $|\Psi_0\rangle$, whereas the diffusion matrix D mirrors the outcome of O about the axis of the initial state $|\psi\rangle$.

Having provided a closed-form relationship between the state amplitudes and the number of $\mathcal{G}$ applications, we may now proceed by evaluating the specific number L_{opt} of $\mathcal{G}$ applications, which maximizes the probability of successfully detecting the correct solutions during the observation of measurement operation. Based on Eq. (3.65) it becomes clear that the detection error probability is equal to the probability of detecting the eigenvector $|\Psi_0\rangle$. Consequently, the particular number $\tilde{m}$ of $\mathcal{G}$ applications minimizing this specific unsuccessful search probability occurs is given by [80]:

$$cos^2[(2\tilde{m}+1)\theta] = 0, \tag{3.66}$$

which in turn yields:

$$\tilde{m} = \frac{\pi - 2\theta}{4\theta}. \tag{3.67}$$

However, this number $\tilde{m}$ has to be an integer. Let us now define $m = \lfloor \frac{\pi}{4\theta} \rfloor$, where $|m - \tilde{m}| \leq 1/2$. Then, following the methodology proposed in [80] we get:

$$|(2m+1) - (2\tilde{m}+1)| \leq 1,$$

$$|(2m+1)\theta - (2\tilde{m}+1)\theta| \leq \theta.$$

However, based on the the definition of $\tilde{m}$ we have $(2\tilde{m}+1)\theta = \pi/2$, thus yielding [80]:

$$|(2m+1)\theta - \pi/2| \leq \theta,$$

$$|\sin\left((2m+1)\theta - \pi/2\right)| \leq |\sin\theta|,$$

$$\cos^2\left[(2m+1)\theta\right] \leq \sin^2\theta = t/N \approxeq 0, \text{ when } t \ll N. \tag{3.68}$$

Therefore, the probability of detecting an invalid solution is minimized for $L = m$. Furthermore, by exploiting $\theta \geq \sin\theta$, the optimal number L_{opt} of $\mathcal{G}$ applications is given by [80]:

$$L_{opt} = m = \left\lfloor \frac{\pi}{4\theta} \right\rfloor \leq \frac{\pi}{4\theta} \leq \frac{\pi}{4}\sqrt{\frac{N}{t}}, \tag{3.69}$$

$$L_{opt} = \left\lfloor \frac{\pi}{4}\sqrt{\frac{N}{t}} \right\rfloor. \tag{3.70}$$

Eq. (3.70) indicates that Grover's QSA imposes a computational complexity, which is quantified in terms of O queries on the order of $O\left(\sqrt{N/t}\right)$, providing a *quadratic* complexity reduction [79]. In fact, all the QAAAs using the operator $\mathcal{G}$ benefit from this substantial reduction. Naturally, observe from Eq. (3.66) that it is possible to derive the probability P_s of finding a valid solution as a function of the number of $\mathcal{G}$ applications as follows [80]:

$$P_s = \sin^2\left[\left(2L_{opt}+1\right)\theta\right], \tag{3.71}$$

where we have $\theta = \arcsin\sqrt{t/N}$ and L_{opt} corresponds to the number of $\mathcal{G}$ applications.

3.3.3 An Illustrative Example

Let us now provide a multiple-objective routing example for making the somewhat abstract arithmetic examples used in the previous subsections more plausible. We will consider the fully interconnected 4-node WMHN portrayed in Fig. 3.6, which obeys the assumptions summarized in Table 2.1 and relies on the UV $\mathbf{f}(x) = [P_{e,x}, CL_x]$. Note that we have utilized the first two UF of the UV of Eq. (2.5) for the sake of simplicity. In terms of this tutorial, the evaluation of the routes is carried out through their Pareto distance $P_d(x)$ defined in Definition 5. We will assume that we have the perfect *a priori* knowledge of the

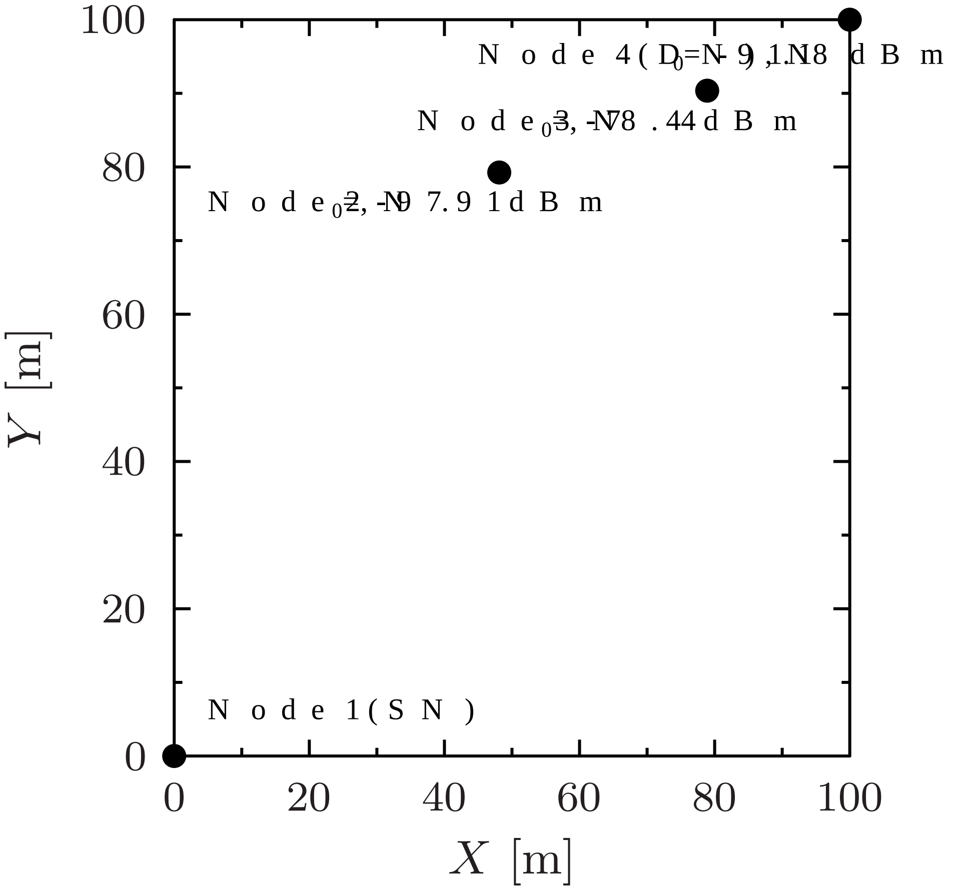


Figure 3.6: Exemplified 4-node WMHN topology based on the system model of Table 2.1 and relying on the UV $\mathbf{f}(x) = [P_{e,x}, CL_x]$. The associated routes as well as their their Pareto distances are shown in Table 3.1.

number of solutions t, which is equal to $t = 2$ in our scenario . In the Pareto Optimality problem defined in Eq. (2.7) we have prior knowledge of the optimal Pareto distance, which by definition would be equal to $\delta = 0$. Moreover, it should be noted that since the total number of routes is equal to 5, the length of the QIR would be equal to 3 qubits, since $2^3 = 8$ is the next integer power of 2 larger than 5. The non-existent routes, which are marked as "N/A" in Table 3.1 are assumed to be dominated by all other routes, hence their Pareto distance would be 1. Since we have the knowledge of $t = 2$, the required number of Grover operator with the aid of applications Eq. (3.70) is given by :

$$L = \left\lfloor \frac{\pi}{4}\sqrt{\frac{N}{t}} \right\rfloor = \left\lfloor \frac{\pi}{4}\sqrt{\frac{8}{2}} \right\rfloor = \left\lfloor \frac{\pi}{2} \right\rfloor = \lfloor 1.57 \rfloor = 1. \tag{3.72}$$

Therefore, a single application of $\mathcal{G}$ operator is sufficient for maximizing the probability of successfully detecting a valid solution. Based on Eq. (3.63), the angle θ is calculated as follows:

$$\theta = \sin^{-1}\sqrt{\frac{t}{N}} = \sin^{-1}\sqrt{\frac{2}{8}} = \frac{\pi}{6}. \tag{3.73}$$

Hence, we may readily calculate the amplitudes of the states by substituting Eq. (3.73)

Table 3.1: Routes and their Pareto distances for the 4-node WMHN of Fig. 3.6 based on the system model of Table 2.1 and relying on the UV $\mathbf{f}(x) = [P_{e,x}, CL_x]$.

Routes	x	$P_{e,x}(\times 10^{-4})$	CL_x [dBm]	$P_d(x)$	$f(x,0)$
$1 \to 4$	0	0.99	74.15	0.4	0
$1 \to 3 \to 4$	1	11.29	72.03	0.4	0
$1 \to 2 \to 4$	2	0.12	69.51	0.0	1
$1 \to 2 \to 3 \to 4$	3	0.29	68.90	0.0	1
$1 \to 3 \to 2 \to 4$	4	11.35	72.50	0.6	0
N/A	5	$+\infty$	$+\infty$	1.0	0
N/A	6	$+\infty$	$+\infty$	1.0	0
N/A	7	$+\infty$	$+\infty$	1.0	0

into Eq. (3.64), which leads to:

$$\left.\begin{array}{rcl} k_1 & = & \frac{1}{\sqrt{2}}\sin(3\pi/6) \\ l_1 & = & \frac{1}{\sqrt{8-2}}\cos(3\pi/6) \end{array}\right\} \Leftrightarrow \left.\begin{array}{rcl} k_1 & = & \frac{1}{\sqrt{2}} \\ l_1 & = & 0 \end{array}\right\}, \tag{3.74}$$

yielding the state final state $|\psi_f\rangle$ before the observation or measurement operation equal to:

$$|\psi_f\rangle = |\psi_1\rangle = \left(0, 0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, 0, 0, 0\right)^T. \tag{3.75}$$

Thus, each of the states $|2\rangle$ and $|3\rangle$, which are Pareto Optimal, will be detected with a probability of 0.5 each. We note that the outcome is optimal, since we have $l_1 = 0$, resulting in a zero probability of unsuccessful detection. Despite its optimality, there some further issues to be considered in the context of Grover's QSA. On the one hand, the number t of valid solutions has to be known beforhand; otherwise, it is not possible to evaluate the exact number L_{opt} of $\mathcal{G}$ applications. Hence, satisfying this condition is vital for the optimality of the Grover's QSA. On the other hand, it is not possible to guarantee that all the routes belonging to the OPF will be identified, since the observation or measurement operation is a random process, which depends on the state amplitudes.

3.4 Boyer-Brassard-Høyer-Tapp Quantum Search Algorithm

As discussed in Section 3.3, Grover's QSA has the substantial limitation of requiring *a priori* knowledge of both the solution value and of the number t of solutions. Even though Brassard *et al.* [85] proposed a quantum algorithm for counting the number of valid solutions, this process involves a rather excessive complexity overhead. The BBHT-QSA [80] eliminates this limitation albeit the actual value of the solution still has to be known, similar to Grover's QSA.

In simplified terms, the BBHT-QSA involves a number of Grover iterations. In contrast to Grover's QSA, the BBHT-QSA makes use of an random process which selects the number L of $\mathcal{G}$ applications from an integer range of $\{0, ..., \lfloor m \rfloor\}$, for a particular positive real number m. Grover's QSA is then activated L times and the measured state $|j\rangle$ is exported. Subsequently, the algorithm checks as to whether the exported state $|j\rangle$ belongs to the set S_1 of valid solutions, i.e. whether $f(j, \delta) = 1$ is satisfied. If $j \in S_1$ or the maximum affordable number of $\mathcal{G}$ applications $L^{\max}_{BBHT}$ has been exhausted, the algorithm is terminated by exporting this particular solution. Otherwise, the integer range maximum is increased by a factor λ – which is a real number strictly between 1 and 4/3 – and a new number of $\mathcal{G}$ applications is selected from the updated range. We note that since the maximum number of Grover iterations has been shown in Eq. (3.70) to be equal to $\sqrt{N}$ corresponding to the case, where only a single valid solution is present, the maximum of the range is given by this number $\mathcal{G}$ applications. Consequently, each time the upper bound m of the range is updated, it constrains the maximum of the range to be the minimum between λm and $\sqrt{N}$. This process is repeated, until either a valid solution is detected or the maximum number of Grover iterations $L^{\max}_{BBHT}$ has been exhausted. As far as the initialization parameters are concerned, following the approach of [85], λ is set equal to 6/5 and m is to 1. Based on these parameters, the maximum required[1] number of Grover iterations $L^{\max}_{BBHT}$ was shown to be [85]:

$$L^{\max}_{BBHT} = 4.5\sqrt{\frac{N}{t}} \leq 4.5\sqrt{N}, \tag{3.76}$$

where the upper bound corresponds to the case, where only a single valid solution is present in the database. Since no knowledge of the number of solutions is provided beforehand, the maximum affordable number of $\mathcal{G}$ applications is set to:

$$L^{\max}_{BBHT} = \left\lfloor 4.5\sqrt{N} \right\rfloor. \tag{3.77}$$

Observe in Eq (3.77) that the BBHT-QSA's complexity quantified in terms of the O queries is on the order of $O(\sqrt{N})$; hence, this algorithm succeeds in providing us with a substantial complexity reduction. The detailed steps of the BBHT-QSA are shown in Alg. 3.2, while its flowchart is portrayed in Fig. 3.7.

Algorithm 3.2 BBHT Quantum Search Algorithm [85]

```
1: Set $m \leftarrow 1$, $\lambda \leftarrow 6/5$ and $L_{BBHT} \leftarrow 0$.
2: Choose $L$ uniformly from the set $\{0, \ldots, \lfloor m \rfloor\}$.
3: Apply the $\mathcal{G}$ operator $L$ times starting from the initial state $|\psi\rangle$ in (3.36), resulting in
   the final state $|\psi_f\rangle = \mathcal{G}^L |\psi\rangle$.
4: Observe $|\psi_f\rangle$ and obtain $|j\rangle$.
5: Update $L_{BBHT} \leftarrow L_{BBHT} + L$.
6: if $f(j) = 1$ or $L_{BBHT} \geq L^{\max}_{BBHT}$ then
7:     Set $x_s \leftarrow j$, output $x_s$ and exit.
8: else
9:     Set $m \leftarrow \min\left\{\lambda m, \sqrt{N}\right\}$ and go to Step 2.
10: end if
```

[1]Required for achieving ~100% success probability.

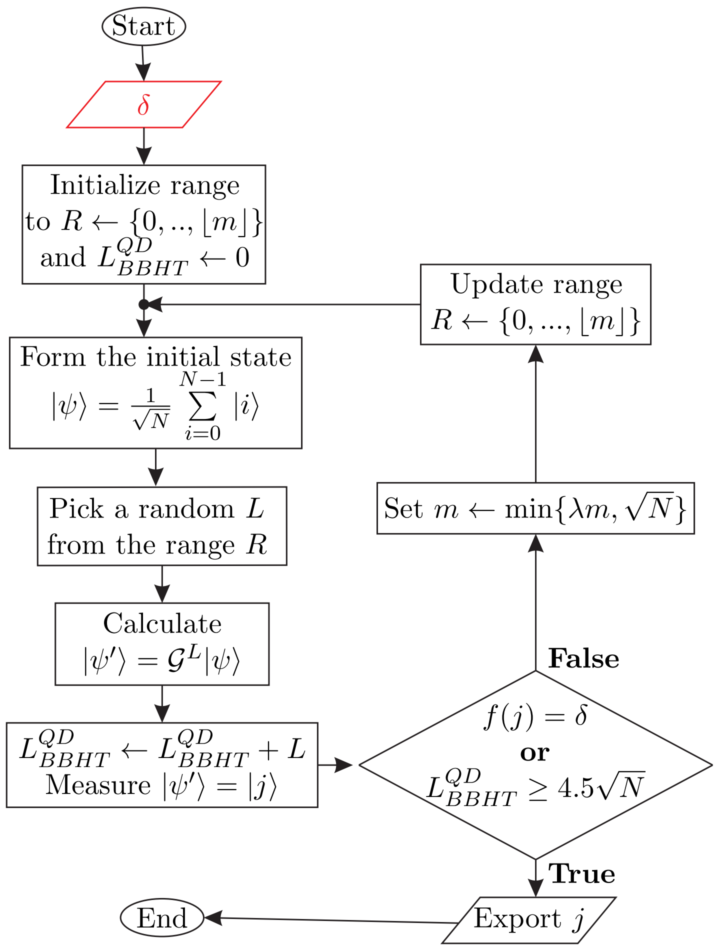

Start
δ
Initialize range to $R \leftarrow \{0, .., \lfloor m \rfloor\}$ and $L^{QD}_{BBHT} \leftarrow 0$
Form the initial state $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle$
Pick a random L from the range R
Calculate $|\psi'\rangle = \mathcal{G}^L |\psi\rangle$
$L^{QD}_{BBHT} \leftarrow L^{QD}_{BBHT} + L$ Measure $|\psi'\rangle = |j\rangle$
$f(j) = \delta$ or $L^{QD}_{BBHT} \geq 4.5\sqrt{N}$
False
Set $m \leftarrow \min\{\lambda m, \sqrt{N}\}$
Update range $R \leftarrow \{0, ..., \lfloor m \rfloor\}$
True
Export j
End

Figure 3.7: BBHT-QSA's flowchart. Note that the **red** input block hightlights the assumption of knowing the sought value δ.

Let us now consider again the routing problem example of Section 3.3.3, as characterized in Table 3.1. The only difference in terms of the assumptions compared to the Grover's QSA example is that the actual number t of valid solutions is now unknown. During the initialization process, λ is set to $\lambda = 6/5$, m is set to $m = 1$ and the maximum affordable number of $\mathcal{G}$ applications to $L^{\max}_{BBHT} = \lfloor 4.5\sqrt{8} \rfloor = 12$. In the first BBHT-QSA iteration a random number is selected from the range of $\{0, 1\}$. Assuming that we opted for $L = 0$ $\mathcal{G}$ applications, no $\mathcal{G}$ applications are used. Hence, the observation outcome is given by any of the states, since the initial state $|\psi\rangle$ is in the equal-wighted superposition of all the possible states. Note that the probability of successful detection is equal to $P_s = t/N = 0.25$. Let us assume that the state $|6\rangle$ is observed, for which we have $f(6, 0) = 0$. Since $f(6, 0) \neq 1$ and $L_{BBHT} = 0 < L^{\max}_{BBHT}$, the range upper bound m is increased by a factor of $6/5$. Therefore, now m would be equal to $6/5$; nevertheless, the integer range still remains the same, i.e. equal to $\{0, 1\}$. Assuming now that L is chosen to be 1, the probability of successful search after a single application of $\mathcal{G}$ becomes equal to 1, since the $|2\rangle$ and $|3\rangle$ states' amplitudes and probabilities become equal to $1/\sqrt{2}$ and $1/2$, respectively, based on Eq. (3.74). Hence, let us assume that the state $|2\rangle$ is detected; since we have $f(2, 0) = 1$, the algorithm will output the solution and will exit. Finally, we note that since the algorithm is probabilistic, the actual simulation produces different outcomes each time it is invoked.

3.5 Dürr-Høyer Algorithm

In Sections 3.3 and 3.4, we presented QSAs tailored for addressing specific search problems, where explicit knowledge of the value sought is available. Nevertheless, when performing optimization this is scarcely the case. Naturally, when attempting to minimize or maximize an arbitrary function $f(x)$, the actual optimal value $f(x_{\text{opt}})$ is unknown to the optimization process, hence making the employment of the BBHT-QSA infeasible. In classical computing, the *Exhaustive Search* (ES) method, often referred to as *Brute-Force* (BF) method as well, compares the respective value of each possible input to an appropriately initialized buffer, which is then updated to the hitherto best value after every comparison, resulting in a complexity of N comparisons.

Against this background, an algorithm was proposed by Dürr and Høyer in [81] for the sake of addressing these optimization problems, whilst imposing a complexity on the order of $O(\sqrt{N})$, i.e. providing a substantial complexity reduction. This algorithm is often referred to as the *Dürr-Høyer Algorithm* (DHA). The DHA is formally presented in Alg. 3.3, while its respective flowchart is shown in Fig. 3.9. To elaborate further, the DHA uses a buffer in the same manner as the classical ES method. However, in the DHA this buffer is not compared to every possible input in a serial fashion. Instead, a BBHT-QSA process is activated searching for a better solution than the buffered value. In this fashion, the optimization problem is decomposed into a series of search problems, which can be solved using the BBHT-QSA. If the BBHT-QSA succeeds in finding a better solution, the buffer is updated to this specific solution and a new BBHT-QSA iteration is invoked with its input δ set to the updated buffer value. In any other case, a new BBHT-QSA is invoked using the same buffer input until the maximum affordable number of CFEs[2] is exhausted. Using this limit is essential for avoiding infinite loops, when reaching the finding optimal solution. Explicitly, this limit was shown to be [81]:

$$L_{DHA}^{QD,\max} = 22.5\sqrt{N}, \tag{3.78}$$

which provides us with ~100% probability of successfully finding the optimal solution.

In terms of its implementation, the DHA iteratively invoked the BBHT-QSA, which in turn utilizes a modified quantum oracle gate O in the sense that the buffer input has to be updated before it is taken into account by the oracle gate. For this reason, the controlled input unitary operator U_f is used, which invokes a binary function $f(x,i)$ defined as follows:

$$f(x,i) = \begin{cases} 1 & f(x) < f(i), \\ 0 & f(x) \geq f(i), \end{cases}, \tag{3.79}$$

where the input i denotes the index stored in the buffer, which is referred to as the *reference state*, while x accounts for all the legitimate inputs. We note that Eq. (3.79) corresponds to the minimization case, while for the maximization one the logical operators will have

[2]We have considered as a single CFE the event of applying the $\mathcal{G}$ operation once.

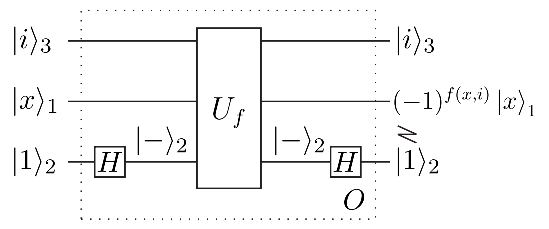


Figure 3.8: Quantum circuit of the quantum oracle gate O, which implements the binary function $f(x,i)$; even though O would alter the phase of $|-\rangle_2$ due to the *entanglement*, it would be valid to assume that $|x\rangle_1 [(-1)^{f(x,i)}|-\rangle_2] = [(-1)^{f(x,i)}|x\rangle_1]|-\rangle_2$ [112]. These quantum oracle gates belong to the family of *Phase-Kickback Quantum Circuits* [67] since they flip the phase of some selected states.

the opposite direction. The new O quantum circuit is shown in Fig. 3.8.

Algorithm 3.3 DHA for minimization problems [81]

```
1: Choose a threshold index $0 \leq y \leq N-1$ randomly using uniform distribution.
2: Set $L_{DHA}^{QD} \leftarrow 0$.
3: repeat
4:     Define the quantum oracle implementing the binary function $f(x,i)$ of Eq. (3.79)
       and set $i \leftarrow y$.
5:     Invoke the BBHT-QSA process of Alg. 3.2 with input the function $f(x,y)$ and output
       $y'$ using $L_{BBHT}^{QD}$ CFEs.
6:     Set $L_{DHA}^{QD} \leftarrow L_{DHA}^{QD} + L_{BBHT}^{QD}$
7:     if $f(y') < f(y)$ then
8:         Set $y \leftarrow y'$.
9:     end if
10: until $L_{DHA}^{QD} < \lceil 22.5\sqrt{N} \rceil$.
11: Output $y$ and exit.
```

Let us now provide an example for illustrating the operation of the DHA. We note that we will not refer to the inner BBHT-QSA iterations of the DHA in this tutorial part for the sake of avoiding redundancy. Once again, we will consider the 4-node WMHN of Fig. 3.6 relying on the assumptions of Table 2.1 and the multiple-objective routing example of Table 3.1. Since we have $N = 8$, the DHA timeout will be set to $L_{DHA}^{QD,\max} = 64$ CFEs based on Eq. 3.78. Naturally, in our example the function $f(x)$ of Eq. (3.79) corresponds to the Pareto distance function $P_d(x)$ defined in Eq. 2.6. Assuming that the minimum Pareto distance P_d is not known, the DHA randomly selects a route obeying a uniform distribution from all the possible route-solutions, as stated in Step 3.3.1 of Alg. 3.3. Let us assume that we have opted[3] for $y = 4$, i.e. the DHA has selected the route 1→3→2→4 with $P_d(4) = 0.6$. Subsequently, the CFE counter is initialized to zero in Step 3.3.2, i.e. we have $L_{DHA}^{QD} = 0$, and a BBHT-QSA process is invoked in order to search for routes satisfying the condition $P_d(x) < P_d(4) = 0.6$ in Step 3.3.4. In fact, there exist four routes, implying that half the routes are valid solutions. In the BBHT-QSA process there will be a 50% probability according to Eq. (3.71) of finding a solution regardless of the number of $\mathcal{G}$ applications. For simplicity, let us assume that a valid solution is found after $L_{BBHT}^{QD} = 1$

[3]We have assumed that the route index numbering starts from the value 0.

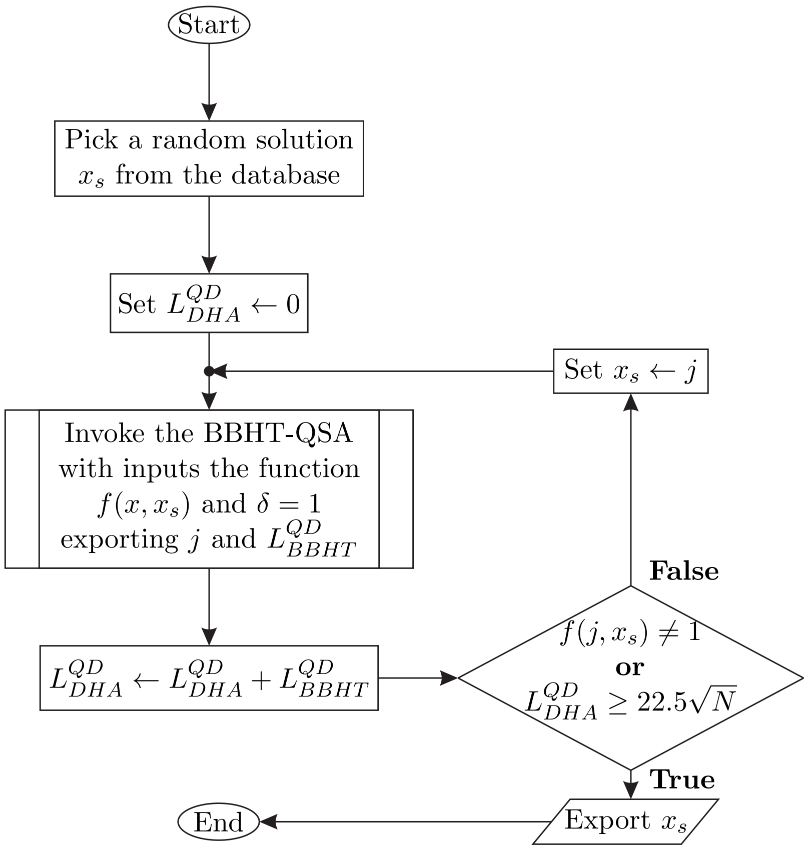

Start
Pick a random solution x_s from the database
Set $L^{QD}_{DHA} \leftarrow 0$
Set $x_s \leftarrow j$
Invoke the BBHT-QSA with inputs the function $f(x, x_s)$ and $\delta = 1$ exporting j and L^{QD}_{BBHT}
False
$L^{QD}_{DHA} \leftarrow L^{QD}_{DHA} + L^{QD}_{BBHT}$
$f(j, x_s) \neq 1$ or $L^{QD}_{DHA} \geq 22.5\sqrt{N}$
True
End
Export x_s

Figure 3.9: DHA's flowchart.

CFEs, namely the route 1→3→4 associated with $y' = 1$ and $f(y') = 0.4$. The total number of $\mathcal{G}$ applications is updated, i.e. we have $L^{QD}_{DHA} = 1$ in Step 3.3 and, since $P_d(1) < P_d(4)$ a new BBHT-QSA process is activated in conjunction with $y = 1$. Observe in Table 3.1 that only two routes exist out of the eight in total with a Pareto distance less than that of the route associated with $y = 1$. Hence, based on Eq. (3.71), a single $\mathcal{G}$ application would be sufficient for having a 100% probability of observing a valid route-solution. In its initialization process, the BBHT-QSA selects the number of $\mathcal{G}$ applications to be used from the range $\{0, 1\}$. Let us assume that the process opts for applying $L = 0$ $\mathcal{G}$ applications. Then, based on Eq. (3.71) the probability of observing a valid route-solution would be equal to $P_s = 0.25$. Let us assume that the route with index $j = 7$ is observed in the first inner BBHT-QSA iteration. Since $P_d(7) \not< P_d(0)$ the selection range is expanded by a factor $\lambda = 6/5$ and the process attempts to select from the range $\{0, .., \lfloor 6/5 \rfloor\} \equiv \{0, 1\}$. Then, let us assume that the BBHT-QSA opts for applying the $\mathcal{G}$ operator $L = 1$ times, which happens to be optimal. Then the route 1→2→4 with index $y = 2$ is observed, which is a valid solution, since we have $g(2, 1) = 1$, and the BBHT-QSA exits and outputs $y' = 0$. We have $L^{QD}_{BBHT} = 1$ CFE in Step 3.3.4. Since we have $P_d(y' = 2) < P_d(y = 1)$, the buffer value is updated in Step 3.3.8, i.e. we have $y = 2$. Observe now in Table 3.1 that the new reference route index corresponds to an optimal route, since we have $P_d(2) = 0$. Nevertheless, the DHA is unable to ascertain that it has found optimal and, instead, a new BBHT-QSA process is activated in conjunction with $y = 2$. However, in the absence of valid solutions the BBHT-QSA will exhaust the maximum affordable number of $\mathcal{G}$ applications, which is equal to $L_{BBHT}^{QD} = L_{BBHT}^{\max} = \lceil 4.5\sqrt{N} \rceil = 13$ CFEs and will return a random invalid solution. This process is repeated until the DHA has reached its complexity limit. The DHA then terminates by outputting the optimal solution stored in its buffer. Finally, the interactions between the quantum and classical processors in the context of the DHA used in the aforementioned tutorial are encapsulated in Fig. 3.10, while the reference route update process is portrayed in Fig. 3.11.

3.6 Chapter Summary

In this chapter, we provided an introduction to the family of quantum search algorithms. More specifically, the aspects of Grover's QSA [79] have been analyzed in detail in Section 3.3, since this algorithm constitutes the cornerstone of more sophisticated algorithms, such as the BBHT-QSA and DHA of Sections 3.4 and 3.5, respectively. In a nutshell, Grover's QSA is applicable to search problems, where both the number of desired entries and the actual desired entry value are known to the optimization process. For example, this specific QSA is applicable to the unsorted phone book search, in which the search process seeks for a specific person's telephone number and it has the knowledge of how many different telephone numbers are assigned to this specific person. Its direct extension would be the case, where the actual number of solutions is unknown, but the desired value (the person's name) is still known. This problem can be solved by the BBHT-QSA [80] at the expense of some additional CFEs to compensate for this information loss. Note that in the context of our multi-objective routing, Grover's QSA can be invoked in a database containing the Pareto distances of all the legitimate routes in order to search for Pareto-optimal routes associated with $P_d(x) = 0$, while having explicit knowledge of the number of Pareto-optimal routes. The BBHT-QSA is also applicable in this scenario only requiring knowledge of the Pareto distance value associated with Pareto-optimal routes.

The final extension, corresponding to the generalization of the search problem has been made possible in the DHA [81], where only a specific attribute of the desired entry is known, e.g. the fact that it corresponds to the maximum or the minimum of the database, but no other information is available. Naturally, this reduction of the riddle requires further additional CFEs. The DHA is readily applicable to our multi-objective routing problem, where it searches for route-solutions that are associated with the attribute of having minimum Pareto distance. The basic concepts of all three algorithmsare summarized in Fig. 3.12, which provide a substantial complexity reduction. Based on Eqs. (3.70), (3.77) and (3.78), the complexities of all three algorithms are [79, 80, 81]:

$$L_{\text{Grover's QSA}} \equiv L_{opt} = \left\lfloor \frac{\pi}{4}\sqrt{\frac{N}{t}} \right\rfloor = O(\sqrt{N}), \tag{3.80}$$

$$L_{BBHT-QSA} \equiv L_{BBHT}^{QD} = 4.5\sqrt{N} = O(\sqrt{N}), \tag{3.81}$$

$$L_{DHA} \equiv L_{DHA}^{QD} = 22.5\sqrt{N} = O(\sqrt{N}). \tag{3.82}$$

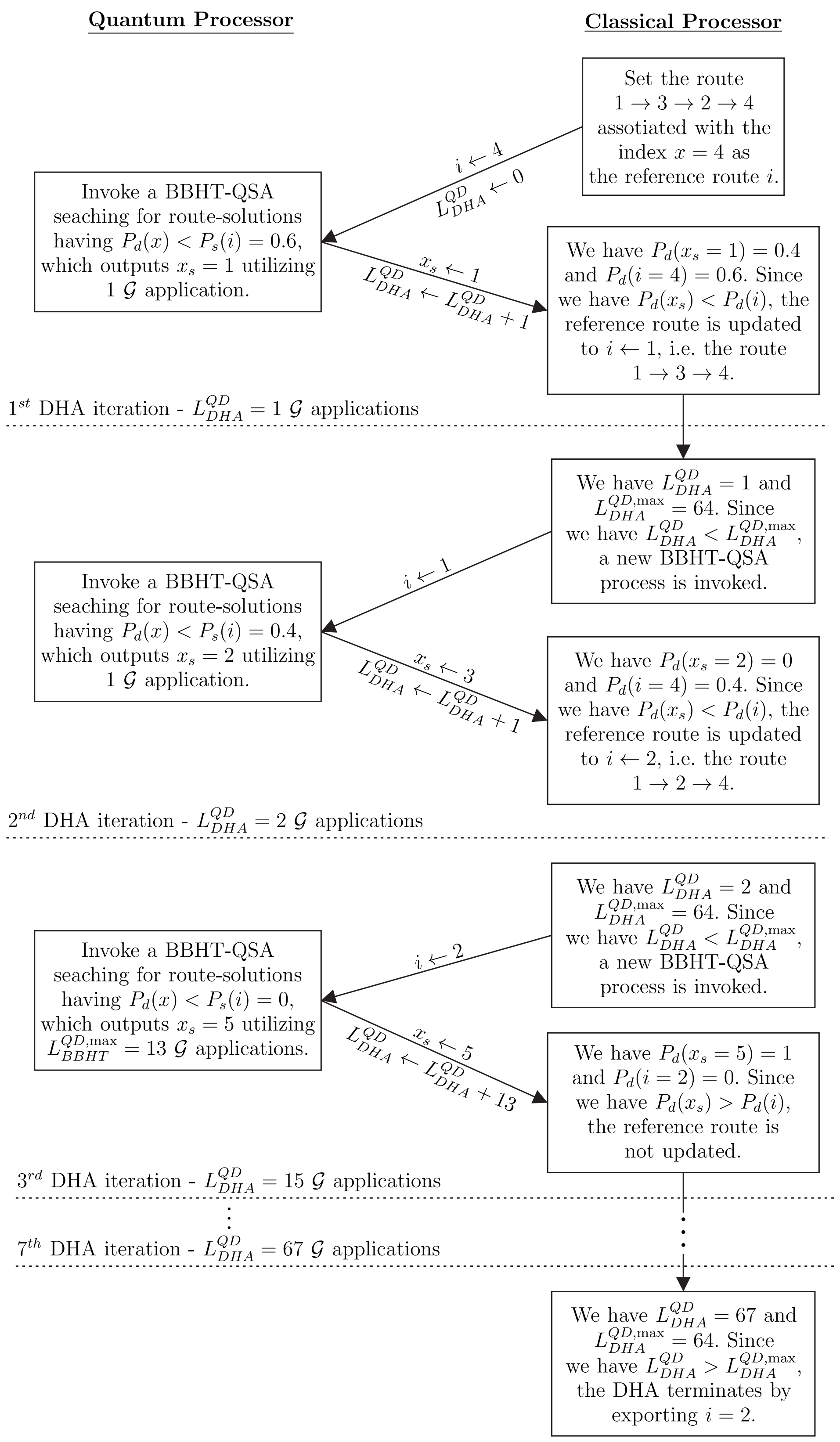

Quantum Processor
Classical Processor
Set the route $1 \rightarrow 3 \rightarrow 2 \rightarrow 4$ assotiated with the index $x = 4$ as the reference route i.
$i \leftarrow 4$
$L_{DHA}^{QD} \leftarrow 0$
Invoke a BBHT-QSA seaching for route-solutions having $P_d(x) < P_s(i) = 0.6$, which outputs $x_s = 1$ utilizing 1 $\mathcal{G}$ application.
$x_s \leftarrow 1$
$L_{DHA}^{QD} \leftarrow L_{DHA}^{QD} + 1$
We have $P_d(x_s = 1) = 0.4$ and $P_d(i = 4) = 0.6$. Since we have $P_d(x_s) < P_d(i)$, the reference route is updated to $i \leftarrow 1$, i.e. the route $1 \rightarrow 3 \rightarrow 4$.
1st DHA iteration - $L_{DHA}^{QD} = 1$ $\mathcal{G}$ applications
We have $L_{DHA}^{QD} = 1$ and $L_{DHA}^{QD,\max} = 64$. Since we have $L_{DHA}^{QD} < L_{DHA}^{QD,\max}$, a new BBHT-QSA process is invoked.
$i \leftarrow 1$
Invoke a BBHT-QSA seaching for route-solutions having $P_d(x) < P_s(i) = 0.4$, which outputs $x_s = 2$ utilizing 1 $\mathcal{G}$ application.
$x_s \leftarrow 3$
$L_{DHA}^{QD} \leftarrow L_{DHA}^{QD} + 1$
We have $P_d(x_s = 2) = 0$ and $P_d(i = 4) = 0.4$. Since we have $P_d(x_s) < P_d(i)$, the reference route is updated to $i \leftarrow 2$, i.e. the route $1 \rightarrow 2 \rightarrow 4$.
2nd DHA iteration - $L_{DHA}^{QD} = 2$ $\mathcal{G}$ applications
We have $L_{DHA}^{QD} = 2$ and $L_{DHA}^{QD,\max} = 64$. Since we have $L_{DHA}^{QD} < L_{DHA}^{QD,\max}$, a new BBHT-QSA process is invoked.
$i \leftarrow 2$
Invoke a BBHT-QSA seaching for route-solutions having $P_d(x) < P_s(i) = 0$, which outputs $x_s = 5$ utilizing $L_{BBHT}^{QD,\max} = 13$ $\mathcal{G}$ applications.
$x_s \leftarrow 5$
$L_{DHA}^{QD} \leftarrow L_{DHA}^{QD} + 13$
We have $P_d(x_s = 5) = 1$ and $P_d(i = 2) = 0$. Since we have $P_d(x_s) > P_d(i)$, the reference route is not updated.
3rd DHA iteration - $L_{DHA}^{QD} = 15$ $\mathcal{G}$ applications
7th DHA iteration - $L_{DHA}^{QD} = 67$ $\mathcal{G}$ applications
We have $L_{DHA}^{QD} = 67$ and $L_{DHA}^{QD,\max} = 64$. Since we have $L_{DHA}^{QD} > L_{DHA}^{QD,\max}$, the DHA terminates by exporting $i = 2$.

Figure 3.10: Interactions between the quantum and the classical processors in the context of employing the DHA for minimizing the Pareto distance in the routing problem of Fig. 3.6 and relying on Table 3.1.

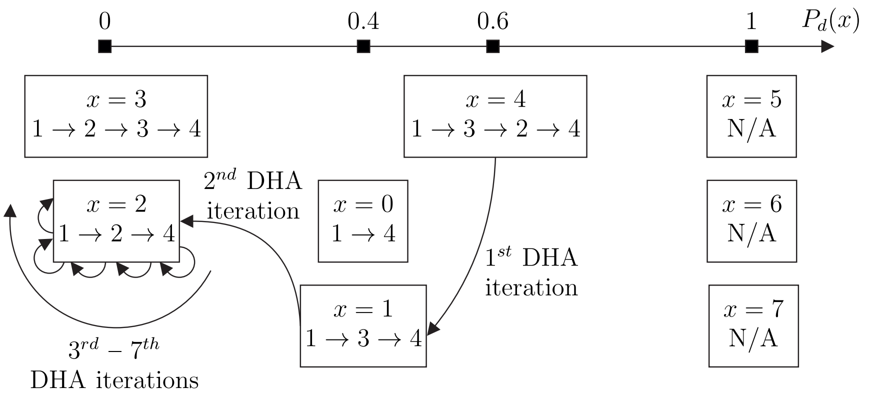


Figure 3.11: Reference route update process in the context of employing the DHA for minimizing the Pareto distance in the routing problem of Fig. 3.6 and relying on Table 3.1.

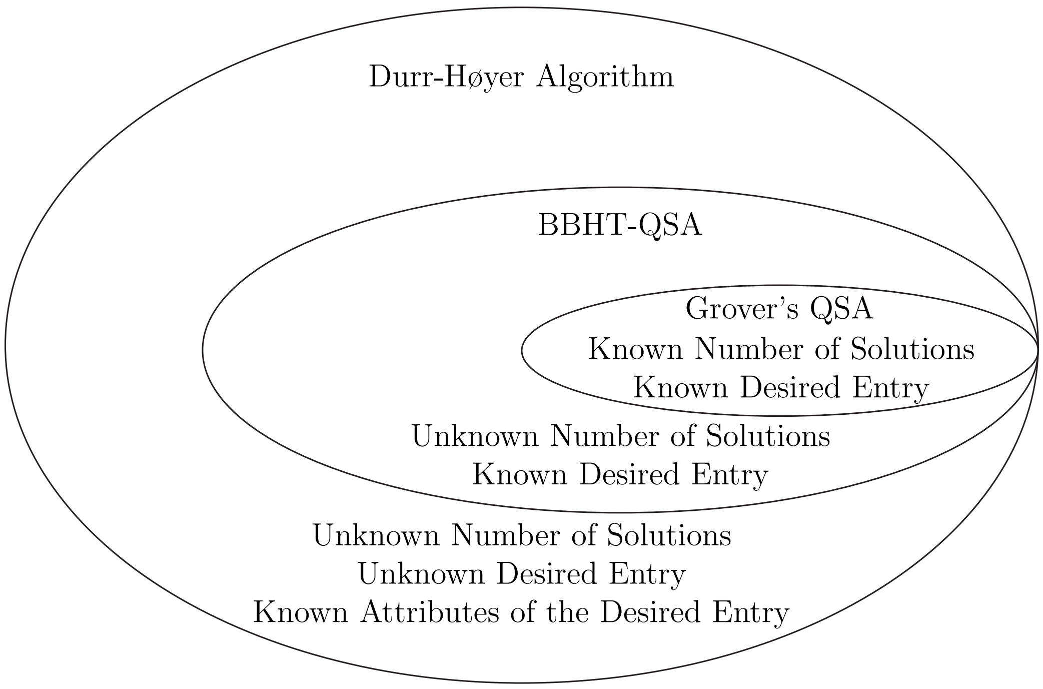

Durr-Høyer Algorithm
BBHT-QSA
Grover's QSA
Known Number of Solutions
Known Desired Entry
Unknown Number of Solutions
Known Desired Entry
Unknown Number of Solutions
Unknown Desired Entry
Known Attributes of the Desired Entry

Figure 3.12: Summary of the existing QSAs.

According Eqs. (3.81) and (3.82), the computational complexity imposed by the BBHT-QSA and the DHA respectively, has been set to the total number of $\mathcal{G}$ operator applications. This assumption in [80, 81] exclusively considers the *Quantum Domain* (QD) CFEs, whilst disregarding the *Classical Domain* (CD) ones, which are imposed in the classical checks undertaken by the Steps 3.2.6 and 3.3.7 for the BBHT-QSA and the DHA, respectively, as it has been mentioned by Botsinis *et al.* in [93] and [94]. In the next chapters, where our quantum-assisted algorithms are presented, we will characterize both the QD-CFEs and CD-CFEs of both the BBHT-QSA and the DHA for quantifying their upper and lower complexity bounds in our multi-objective applications.

Chapter 4

Non-Dominated Quantum Optimization

4.1 Introduction

In Chapter 3, we discussed the specifics of some popular QSAs, namely Grover's QSA [79] and the BBHT-QSA [80] in Sections 3.3 and 3.4, respectively. Furthermore, we provided a brief introduction to quantum-assisted optimization by presenting the DHA in Section 3.5. We have also addressed the multi-objective routing problem using the tutorial presented in Table 3.1 for the sake of identifying a single Pareto-optimal route. In this tutorial example, we postulated that the quantum oracle gate O of the Grover operator $\mathcal{G}$, which is defined in Eq. (3.24), is capable of evaluating the x-th route's Pareto distance $P_d(x)$, which is introduced in Definition 5, while imposing a complexity on the order of $O(1)$. Naturally, this assumption is unrealistic, since the evaluation of the Pareto distance $P_d(x)$ is a rather complex process, imposing a classical complexity on the order of $O(N)$, should classical computation processes be invoked for the sake of calculating the fraction of Eq. (2.6). Fortunately, it is feasible to approximate its value for a specific route using *Quantum Phase Estimation* techniques [85], such as the *Quantum Mean Algorithm* (QMA) [86] or the so-called *Quantum Weighted Sum Algorithm* (QWSA), which was introduced in [87]. Explicitly, both the QMA and the QWSA are capable of estimating a specific route's Pareto distance value at a precision of l bits, while imposing a complexity on the order of $O(2^l)$. Apart from the increased complexity imposed by the QWSA, which scales exponentially by increasing the intended accuracy, the QWSA involves a measurement or observation operation, the specifics of which were detailed in Section 3.2.3. Naturally, this operation acts as a partial measurement [87] on the superimposed composite system state, which consists of the entire set of legitimate routes as well as their respective Pareto distance values. Consequently, the superimposed system state collapses into a pure basis state. Explicitly, this specific operation has a detrimental effect on the QP, thus prohibiting its potential use in conjunction with the Grover operator $\mathcal{G}$.

In addition to the design issues arising from the estimation of the Pareto distance, the entire set of quantum heuristics presented in Chapter 3 is only capable of identifying a single Pareto-optimal route. However, identifying the entire set of Pareto-optimal routes is of great importance, since these specific routes reveal the underlying trade-offs among contradictory optimization objectives [47]. Despite these shortcomings, the aforementioned algorithms constitute an attractive framework due to exhibiting near-optimal accuracy with respect to full-search-based methods, while benefiting from a quadratic complexity reduction upon exploiting the QP. More specifically, in this chapter we present the so-called *Non-Dominated Quantum Optimization* (NDQO) algorithm, which utilizes the BBHT-QSA for the sake of identifying the entire OPF associated with the multiple-objective routing problem in WMHNs, as defined in Eq. (2.5) using the assumptions summarized in Table 2.1. Note that in the system model of Eq. (2.5) we opted for designing the NDQO algorithm for the sake of addressing the weak Pareto optimality problem. Our choice can be justified by the fact that solving the weak Pareto optimality problem is inherently more complex than the strong Pareto optimality, since the set of strongly Pareto-optimal routes is a subset of the set of weakly Pareto-optimal routes [47]. Hence, by addressing the weak Pareto optimality we account for the strong Pareto optimality as well.

Having described our motivation, let us proceed by presenting an overview of this chapter, which analyzes our contribution in [1]. Before delving into the intricacies of the NDQO algorithm, in Section 4.2 we will first define the unitary operator U_g, which implements a single strong Pareto dominance comparison and will be employed as the quantum oracle gate of the Grover operator $\mathcal{G}$. In Section 4.3, we will then invoke the modified Grover operator $\mathcal{G}$ in the context of BBHT-QSA sub-processes, which searches for routes that dominate a specific appropriately initialized reference route. Explicitly, our goal is not only to disregard routes that are sub-optimal, i.e. routes that are dominated by at least one route, but also to successively approach a single Pareto-optimal route. In Section 4.4, we will present the NDQO algorithm and demonstrate how the entire OPF can be identified using this series of BBHT-QSA sub-processes, followed by a 5-node tutorial example on WHMN routing using the NDQO algorithm in Section 4.5. We will then assess its performance in terms of it complexity and its accuracy with respect to full-search-based methods in Sections 4.6.1 and 4.6.2, respectively. Let us now proceed with some discussions concerning the appropriate design of the quantum oracle gate.

4.2 Quantum Oracle Design

In this section, we will present the design of the quantum oracle gate utilized within the Grover operator $\mathcal{G}$ for the sake of marking the appropriate routes by flipping their phase. Before delving into its design guidelines, we will first define the unit of computational cost, which we will refer to as *Cost Function Evaluation* (CFE). Similar to the methodology presented in Section 2.6, we will define the activation of the operator U_g as a single CFE, implementing a Pareto dominance comparison between two routes. Calculating the Pareto distance for each specific route is a rather demanding task in terms of the number of

CFEs, since it would require N^2 CFEs, which is equal to the classic exhaustive search complexity. Although some complexity reduction is offered by the QMA [86], the accuracy of the Pareto distance evaluation is affected. To elaborate further, assuming l qubits to be in the *Quantum Control Register* (QCR), which is used for the parallel calculation of the nominator in Eq . (2.6), an error $\epsilon \in O(1/l)$ [86] would be introduced, while increasing l for the sake of minimizing this error would result in an excessive complexity.

This imperfection may be mitigated, if we inspect the concept of Pareto optimality in a more meticulous manner, given in Definition 3. In fact, the calculation of the Pareto distance itself is unnecessary. In other words, a particular route would belong to the OPF, if and only if there is no other route which would dominate it. Under this perspective, our problem is simplified to an existence problem and all we have to find is a route, which would dominate the examined one. If the search is unsuccessful and, thus, there is no legitimate route which would dominate the examined one, the latter would belong to the OPF.

Our new approach would involve "asking" the oracle gate O whether a particular route-solution vector is dominated by the other route-solution vectors. Assuming that the particular route, which from now on will be referred to as the *reference route*, corresponds to the state $|i\rangle$, let us now define the oracle function $g(x,i)$, inspired by the one of Eq. (3.79), which the QSA would query as follows:

$$g(x,i) = \begin{cases} 1 & \mathbf{f}(x) \succ \mathbf{f}(i), \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

It may be observed from Eq. (4.1) that the oracle function would require two inputs: a classic state $|i\rangle$, which would point to a specific route, and a quantum state $|x\rangle$ initialized to the equally weighted superposition of all the states. Hence, the oracle gate O implementing $g(x,i)$ would map the state $|x\rangle$ to $-|x\rangle$, if and only if the respective route dominates the reference route i. The quantum circuit of O is similar to the one of the DHA oracle, which is shown in Fig. 3.8. However, the implementation of the unitary operator U_g would differ. According to Eq. (4.1) and Definition 2, simultaneous comparisons have to be conducted for the application of the strong dominance operator. This imposes an additional difficulty, since these comparisons need to be entangled with each other. Therefore, we propose the employment of three unitary operators U_{f_k}, where we have $k \in \{1,2,3\}$, one for every UF considered in the UV of Eq. (2.5). The functions $f_k(x,i)$ correspond to the less-comparison outcome between the x-th and the i-th routes in terms of their k-th UF and they are defined as:

$$f_k(x,i) = \begin{cases} 1, & f_k(x) < f_k(i), \\ 0, & f_k(x) \geq f_k(i), \end{cases} \tag{4.2}$$

where the function $f_k(x)$ corresponds to the k-th UF value of the x-th route. We have opted for connecting the QCR, where the reference route is stored, and the QIR inputs of three unitary operators U_{f_i} in a cascade formation as shown in Fig. 4.1, where we note that the control input $|i\rangle_1$ corresponds to the index of the reference route. This formation

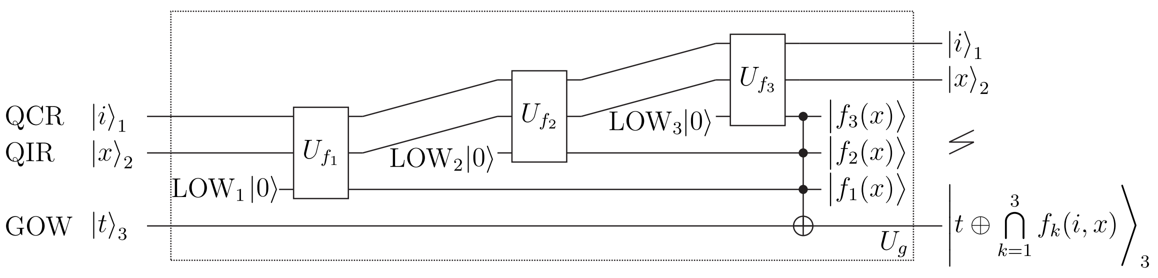


Figure 4.1: Quantum circuit of the unitary operator U_g implementing the controlled operation $g(x, i)$, corresponding to application the dominance operator. Three unitary operators U_{f_i} are employed, each corresponding to a single comparison between the reference path stored in the QCR and the states of the QIR in terms of the respective optimization objective.

entangles the *Local Oracle Workspace* (LOW) output of each U_{f_i} with the QIR output, resulting in the three LOWs to be entangled with the same QIR, yielding that the LOWs are entangled together. We note that the LOW input state used for each U_{f_i} is the $|0\rangle$, since we opted for mapping the binary output of $g(x, i)$ into the LOW output. The three LOW outputs are combined together using a 3-qubit *Tofolli* quantum gate [62], which performs the intersection (AND) operation among all three LOWs and then performs a XOR operation between the *Global Oracle Workspace* (GOW) input $|t\rangle_3$ and the outcome of the AND operation. Explicitly, the n-qubit Toffoli gate T_n [143] has a transfer matrix equal to [62]:

$$T_n = \begin{bmatrix} I_{2^n-2} & \mathbf{0}_{2^n-2,1} & \mathbf{0}_{2^n-2,1} \\ \mathbf{0}_{1,2^n-2} & 0 & 1 \\ \mathbf{0}_{1,2^n-2} & 1 & 0 \end{bmatrix}, \tag{4.3}$$

where the I_{2^n-2} sub-matrix denotes the a $[(2^n - 2) \times (2^n - 2)]$ identity matrix and the vector $\mathbf{0}_{1,2^n-2} = \mathbf{0}^T_{2^n-2,1}$ contains $2^n - 2$ zero elements. Furthermore, if the GOW input is set to the $|-\rangle$ state, then the U_g operator acts as an Oracle Gate in the same way as in Fig 3.8. Therefore, for each examined route it is possible to construct a database containing the dominance relationship of the rest of the routes with respect to the examined one.

Additionally, we note that a single activation of the oracle gate O would impose a single CFE, since the dominance operator would be employed only once and the comparisons are carried out serial due to the cascade formation. The function of this gate can be simulated in a classical computer. To elaborate further, simulating the oracle gate in a classical computer results in checking serially whether each of the legitimate routes dominates the reference route with index i and "marking" the specific routes that indeed dominate it. However, in this case the actual number of CFEs imposed would be equal to the number of the legitimate routes in the absence of QP.

4.3 The Quest for Optimal Routes

Having constructed a database containing the specific routes which dominate each route by the application of the quantum oracle gate O, we may employ a QSA [79, 80, 112] for finding the routes that correspond to the minimum Pareto distance. A search algorithm succeeds in finding the specific index x in a database or the argument for which the function g satisfies $g(x, i) = \delta$, which is termed as the *solution*. As we elaborated in Section 3.3, in a search maze of size N, Grover's QSA [79] finds a solution with $\sim 100\%$ probability after $O(\sqrt{N})$ database queries or function evaluations, provided that the number of solutions t is equal to $t = 1$. Hence, it achieves a quadratic reduction in the computational complexity, when compared to the optimal BF search in unsorted databases. By contrast, the BBHT-QSA [80], detailed in Section 3.4, is based on a successive application of Grover's QSA and finds a solution to the search problem with $\sim 100\%$ probability, even if multiple solutions exist, which is achieved without requiring *a priori* knowledge about the exact value of t. Explicitly, the BBHT-QSA succeeds in finding a solution after $4.5\sqrt{N}$ database queries in the worst-case scenario. Naturally, both in Grover's QSA and in the BBHT-QSA, the value δ has to be known. However, in many communication applications where the index $x_{\min}$ minimizing the cost function f is required to be found, the exact value of $f(x_{\min})$ cannot be known until after all the possible CF values have been evaluated. As for the solution, the DHA [81], detailed in Section 3.5, employs the BBHT-QSA multiple times and manages to find $x_{\min}$ representing the minimum entry of a database, even if the value of that particular entry is not known beforehand and also if we have $t \geq 1$. The DHA performs a minimum of $4.5\sqrt{N}$ and a maximum of $22.5\sqrt{N}$ database queries in the best-case and the worst-case scenario, respectively.

In our specific application we have to find a route that dominates the reference route in the previously constructed database. Because of the particular nature of our problem, we know that the intended entry is equal to $\delta = 1$ and that multiple routes may dominate a single route. Hence, the BBHT-QSA is the most appropriate quantum algorithm for finding a solution x_s in our application. It should be noted that the DHA will also succeed in finding a solution, since we have $x_s = x_{\min}$ but it will introduce more unnecessary database queries. Let us now proceed by introducing the improved BBHT-QSA used in the NDQO and then the NDQO algorithm.

The number of solutions t of a problem does affect the number L_{opt} of optimal Grover iterations. Since the BBHT-QSA [79, 87] assumes having no *a priori* knowledge about the number of solutions S in the database, it employs Grover's operator a pseudo-random number of consecutive times in a structured way. In our routing application we have $\delta = 1$, but we are unaware of the number of routes t that dominate the i-th route. The BBHT-QSA applied in our system is formally stated in Algorithm 4.1 [87]. The BBHT-QSA applies the Grover operator L consecutive times to the initial equiprobable superposition of states in (3.36) (Step 4.1.4[1]) and then measures the resultant QR $|x_f\rangle$ (Step 4.1.5). The extraction of the database's entry that corresponds to the observed index will verify whether the

[1]The notation refers to Step 4 of Alg. 4.1.

observed state $|j\rangle$ is a solution or not (Step 4.1.8). If the latter case is true, the process described is repeated after an update of the parameters (Steps 1.11–1.16) until a solution is found or a predetermined maximum number $L_{BBHT}^{QD,\text{ max}}$ of affordable Grover iterations has been reached. The algorithm keeps track of the total number L_{BBHT}^{QD} of CF evaluations in the *Quantum Domain* (QD) and the total number L_{BBHT}^{CD} of CF evaluations in the *Classical Domain* (CD) (Step 4.1.7). Naturally, the QD CFEs would be increased from the total number of Grover iterations L with each BBHT-QSA iteration, and the number of CD CFEs would are based on the check of Step 4.1.8. The function implemented by the Oracle's action is invoked in the CD.

An improvement the original BBHT-QSA in [80] is proposed, as detailed in Section 4.6. To elaborate briefly, as soon as the upper limit m of the range, from which the number of Grover iterations is selected, becomes higher than or equal to $\sqrt{N}$ (Step 4.1.11), i.e. higher than the value which corresponds to the worst case scenario of having only a single solution, a bias is imposed on it for excluding the value 0 from the range (Step 4.1.13). This stage of the BBHT-QSA is often referred as the *critical stage* [80], since there is at least 25% probability of finding a solution, as long as there exists one.

By applying the BBHT-QSA to the database constructed by the oracle gate O of Eq. (4.1), we will be able to identify a route, if there exists one, that would dominate the input one with $\sim 100\%$ probability, while using a number of $O(\sqrt{N})$ database queries. If there is no route that dominates the input one, the BBHT-QSA will output a random route $x_{s,r}$ from the search database, which could be disregarded with a simple final check. This check, which is similar to the first check of Step 4.1.8, may be invoked after the completion of the BBHT-QSA process for identifying at the expense of a single CFE as to whether the index exported from the BBHT-QSA corresponds to a valid route-solution, i.e. whether the condition $g(x_{s,r}, i) = \delta = 1$ is satisfied. Therefore, it is possible to avoid the identification of false solutions stemming from a potential BBHT-QSA QD-CFE time-out.

4.4 The Non-Dominated Quantum Optimization Algorithm

Having defined the core procedure of finding a route, which dominates the examined one, we may now proceed by presenting our proposed approach. We will exploit the observation [79] that if there is no solution for which we have $\delta = 1$, then O will mark no solutions and a single application of the $\mathcal{G}$ operator will leave the probabilities of the routes unaltered. Consequently, as the BBHT-QSA will not find any solutions, it will reach its time-out after $L_{BBHT}^{\max} = \left\lfloor 4.5\sqrt{N} \right\rfloor$ Grover iterations and output a random route from the total set of routes chosen randomly. At this point, let us check as to whether the output of the BBHT-QSA dominates the reference route i; the i-th route will belong to the OPF if and only if the check outcome is false, i.e. we have $g(x_s, i) = 0$, which implies that there is no solution dominating it. Otherwise, if the BBHT-QSA outcome dominates the solution examined, we can proceed with checking as to whether the outcome is optimal. This action is repeated until a Pareto-optimal route-solution is extracted, which would then terminate this *chain of BBHT-QSA activations*. Note that from now on we will refer to this process as *BBHT-*

Algorithm 4.1 Improved BBHT-QSA in NDQO Algorithm

```
1: Import reference route index $i$.
2: Set $m \leftarrow 1$, $\lambda \leftarrow 6/5$ and $L_{BBHT}^{QD} \leftarrow 0$, $L_{BBHT}^{CD} \leftarrow 0$.
3: Choose $L$ uniformly from the set $\{0, \dots, \lfloor m \rfloor\}$.
4: Apply the $\mathcal{G}$ operator $L$ times starting from the initial state $|\psi\rangle$ in (3.36), resulting in
   the final state $|x_f\rangle = \mathcal{G}^L |\psi\rangle$.
5: Observe $|x_f\rangle$ in the QD and obtain $|j\rangle$.
6: Compute $g(j, i)$ in the CD.
7: Update $L_{BBHT}^{CD} \leftarrow L_{BBHT}^{CD} + 1$ and $L_{BBHT}^{QD} \leftarrow L_{BBHT}^{QD} + L$.
8: if $g(j, i) = \delta = 1$ or $L_{BBHT}^{QD} \geq L_{BBHT}^{QD,\,\max}$ then
9:    Set $x_s \leftarrow j$, output $x_s$, $L_{BBHT}^{CD}$, $L_{BBHT}^{QD}$ and exit.
10: else
11:    Set $m \leftarrow \min\left\{\lambda m, \sqrt{N}\right\}$.
12:    if $m = \sqrt{N}$ then
13:       Choose $L$ uniformly from the set $\{1, \dots, \lfloor m \rfloor\}$ and go to step 4.
14:    else
15:       Go to step 3.
16:    end if
17: end if
```

QSA chain. Having extracted either a single or multiple route-solutions from the OPF, it becomes possible to avoid an excessive number of CFEs by checking as to whether the reference route-solution is dominated by the hitherto generated OPF. Explicitly, the specific route-solutions that are dominated by the OPF generated so far have a high probability of reaching an already generated OPF route-solution, which would imply having unnecessary BBHT-QSA chain activations

Algorithm 4.2 NDQO Algorithm

```
1: Initialize solution flag vector, $\mathcal{F}$, to zero.
2: Initialize $OPF = \varnothing$.
3: for $i = 0$ to $N - 1$ do
4:    if $\mathcal{F}_i = 0$ then
5:       if $\nexists j \in OPF : \mathbf{f}(j) \succ \mathbf{f}(i)$ then
6:          Set $l \leftarrow i$.
7:          repeat
8:             Set $k \leftarrow l$.
9:             Define the oracle function $g(x, k)$ from (4.1).
10:            Invoke the BBHT-QSA with input $g(x, k)$ and output $x_s$.
11:            Set $l \leftarrow x_s$ and $\mathcal{F}_k \leftarrow 1$.
12:         until $\mathbf{f}(l) \not\succ \mathbf{f}(k)$.
13:         Append $x_k$ into the $OPF$.
14:      end if
15:   end if
16: end for
17: Output the $OPF$ and exit.
```

Having provided all the necessary discussions concerning the NDQO subroutines, we may now proceed to its detailed description relying on Alg. 4.2 as well as on the flowchart portrayed in Fig. 4.2. We will define a binary check flag vector $\mathcal{F}$ which indicates whether

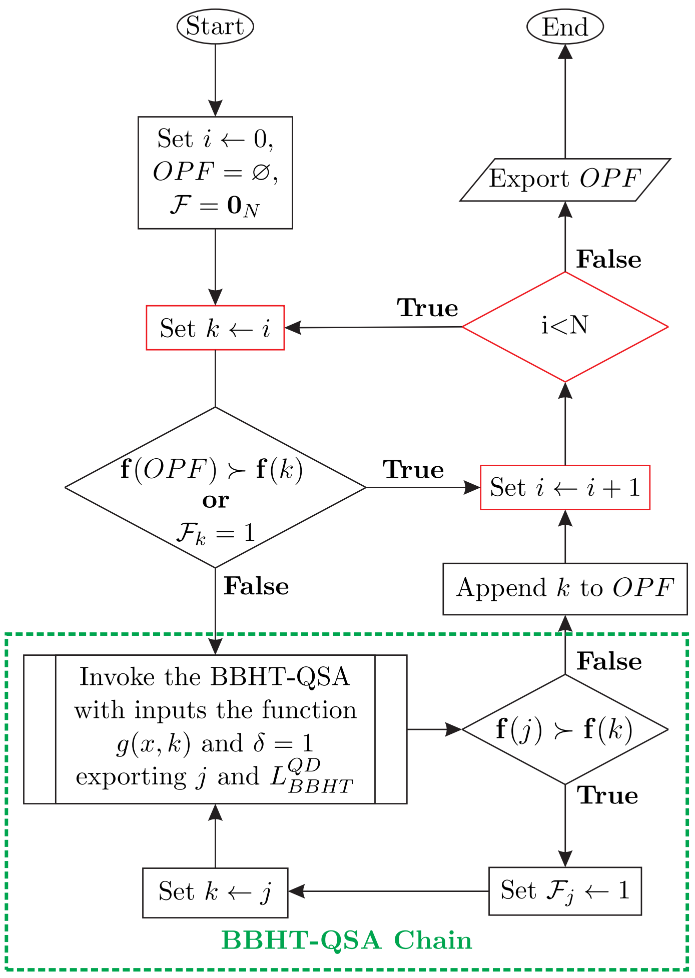

Start
End
Set $i \leftarrow 0$, $OPF = \varnothing$, $\mathcal{F} = \mathbf{0}_N$
Export OPF
False
True
Set $k \leftarrow i$
i<N
$\mathbf{f}(OPF) \succ \mathbf{f}(k)$ or $\mathcal{F}_k = 1$
True
Set $i \leftarrow i + 1$
False
Append k to OPF
False
Invoke the BBHT-QSA with inputs the function $g(x, k)$ and $\delta = 1$ exporting j and L_{BBHT}^{QD}
$\mathbf{f}(j) \succ \mathbf{f}(k)$
True
Set $k \leftarrow j$
Set $\mathcal{F}_j \leftarrow 1$
BBHT-QSA Chain

Figure 4.2: NDQO algorithm's flowchart.

a specific route has already been processed and it is initialized to an all-zero vector. Subsequently, for each route we will check as to whether it has already been processed. In this case, the specific route is checked as to whether it is dominated by the OPF already generated in Step 4.2.5. We note that we will take into account the classical complexity imposed by these specific dominance comparisons. If it is not dominated by the hitherto generated OPF, then a *BBHT-QSA chain* is activated relying on the examined route as its reference route. The BBHT-QSA chain will be then terminated, when a Pareto-optimal route-solution is examined and the flags of all the routes processed by the BBHT-QSA chain are set equal to 1. This procedure is repeated until all the legitimate routes have been processed either by the OPF dominance check or by the BBHT-QSA chain and, thus, all the Pareto-optimal route-solutions are exported. The goal of this procedure is to identify the entire OPF as promptly as possible.

4.5 A Detailed 5-Node Example

Let us now provide an illustrative example portraying the main concepts of our proposed algorithm. We will consider a 5-node network having the architecture of Fig. 4.3(a), where

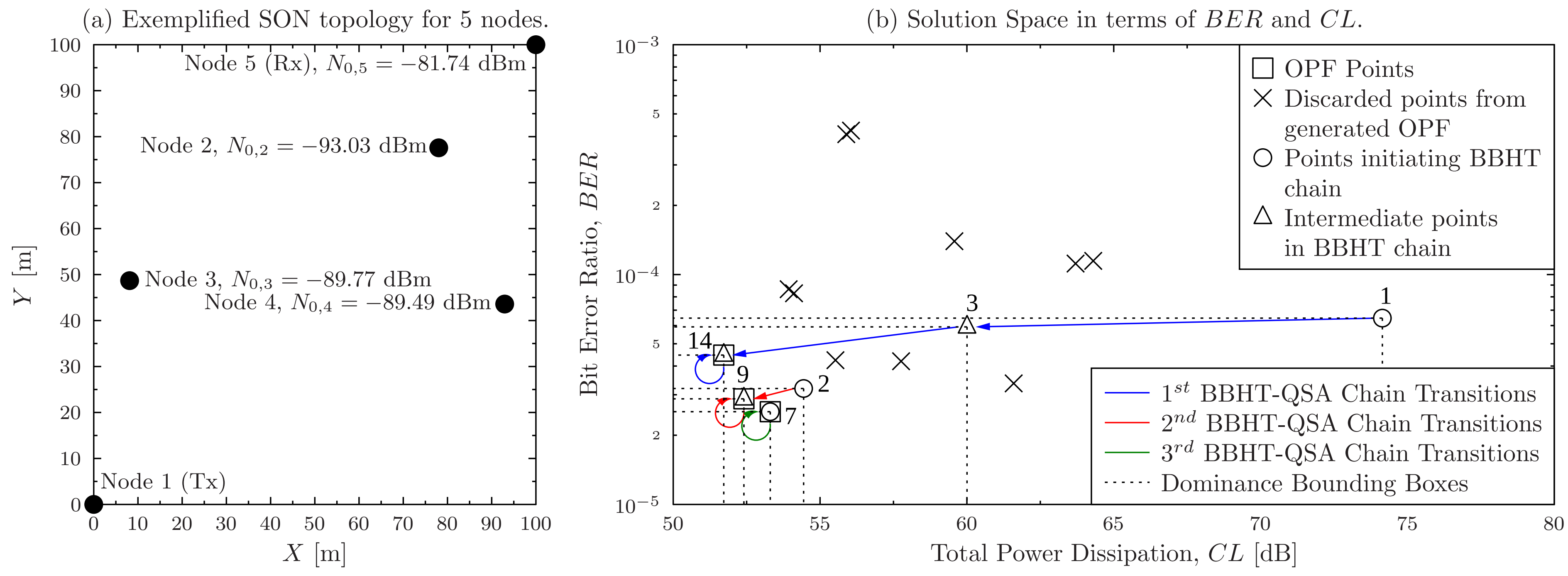


Figure 4.3: (a) Exemplified architecture for a 5-node WMHN, and (b) its optimization process using the NDQO algorithm. In this example only two UF are used per solution for the sake of simplicity relying on the system model of Table 2.1. The routes that belong to the OPF are noted with a square marker (□), the routes that have already been processed as intermediate points in BBHT-QSA chains and will be skipped in the serial parsing step (Step 4.2.4) are marked with a triangle (△), whereas those that have not been processed and, at the same time, are not dominated by the generated OPF initiating a BBHT-QSA chain are marked with a circle (○). Moreover, the route-solutions, which have not already been processed but are dominated by the hitherto generated OPF and thus they will be skipped, are marked with a cross (×). Moreover, the indices of the routes as shown in Table 4.1 are marked in (b). Finally, the round arrows in (b) denote that a BBHT-QSA has been activated with input the respective point but in the absence of potential route-solutions a random route is output by the BBHT-QSA, classifying the input route-solution as Pareto Optimal (Step 4.2.13). The current problem solution is not the unique one; different solutions could be derived depending on the BBHT-QSA chain intermediate outcomes.

the interference experienced by each of the nodes is also visible. Explicitly, the topology of Fig. 4.3(a) has been generated based on the assumptions presented in Table 2.1. The legitimate route-solutions produced by this setup along with their UF values are shown in Table 4.1. We note that only the BER and the CL are taken into account for the sake of simplifying the solutions' graphical representation, which is shown in Fig. 4.3(b). Moreover, the actual routes are assumed to be stored on a list in ascending lexicographical order, thus facilitating the employment of Lehmer Encoding/Decoding [137], which is highlighted in Appendix A. Furthermore, in Fig. 4.4 we present the probability P_s of successfully finding a route-solution, when using Grover's QSA, versus the number of solutions t present and the number of $\mathcal{G}$ applications for our 5-node WMHN routing problem of Fig. 4.3(a). Explicitly, the respective probabilities portrayed in Fig. 4.4 have been calculated with the aid of Eq. (3.71).

Table 4.1: Routes along with their UFs and indices for the 5-node WMHN example of Fig. 4.3, relying on the system model of Table 2.1.

Index i	Route	$P_{e,i}$ ($\times 10^{-4}$)	CL_i [dB]	Index i	Route	$P_{e,i}$($\times 10^{-4}$)	CL_i [dB]
1	{1 5}	0.646	74.147	9	{1 4 2 5}	0.288	52.407
2	{1 2 5}	0.319	54.440	10	{1 4 3 5}	1.147	64.302
3	{1 3 5}	0.592	60.004	11	{1 2 3 4 5}	1.397	59.575
4	{1 4 5}	0.336	61.593	12	{1 2 4 3 5}	4.230	56.053
5	{1 2 3 5}	1.117	63.700	13	{1 3 2 4 5}	0.829	54.113
6	{1 2 4 5}	0.424	55.524	14	{1 3 4 2 5}	0.446	51.721
7	{1 3 2 5}	0.253	53.304	15	{1 4 2 3 5}	4.079	55.893
8	{1 3 4 5}	0.420	57.759	16	{1 4 3 2 5}	0.863	53.931

Moreover, all the steps carried out by the NDQO algorithm for exporting the OPF are shown in Fig. 4.3(b), where the solution[2] transitions that are facilitated by the BBHT-QSA are represented by the arrows. Moreover, the routes that belong to the OPF are indicated by a square marker (□). Furthermore, the points that have already been processed as intermediate points in the BBHT-QSA chains and hence will be skipped during the serial parsing step (Step 4.2.4) are marked by a triangle (△). Still referring to Fig. 4.3(b), those points that have not been processed and, at the same time, are not dominated by the generated OPF initiating a BBHT-QSA chain are marked with a circle (○). Additionally, the points that have not yet been processed but are dominated by the generated OPF and thus are skipped are marked by a cross (×). Moreover, the transitions that are carried out by the BBHT-QSA chains of Steps 4.2.6 – 4.2.11 are indicated by arrows. Additionally, arrows of different color has been used for each BBHT-QSA chain. Let us now proceed with a more detailed description of the NDQO algorithm.

The algorithm will initialize the binary check flag vector $\mathcal{F}$ to a vector of zeros and the OPF to an empty set ($\varnothing$), according to Step 4.2.2. Then, the 1st route of Table 4.1, {1 5}, is checked. Since its flag value is equal to zero and the OPF set is empty, the BBHT-QSA process of Alg. 4.1 is initiated with this specific route as its reference route. The legitimate successful outputs of the BBHT-QSA, i.e. the solutions that dominate {1 5}, are located

[2]We define a solution as an output route of the BBHT-QSA that dominates the input one.

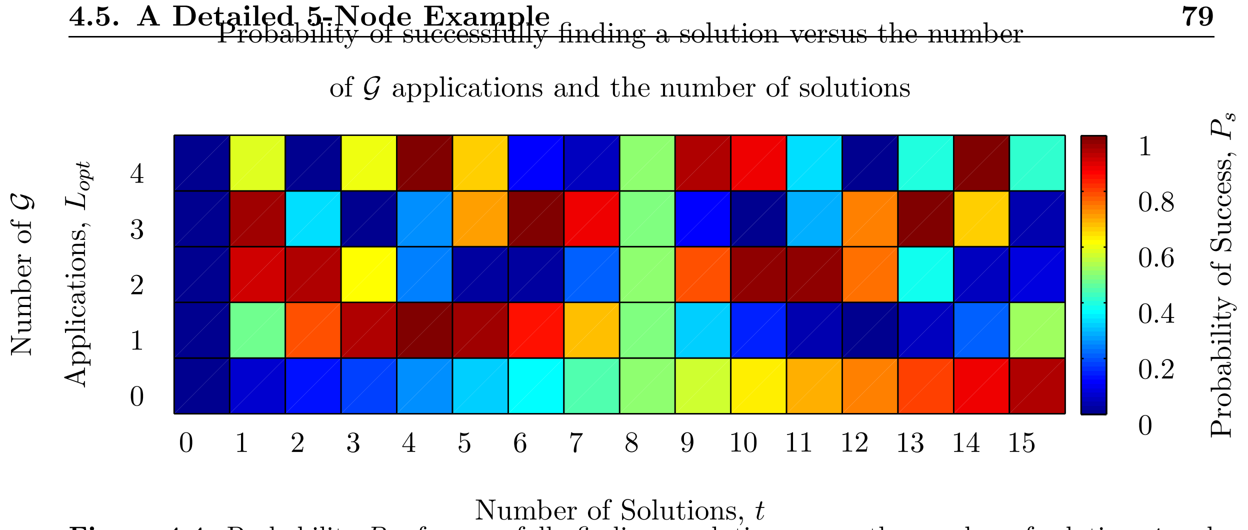


Figure 4.4: Probability P_s of successfully finding a solution versus the number of solutions t and the number of Grover iterations L_{opt} for the database of Table 4.1, based on Eq. (3.71). The number of Grover iterations L_{opt} is varied in the range $\{0, 1, .., \sqrt{N}\}$; the upper bound of the range corresponds to the optimal number of Grover iterations, where only a single solution is available. In our example, we have $\sqrt{N} = \sqrt{16} = 4$.

within the rectangle of Fig. 4.3(b) which has the solution argument and the zero point of coordinates as its opposite corners, as indicated by the doted lines. Our database length is equal to $N = 16$, while the number of present solutions, which are located within the aforementioned rectangle, is equal to $t = 7$. Each of these solutions will have the same probability of becoming the single output. According to Fig. 4.4, the optimal number of $\mathcal{G}$ applications would be for $L_{opt} = 3$, which gives a 88.45% probability of finding a route-solution that dominates the direct route. The BBHT-QSA process initializes the upper bound of the L_{opt} selection range to $m = 1$ (Step 4.1.2) and the specific L_{opt} value is chosen from the set $\{0, 1\}$. Assuming that $L_{opt} = 0$ is chosen, based on Eq. (3.71), the probability of successfully finding a solution would be equal to $P_s = 43.75\%$. The quantum algorithm then observes its QIR (Step 4.1.5) and outputs $j = 13$, i.e the route $\{1\ 4\ 3\ 2\ 5\}$. A check is then performed whether the 13th route dominates the direct one (Step 4.1.8), which is unsuccessful, since $P_{e,1} < P_{e,13}$ and $CL_1 < CL_{13}$ and hence $g(13, 1) = 0$. Then parameter m is increased to $m = \lambda m = 6/5$ (Step 4.1.11) and the upper bound is modified to $\lfloor m \rfloor = \lfloor 6/5 \rfloor = 1$, while the range remains the same as in the previous iteration, i.e. the parameter L_{opt} will be selected from the set $\{0, 1\}$. At this point, assuming that $L_{opt} = 1$ is chosen, the probability of successfully finding a route-solution that dominates the input one would be equal to $P_s = 68.36\%$. Let us assume that the output of the BBHT-QSA is the 3rd route ($i = 3$), $\{1\ 3\ 5\}$, of Table 4.1. Once again, this route will be checked whether it dominates the input route and since it lies within the rectangle of Fig. 4.3(b), it will indeed dominate it. The transition between the direct route and the third one is noted using a blue colored arrow in Fig. 4.3(b).

Subsequently, the input solution flag value is set to $\mathcal{F}_1 = 1$ and, hence, a new BBHT-QSA process is invoked having as its input the output of the previous BBHT-QSA. Again, the successful BBHT-QSA outputs will be located within the rectangle defined by the new reference route and the center of coordinate axes as its opposite corners; after the completion of the BBHT-QSA of Alg. 4.1, the flag value of the input route is set to one, i.e. to $\mathcal{F}_3 = 1$. Following a similar procedure as in the first BBHT-QSA iteration, the algorithm outputs the 14th route, $\{1\ 3\ 4\ 2\ 5\}$, of Fig 4.3(b) which dominates the input and happens to be a Pareto Optimal solution. A BBHT-QSA will be invoked with this route as its input and since no solutions would exist which dominate the new reference, the BBHT-QSA will reach its time-out after at a minimum of $L_{BBHT}^{QD,\max}$ Grover operator $\mathcal{G}$ applications and it will hence output a random route-solution selected from the set of all the legitimate ones. Upon applying the dominance operator to the BBHT-QSA output, its response will become false, which would indicate that no point dominating the input route exists, which, in turn, would imply that the route belongs to the OPF. Hence, the route with index $i = 14$ in Fig. 4.3(b) is appended to the OPF. Moreover, its flag value will be set to $\mathcal{F}_{14} = 1$. This transition is noted in marked Fig. 4.3(b) with the blue round arrow, since the NDQO algorithm will return to the input route-solution and classify it as optimal.

Then, the second route with index $i = 2$ of Table 4.1 will be checked (Step 4.2.3). Since we have $\mathcal{F}_2 = 0$ and this route is not dominated by the route with index $i = 14$ (Steps 4.2.4 and 4.2.5), the BBHT-QSA process of Alg. 4.1 will be initiated with the second route as its input. The possible successful outcomes of the BBHT-QSA would be the 7th ($i = 7$) and the 9th ($i = 9$) routes with 50% probability, both of which happen to be Pareto Optimal. Following the same process as the very first BBHT-QSA activation, we may now assume that the 9th route is the output of the BBHT-QSA, and since it will indeed dominate the input route, the flag value of the input route is toggled to $\mathcal{F}_2 = 1$. The respective transition is noted in Fig. 4.3(b) using a red arrow. A new BBHT-QSA process will be invoked with the 9th route of Fig 4.3(b) as its input, which would exhaust the maximum number of $\mathcal{G}$ applications, therefore outputting a random solution from the solution space. Hence, the 9th route is incorporated into the OPF and its flag value will be modified accordingly. This operation corresponds to the round red arrow seen in Fig. 4.3(b). Afterwards, the 3rd route will be skipped, because we have $\mathcal{F}_3 = 1$. Additionally, the rest of the routes until the 7th route of Fig. 4.3(b) will be discarded as they are dominated by the 9th route and, thus, are not Pareto Optimal (Step 4.2.5). As for the 7th route, since it has not been processed ($\mathcal{F}_7 = 0$) and will not be dominated by any solution due to the fact that it is Pareto Optimal, the BBHT-QSA process of Alg. 4.1 will be invoked with its input given by this route. This process will exhaust the maximum number of $\mathcal{G}$ applications and will hence output a random solution, thus leading to appending the 7th route-solution to the OPF. Moving back to Fig. 4.3(b), this step corresponds to the round green arrow. Finally, the rest of the solutions will be discarded, since albeit they have not been processed, they will be dominated by the OPF generated.

During the exemplified description of the NDQO algorithm, we have not quantified the complexity imposed by the NDQO in terms of the number of CFEs. Thus, we will elaborate on this issue in the next section, where a variety of performance metrics will be introduced.

4.6 Computational Accuracy versus Complexity

In this section, we will provide simulation results concerning the accuracy of the NDQO algorithm versus its complexity. Before delving into the presentation of the results, the complexity of the classical BF algorithm should be quantified. In the classical domain, each route is compared to all the other legitimate routes for determining whether it is dominated by any solutions. Based on Eq. (2.9), this would impose a complexity on the order of $O(N^2)$. However, this operation, which we will refer to as *naive-BF*, would find the OPF route-solutions along with sorting all legitimate routes into PFs. The latter operation is not performed by our proposed algorithm, making their comparison rather unfair.

For the sake of fairness, we will also use another version of the BF, which exports only the OPF, as formally presented in Alg. 4.3. This specific algorithm functions in the same fashion as the NDQO algorithm presented in Alg. 4.2. The only difference would be that a BF serial search is used instead of the BBHT-QSA chains for identifying whether a solution is optimal. The actual complexity of this BF method is random and would explicitly depend on the number of OPF route-solutions and on their order of appearance in the solution space. Therefore, due to the complex structure of the WMHNs examined we will derive its complexity using *Monte Carlo* simulations and compare it to the respective complexity of the NDQO algorithm for the same WMHN setups. Moreover, both the upper and the lower bounds of this BF method may be derived. The upper bound corresponds to the rather unrealistic case, where all the legitimate routes are optimal. In this case, the direct route would require N CFEs for its determining whether or not a route-solution is optimal, the second route in the database will require in turn $(N+1)$ CFEs, since it will be checked against the OPF generated, which consists of a single solution. Finally, the last one will be checked against all the solutions forming part of the OPF, which would consist of $(N-1)$ route-solutions and another N CFEs would be required for its identification as an optimal route-solution. Therefore, the resultant maximum complexity may be derived by exploiting the following property of the sum of arithmetic series as:

$$L_{BF}^{\max} = N^2 + \sum_{i=0}^{N-1} i = N^2 + \frac{N}{2}(N-1) = \frac{3}{2}N^2 - \frac{1}{2}N. \tag{4.4}$$

Hence, the upper bound of the BF complexity is still on the order of $O(N^2)$. On the other hand, assuming that there is only a single optimal path, which happens to be the first in our database, namely the direct route, the BF method would require N CFEs for classifying this route as an optimal one (Steps 4.3.5-4.3.12) and another $(N-1)$ CFEs for classifying the rest of the routes as suboptimal. As a result, the lower bound of the BF complexity is equal to:

$$L_{BF}^{\min} = 2N - 1 = O(N). \tag{4.5}$$

Therefore, the BF method would involve a complexity on the order of $O(N)$ and $O(N^2)$ for the best- and the worst-case scenario, respectively.

Additionally, we note that the simulation results presented in this section have been

Algorithm 4.3 BF method

```
1: Initialize $OPF = \varnothing$.
2: for $i = 0$ to $N - 1$ do
3:   Set $f \leftarrow 0$
4:   if $\nexists j \in OPF : \mathbf{f}(j) \succ \mathbf{f}(i)$ then
5:     for $k = 0$ to $N - 1$ do
6:       if $\mathbf{f}(k) \succ \mathbf{f}(i)$ then
7:         Set $f \leftarrow 1$ and terminate inner loop.
8:       end if
9:     end for
10:    if $f = 0$ then
11:      Append $i$ into the $OPF$.
12:    end if
13:  end if
14: end for
15: Output the $OPF$ and exit.
```

generated using the *Monte Carlo* simulation method and they have been averaged over 10^8 runs. Finally, since we had no quantum computer at our disposal, the simulations of the QSAs were carried out using a classical cluster. Explicitly, since the quantum Oracle O calculates in parallel the UF vectors of all the legitimate routes in the QD, they were pre-calculated. We note that this results in an actual complexity higher than that of the BF method. Therefore, the deployment of the NDQO in a quantum computer is essential for observing a complexity reduction stemming from the QP. Hence, in our simulations, we have made the assumption of employing a quantum computer for our algorithm and we count the total number of O-activations for quantifying the NDQO algorithm's complexity. This number would be the same for both classical and quantum implementations. Let us now proceed by characterizing the complexity of the NDQO algorithm.

4.6.1 NDQO Complexity Performance

Within the BBHT-QSA iterations, each quantum oracle application would result in a single QD-CFE, whereas the validation check of Step 4.1.4[3] would require a single CD-CFE. We note that the complexity of a single CD-CFE and of a single QD-CFE are assumed to be identical for simplicity. The total complexity L_{NDQO}^{tot} may be derived as the sum of the number of $\mathcal{G}$ applications L_{NDQO}^{QD}, that of the total classic comparison activations within the BBHT-QSA iterations L_{NDQO}^{CD} and that of the comparisons with the route-solutions of the already generated OPF $L_{\text{NDQO}}^{\text{OPF}}$, yielding:

$$L_{\text{NDQO}}^{tot} = L_{\text{NDQO}}^{QD} + L_{\text{NDQO}}^{CD} + L_{\text{NDQO}}^{\text{OPF}}. \tag{4.6}$$

Therefore, in order to derive both the upper and lower bounds of complexity in terms of the number of CFEs in both the quantum and classic domains, we have to consider two extreme cases, which are identical to the ones considered for the BF method. For the

[3]We remind that the notation refers to Step 4 of Alg. 4.1.

lower bound, we will assume that the optimization problem has only a single solution, which happens to be the first route in the solution database, namely the direct route. The NDQO algorithm will exhaust the maximum affordable complexity of $L_{BBHT}^{QD,\ \max} = \left\lfloor 4.5\sqrt{N} \right\rfloor$ for the first route and the rest of the routes will be discarded, since they will be dominated by the first one. Since we are examining the lower bound, each of the terms in the sum of (4.6) needs to be minimized. In terms of L_{NDQO}^{QD}, only the first route will invoke the BBHT-QSA process which will reach the maximum number of $\mathcal{G}$ applications and the lowest possible number would be:

$$L_{\text{NDQO}}^{QD,\ \min} = \left\lfloor 4.5\sqrt{N} \right\rfloor + 1 > 4.5\sqrt{N}. \tag{4.7}$$

Hence, the minimum number of the $\mathcal{G}$ applications would be for $L_{\text{NDQO}}^{QD,\ \min} = 4.5\sqrt{N}$. As for the CD-CFEs, we could can a greedy approach in order to find the associated minimum value: we may assume that the maximum number of $\mathcal{G}$ iterations $\lceil m \rceil$ is selected in Step 4.1.2. In this way, the maximum number of $\mathcal{G}$ applications will be reached with as few CD-CFEs as possible. Under this perspective, we would get:

$$\sum_{i=0}^{L_{\text{NDQO}}^{QD,\min}-1} \lambda^i m \geq L_{BBHT}^{QD,\ \max} \equiv 4.5\sqrt{N}, \tag{4.8}$$

where λ and m are the BBHT-QSA initialization parameters. Therefore, in order to find the minimum value, all that has to be done is to solve Eq. (4.8) in terms of $N_{QD}^{\min}$, yielding:

$$L_{\text{NDQO}}^{CD,\min} = \log_{\lambda}\left(4.5\ \frac{\lambda-1}{m}\ \sqrt{N}+1\right)+1. \tag{4.9}$$

As for the comparisons with the hitherto generated OPF, all the routes except for the first one will be dominated by the direct route leading to $L_{\text{NDQO}}^{\text{OPF},\ \min} = N-1$ and hence all but the first routes will be discarded. Finally, the lower bound of the NDQO algorithm's complexity may be expressed as:

$$\begin{aligned} L_{\text{NDQO}}^{tot,\min} &= 4.5\sqrt{N} + \log_{\lambda}\left(4.5\ \tfrac{\lambda-1}{m}\ \sqrt{N}+1\right) + N \\ &= O(N). \end{aligned} \tag{4.10}$$

Consequently, the lower bound complexity of the NDQO algorithm is on the order of $O(N)$ providing a quadratic speed-up down from $O(N^2)$.

As far as the NDQO algorithm's upper bound of complexity is concerned, we will consider the extreme case, where all the routes are Pareto-optimal. Naturally, having that many route-solutions in the OPF would result in excessive an excessive number of CFEs in the sub-process of the NDQO algorithm, where a particular route-solution is examined to ascertain whether it is dominated by the OPF generated. Hence, a restriction should be imposed on the grounds that this process should not exceed the BBHT-QSA maximum number of activations. To elaborate further, since the BBHT-QSA involves in the worst case $4.5\sqrt{N}$ CFEs in the quantum domain, it is reasonable to impose an upper bound also on the number of the classic domain CFEs, which may be set to the number of comparisons with the generated OPF for maintaining the number of classic CF evaluations. In our approach this upper bound was set to the BBHT-QSA time-out of $L_{BBHT}^{QD,\text{ max}} = 4.5\sqrt{N}$. Therefore, if the length of the generated OPF is higher than or equal to the BBHT-QSA time-out, the examined solution will not be compared to the generated OPF and a BBHT-QSA chain will be invoked directly, since it would involve fewer CF evaluations. Under this perspective, the number of CD-CFEs due to OPF comparisons would be upper bounded by:

$$L_{\text{NDQO}}^{\text{OPF, max}} = \sum_{i=1}^{4.5\sqrt{N}} i = 2.25\sqrt{N}\left(1 + 4.5\sqrt{N}\right). \tag{4.11}$$

Based on Eq. (4.11) the upper bound involves a complexity on the order of $O(N)$. As for the BBHT-QSA, an additional restriction should be imposed. There exists an extreme case, when only zero $\mathcal{G}$ applications are selected to be applied. In this case, since the time-out is quantified in terms of the number of oracle queries, there is an extremely low probability that the algorithm will fall into an infinite loop, where $L = 0$ is continuously chosen in Step 4.1.2. This event would yield an upper bound of infinity. In fact, the effect of this exceptional case could be mitigated. Upon reaching the *critical stage*[4] [80] of the algorithm, a valid solution would be output with a probability equal to 25%. Hence, it may seem reasonable to exclude from the range the specific event of "choosing" 0 $\mathcal{G}$ applications for avoiding the infinite loop trap. Additionally, the probability of success upon reaching the *critical stage* will remain unaltered. Under this perspective, the worst case scenario, as far as the CD-CFEs of the inner BBHT-QSA iterations are concerned, would be to "choose" $L = 0$ in Step 4.1.3, until the *critical stage* (condition in Step 4.1.12) is reached and where $L = 1$ is selected in Step 4.1.13 $4.5\sqrt{N} - 1$ times, or in other words until we are a single QD-CFE away from the time-out condition, and at $\sqrt{N}$ $\mathcal{G}$ applications during the last iteration, following a greedy approach. Consequently, this operation will be repeated for each solution yielding a complexity of:

$$L_{\text{NDQO}}^{CD,\text{ max}} = N\left(\log_\lambda \sqrt{N} + 4.5\sqrt{N}\right), \tag{4.12}$$

$$L_{\text{NDQO}}^{QD,\text{ max}} = N\left(5.5\sqrt{N} - 1\right). \tag{4.13}$$

Finally, the upper bound of the NDQO algorithm's complexity may be quantified by substituting Eqs. (4.11-4.13) into (4.6), yielding:

$$\begin{aligned} L_{\text{NDQO}}^{tot,\max} &= 10N\sqrt{N} + N\log_\lambda \sqrt{N} + 9.125N + 2.25\sqrt{N} \\ &= O(N\sqrt{N}) \\ &= O(N^{3/2}). \end{aligned} \tag{4.14}$$

[4]The *critical stage* will be reached at exactly $\left\lceil \log_\lambda \sqrt{N} \right\rceil$ BBHT-QSA iterations [80].

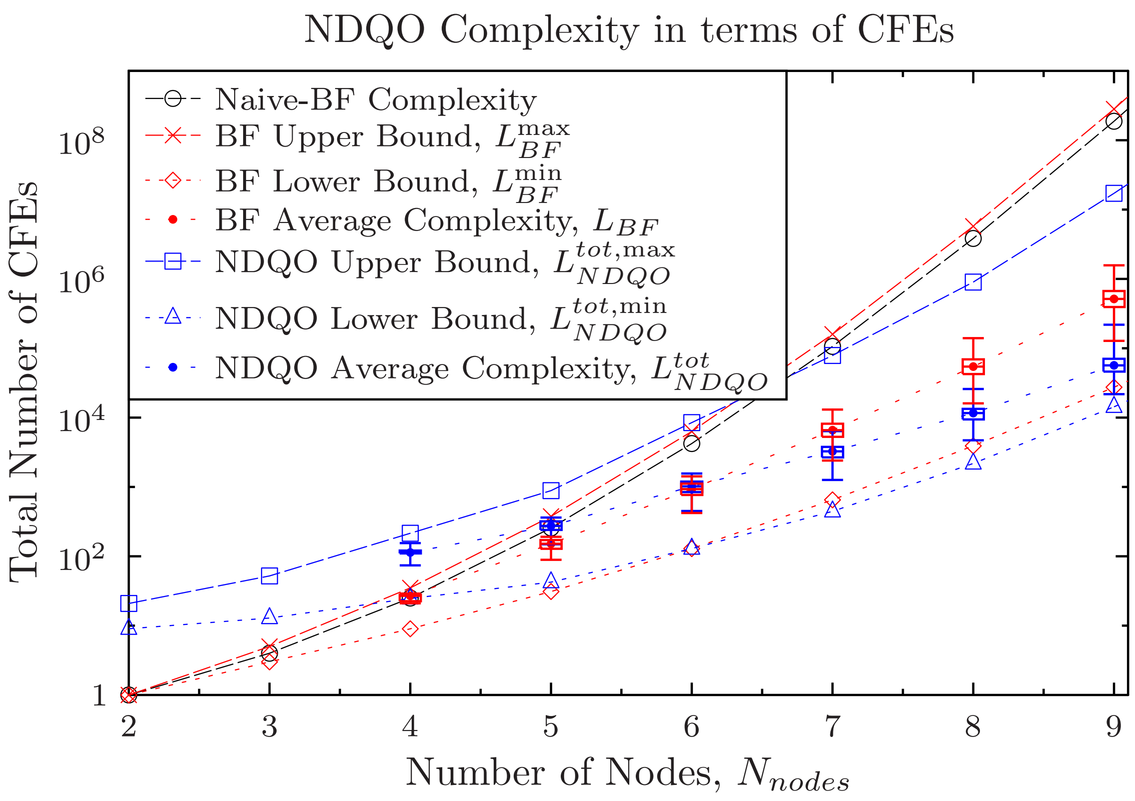


Figure 4.5: Complexity of the NDQO algorithm compared to that imposed by the BF method of Alg. 4.3; the mean number of CFEs is shown along with the upper and the lower bounds, as they were derived in Eqs. (4.10) and (4.14). They are compared to the naive-BF complexity as well as to the upper and lower bounds of the BF method of Alg. 4.3, based on Eqs. (5.24) and (5.25), respectively. Both the NDQO algorithm's and the BF method's average complexities are presented using box plots; the upper and lower bounds of the boxes correspond to the 75% and 25% quartiles, respectively. In addition, the observed maximum and minimum complexity values are presented using horizontal lines. The mean complexity results have been averaged over 10^8 runs for WMHNs based on the optimization problem of Eq. (2.7) relying on the UV defined in Eq. (2.5) and on the assumptions of Table 2.1.

Therefore, the upper bound would involve a complexity on the order of $O(N^{3/2})$, which is still lower than that of the classical BF, which is $O(N^2)$.

The average complexity of the NDQO algorithm $E\left[L_{\text{NDQO}}^{tot}\right]$ is shown in Fig. 4.5 for WMHNs consisting of $N_{nodes} = 2$ to $N_{nodes} = 9$ nodes. Recall that the Pareto optimality routing problem of Eq. (2.7) is associated with the UV of Eq. (2.5) relying on the assumptions in Table 2.1. These average complexities are also compared to both the upper and the lower bound of Eqs. (4.14) and (4.10) respectively, as well as to the respective complexity of the classical BF algorithm, as they were derived in Eqs. (5.24) and (5.24). The average complexity of the NDQO algorithm is presented in Fig. 4.5 with the aid of boxes surrounding the bold dots and having different aspect ratios in order to portray its stochastic nature. Explicitly, the upper and lower edges of the box boundaries at each WMHN size represent the 25% and 75% quartiles, while the vertical bars correspond to the maximum and minimum value of L_{NDQO}^{tot} found by our simulations. Observe in Fig. 4.5 that, naturally, the interquartile distance will increase as the number of WMHN nodes increases. In fact, the increase in the total number of routes would entail a higher variation in the number of routes belonging to the OPF yielding a higher variation in the number of BBHT-QSA time-outs needed in order to check the entire solution space. Additionally, the average NDQO complexity tends to be closer to its upper bound for WMHNs up to $N_{\text{nodes}} = 5$ nodes. This could be justified by the fact that, since three optimization objectives were used in our case based on Eq. (2.5), the OPF would be formed by at least three routes, namely one for the minimum of each objective. Furthermore, the 4-node and 5-node WMHNs involve $N = 5$ and $N = 16$, routes in total respectively based on Eq. (2.8), hence then approach the worst case scenario. However, as the network size is increased, the ratio of the number of optimal routes over the number of the total legitimate ones will decay, hence approaching the lower bound.

Furthermore, it may be observed from Fig. 4.5 that the NDQO tends to require less CFEs than the classical BF method for WMHNs having more than $N_{\text{node}} = 6$ nodes. Explicitly, the NDQO algorithm becomes more efficient for realistic practical databases, where a complexity reduction would be achieved by the use of the BBHT-QSA quantified in terms of the QD-CFEs. This is achieved, if the BBHT-QSA's complexity is lower than that of its respective classical BF counterpart, which would serially check whether there exists a route-solution that dominates the examined one, implying that the minimum required size of a database should satisfy the condition of $L_{BBHT}^{tot} < N$, where L_{BBHT}^{tot} corresponds to the complexity imposed by a single BBHT-QSA activation. Based on Eqs. (4.7) as well as (4.9) for the lower bound and on Eqs. (4.13) as well as (4.12) for the upper bound, the respective complexities of a single BBHT-QSA activation will be equal to:

$$L_{BBHT}^{tot,\min} = 4.5\sqrt{N} + \log_{\lambda}\left(4.5\,\frac{\lambda-1}{m}\,\sqrt{N}+1\right)+1, \tag{4.15}$$

$$L_{BBHT}^{tot,\max} = 10\sqrt{N} + \log_{\lambda}\sqrt{N} - 1. \tag{4.16}$$

Therefore, based on Eqs. (4.15) and (4.16), the minimum database required size for achieving a complexity reduction would be $N_{\min}^{best} > 39$ and $N_{\min}^{worst} > 123$ routes for the best- and the worst-case scenarios, respectively. In our routing application, this condition becomes valid for the best-case scenario in WMHNs having six or more nodes, where the total number of routes is $N = 65$[5], whereas in the 4-node and 5-node WMHNs it would be equal to $N = 5$ and $N = 16$ routes, respectively, according to Eq. (2.8). Hence, we observe in Fig. 4.5 that some complexity reduction is offered by the NDQO lower bound for WMHNs consisting of seven nodes.

Moreover, as far as the complexity upper bound is concerned, the condition for achieving a complexity reduction by the BBHT-QSA is satisfied for WMHNs having seven or more nodes. This may be verified by the change in trend of the upper bound complexities observed in Fig. 4.5 for the WMHNs consisting of $N_{nodes} = 7$ nodes. Explicitly, the upper

[5]A database of $N = 128$ entries is used for the 6-node WMHN, since the database length has to be explicitly a power of 2 due to the binary nature of qubits.

bound of the NDQO algorithm is seen to impose a lower complexity quantified in terms of CFEs compared to the BF and the naive-BF methods. Furthermore, we observe in Fig. 4.5 that the BF method's complexity upper bound imposes a higher number of CFEs compared to the naive-BF one, owing to the number of excessive dominance comparisons invoked by Step 4.3.4. As for the NDQO lower bound, where the only optimal route is the direct one, we will achieve a better performance with respect to the BF method for WMHNs having more than four nodes.

Additionally, as for the average complexity, observe in Fig 4.5 that the same average complexity is imposed for both the average NDQO algorithm and the BF method for WMHNs consisting of six nodes. This is justified by considering the fact that even though a beneficial complexity reduction is offered by the BBHT-QSA chains, there is a computational overhead which is imposed by padding our database so that it has a size equal to a power of 2. Hence, the NDQO algorithm has a slightly higher average complexity than the BF method. On the other hand, a substantial complexity reduction of about 50.5% is offered on average for a 7-node WMHN, while for the 8-node and 9-node WMHNs we have a complexity reduction of about 78.6% and 89% respectively, and it increases as the WMHN becomes larger. This complexity reduction may be translated into a routing-latency reduction of 202%, 466% and 908% for WMHNs consisting of seven, eight and nine nodes, respectively.

Last but not least, a substantial complexity reduction is offered by the NDQO algorithm compared to the naive-BF method for WMHNs consisting of five or more nodes. According to Fig. 4.5, a complexity reduction of about an order of magnitude is offered by the NDQO algorithm for 6-node WMHNs, which is increased to several orders of magnitude for WMHNs supporting more than six nodes.

4.6.2 NDQO Computational Accuracy

Before delving into the related NDQO accuracy discussions, the three metrics of computational accuracy that were used should be defined. To begin with, the optimization accuracy may be quantified by the distance from the OPF, which is equal to the average Pareto distance $E[P_d(x)]$ of the OPF exported from the true OPF. Assuming that the exported OPF routes form a set S^{OPF} having a length of $\left|S^{\text{OPF}}\right|$, the *average Pareto Distance* $E[P_d(x)]$ from the OPF becomes:

$$E[P_d(x)] = \sum_{x\in S^{\text{OPF}}} \frac{P_d(x)}{|S^{\text{OPF}}|}. \tag{4.17}$$

Its physical interpretation is given by the average probability of a route belonging to the hitherto generated OPF being dominated by the rest of the legitimate routes. Inherently, if the generated OPF consists exclusively of routes of the true OPF, which is generated by the BF method, the value of this metric would be equal to zero. On the other hand, should the OPF consist of suboptimal points, its value will be bounded by the range $(0, 1]$. Consequently, it becomes plausible that as its value decays and tends to zero, the generated

OPF approaches the true OPF.

Additionally, the accuracy may also be quantified in terms of the *average normalized euclidean distance* between the exported routes, that are erroneously included in the exported OPF, and the specific routes of the true OPF that are closer to the particular erroneous one and they dominate them at the same time. Assuming an exported route x_i from the OPF and another x_j, which corresponds to its counterpart from the true OPF, the related error function $e(x_i)$ can be formulated as:

$$e(x_i) = \frac{1}{\sqrt{K}} \sqrt{\sum_{k=1}^{K} \left(\frac{f_k(x_i) - f_k(x_j)}{f_k(x_j)} \right)^2}, \tag{4.18}$$

where K is the total number of UFs. From this perspective, the average error $E[e(x)]$ would be equal to:

$$E[e(x)] = \sum_{x \in S^{\mathrm{OPF}}} \frac{e(x)}{|S^{\mathrm{OPF}}|}. \tag{4.19}$$

This metric has the advantage that its value becomes independent from the distribution of the routes within the higher-rank Pareto Fronts. This is desirable, because there may exist route-solutions, which would potentially belong to a suboptimal PF, which is pretty close to the OPF, while their solution vectors are rather distant from those of their OPF counterparts. However, its value in Eq. (4.19) would inherently depend on the actual values of the route solution vector. This is in contrast to the average Pareto distance. Explicitly, the value of Eq. (4.19) would be bounded by the range $[0, 1]$.

The third metric considered is what we refer to as the *Optimal Pareto Front Completion* C, which is defined as follows. Let us that the optimization process has generated the set S^{OPF} of Pareto-optimal routes of length equal to $|S^{\mathrm{OPF}}|$, while the truly Pareto-optimal routes form the set S^{TOPF}, which we will refer to as the *true OPF* (TOPF), and that the sub-optimal routes not belonging to the true OPF form the set S_e^{OPF} with $S_e^{\mathrm{OPF}} \subseteq S^{\mathrm{OPF}}$, the Optimal Pareto Front Completion C may be defined as:

$$C = \frac{|S^{\mathrm{OPF}}| - |S_e^{\mathrm{OPF}}|}{|S^{\mathrm{TOPF}}|}, \tag{4.20}$$

where $|S^{\mathrm{TOPF}}|$ is the length[6] of the TOPF. Naturally, this metric is bounded to the range $[0, 1]$ and should the entire true OPF be successfully generated it will be equal to unity.

Having defined these metrics, let us now carry out a comparative study. We will consider the Pareto-optimal routing problem of Eq. (2.7) associated with the UV of Eq. (2.5). Subsequently, we will evaluate the OPF generated by the NDQO algorithm and compare its accuracy to that of the NSGA-II and of the MO-ACO algorithm, which are presented in Sections 2.4 and 2.5, respectively. For the NSGA-II and the MO-ACO the evaluation of

[6]The length of a PF may be defined as the number of route-solutions, which it consists of.

the hitherto generated OPF will be provided at the end of each generation process. As for the NDQO algorithm, there is no notion of generations, hence the evaluation process will be invoked each time a route is included to the OPF (right after Step 4.2.13). However, since the total number of CFEs required by the BBHT-QSA chains is a rather stochastic process upper bounded by $L_{BBHT}^{\max}$ CFEs as defined in Eq. (4.16), the evaluation process will be activated at different L_{NDQO}^{tot} values. We will assume that between these evaluation processes the aforementioned accuracy metrics remain constant, which results in a sum of step functions for each simulation. We can then extract a continuous distribution of these metrics versus the number of CFEs by performing an averaging operation. The simulation parameters of the NSGA-II and of the MO-ACO algorithm are presented in Table 2.2, where the parameter L_{CFE}^{tot} corresponds to the maximum complexity observed in terms of the number of CFEs imposed by the NDQO algorithm in our simulations. Explicitly, we have set L_{CFE}^{tot} equal to 1728, 6859 and 25852 CFEs for WMHNs consisting of 6 and 7 nodes, respectively, based on Fig. 4.5.

The accuracy metrics are shown in Figs. 4.6, 4.7 and 4.8 for 6-node, 7-node and 8-node WMHNs, respectively. For all the WMHNs considered, as far as the average Pareto distance $E[P_d(x)]$ is concerned, it becomes clear from Figs. 4.6(a), 4.7(a) and 4.8(a) that the NDQO algorithm exhibits a far better performance than the NSGA-II and the MO-ACO algorithm. Explicitly, observe in Fig. 4.7(a) that for the 7-node WMHNs the NDQO performs optimally for 502 CFEs and then the average Pareto distance $E[P_d(x)]$ would be about 10^{-8}. Additionally, the same effect is visible from Fig. 4.6(a) for the 6-node WMHNs scenario, where the NDQO performs optimally for 288 CFEs. The order of these values suggests that our NDQO algorithm attains a near-optimal performance compared to the BF method, while its complexity is about an order of magnitude lower than the complexity of the BF method for the 7-nodes WMHNs, according to Fig. 4.5. Moreover, according to Figs. 4.6(a), 4.7(a) and 4.8(a), the computational accuracy in terms of the average Pareto distance is several orders of magnitude lower than that of both the NSGA-II and of the MO-ACO algorithm. It should be noted that the associated errors of the NDQO algorithm arise from the inclusion of suboptimal route-solutions into OPF. To elaborate further, the BBHT-QSA involves a small arbitrary error [80], which would imply that there is a slight possibility that a BBHT-QSA time-out will result in a suboptimal route owing to its inability to find another dominating route, which results in misinterpreting it as the optimal one. However, the inclusion of the entire true OPF route set is guaranteed, because all the routes will be explicitly considered. This leads to a generated OPF which consists of all the true OPF routes along with with some low-probability suboptimal ones.

This trend is shown in Figs. 4.6(c), 4.7(c) and 4.8(c) in terms of their Optimal Pareto Front Completion C. More specifically, observe in Fig. 4.7 (c) that although the NSGA-II and the MO-ACO algorithm fail to converge to unity, the NDQO algorithm succeeds in exporting all the routes consisting the true OPF after 5575 CFEs, yielding a 234.22% and 1906.30% improvement for our 7-node WMHN compared to the BF and the naive-BF methods, respectively. This gain is further increased the number of nodes increases. This property of the NDQO algorithm is of great importance, since it enables the reconstruction of the OPF by invoking a classical NDS for discarding the erroneous route-solutions. Nevertheless, this operation would require additional CFEs. This action cannot be invoked for our benchmarking algorithms, since their completeness does not converge to unity and, consequently, they will fail to export all the true OPF route-solutions. As for the 8-node WMHNs, observe in Fig. 4.8(c) that the NDQO algorithm succeeds in identifying the entire OPF after 25852 CFEs, yielding a 540.82% and 14981.30% improvement against the BF and the naive-BF methods, respectively, On the other hand, it is clear from Fig. 4.6 (c) that the entire true OPF is explicitly identified after 1583 CFEs for 6-node WMHNs, yielding the same complexity as the one of the BF method, as it can be seen in Fig. 4.5, yet we need to stress out that the NDQO still outperforms the performances in terms of C of the NSGA-II and the MO-ACO algorithm for 6-node WMHNs.

As far as the average error $E[e(x)]$ is concerned, identical trends to these seen for the average Pareto distance are observed in Figs. 4.6(b), 4.7(b) and 4.8(b); however, the value of this metric is higher than that of the $E[P_d(x)]$. Quantitatively, they are on the order of 10^{-5} for WMHNs considered. This error is about four orders of magnitude lower than the the respective error of both the NSGA-II and the MO-ACO algorithm. Explicitly, our proposed algorithm has a near-optimal accuracy, since each generated OPF route would differ from the closest true OPF, which would potentially dominate it, by about 0.001%.

Last but not least, note that there is a discrepancy between the value of the average error and that of the average Pareto distance. Explicitly, a relatively low Pareto distance value does not imply having an average error on the same order of magnitude. This is justified by the fact that a specific suboptimal route-solution that has been erroneously included in the OPF by any of the NSGA-II, MO-ACO and NDQO Algs. might belong to a rather low rank PF, i.e. to a PF whose routes are dominated by a relatively low number of route-solutions. By contrast, there is a high probability that this specific suboptimal route exhibits a relatively high average error, since the Pareto-optimal route, which dominates it, exhibits far lower UFs. In a nutshell, we can surmise that the trend is the same for both metrics. However, the two metrics give different information about the OPF. More specifically, the Pareto distance demonstrates the rank of the identified OPF with respect to the TOPF. By contrast, the average error demonstrates the average potential improvement in terms of the UFs that the TOPF offers, when compared to the identified OPF.

4.7 Chapter Summary

In this chapter we have presented our contribution in [1]. In particular, we have proposed a near-optimal algorithm for multi-objective routing in WMHNs using Pareto Optimality. The theoretical upper and lower complexity bounds of the NDQO algorithm have been analytically derived, yielding a complexity between $O(N)$ and $O(N\sqrt{N})$. This implies a significant CFE reduction compared to the classical BF method, which exhibits a complexity on the order of $O(N^2)$ in the worst-case scenario. Naturally, this complexity reduction becomes more significant, as the number of nodes increases. As far as its accuracy is concerned, we have demonstrated that the NDQO algorithm exhibits a near-optimal performance whilst attaining several orders of magnitude better accuracy than the state-of-the-art classical evolutionary NSGA-II and MO-ACO algorithms.

Despite the substantial reduction in computational complexity achieved by the NDQO, while retaining a near-optimal performance, when compared to the BF method, the NDQO algorithm seems to provide some room for improvement. To elaborate further, its main issue lies in the fact that the serial processing of the routes, invoked in Step 4.2.3, imposes a complexity equal to N CFEs, which is substantially high for densely populated WMHNs. This issue leads to the need of implementing a sophisticated method with reduced complexity for finding the next initial route-solution that initiates a BBHT-QSA chain, which will in turn reduce the lower bound potentially below $O(N)$.

Moreover, serially processing all the legitimate routes imposes an additional obstacle. Explicitly, this process is stochastic and hectic in certain cases, where there is a need for invoking the NDQO algorithm with a certain number of CFEs. Then, there is no guarantee from the characterization of this algorithm how many OPF route it would generate. Therefore, this process oughts to be improved so that it becomes iterative, giving in turn the NDQO an iterative nature. Additionally, we have mentioned in the Subsection 4.6.2 that the NDQO is capable of identifying the entire true OPF, as it can be observed from Figs. 4.6(c), 4.7(c) and 4.8(c). This provides us with motivation for employing a reduced-complexity mechanism for repairing the OPF generated by the NDQO algorithm, resulting in optimal an performance at the expense of some additional CFEs.

Finally, the issues arising from the characterization of the NDQO algorithm provide us with all the necessary motivation for improving the existing algorithm, while ensuring that its function will be iterative. These improvements are discussed in detail in the next chapter, where we present the so-called *Non-Dominated Quantum Iterative Optimization* (NDQIO) algorithm, which utilizes hybrid synergy of both the QP and the hardware parallelism.

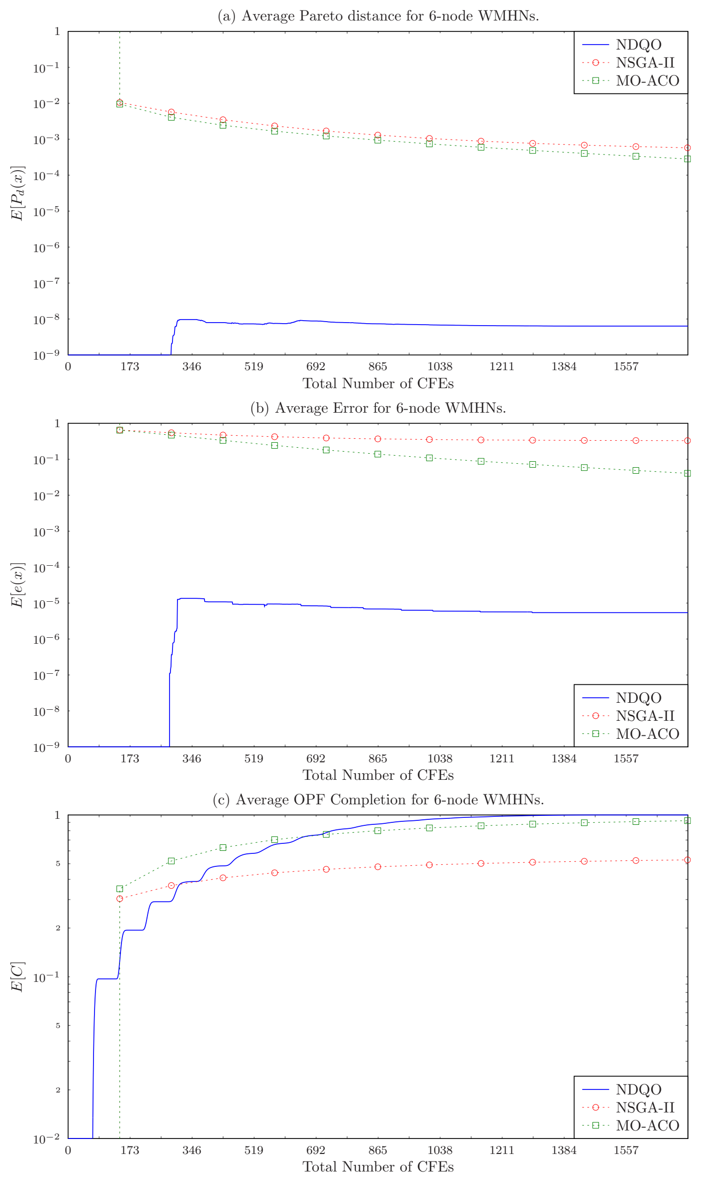


Figure 4.6: Perfomance comparison between the NDQO and the state-of-the-art benchmarking algorithms NSGA-II and MO-ACO for 6-node WMHNs in terms of (a) the Average Pareto Distance $E[P_d(x)]$, (b) the Average Error $E[e(x)]$ and (c) Optimal Pareto Front Completion $E[C]$. For the sake of fairness, the comparisons are made in terms of the number of CFEs for all the algorithms examined. The number of agents has been set equal to the number of the generations for both the NSGA-II and the MO-ACO, which in turn is equal to the cubic root of the maximum NDQO complexity. Therefore, they will be set to 12 for 6-node WMHNs. The results have been averaged over 10^8 runs for WMHNs based on the optimization problem of Eq. (2.7) relying on the UV defined in Eq. (2.5) and on the assumptions of Table 2.1. Finally, the rest of the simulation parameters for both the NSGA-II and the MO-ACO are presented in Table 2.2.

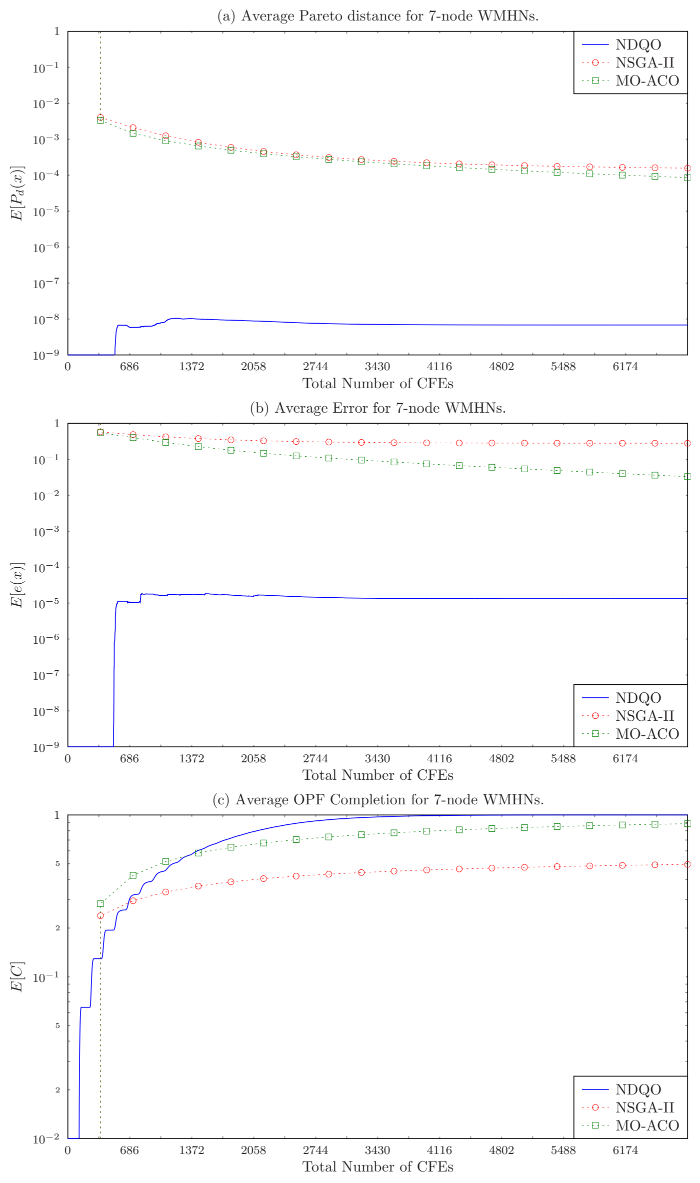


Figure 4.7: Perfomance comparison between the NDQO and the state-of-the-art benchmarking algorithms NSGA-II and MO-ACO for 7-node WMHNs in terms of (a) the Average Pareto Distance $E[P_d(x)]$, (b) the Average Error $E[e(x)]$ and (c) Optimal Pareto Front Completion $E[C]$. For the sake of fairness, the comparisons are made in terms of the number of CFEs for all the algorithms examined. The number of agents has been set equal to the number of the generations for both the NSGA-II and the MO-ACO, which in turn is equal to the cubic root of the maximum NDQO complexity. Therefore, they will be set to 19 for the 7-node WMHNs. The results have been averaged over 10^8 runs for WMHNs based on the optimization problem of Eq. (2.7) relying on the UV defined in Eq. (2.5) and on the assumptions of Table 2.1. Finally, the rest of the simulation parameters for both the NSGA-II and the MO-ACO are presented in Table 2.2.

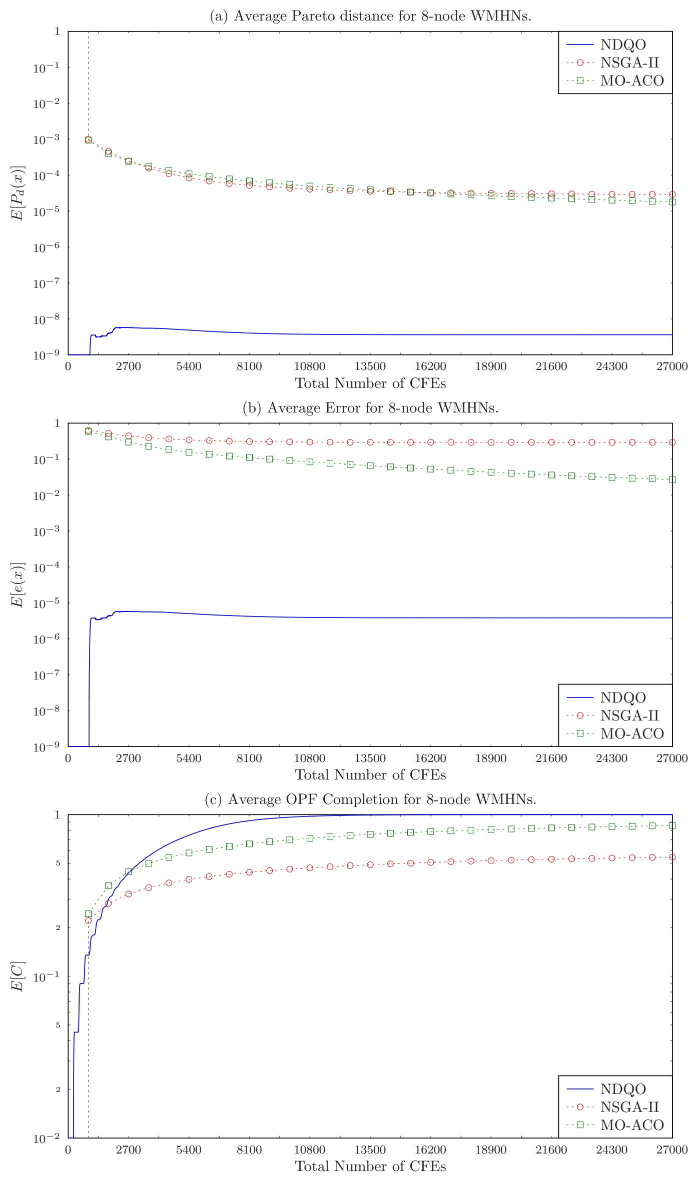


Figure 4.8: Perfomance comparison between the NDQO and the state-of-the-art benchmarking algorithms NSGA-II and MO-ACO for 8-node WMHNs in terms of (a) the Average Pareto Distance $E[P_d(x)]$, (b) the Average Error $E[e(x)]$ and (c) Optimal Pareto Front Completion $E[C]$. For the sake of fairness, the comparisons are made in terms of the number of CFEs for all the algorithms examined. The number of agents has been set equal to the number of the generations for both the NSGA-II and the MO-ACO, which in turn is equal to the cubic root of the maximum NDQO complexity. Therefore, they will be set to 30 for 8-node WMHNs. The results have been averaged over 10^8 runs for WMHNs based on the optimization problem of Eq. (2.7) relying on the UV defined in Eq. (2.5) and on the assumptions of Table 2.1. Finally, the rest of the simulation parameters for both the NSGA-II and the MO-ACO are presented in Table 2.2.

Chapter 5

Non-dominated Quantum Iterative Optimization

5.1 Introduction

As we pointed out in Section 4.7, the NDQO algorithm exhibits some limitations, which do not allow the complexity reduction to reach its full potential. More specifically, we have mentioned that serially processing all the legitimate routes provides us with an explicit asymptotic lower bound on the order of $O(N)$, where N denotes the total number of legitimate routes. This lower bound makes the employment of the NDQO algorithm gradually more infeasible, as the number of nodes that populate the WMHN increases, leading to an exponential increase in the total number of legitimate routes. For example, for a 12-node WMHN, based on Eq. (2.8) the resultant number of legitimate routes becomes $N = 2^{24} = 16,777,216$, making its multi-objective routing optimization solely feasible by employing sub-optimal heuristic algorithms such as the NSGA-II [29] and the MO-ACO [52] of Sections 2.4 and 2.5, respectively. Clearly, the BF method cannot be applied due to its excessive requirements in both computations and memory. Therefore, our main challenge is to further reduce the computational complexity, so that the lower bound tends to a lower order, such as $O(\sqrt{N})$, which corresponds to the complexity imposed by Grover's QSA, as presented in Section 3.3.2. This may be achieved by converting the serial search into a predominantly parallel quantum-assisted sub-process.

In addition to this computational complexity reduction, our algorithm has to perform optimally in terms of its heuristic accuracy. Therefore, based on the fact that the NDQO algorithm is able to identify the entire true OPF by reaching a Pareto Completion Ratio C equal to unity based on Figs. 4.6(c), 4.7(c) and 4.8(c) of Section 4.6.2, we have to employ an OPF *self-repairing process* for addressing this issue. However, this process would impose an additional amount of CFEs, which would be traded for the sake of attaining am increased accuracy. Moreover, we intend to structure our new algorithm so that it exhibits an iterative nature. This design requirement is of great importance for the case, when a time- or power-dissipation constraint is imposed in conjunction with a requirement for the minimum number of identified OPF route-solutions.

Furthermore, another issue that has not been considered in Section 4.7 is how route-processing is being conducted. Explicitly, there are several contributions [144,145,146,147, 148,149,150], which use another realm of parallelism, namely that of the *Hardware Parallelism* (HP), for achieving a complexity reduction, while addressing the routing problem. The complexity reduction offered by HP has been mainly enabled through the use of *Graphics Processing Units* (GPU) [151] architectures for general purpose programming [152] apart from *Central Processing Units* (CPU). As for the routing problem, Han *et al.* [144] introduced a hybrid GPU-CPU concurrent framework for routers, which offered a substantial complexity reduction in the context of the global routing problem. Additionally, both Mu *et al.* [145] and Zhao *et al.* [146] proposed their routing algorithms, which were tailored for *Internet Protocol* (IP) routers having GPU architectures. In a similar context, namely that of the *Travelling Salesman Problem* (TSP), Uchida *et al.* [148] conceived a parallel implementation of the *Ant Colony Optimization* (ACO) algorithm, while Cekmez *et al.* [149] deployed a parallel version of the GA, both for addressing the TSP problem using GPUs.

Consequently, prior to setting the design goal for our new algorithm, we will present a brief overview of the prevalent processing techniques used for solving combinatorial problems. In particular, these techniques may be classified into three major categories [87], - namely *serial processing*, *parallel processing* relying on hardware parallelism and *quantum processing* relying on quantum parallelism - as portrayed in Fig. 5.1, which might be visualized for the sake of simplicity using the paradigm of unlocking a specific lock. We note that in Fig. 5.1 the keys symbolize the set of routes, while the event of inserting a specific key in the keyhole represents processing the corresponding route.

In *serial processing* the full set of available keys has to be checked sequentially for each keyhole to ascertain as to whether they do or do not unlock the door. This type of processing is referred to as "Pure Serial Processing" in Fig. 5.1. The employment of graphics processing units in routing [145,146,147,148,149] has unveiled a new perspective, where all the keys can be simultaneously inserted into identical keyholes for unlocking the door; this type of process is referred to as "Pure Parallel Processing" in Fig. 5.1. Nevertheless, there are practical cases, where the parallel processes have to be synchronized [144, 148, 149], leading to an inevitable decomposition of the task into a series of carefully coordinated parallel steps. This type of processing is referred to as "Hybrid Parallel-Serial Processing" in Fig. 5.1, where the consecutive doors denote the decomposition of the task.

Subsequently, quantum processing has been brought to the limelight, as a benefit of the advances in quantum computing [74, 75, 76, 77, 79, 80, 81, 78, 82, 83, 85, 89, 86]. In fact, there are algorithms relying solely on the philosophy of QP [76, 77, 79, 78, 83], such as Grover's QSA presented in Section 3.3, which are classified as "Pure Quantum Processing" in Fig. 5.1. Explicitly, a "quantum" keyhole is represented as an elaborate single lock having multiple keyholes for portraying the principles of the QP [62]. On the other hand, some other quantum algorithms, such as the BBHT-QSA [80], the DHA [81] and the NDQO algorithm [1] presented in Sections 3.4, 3.5 and 4.4, respectively, decompose the overall task

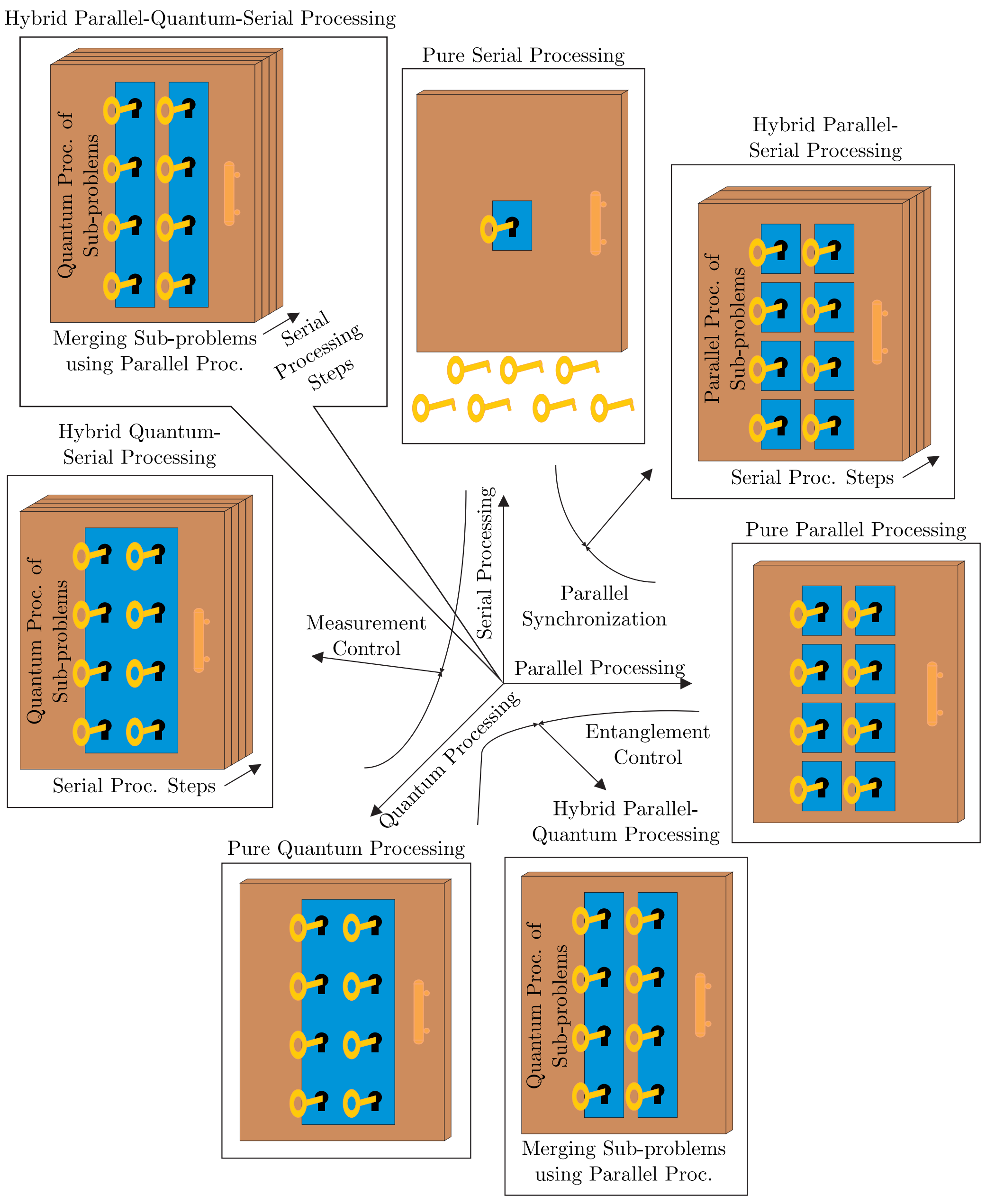

Hybrid Parallel-Quantum-Serial Processing
Quantum Proc. of Sub-problems
Merging Sub-problems using Parallel Proc.
Serial Processing Steps
Pure Serial Processing
Hybrid Parallel-Serial Processing
Parallel Proc. of Sub-problems
Serial Proc. Steps
Hybrid Quantum-Serial Processing
Quantum Proc. of Sub-problems
Serial Proc. Steps
Serial Processing
Parallel Synchronization
Measurement Control
Parallel Processing
Pure Parallel Processing
Entanglement Control
Quantum Processing
Hybrid Parallel-Quantum Processing
Pure Quantum Processing
Quantum Proc. of Sub-problems
Merging Sub-problems using Parallel Proc.

Figure 5.1: Pure processing categories along with their respective hybrid cases. The example used in the illustation was inspired by [87].

into sequential sub-problems, which are individually handled with the aid of the QP. This decomposition is inherently necessary, since these algorithms involve the *measurement* or *observation* operation [67], which terminates the quantum processes by collapsing the effect of the QP [62]. This type of processing is termed as “Hybrid Quantum-Serial Processing” in Fig. 5.1. Additionally, some independent processes may be simultaneously invoked for the sake of achieving either a further complexity reduction by benefiting both from the QP and from the HP, or a more coherent *entanglement* [62] of the outputs of the independent processes. This specific case is referred to as “Hybrid Parallel-Quantum Processing” in Fig. 5.1. Nevertheless, further decomposition of the overall task into sequential steps may be inevitable due to the *measurement* or *observation* operations that the task may involve. Therefore, this latter type of processing involves a synergy of all the potential processing types and it is hence referred to as “Hybrid Parallel-Quantum-Serial Processing” in Fig. 5.1.

This hybrid case will be our lead for designing the quantum oracle gates of our new algorithm, namely for *Non-Dominated Quantum Iterative Optimization* (NDQIO). The discussions regarding quantum oracles benefiting from the HP are presented in Section 5.2. Subsequently, we will elaborate on the NDQIO's algorithmic step in Section 5.3. We note that in this chapter we will consider the same network model as that used in Chapter 4, namely the Pareto optimality problem of Eq. (2.7) using the UV defined in Eq. (2.5) and complying with the specifications of Table 2.1. In addition to these assumptions, we consider the case of weak Pareto optimality presented in Definition 3, as in the NDQO algorithm of Chapter 4. We will then utilize the same 5-node WMHN used in the tutorial of Section 4.5 for the sake of presenting a low-paced tutorial on the NDQIO algorithm's sub-processes in Section 5.4. Finally, we will analytically characterize the complexity imposed by the NDQIO algorithm and evaluate its heuristic accuracy in Sections 5.5.1 and 5.5.2, respectively. Note that we will compare the NDQIO algorithm's heuristic search to that of the NSGA-II, the MO-ACO and the NDQO algorithms of Sections 2.4, 2.5 and 4.4, respectively. Let us now proceed with a detailed description of the HP-empowered quantum oracle gates, which we will utilize in the NDQIO algorithm.

5.2 Parallel Oracle Design

In the introduction of this chapter, we have mentioned that it is possible to perform independent quantum operations in parallel. However, we pointed out that an entanglement control process is necessary for the synchronization of independent quantum processes. Before delving into the details of the design process, let us provide a brief description of the quantum entanglement phenomenon [153]. For this reason, let us assume a unitary operator U_f, which implements a binary function $f(x) : \{0, 1, \ldots, N-1\} \rightarrow \{0, 1\}$, as shown in Fig. 5.2. We note that the unitary operator U_f is similar to that presented in Fig. 3.1; their difference relies upon the fact that the OW quantum register is initialized to the arbitrary state $|t\rangle_2$. Additionally, let us consider that the QIR input $|x\rangle_1$ is set to the superimposed state:

$$|x\rangle_1 = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} |n\rangle \, , \tag{5.1}$$

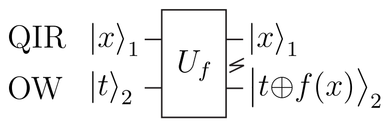


Figure 5.2: Quantum circuit of a unitary operator U_f implementing a binary function $f(x) : \{0, 1, \ldots, N-1\} \rightarrow \{0, 1\}$. The non-staight lines between the QIR and OW outputs denote quantum entanglement of their states.

where $\{|n\rangle\}_{n=0}^{N-1}$ corresponds to all the possible inputs of the function $f(x)$, Then, the unitary operator U_f would have the following effect on the input states:

$$|x\rangle_1 |t\rangle_2 \xrightarrow{U_f} |x\rangle_1 |t \oplus f(x)\rangle_2 = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} |n\rangle_1 |t \oplus f(n)\rangle_2. \tag{5.2}$$

The outcome of Eq. (5.2) is indeed a superposition of composite states. Let us now provide an example to portray the concepts of these states. Assuming that we have $N = 4$ and $t = 0$ and that the function $f(x)$ is defined as follows:

$$f(x) = \mod_2(x),\ x \in \{0, 1, 2, 3\}, \tag{5.3}$$

based on Eq. (5.3), the Eq. (5.2) is then transformed into:

$$\begin{aligned} |x\rangle_1 |t\rangle_2 \xrightarrow{U_f} |x\rangle_1 |0 \oplus f(x)\rangle_2 &= \frac{1}{2} \sum_{n=0}^{3} |n\rangle_1 |t \oplus f(n)\rangle_2 &(5.4)\\ &= \frac{1}{2} |0\rangle_1 |0\rangle_2 + \frac{1}{2} |1\rangle_1 |1\rangle_2 + \frac{1}{2} |2\rangle_1 |0\rangle_2 + \frac{1}{2} |3\rangle_1 |1\rangle_2 &(5.5)\\ &= \frac{1}{2} (|0\rangle_1 + |2\rangle_1) |0\rangle_2 + \frac{1}{2} (|1\rangle_1 + |3\rangle_1) |1\rangle_2 &(5.6) \end{aligned}$$

At this stage, let us assume that we attempt to perform a measurement on the QIR output, i.e. that we attempt to observe the output of the register $|x\rangle_1$. This type of observation is often referred to as *partial measurement* [87]. Based on Eqs. (3.12) and (5.5), the probability $p(n)$ of observing the state $|n\rangle_1$, where we have $n \in \{0, 1, 2, 3\}$, is equal to:

$$p(n) = \sum_{\forall y \in \{0,1\}} \left| a_{|n\rangle_1 |y\rangle_2} \right|^2 = 1/4,\ \forall n \in \{0, 1, 2, 3\}, \tag{5.7}$$

where $a_{|n\rangle_1 |y\rangle_2}$ corresponds to the amplitude of the composite state $|n\rangle_1 |y\rangle_2$. Based on Eqs. (3.16), (3.17) and (5.5), the post-measurement product $|q_{new}\rangle$ is equal to [87]:

$$|q_{new}\rangle|_{n=0} = \frac{\sum_{\forall y \in \{0,1\}} a_{|0\rangle_1 |y\rangle_2} |0\rangle_1 |y\rangle_2}{\sqrt{p(0)}} = |0\rangle_1 |0\rangle_2, \tag{5.8}$$

$$|q_{new}\rangle|_{n=1} = \frac{\sum_{\forall y \in \{0,1\}} a_{|1\rangle_1 |y\rangle_2} |1\rangle_1 |y\rangle_2}{\sqrt{p(1)}} = |1\rangle_1 |1\rangle_2, \tag{5.9}$$

$$|q_{new}\rangle|_{n=2} = \frac{\sum_{\forall y \in \{0,1\}} a_{|2\rangle_1 |y\rangle_2} |2\rangle_1 |y\rangle_2}{\sqrt{p(2)}} = |2\rangle_1 |0\rangle_2, \tag{5.10}$$

$$|q_{new}\rangle|_{n=3} = \frac{\sum_{\forall y \in \{0,1\}} a_{|3\rangle_1 |y\rangle_2} |3\rangle_1 |y\rangle_2}{\sqrt{p(3)}} = |3\rangle_1 |1\rangle_2. \tag{5.11}$$

If we assume the observable $|n\rangle_1$ and take into account the Eq. (5.3), the above Eqs. can be grouped into a single one, which is equal to:

$$|q_{new}\rangle|_n = \frac{a_{|n\rangle_1|f(n)\rangle_2}|n\rangle_1|f(n)\rangle_2}{\sqrt{p(n)}} = |n\rangle_1|f(n)\rangle_2. \tag{5.12}$$

Observe in Eq. (5.12) that although a partial measurement has been conducted on the QIR, the OW register also collapses to the respective classical state depending on the outcome of Eq. (5.3), due to the linkage caused by quantum entanglement introduced by U_f.

Moving on to the design of the NDQIO algorithm oracle gates, let us now assume that we have K different unitary operators U_{f_k}, where we have $k \in \{1, \dots, K\}$. We have mentioned in the introduction of this chapter that we opt for fully parallelizing our independent quantum procedures. Thus, let us assume that we activate these unitary operators in parallel, as portrayed in Fig. 5.3. We note that the QIR input is set equal to the equal superposition of all the possible inputs, i.e. we have:

$$|x\rangle_{k,1} = \frac{1}{\sqrt{N}}\sum_{n=0}^{N-1}|n\rangle_{k,1}. \tag{5.13}$$

Hence, the input state $|y_{in}\rangle$ of the parallel formation of these unitary gates is equal to:

$$|y_{in}\rangle = \frac{1}{\sqrt{N^K}}\sum_{n=0}^{N-1}|n\rangle_{1,1}|t\rangle_{1,2}\cdots\sum_{n=0}^{N-1}|n\rangle_{k,1}|t\rangle_{k,2}\cdots\sum_{n=0}^{N-1}|n\rangle_{K,1}|t\rangle_{K,2}. \tag{5.14}$$

Subsequently, applying the parallel formation of the unitary operators $\{U_{f_k}\}_{k=1}^{K}$ into the input state of Eq. (5.14) results in the following output state $|y_{out}\rangle$:

$$|y_{out}\rangle = \frac{1}{\sqrt{N^K}}\sum_{n=0}^{N-1}|n\rangle_{1,1}|t\oplus f_1(n)\rangle_{1,2}\cdots\sum_{n=0}^{N-1}|n\rangle_{k,1}|t\oplus f_k(n)\rangle_{k,2}\cdots\sum_{n=0}^{N-1}|n\rangle_{K,1}|t\oplus f_K(n)\rangle_{K,2}. \tag{5.15}$$

The output state $|y_{out}\rangle$ of Eq. (5.15) is clearly not an entangled composite state, since the

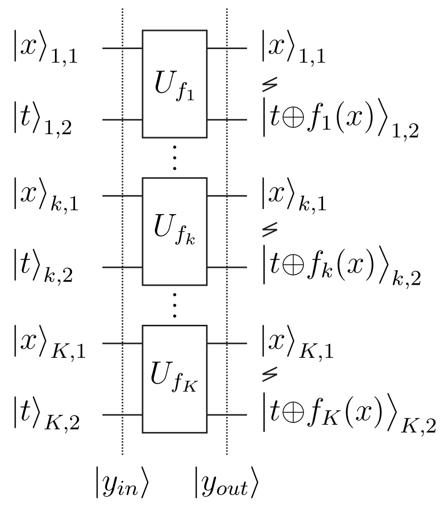


Figure 5.3: Unitary operators implementing the functions $\{f_k(x)\}_{k=1}^{K}$ in parallel.

output states of different unitary operators are not correlated in the absence of quantum entanglement. This setup poses a strong limitation, since it is not possible to combine the operator outputs together in the same fashion as the NDQO quantum oracle of Fig. 4.1, where a Tofolli gate [62] was used for this reason. However, if the input QRs $\left\{|x\rangle_{k,1}\right\}_{k=1}^{K}$ are indeed entangled together, the output state $|y_{out}\rangle$ becomes equal to:

$$|y_{out}\rangle = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} |n\rangle_{1,1} |t \oplus f_1(n)\rangle_{1,2} \dots |n\rangle_{k,1} |t \oplus f_k(n)\rangle_{k,2} \dots |n\rangle_{K,1} |t \oplus f_K(n)\rangle_{K,2} . \tag{5.16}$$

Observe now that the output state $|y_{out}\rangle$ of Eq. (5.16) is indeed an entangled composite state and, thus, it is possible to combine the outputs $\left\{|t \oplus f_k(n)\rangle_{k,2}\right\}_{k=1}^{K}$ in the same fashion as in the NDQO oracle gate using a *Toffoli Gate* [62]. We remind that the n-qubit Toffoli gate T [143] has been defined in Eq. (4.3). In our specific scenario, each operator U_{f_k} has a single-qubit LOW output and the GOW is also comprised of a single qubit, while the UV of Eq (2.5) is comprised by $K = 3$ UFs. Therefore, we will use a 4-qubit Toffoli gate, i.e. we have $n = 4$ qubits in Eq. (4.3).

Having this observation as our motivation, let us now proceed by presenting the parallel implementation of the strong dominance operator. More specifically, we propose the employment of the unitary operator $U_{g'}$, the quantum circuit of which is shown in Fig. 5.4. Its main difference to the unitary operator U_g of the NDQO algorithm, which is presented in Fig 4.1, lies in the use of CNOT gates right before the input of the unitary operators U_{f_k} that implement the low-comparison-check. By contrast, in the unitary operator U_g Fig. 4.1 the QIR and QCR outputs of the unitary operator U_{f_k} are fed forward to the $U_{f_{k+1}}$ operator. These CNOT gates are employed for the sake of entangling the states of both the *Global Quantum Control Register* (GQCR) and the *Global Quantum Index Register* (GQIR) to the respective local ones, i.e to the *Local Quantum Control Registers* (LQCRs) and to the *Local Quantum Index Registers* (LQIRs), for creating entangled composite states. Their employment is essential for forming the entangled composite state:

$$|y\rangle_{LOW}^{out} = \sum_{x=0}^{N-1} |f_1(x,i)\rangle_{1,3} |f_2(x,i)\rangle_{2,3} |f_3(x,i)\rangle_{3,3} \tag{5.17}$$

at the LOW outputs. In the absence of entanglement, the respective LOW output state will be:

$$|y'\rangle_{LOW}^{out} = \sum_{x_1=0}^{N-1} \sum_{x_2=0}^{N-1} \sum_{x_3=0}^{N-1} |f_1(x_1,i)\rangle_{1,3} |f_2(x_2,i)\rangle_{2,3} |f_3(x_3,i)\rangle_{3,3} , \tag{5.18}$$

making the implementation of the dominance operator infeasible.

With the entanglement achieved by the employment of the series of the CNOT gates, the $U_{g'}$ operator has identical effect as the U_g operator, i.e. we have:

$$\sum_{x=0}^{N-1} |i\rangle_1 |x\rangle_2 |t\rangle_3 \overset{U_g,U_{g'}}{\longrightarrow} \sum_{x=0}^{N-1} |i\rangle_1 |x\rangle_2 |t \oplus g(x,i)\rangle_3 , \tag{5.19}$$

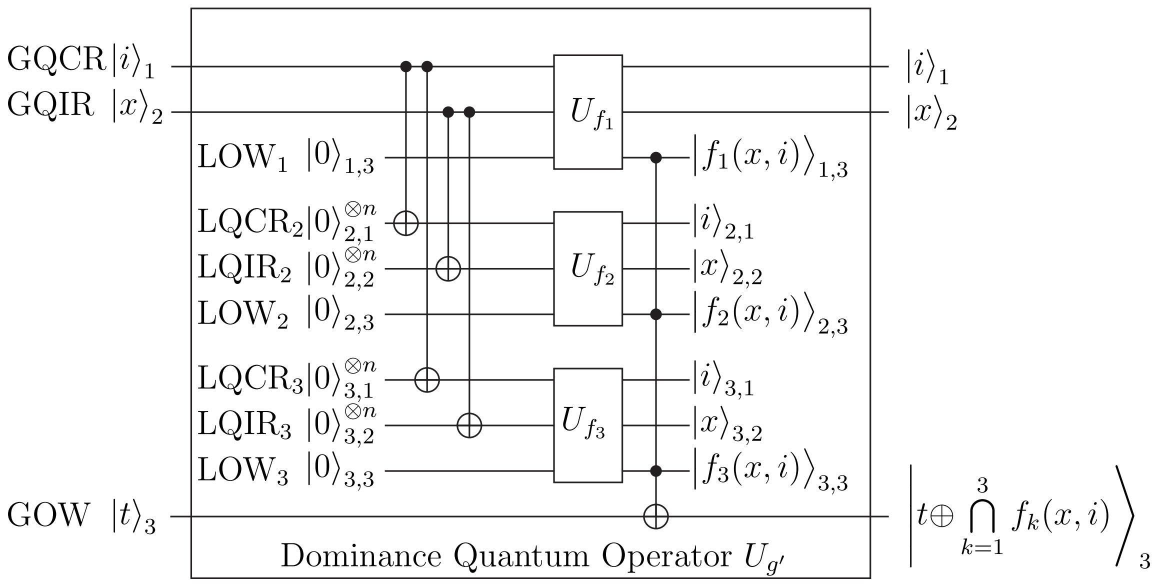

GQCR $|i\rangle_1$
GQIR $|x\rangle_2$
LOW$_1$ $|0\rangle_{1,3}$
LQCR$_2$ $|0\rangle_{2,1}^{\otimes n}$
LQIR$_2$ $|0\rangle_{2,2}^{\otimes n}$
LOW$_2$ $|0\rangle_{2,3}$
LQCR$_3$ $|0\rangle_{3,1}^{\otimes n}$
LQIR$_3$ $|0\rangle_{3,2}^{\otimes n}$
LOW$_3$ $|0\rangle_{3,3}$
GOW $|t\rangle_3$
U_{f_1}
U_{f_2}
U_{f_3}
$|i\rangle_1$
$|x\rangle_2$
$|f_1(x,i)\rangle_{1,3}$
$|i\rangle_{2,1}$
$|x\rangle_{2,2}$
$|f_2(x,i)\rangle_{2,3}$
$|i\rangle_{3,1}$
$|x\rangle_{3,2}$
$|f_3(x,i)\rangle_{3,3}$
$\left|t\oplus \bigcap_{k=1}^{3} f_k(x,i)\right\rangle_3$
Dominance Quantum Operator $U_{g'}$

Figure 5.4: Parallel implementation of the domincance operator used in NDQIO algorithm, when considering 3 UFs.

where the function $g(x, i)$, which is defined in Eq (4.1), corresponds to the strong Pareto dominance comparison between the route associated with the index x and the reference route associated with the index i. Consequently, if we set the GOW register to the state $|0\rangle_3$, the $U_{g'}$ operator would return the outcome of the dominance operator and store it in the GOW output. Otherwise, if the GOW register is set to the state $|-\rangle_3$, then the $U_{g'}$ operator operates identically to the Grover's QSA Quantum Oracle Gate O, flipping the phase of the route-solutions, which satisfy the condition of having $g(x, i) = 1$. Furthermore, the main improvement of the new design relies on the fact that we have achieved parallel activation of the unitary operators U_{f_k}.

As for the computational complexity imposed by the $U_{g'}$ operator, we will utilize two distinct complexity metrics, namely the *parallel complexity* and the *sequential complexity*. To elaborate further, the parallel complexity considers the degree of hardware parallelism in the quantum circuits and it is deemed to be commensurate with the circuit's normalized time execution. As for the sequential complexity, it does not take into account the degree of parallelism offered by the quantum circuit's architecture. Hence, the latter is deemed to be commensurate with the circuit's normalized power consumption. Both these complexity metrics rely on the number of CFEs. As for the definition of a single CFE, we will assume that a single CFE is imposed by a single activation of the U_g shown in Fig. 4.1 both in the parallel and in the sequential complexity domains. Note that we have opted for this specific definition for the sake of complying with the complexity analysis of both the U_g operator and of the NDQO algorithm presented in Sections 4.2 and 4.6.1, respectively. Based on the aforementioned assumptions, the $U_{g'}$ operator requires a single CFE in terms of its sequential complexity, since the same number of K U_{f_k} operators is activated as in the U_g operator. By contrast, due to the parallel activation of the U_{f_k} operators within the $U_{g'}$ operator of Fig. 5.4, a single activation of the $U_{g'}$ imposes a parallel complexity equal to $1/K$ CFEs. In our specific application, namely in the Pareto optimaltiy problem of

Eq. (2.7) associated with the UV of Eq. (2.5), we have set $K = 3$, since the UV considered consists of 3 UFs. We note that for the sake of simplicity we have assumed that both the series of CNOT gates and the Toffoli gate consume negligible power compared to the U_{f_k} operators and that their response time is negligible. Having presented our novel parallel oracle design, let us now proceed with our detailed discussions of the NDQIO algorithm.

5.3 Non-Dominated Quantum Iterative Optimization

Our ultimate target is to reduce the lower bound of the complexity below the $O(N)$ complexity dependence, yielding a further reduction of the average complexity. Therefore, we have revisited the framework presented in Section 4.4 with the objective of conceiving a hybrid design relying both on hardware parallelism and on quantum parallelism. More specifically, we will introduce a low-complexity initialization process for identifying the globally optimal routes, along with a sophisticated quantum-assisted process for finding new and potentially optimal routes. Furthermore, we have introduced an OPF *Self-Repair* (SR) process, which discards the suboptimal routes that have been erroneously included in the OPF, hence providing the NDQIO with an improved accuracy compared to the near-optimal NDQO algorithm's accuracy, as it can be readily verified from Figs. 4.6(a,b), 4.7(a,b) and 4.8(a,b).

The NDQIO algorithm is formally stated in Alg. 5.1, where each distinct block is annotated using a comment starting with the character "#". Additionally, we provide its flowchart, which is shown in Fig. 5.5. In a nutshell, the NDQIO algorithm initializes the OPF to an empty set during Step 5.1.1 and then invokes the initialization process of Steps 5.1.3-6, where the DHA is activated as many times as the number of optimization objectives for the sake of identifying the globally optimal routes in terms of each objective. Subsequently, the iterative part of the algorithm is activated. At each iteration of Steps 5.1.8-30, the algorithm initially searches for a route, which is not dominated by the hitherto generated OPF using the BBHT-QSA process of Step 5.1.12. Should it succeed in identifying an appropriate route, it activates the BBHT-QSA chain of Steps 5.1.20-25, in the same fashion as the one in the NDQO algorithm. After the completion of the chain, the OPF Self-Repair (OPF-SR) process is invoked in Steps 5.1.27-28, where the routes of the OPF generated so far are checked to ascertain, whether they are dominated by the optimal route identified by the current iteration of the BBHT-QSA chain of Steps 5.1.20-25. The algorithm terminates and outputs the OPF, when the BBHT-QSA fails to identify a new potentially optimal route as formally defined by the condition of Step 5.1.30, concluding that there exists no other Pareto optimal route. Last but not least, we note that all the single-objective comparisons as well as the dominance operator activation have been carried out using the same quantum unitary operators as those used for forming the QSAs quantum oracles; the only difference relies upon the fact that in the OW register the input is initialized to the $|0\rangle$ state. Therefore, in constrast to the NDQO algorithm formally presented in Alg.4.2, in the NDQIO algorithm there is no distinction between the CD- and the QD-CFEs, since they are exclusively undertaken in the QD. Let us now proceed with

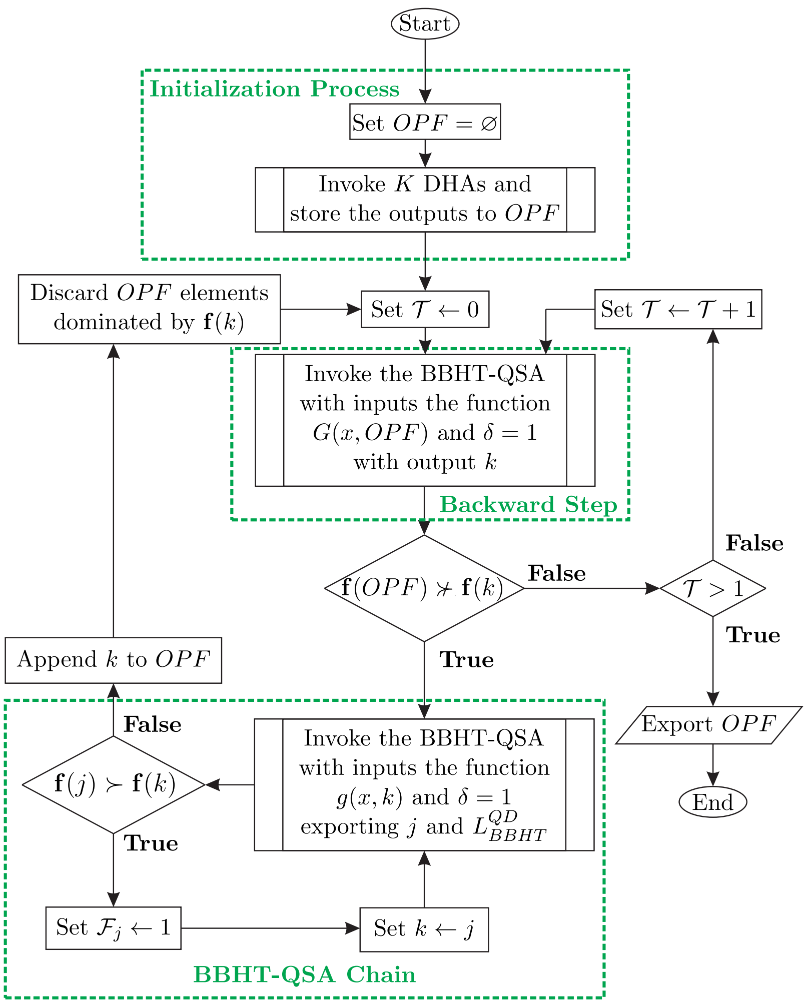

Start
Initialization Process
Set $OPF = \varnothing$
Invoke K DHAs and store the outputs to OPF
Discard OPF elements dominated by $\mathbf{f}(k)$
Set $\mathcal{T} \leftarrow 0$
Set $\mathcal{T} \leftarrow \mathcal{T} + 1$
Invoke the BBHT-QSA with inputs the function $G(x, OPF)$ and $\delta = 1$ with output k
Backward Step
$\mathbf{f}(OPF) \nsucc \mathbf{f}(k)$
False
$\mathcal{T} > 1$
False
True
True
Export OPF
End
Append k to OPF
False
$\mathbf{f}(j) \succ \mathbf{f}(k)$
True
Invoke the BBHT-QSA with inputs the function $g(x, k)$ and $\delta = 1$ exporting j and L_{BBHT}^{QD}
Set $\mathcal{F}_j \leftarrow 1$
Set $k \leftarrow j$
BBHT-QSA Chain

Figure 5.5: NDQIO algorithm's flowchart.

our detailed discussions on each distinct sub-process of the NDQIO algorithm.

5.3.1 Initialization Process

In the NDQO algorithm, which is formally stated in Alg. 4.2, no initialization process has been used; instead, the algorithm considers by default the first index of the legitimate route list as the first reference route and then initiates a BBHT-QSA chain. Despite the reduction offered by the hardware parallelization, the power consumption remains the same as that of the NDQO algorithm, as we demonstrated in Section 5.2. In fact, it is possible to achieve the same reduction in the power consumption as well by using the DHA for identifying the globally optimal routes in terms of each objective. To elaborate further, a single unitary operator U_{f_k}, which is defined in Eq. (4.2), is used for implementing a comparison in terms of the k-th objective, yielding a reduction in the sequential complexity per DHA activation, which is proportional to the number of optimization objectives. In fact, assuming K optimization objectives, this sequential complexity reduction results in identifying K OPF routes, while consuming the same amount of parallel and sequential complexity as in a single NDQO BBHT-QSA chain, which would only identify a single

Algorithm 5.1 Non-Dominated Quantum Iterative Optimization Algorithm

```
1: Set $OPF \leftarrow \varnothing$.
2: # Initialization Process:
3: for $k = 1$ to $K$ do
4:     Invoke the DHA of Alg. 5.2 with input function $f_k(x, i)$, where $i$ is the index of a
       random legitimate route, and output $x_s$.
5:     Append $x_s$ to the $OPF$.
6: end for
7: # Iterative Step:
8: repeat
9:     # Backward BBHT-QSA Step:
10:    Set $\mathcal{F} \leftarrow 0$ and $\mathcal{T} \leftarrow 0$.
11:    repeat
12:        Invoke the BBHT of Alg. 4.1 with input the function $G(x, OPF)$ defined in
           Eq. (5.20) and output $x_s$.
13:        if $G(x_s, OPF) = 1$ then
14:            Set $\mathcal{F} \leftarrow 2$ and $\mathcal{T} \leftarrow 1$.
15:        else
16:            Set $\mathcal{F} \leftarrow \mathcal{F} + 1$.
17:        end if
18:    until $\mathcal{F} = 2$
19:    if $\mathcal{T} = 1$ then
20:        # BBHT-QSA Chain:
21:        repeat
22:            Set $i \leftarrow x_s$.
23:            Define the oracle function $g(x, i)$ from (4.1).
24:            Invoke the BBHT-QSA of of Alg. 4.1 with input $g'(x, i)$ and output $x_s$.
25:        until $\mathbf{f}(x_s) \not\succ \mathbf{f}(i)$.
26:        # Self-Repair Mechanism:
27:        Discard the routes from the OPF that are dominated by the $i$-th one.
28:        Append $i$ to the OPF.
29:    end if
30: until $\mathcal{T} = 0$.
31: Export the $OPF$ and exit.
```

OPF route. More explicitly, a single DHA activation imposes the same amount of parallel complexity, when compared to a single NDQO BBHT-QSA chain, while it simultaneously is also offering a sequential complexity reduction by a factor of $1/K$. We note that in our case study we have considered $K = 3$ optimization objectives according to Eq. (2.5). However, the application of the DHA is limited to identifying only globally optimal routes in terms of a single objective and not the Pareto optimal routes in general. Hence, the number of DHA activations is strictly limited to a maximum K routes. Additionally, we note that we have used an improved version of the DHA, which has been initially proposed in [94], and terminates the algorithm as soon as the BBHT-QSA fails to spot a legitimate solution, while exhausting its maximum affordable number of $\mathcal{G}$ applications. This improved version of the DHA is formally stated in Alg. 5.2.

Algorithm 5.2 Improved Durr-Høyer Algorithm [94]

```
1: Choose a reference index $0 \leq y' \leq N-1$ randomly from the uniform distribution.
2: Set $L_{DHA}^{QD} \leftarrow 0$.
3: repeat
4:     Set $y \leftarrow y'$.
5:     Define the quantum oracle implementing the binary function $f_k(x,i)$ of Eq. (4.2)
       and set $i \leftarrow y$.
6:     Invoke the BBHT-QSA process of Alg. 4.1 with input the function $f_k(x,y)$ and output
       the index $y'$, using $L_{BBHT}^{QD}$ CFEs.
7:     Set $L_{DHA}^{QD} \leftarrow L_{DHA}^{QD} + L_{BBHT}^{QD}$.
8: until $f_k(y',y) \neq 1$ or $L_{DHA}^{QD} \geq \lceil 22.5\sqrt{N} \rceil$
9: Output $y$ and exit.
```

5.3.2 Seeking the Next Optimal Route

After the completion of the initialization process we will acquire an OPF consisting of k OPF routes, where we have $k \in \{1, 2, ..., K\}$. The maximum value of $k = K$ corresponds to the case, where each objective is optimized by finding different routes, while the minimum value of $k = 1$ corresponds to the case, where only a single optimal route-solution exists, which is globally optimal for all the objectives considered. In the latter case, based on Definition 2, the true OPF will solely be comprised of this route. Nevertheless, since the BBHT-QSA and, inherently, the DHA exhibit a small but non-negligible probability of failing to identify a valid solution [80], the algorithm has to ensure that there are no unidentified Pareto optimal routes.

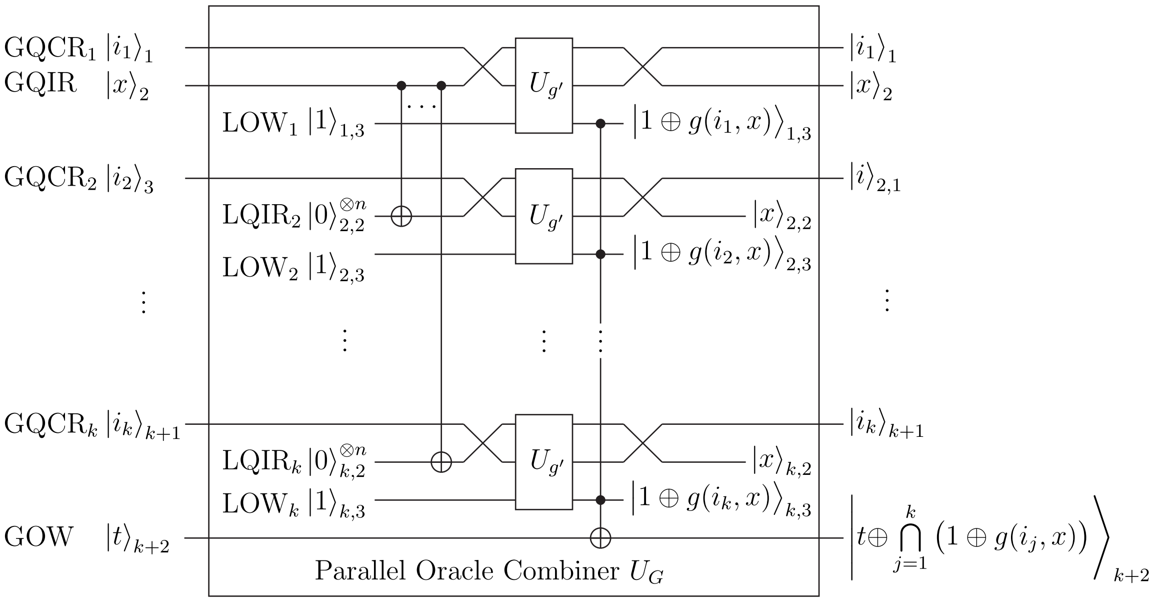
In the NDQIO algorithm, we have avoided the serial processing of the routes by employing a BBHT-QSA for finding the next potentially Pareto optimal route, hence achieving some complexity reduction. To elaborate further, our algorithm searches for route-solutions, which are not dominated by the OPF generated so far. Explicitly, the potentially Pareto-optimal route-solutions should satisfy the condition $G(x, OPF) = 1$, where the function $G(x, OPF)$ is defined as follows:

$$G(x, OPF) = \begin{cases} 1, \ \nexists j \in OPF : g(j,x) = 1 \\ 0, \ \text{otherwise} \end{cases} = \bigcap_{j=1}^{|OPF|} [1 \oplus g(j,i)]. \tag{5.20}$$

For this reason, we can use our novel operator $U_{g'}$ for checking as to whether a route is or is not dominated by a reference route. This is realized by performing a swap between the states stored in the GQCR and the GQIR resulting in the state $|t \oplus g(i,x)\rangle_3$ at the GOW output of the $U_{g'}$ operator. In this case, we initialize the GOW input to the state $|t\rangle_3 \leftarrow |1\rangle_3$, while the respective GOW output becomes $|1 \oplus g(i,x)\rangle$, resulting in invoking the non-dominance operator. Explicitly, based on Eq. (4.1) the binary function $g(i,x)$ returns whether the i-th route does or does not dominate the x-th route and the operation $[1 \oplus g(i,x)]$ corresponds to the binary complement[1] of this function, implementing the

[1]It returns whether the x-th route is dominated by the i-th one.

non-dominance operator. Additionally, since the OPF is comprised of multiple routes, we



GQCR$_1$ $|i_1\rangle_1$
GQIR $|x\rangle_2$
LOW$_1$ $|1\rangle_{1,3}$
GQCR$_2$ $|i_2\rangle_3$
LQIR$_2$ $|0\rangle^{\otimes n}_{2,2}$
LOW$_2$ $|1\rangle_{2,3}$
GQCR$_k$ $|i_k\rangle_{k+1}$
LQIR$_k$ $|0\rangle^{\otimes n}_{k,2}$
LOW$_k$ $|1\rangle_{k,3}$
GOW $|t\rangle_{k+2}$
$U_{g'}$
$|1 \oplus g(i_1,x)\rangle_{1,3}$
$|x\rangle_{2,2}$
$|1 \oplus g(i_2,x)\rangle_{2,3}$
$|x\rangle_{k,2}$
$|1 \oplus g(i_k,x)\rangle_{k,3}$
$|i_1\rangle_1$
$|x\rangle_2$
$|i\rangle_{2,1}$
$|i_k\rangle_{k+1}$
$\left|t \oplus \bigcap_{j=1}^{k} \left(1 \oplus g(i_j,x)\right)\right\rangle_{k+2}$
Parallel Oracle Combiner U_G

Figure 5.6: Quantum circuit of the BBHT-QSA unitary operator U_G used in the BBHT-QSA Oracle of Step 5.1.12 . Each activation of the U_G operator would impose 1/3 CFEs in execution time domain due to the parallel activation of the unitary operators $U_{g'}$ as well as the parallel activation of the U_{f_k} operators within each $U_{g'}$. In the power consumption domain, a single activation of the U_G imposes as many CFEs as the number of reference routes considered, i.e. the number OPF routes that have been so far generated.

have to use multiple $U_{g'}$ operators, each having a different OPF route as the reference route. Subsequently, using the novel framework presented in the Subsection 5.2, we can still achieve the parallel activation of the $U_{g'}$ operators by employing the series of CNOT gates for entangling the LQIRs state with the state of the GQIR at the input of the $U_{g'}$ operators along with a $(k+1)$-qubit Toffoli Gate, assuming having k reference routes, as portrayed in Fig. 5.6. As for the complexity imposed by a single activation of the U_G operator, it may be deemed to impose a parallel complexity equal to $1/k$ CFEs[2] due to the parallel activation of the $U_{g'}$ unitary operators, which in turn activates the U_{f_k} unitary operators in parallel. As for the sequential complexity imposed by the U_G operator, a single activation of this specific operator imposes as many CFEs as the number of reference routes considered, which corresponds to the number of OPF routes that have been generated so far.

Moving on to the BBHT-QSA for identifying a specific route, which potentially belongs to the OPF, the U_G operator of Fig. 5.6 is used with its GOW register initialized to the state $|-\rangle_{k+2}$ and a BBHT-QSA is invoked with the OPF routes generated so far as reference ones, as stated in Step 5.1.4 . By contrast, should the GOW register be initialized to the state $|0\rangle_{k+2}$, the operator U_G returns the non-dominance outcome at the output of the GOW register. Hence, we will utilize this initialization for performing the CD checks of the BBHT-QSA. Again, the BBHT-QSA exhibits a small but non-negligible probability

[2]We note that we have assumed that a single CFE corresponds to the activation of the serial unitary operator U_g of the NDQO algorithm.

of failing to identify a valid solution [80], while exhausting the maximum number of $\mathcal{G}$ applications. We have mitigated this effect by repeating the BBHT-QSA process for one more additional iteration (Steps 5.1.16 and 5.1.18), for the sake of avoiding the premature termination of the NDQIO algorithm. Explicitly, an erroneous timeout would terminate unexpectedly the NDQIO algorithm, leading to its inability to identify the entire OPF. After identifying a potential OPF route, we may employ a BBHT-QSA chain (Steps 5.1.20-26) as in the NDQIO algorithm. The only difference lies in the employment of the $U_{g'}$ operator in the respective quantum Oracle gate, which provides the sub-process with a complexity reduction by a factor of $1/K$ in the execution time domain, albeit no reduction in the power consumption domain. Let us now proceed with the detailed description of the OPF-SR process.

5.3.3 Self-Repair Process

Searching for the next potential route guarantees that the exported route $x_{s,2}$ will not be dominated by the OPF generated so far, as ensured by the check performed in Step 5.1.13. Consequently, the route $x_{s,1}$ identified by the BBHT-QSA chain in conjunction with the initial reference route $x_{s,2}$ being as optimal, will not be dominated either based on Definition 2. However, the event when $x_{s,1}$ may dominate one or more routes of the OPF is not mutually excluded due to the dominance operator being non-commutative. Consequently, there may exist suboptimal routes that have been erroneously included into the OPF, owing to a BBHT-QSA failure. Hence, we may readily check whether there is any OPF route from the previous iterations, which is dominated by the identified OPF route of the current iteration, and discard it from the OPF. This check may be implemented using the U_{g^*} operator. The global registers ought to be initialized to:

$$|i\rangle_1 \leftarrow |x_{s,2}\rangle_1 \,,\; |x\rangle_2 \leftarrow |OPF_j\rangle_2 \,,\; |t\rangle_3 \leftarrow |0\rangle_3 \,. \tag{5.21}$$

Then, we only have to observe the state of the GOW register output. This process is repeated for all the routes belonging to the OPF, as it was formally stated in the loop of Step 5.1.28. This repair process ensures that the NDQIO algorithm performs at its best attainable accuracy in terms the average Pareto distance $E[P_d]$, as long as the entire true OPF has been identified. Moving on to the consideration of the computational complexity imposed by the OPF-SR process, assuming that the multiple DHA activations provide us with k OPF routes and that the total number iterations carried out by the algorithm is equal to N_{OPF}, then in the first iteration we will have to invoke the $U_{g'}$ operator k times, while in the second iteration it will be activated $(k+1)$ times and so on, yielding a total number of CFEs that is equal to:

$$L_{SR}^{P} = \frac{1}{K} \sum_{i=k}^{N_{OPF}-1} i = \frac{1}{2K}(N_{OPF}^2 - k^2 - N_{OPF} + k), \tag{5.22}$$

$$L_{SR}^{S} = \sum_{i=k}^{N_{OPF}-1} i = \frac{1}{2}(N_{OPF}^2 - k^2 - N_{OPF} + k), \tag{5.23}$$

where L_{SR}^{P} and L_{SR}^{S} correspond to the parallel and the sequential complexities, respectively.

5.4 A Detailed 5-Node Example

Having provided all the necessary discussions about the NDQIO algorithm's sub-processes, let us now provide an illustrative example for portraying the main concepts of our proposed algorithm. The exemplified WMHN structure is shown in Fig. 5.7(a), where the same 5-node WMHN structure is utilized as that of the tutorial example presented in Section 4.5. Following a similar approach to that of Section 4.5, we will solely utilize two UFs for each route-solution, namely the BER and the CL, for facilitating the graphical representation of the route-solutions. Furthermore, the solution vectors and their respective indices, which correspond to all the legitimate routes, are presented in Table 4.1, while their graphical representation is shown in Fig. 5.7(b).

Additionally, we present all the necessary steps undertaken by the NDQIO algorithm in Fig. 5.7(b), where the route-solution transitions realized by the DHA or the BBHT-QSA chain activations are represented with the aid of arrows. Distinct colors have been used for representing the different sub-processes. In particular, as noted in the top legend of Fig. 5.7(b), the first and the second DHA activation transitions are annotated with red and blue arrows, respectively, while the BW-BBHT-QSA process and the BBHT-QSA chain transitions are indicated by green dashed and straight arrows, respectively. Moreover, the routes that belong to the OPF are marked by a square marker (□), the routes that initiate either a BBHT-QSA chain or a DHA activation are marked by a triangle (△), while the route-solutions output by each DHA or BBHT-QSA chain iteration, which are used as the new reference routes in the next iteration, are marked by a circle (○). Still referring to Fig. 5.7(b), the boundaries of the space, where valid route-solutions lie, are annotated by the long- and short-dashed lines. Let us now proceed with a more detailed description of the NDQIO algorithm's operation. We note that in terms of this tutorial example, the reader is assumed to be familiar with the concepts of the BBHT-QSA process.

Initially, the NDQIO algorithm sets the OPF set containing the optimal route-solution indices to an empty set, as formally stated in Step 5.1.1. Then, the algorithm's initialization process takes place (Steps 5.1.3–6), where the global minima are determined in terms of each optimization objective. To elaborate further, in this tutorial example we try to minimize both the BER and the CL, for the sake of simplicity, and, thus, we have $K = 2$ in Step 5.1.3. Therefore, we will activate the DHA process (Step 5.1.4) of Alg. 5.2 twice, i.e. once for each objective. Firstly, a DHA process is activated for minimizing the route's BER. More particularly, according to Step 5.2.1, a random route is chosen as the initial reference one and, then, a BBHT-QSA process is activated seeking a potential route-solution, which exhibits a lower BER. Let us assume that the initial reference route chosen is the one with index $i = 3$, i.e. the first DHA has chosen the route $\{1\ 3\ 5\}$ according to Table 4.1, which is annotated with the red triangle (△) marker in Fig. 5.7(b). The arguments of the valid route-solutions of the BBHT-QSA process lie below the red horizontal long- and short-dashed line that crosses the argument of the route with index $i = 3$ in Fig. 5.7(b).

Let us assume that the BBHT-QSA process outputs the route with index $i = 6$, i.e. the route $\{1\ 2\ 4\ 5\}$ according to Table 4.1, which is marked with a red circle in Fig. 5.7(b).

Observe in Table 4.1 that the BER exhibited by the route with index $i = 6$ is lower than that exhibited by the reference route with index $i = 3$, i.e. we have $P_{e,6} < P_{e,3}$ and, thus, after the completion of the BBHT-QSA process invoked by the DHA in Step 5.2.5, the reference route will be updated to the BBHT-QSA output (Steps 5.2.7-8). Then, a new BBHT-QSA process is activated searching for a route with a lower BER that that of the reference one. Observe in Fig. 5.7(b) that the valid route-solutions lie below the red horizontal long- and short-dashed line that crosses the argument of the route with index $i = 6$. Therefore, the valid route-solution indices belong to the set $\{2, 4, 7, 8, 9\}$ and the BBHT-QSA process is capable of identifying any of them with equal probability. Let us assume that the output of the BBHT-QSA process is the route with index $i = 9$, i.e. we have $y' = 6$ in Step 5.2.5, and the reference route is updated, since we have $P_{e,9} < P_{e,6}$ according to Table 4.1. Still referring to Fig. 5.7(b), observe that the new reference route is indeed a Pareto optimal one. Nevertheless, the DHA process is unable to identify the route's property, since it is solely seeking a route with the minimum BER. Therefore, a new BBHT-QSA process is activated with the aid of the updated reference route, in which the only eligible output is the route with index $i = 7$, i.e. the route $\{1\ 3\ 2\ 5\}$. Hence, assuming that the BBHT-QSA process of Step 5.2.6 successfully identifies the latter route, the reference route is once again updated (Steps 5.27-8) and a new BBHT-QSA process is activated. Observe in Fig. 5.7(b) that the new reference route is indeed an optimal one in terms of its BER, hence, the BBHT-QSA will exhaust the maximum number of affordable $\mathcal{G}$ applications in the absence of valid solutions. Since in the design of the improved DHA of Alg. 5.2 we have set a single BBHT-QSA time-out as the termination condition (Steps 5.2.8, 12), the DHA exits and identifies the route associated with $i = 7$ as the optimal one in terms of its BER performance. Then, the NDQIO algorithm appends the DHA output to the OPF set in Step 5.1.5.

Then, a new DHA process is activated in search of the route, which is optimal in terms of CL. A new reference route is selected randomly among all the legitimate ones according to Step 5.2.1. At this stage, let us assume that the route associated with the index $i = 16$ is eventually selected, i.e. the route $\{1\ 4\ 3\ 2\ 5\}$, which is marked in Fig. 5.7(b) with the blue triangular marker. Then, a BBHT-QSA process is activated seeking a route exhibiting a lower CL than that of the reference route (Step 5.2.5). Observe in Fig. 5.7(b) that the valid route-solutions lie at the left-hand side of the vetrical blue long- and short-dashed line crossing the reference route and, in particular, the valid route-solutions have indices that belong to the set $\{7, 9, 14\}$. Assuming that the BBHT-QSA process outputs the route associated with the index $i = 14$, i.e. the route $\{1\ 3\ 4\ 2\ 5\}$, a new BBHT-QSA process is activated by updating the reference route, since the output route exhibits a lower CL than the reference one (Steps 5.2.7-8). The new reference route is the optimal one in terms of its CL performance, as portrayed in Fig. 5.7(b). Consequently, the BBHT-QSA process activated with this route being its input will exhaust the maximum number of affordable $\mathcal{G}$ applications, resulting in the input route's identification as the optimal one through

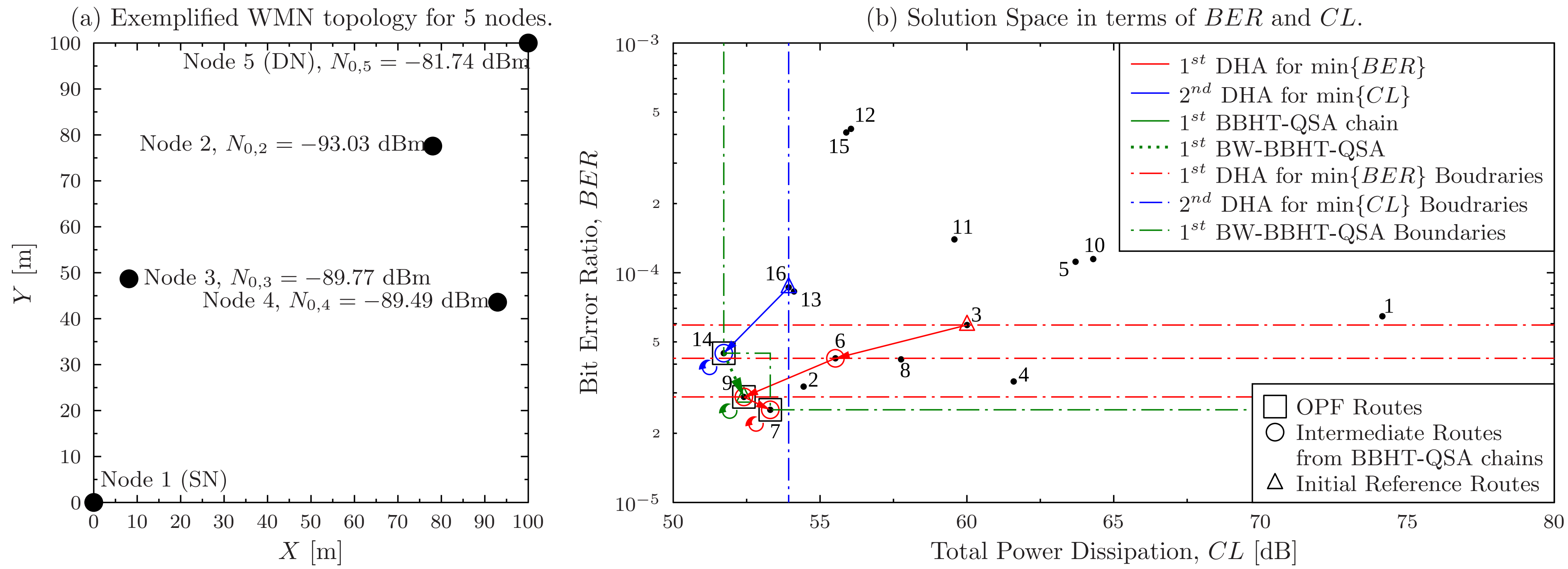


Figure 5.7: (a) Exemplified architecture for a 5-node WMHN, and (b) its optimization process using the NDQIO algorithm. In this example only two UFs are used per route-solution, for the sake of simplicity. The routes that belong to the OPF are marked by a square marker (□), the routes that initiate either a BBHT-QSA chain or a DHA activation are marked with a triangle (△), while the route-solutions output by each DHA or BBHT-QSA chain iteration, which are used as the new reference routes in the next iteration, are marked with a circle (○). Moreover, the indices of the routes as shown in Table 4.1 are marked in (b). Finally, the circural arrows in (b) denote that a BBHT-QSA has been activated with the respective route as its input, yet in the absence of potential route-solutions a random route is output by the BBHT-QSA, classifying the input route-solution as being Pareto Optimal (Step 5.1.25). The portrayed solution is not a unique one; different solutions could be derived depending on the DHA or BBHT-QSA chain's intermediate outcomes. Finally, note that the 5-node WMHN architecture portayed in (a) is identical to the architecture considered in Fig. 5.7(a), where the NDQO algorithm is invoked, for the sake of comparison.

Steps 5.2.7-12. After this operation, the DHA exits and outputs the identified optimal route, which is then incorporated into the OPF by the NDQIO algorithm (Step 5.1.5).

After the completion of the second[3] DHA, the initialization process ends and the iterative process (Steps 5.1.8-30) is activated. In the first part (Steps 5.1.10-18) of the NDQIO algorithm's iterative process, which is referred to as the *Backward BBHT-QSA Step* (BW-BBHT) in Alg. 5.1, a BBHT-QSA process is activated, which seeks a specific route-solution that is not dominated by the hitherto generated OPF and thus may potentially be a Pareto-optimal one. Explicitly, the arguments of the valid route-solutions lie in the area containing the center of the axes and bounded by the green long- and short-dashed lines, as portrayed in Fig. 5.7(b), where it is visible that the only eligible route-solution is the route associated with the index $i = 9$. We note that the BBHT-QSA process exhibits a slight probability of failing in terms of identifying a valid solution[4]. For this reason, this sub-process of the NDQIO algorithm has been designed to repeat the BBHT-QSA process in case of an unsuccessful search (Steps 5.1.15-18), which would prematurely terminate the NDQIO algorithm, hence substantially reducing the probability of an unsuccessful search. At this stage, let us assume that the BBHT-QSA process is able to identify the legitimate route-solution, which is no other than the route having the index $i = 9$, i.e. the route $\{1\ 4\ 2\ 5\}$.

Sequentially, the BBHT-QSA chain process of Steps 5.1.20-25 is activated in the same fashion as in the NDQO algorithm presented in Alg. 4.2, with the reference route being the output of the BW-BBHT-QSA sub-process, which is the route with index $i = 9$. The BBHT-QSA chain process seeks a route-solution, which dominates the reference one. Nevertheless, observe in Fig. 5.7(b) that the initial reference route of the BBHT-QSA chain process is indeed a Pareto optimal route, i.e. there exists no route that dominates it. Therefore, the BBHT-QSA chain will activate only a single BBHT-QSA process, which will exhaust the maximum number of affordable $\mathcal{G}$ applications in the absence of valid route-solutions and hence will terminate the chain, according to condition of Step 5.1.25. Subsequently, the OPF self-repair sub-process of Step 5.1.27 is invoked, where the hitherto generated OPF routes are checked as to whether they are dominated by the Pareto optimal route-solution spotted by the BBHT-QSA chain. Should any of the route-solutions be dominated by the BBHT-QSA chain's output route-solution, they would be disregarded, since they would be suboptimal. In our example, the route-solution with index $i = 9$ does not dominate any of the already generated OPF routes according to Fig. 5.7(b) and thus the OPF remains intact.

After the completion of the OPF self-repair sub-process, the BBHT-QSA output route-solution is incorporated into the OPF and the iterative process of Steps 5.1.8-30 is repeated. Moreover, it is visible from Fig. 5.7(b) that the OPF, which is comprised of the routes-solutions associated with indices $\{7, 9, 14\}$ is identical to the TOPF. Hence, we conclude that the NDQIO algorithm has identified the entire TOPF. However, the NDQIO algorithm has no knowledge of this fact at this stage. By contrast, the BW-BBHT-QSA sub-process

[3]Assuming K optimization objectives, the initialization process ends right after the completion of the K-th DHA.

[4]This occurs when this solution is present in the examined database.

of Steps 5.1.10-18 is once again activated, where both the BBHT-QSA processes invoked by the NDQIO algorithm will exhaust the maximum number of $\mathcal{G}$ applications in the absence of valid route-solutions. This double time-out process would signal to the NDQIO algorithm that the entire OPF has been identified. Finally, the NDQIO algorithm exports the OPF identified and then exits. We note that we have made no mentioning of the NDQIO algorithm complexity, which is the subject of our discussions in the next section.

5.5 Computational Accuracy versus Complexity Discussions

In this section, we will characterize our novel algorithm both in terms of its computational complexity and its accuracy. As benchmarking algorithms, we will employ the quantum-assisted NDQO algorithm, the BF method as well as the NSGA-II and the MO-ACO algorithms, which are presented in Algs. 4.2, 4.3, 2.1 and 2.3, respectively. Let us now proceed with quantifying the complexity imposed by the NDQIO algorithm.

5.5.1 NDQIO Complexity

In contrast to the NDQO algorithm's complexity characterization, which was presented in Section 4.6.1, we will characterize the complexity imposed by the NDQIO algorithm, which is presented in Alg. 5.1, both in terms of its parallel and sequential complexities. Explicitly, these metrics depend on the number of the U_g operators, which is defined in Eq. (4.1). We note that the parallel and sequential complexities imposed by the NDQO are identical due to the absence of hardware parallelism. Hence, relying on the assumption that the operators $\{U_{f_k}\}_{k=1}^{K}$ defined in Eq. (4.2) impose the same parallel and sequential complexity for all $k \in \{1, \ldots, K\}$, we may derive the upper and the lower bounds of the NDQIO algorithm by examining both the worst- and the best-case scenarios considered in Section 4.6.1 for the NDQO algorithm. As far as the BF method is concerned, its parallel and sequential complexities are identical in terms of the number CFEs, since no hardware parallelism is used. Hence, for these two extreme cases, the respective upper- and lower-bounds of the BF method are quantified in terms of the number of CFEs as follows [1]:

$$L_{BF}^{\max} = N^2 + \sum_{i=0}^{N-1} i = \frac{3}{2}N^2 - \frac{1}{2}N = O(N^2), \tag{5.24}$$

$$L_{BF}^{\min} = 2N - 1 = O(N). \tag{5.25}$$

Let us now proceed with characterizing the NDQIO algorithm. For its lower bound, we will assume a scenario, where a single Pareto-Optimal route exists, namely the direct route and all the activated BBHT-QSA processes impose the minimum possible number of CFEs. Explicitly, they impose $L_{BBHT}^{\min}$ CFEs as defined in Eq. (4.15). Let us assume that during the initialization all the DHA processes, which are activated in Step 5.1.4, have their reference route initialized to the direct one in Step 5.2.1. Then, their first invoked BBHT-QSA process formulated in Alg. 4.1 will exhaust the maximum affordable number of $\mathcal{G}$ applications in the absence of valid route-solutions. This results in terminating the respective DHA process of Alg. 5.2, hence imposing a complexity of:

$$\begin{aligned} L^{\min}_{NDQIO,init} &\equiv L^{\min}_{DHA} \\ &= L^{\min}_{BBHT} + 1 \\ &= 4.5\sqrt{N} + \log_{\lambda}\left(4.5\tfrac{\lambda-1}{m}\sqrt{N} + 1\right) + 2 \\ &= O(\sqrt{N}) \end{aligned} \tag{5.26}$$

in terms of the number of U_{f_k} activations, where the unitary operators U_{f_k} are defined in Eq. (4.2).

Therefore, if we take into consideration that the each U_{f_k} activation imposes $1/K$ CFEs in both domains and that K DHA processes are activated during the initialization process of Alg. 5.1, the complexity imposed by the NDQIO initialization process in both domains is equal to:

$$L^{P,\min}_{NDQIO,init} = L^{\min}_{NDQIO,init}, \tag{5.27}$$

$$L^{S,\min}_{NDQIO,init} = L^{\min}_{NDQIO,init}. \tag{5.28}$$

Subsequently, the iterative process of Alg. 5.1 is activated. Nevertheless, since there exist no routes that are not dominated by the direct one, the BBHT-QSA process of Alg. 4.1 that seeks a new potentially optimal route will reach its time-out twice and exit, hence imposing a complexity of:

$$\begin{aligned} L^{\min}_{NDQIO,iter} &= 2(L^{\min}_{BBHT} + 1) \\ &= 9\sqrt{N} + 2\log_{\lambda}\left(4.5\tfrac{\lambda-1}{m}\sqrt{N} + 1\right) + 4 \\ &= O(\sqrt{N}) \end{aligned} \tag{5.29}$$

in terms of $U_{g'}$ activations. Thus, the resultant parallel and sequential complexities imposed by the iterative NDQIO process become:

$$L^{P,\min}_{NDQIO,iter} = \frac{1}{K} L^{\min}_{NDQIO,iter}, \tag{5.30}$$

$$L^{S,\min}_{NDQIO,iter} = L^{\min}_{NDQIO,iter}. \tag{5.31}$$

The SR process of Steps 5.1.27-28 will not be activated, since no OPF route has been identified by the the BBHT-QSA process, which is invoked at Step 5.1.12. Therefore, the total parallel complexity imposed by the NDQIO algorithm is derived by adding up

Eqs. (5.27) as well as (5.30), resulting in:

$$\begin{aligned} L^{P,\min}_{NDQIO,tot} &= \tfrac{K+2}{K}\left[4.5\sqrt{N} + \log_{\lambda}\left(4.5\ \tfrac{\lambda-1}{m}\ \sqrt{N}+1\right)+2\right]. \\ &= O(\sqrt{N}). \end{aligned} \tag{5.32}$$

Equivalently, the total sequential complexity imposed by the NDQIO algorithm is derived by adding up Eqs. (5.28) as well as (5.31),

$$\begin{aligned} L^{S,\min}_{NDQIO,tot} &= 3\left[4.5\sqrt{N} + \log_{\lambda}\left(4.5\ \tfrac{\lambda-1}{m}\ \sqrt{N}+1\right)+2\right]. \\ &= O(\sqrt{N}). \end{aligned} \tag{5.33}$$

Observe in Eqs. (5.32) and (5.33) that the minimum execution time and power consumption imposed by the NDQIO algorithm in both domains is on the order of $O(\sqrt{N})$.

As for the upper bound, we will consider the case, where all the routes are Pareto-Optimal and the BBHT-QSA processes impose the maximum possible complexity of $L^{\max}_{BBHT}$, as quantified in Eq. (4.16). Under this assumption, each DHA process imposes the maximum possible number of U_{f_k} activations, when it activates precisely five BBHT-QSA chains, since we have:

$$4L^{QD,\max}_{BBHT}<22.5\sqrt{N}<5L^{QD,\max}_{BBHT}, \tag{5.34}$$

where $L^{QD,\max}_{BBHT} = 5.5\sqrt{N}-1$ corresponds to the maximum number of QD complexity in terms of U_{f_k} activations imposed by a single BBHT-QSA activation. Therefore, the maximum complexity $L^{\max}_{DHA}$ imposed by the DHA is equal to:

$$L^{\max}_{DHA} = 5(L_{BBHT}+1) = 50\sqrt{N} + 5\log_{\lambda}\left(\sqrt{N}\right). \tag{5.35}$$

Consequently, the parallel and the sequential complexities imposed by the initialization process are equal to:

$$L^{P,\max}_{NDQIO,init}=K\frac{1}{K}L^{\max}_{DHA}=50\sqrt{N}+5\log_{\lambda}\left(\sqrt{N}\right), \tag{5.36}$$

$$L^{S,\max}_{NDQIO,init}=K\frac{1}{K}L^{\max}_{DHA} = 50\sqrt{N}+5\log_{\lambda}\left(\sqrt{N}\right), \tag{5.37}$$

which was quantified as a function of the number of CFEs, resulting in an OPF constituted by exactly K routes. Moving on to the NDQIO iterative process (Steps 5.1.8-30), we will assume that the BBHT-QSA searching for a new potentially optimal route, fails during the initial activation but succeeds during the second one in identifying a valid route-solution. Furthermore, since all the routes are optimal, the BBHT-QSA chain will activate a single BBHT-QSA, which in turn exhausts the maximum affordable number of QD-CFEs, in the absence of valid solutions. Therefore, during each iteration precisely 3 BBHT-QSA processes will be activated. Explicitly, the complexity-dependent power consumption imposed by the BBHT-QSA that seeks a new potential route increases as the number of iterations increases, which is a consequence of increasing in the number of OPF routes used as reference routes. Hence, the maximum parallel and the maximum sequential complexities imposed by this process of Steps 5.1.10-19 are equal to:

$$\begin{aligned} L^{P,\max}_{NDQIO,BW} &= \tfrac{1}{K}\sum_{k=K}^{N}[2(L^{\max}_{BBHT}+1)] \\ &= \tfrac{N-K}{K}\Big[10\sqrt{N}+\log\lambda(\sqrt{N})\Big], \end{aligned} \tag{5.38}$$

$$\begin{aligned} L^{S,\max}_{NDQIO,BW} &= \sum_{k=K}^{N}[2k(L^{\max}_{BBHT}+1)] \\ &= \big(N^2-K^2\big)\Big[10\sqrt{N}+\log\lambda(\sqrt{N})\Big], \end{aligned} \tag{5.39}$$

as a function of the number of CFEs, respectively, while those imposed by the BBHT-QSA chains of Steps 5.1.21-25 are equal to:

$$\begin{aligned} L^{P,\max}_{NDQIO,chain} &= \tfrac{1}{K}\sum_{k=K}^{N-1}(L^{\max}_{BBHT}+1) \\ &= \tfrac{N-K-1}{K}\Big[10\sqrt{N}+\log\lambda(\sqrt{N})\Big], \end{aligned} \tag{5.40}$$

$$\begin{aligned} L^{S,\max}_{NDQIO,chain} &= \sum_{k=K}^{N-1}(L^{\max}_{BBHT}+1) \\ &= (N-K-1)\Big[10\sqrt{N}+\log\lambda(\sqrt{N})\Big], \end{aligned} \tag{5.41}$$

respectively. Subsequently, the OPF-SR process of Steps 5.1.25-39 will be activated precisely $(N-K)$ times, imposing parallel and sequential complexities formulated in Eqs. (5.22) and (5.23), respectively, upon substituting $N_{OPF}=N$ and $k=K$, hence we have:

$$L^{P,\max}_{NDQIO,SR}=\frac{1}{K}\sum_{i=K}^{N-1}i=\frac{1}{2K}(N^2-K^2-N+K), \tag{5.42}$$

$$L^{S,\max}_{NDQIO,SR}=\sum_{i=K}^{N-1}i=\frac{1}{2}(N^2-K^2-N+K), \tag{5.43}$$

respectively. Following a similar approach to the lower bound derivation, the upper bound of parallel complexity imposed by the NDQIO algorithm is derived by adding together Eqs. (5.36), (5.38), (5.40) and (5.42), resulting in:

$$\begin{aligned} L^{P,\max}_{NDQIO,tot} &= \tfrac{2(N-K)-1}{K}\Big[10\sqrt{N}+\log_{\lambda}\Big(\sqrt{N}\Big)\Big]+\tfrac{N^2-K^2-N+K}{2K}, \\ &= O(N^2). \end{aligned} \tag{5.44}$$

Equivalently, the upper bound of the sequential complexity is derived by adding up Eqs. (5.37), (5.39), (5.41) and (5.43),

$$\begin{aligned} L^{S,\max}_{NDQIO,tot} &= \left(N^2-K^2+N-K+4\right)\left[10\sqrt{N}+\log_{\lambda}\left(\sqrt{N}\right)\right]+\tfrac{N^2-K^2-N+K}{2}, \\ &= O(N^2\sqrt{N}). \end{aligned} \tag{5.45}$$

Observe in Eqs. (5.44) and (5.45) that the resultant execution time and power consumption upper bounds are on the order of $O(N^2)$ and $O(N^2\sqrt{N})$, respectively, which are higher than $O(N\sqrt{N})$ imposed by the NDQO algorithm, based on Eq. (4.14). Explicitly, this additional cost is justified by the increased elitism introduced by the NDQIO algorithm compared to the NDQO algorithm. To elaborate further, the NDQIO algorithm is capable of curtailing its operation upon detecting that there are no unidentified OPF routes. In the worst-case scenario, this imposes complexities on the orders of $O(N^2)$ and $O(N^2\sqrt{N})$ in the execution time and the power consumption domains, respectively. By contrast, in the best case-scenario, the lower bound is on the order of $O(\sqrt{N})$ in both domains. Additionally, the OPF-SR process imposes a complexity on the order of $O(N^2)$ in both domains in the worst-case scenario, while no complexity is imposed in the best-case situation. Therefore, the complexity reduction achieved by the NDQIO algorithm is inherently related to the ratio of the total number N_{OPF} of the OPF routes over the total number N of the legitimate routes considered.

The average complexities of the NDQIO, of the NDQO algorithms and of the BF method along with their respective upper and lower bounds in the parallel and the sequential complexity domains are shown in Figs. 5.8(a) and 5.8(b), respectively, for WMHNs consisting of $N_{nodes}=2$ to $N_{nodes}=9$. We note that the respective complexities of the NDQO algorithm and of the BF method will be identical in both domains, since they do not rely on any hardware parallelism techniques. As far as the upper bounds of the parallel complexity are concerned, observe in Fig. 5.8(a) that the NDQIO algorithm involves the same order of complexity as the NDQO algorithm, when considering $N_{nodes}=2$ to $N_{nodes}=7$ nodes. This is justified by the fact that for WMHNs having more than $N_{nodes}=7$ nodes the term predominantly governing the respective complexity is the N^2 term, while for less densely populated WMHNs the upper bound is governed by the term $\frac{20}{K}N\sqrt{N}$ based on Eq. (5.44). Hence, the order of the parallel complexity's upper bound of the NDQIO algorithm for $N_{nodes}<7$ nodes is reduced to $O(N\sqrt{N})$, matching the order of the NDQO algorithm's upper bound. We note that in our case study we have assumed three optimization objectives, i.e. we have $K=3$. Moreover, for WMHNs consisting of more than $N_{nodes}=7$ nodes, we observe in Fig. 5.8(a) that the NDQIO upper bound is lower than those of both the naive-BF and of the BF methods due to the complexity reduction by a factor of $1/K$ offered by the hardware parallelism owing to the employment of the $U_{g'}$ operator. On the other hand, the NDQIO algorithm's upper bound of the sequential complexity is definitely governed by the term $10N^2\sqrt{N}$. Consequently, observe in Fig. 5.8(b) that it involves a several orders of magnitude higher sequential complexity than that of the benchmarking algorithms used for WMHNs having more than $N_{nodes}=3$ nodes. We note that for WMHNs having either $N_{nodes}=3$ or $N_{nodes}=2$ nodes, the NDQIO upper bound

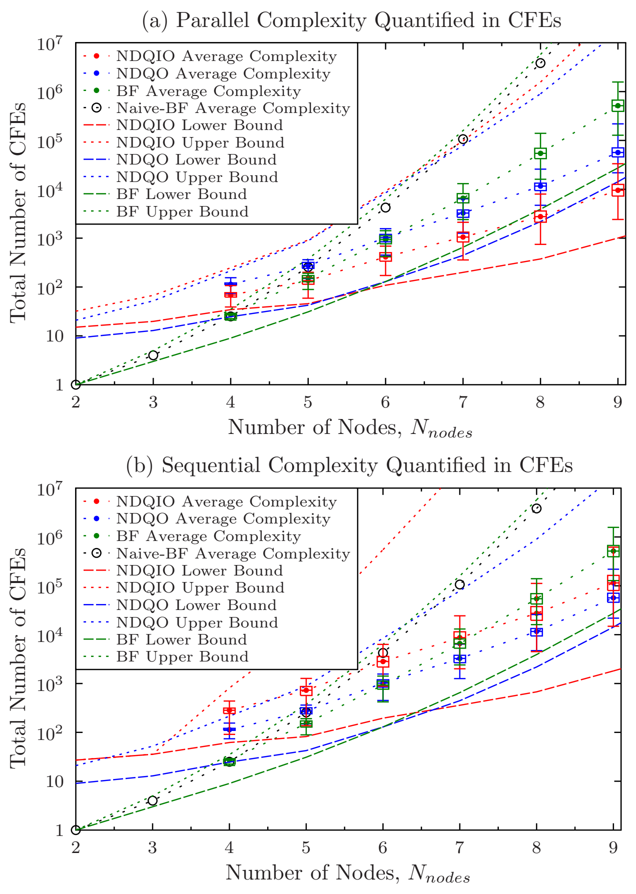


Figure 5.8: Evolution of (a) parallel and (b) sequential complexities of the NDQIO algorithm compared to the respective values imposed by the BF method of Alg. 4.3 and the NDQO algorithm of Alg. 4.2. They are also compared to the naive-BF complexity as well as to the upper and the lower bounds of the aforementioned algorithms. Explicitly, BF method's upper and lower bounds are defined in Eqs. (5.24), (5.25), respectively, while those of the NDQIO algorithm are defined in Eqs. (4.10) and (4.14). For the NDQIO algorithm the respective bounds in terms of its parallel complexity are given in Eqs. (5.32) and (5.44), while those in terms of its sequential complexity are defined in Eqs. (5.33) and (5.45). Additionally, the average complexities are presented using box plots; the upper and lower bounds of the boxes correspond to the 75% and 25% quartiles, respectively. In addition, the maximum and minimum observed complexity values are presented using horizontal lines. The mean complexity results have been averaged over 10^8 runs. Note that the all the aforementioned algorithms have been deployed for identifying the OPF of the weak Pareto optimality problem of Eq. (2.7) associated with the UV of Eq. (2.5) relying on the assumptions of Table 2.1.

matches its lower bound, since in these cases the total number of routes is less than that of the number of objectives, which makes the iterative process unnecessary.

As for the NDQIO algorithm's parallel complexity lower bound, observe in Fig. 5.8(a) that the NDQIO algorithm provides some complexity reduction compared to the benchmarking algorithms for WMHNs having $N_{nodes} = 6$. Explicitly, based on Eqs. (4.10) and (5.32), the NDQIO algorithm begins to outperform the NDQO, when the total number of routes is higher than $N_{routes} = 22$ routes, yielding that the this reduction becomes visible for WMHNs having $N_{nodes} = 6$, where the total number of legitimate route is equal to $N = 65$. In the sequential complexity domain, some complexity reduction is achieved for $N_{routes} > 134$ routes, corresponding to 7-node WMHNs, as shown in Fig. 5.8(b). Additionally, the NDQIO lower bound indicates a complexity reduction of several orders of magnitude, as demonstrated in Figs. 5.8(a,b).

Moving on to the average complexity in terms of the parallel complexity, observe in Fig. 5.8(a) that the NDQIO algorithm outperforms the NDQO for the WMHN sizes considered. In particular, for WMHNs having $N_{nodes} = 4$ and $N_{nodes} = 5$ the sequential complexity imposed by the NDQO algorithm is almost twice as high as that of the NDQIO, since the latter benefits from the hardware parallelism design. However, specifically for 4-node WMHNs it imposes a higher complexity than that of the BF method, since the NDQIO does not benefit from the complexity reduction offered by the QP for small search-spaces as we elaborated on Section 4.6.1, solely relying on the complexity reduction offered by the parallel oracle design. As soon as the complexity reduction of the QP becomes significant, which occurs for WMHNs associated with $N_{nodes} \geq 6$ nodes, the average complexity imposed by the NDQIO algorithm becomes several orders of magnitude lower than that of the NDQO algorithm and that of the BF method, as portrayed in Fig. 5.8(a). We note that this complexity reduction becomes more significant as the number of nodes increases, since the complexity reduction offered by the quantum algorithms is improved as the search-space is increased [80, 87]. As for the NDQIO algorithm's average sequential complexity, observe in Fig. 5.8(b) that it outperforms the BF method for WMHNs having $N_{nodes} = 8$ or more, while in the special case, where we have $N_{nodes} = 7$ the two algorithms impose a sequential complexity of the same order. Compared to the NDQO algorithm, the NDQIO imposes about 2.5 times the respective complexity of the NDQO algorithm for WMHNs having four to seven nodes, while for more nodes the complexity imposed decays to about twice that of the NDQO algorithm.

Therefore, we conclude that both the NDQO and the NDQIO exhibit about the same order of sequential complexity, while at the same time the NDQIO algorithm offers a substantial parallel complexity reduction of several orders of magnitude. This observation unveils a trade-off. Explicitly, based on Figs. 5.8(a,b) the NDQIO algorithm offers about ten times lower parallel complexity at the expense of doubling the number of $U_{g'}$ activations compared to the NDQO, for WMHNs having eight or more nodes. Explicitly, this additional 100% sequential complexity overhead stems from the escalating number of OPF routes included in the BW-BBHT-QSA process of Step 5.1.12. Hence, every BW-BBHT-QSA iteration requires a higher number of $U_{g'}$, due to the inclusion of more reference routes, albeit this is achieved without increasing the respective parallel complexity, owing to the parallel activation of the U_g' operators. However, we expect this 100% in sequential complexity overhead to gradually diminish as the ratio of the number of OPF routes over the total number of routes decreases due to the WMHN becoming more densely populated by nodes. This trend can be inferred from Figs. 5.8(a,b), by observing that both the average execution time and the average power consumption exhibit a larger distance from their respective upper bounds, as the number of nodes in the WMHN increases.

Furthermore, we may observe in Fig. 5.8(b) that the average sequential complexities of the NDQIO and NDQO algorithms become similar, as the number of nodes associated with the WMHN increases. Therefore, a critical question arises, whether the NDQIO algorithm would asymptotically approach the average sequential complexity of the NDQO algorithm or whether it would outperform it. For this reason, let us conduct a further case study, where we will determine both the parallel and the sequential complexity of the algorithms in terms of the number N_{OPF} of the optimal routes belonging to the OPF.

As far as the NDQO algorithm is concerned, its BBHT-QSA chains impose a complexity identical to that of the DHA, since the BBHT-QSA chain constitutes an extension of the DHA for multi-objective problems. Since precisely N_{OPF} BBHT-QSA chain processes will take place, the resultant complexity imposed by this process is equal to:

$$L_{NDQO,chain} = N_{OPF} L_{DHA}. \tag{5.46}$$

For the serial parsing step, let us stipulate the further assumption that the routes initiating a BBHT-QSA chain are distributed uniformly within the route-database and that - on average - the routes require a dominance comparison with half the routes of the hitherto generated OPF. Consequently, a BBHT-QSA chain is invoked for every N/N_{OPF} routes and the resultant complexity becomes:

$$L_{NDQO,sp} = \frac{N}{N_{OPF}} \sum_{i=1}^{N_{OPF}} \frac{i}{2} = \frac{N(N_{OPF}+1)}{4}. \tag{5.47}$$

Hence, the overall complexity imposed by the NDQO algorithm is derived by adding up Eqs. (5.46) and (5.47) yielding:

$$L_{NDQO} = N_{OPF}\left(L_{DHA} + \frac{N}{4}\right) + \frac{N}{4}. \tag{5.48}$$

Since the DHA and inherently the BBHT-QSA chain impose a complexity on the order of $O(\sqrt{N})$, the NDQO complexity will be on the order of $O(N_{OPF}N)$, based on Eq. (5.48). As for the algorithm's normalized execution time and power consumption, they will be identical to the complexity imposed, since the NDQO algorithm does not involve hardware paralellism.

Let us now proceed by characterizing the average parallel and sequential complexities imposed by the NDQIO algorithm. The NDQIO algorithm invokes the DHA K times, per objective, plus $(N_{OPF} - K)$ times a BBHT-QSA chain for the rest of the Pareto-optimal routes, yielding a parallel complexity and sequential complexity of:

$$L^{P}_{NDQIO,chain} = \frac{N_{OPF}}{K} L_{DHA}, \tag{5.49}$$

$$L^{S}_{NDQIO,chain} = (N_{OPF} + 1 - K)\, L_{DHA}, \tag{5.50}$$

respectively. We note that the terms $L^{P}_{NDQIO,chain}$ and $L^{S}_{NDQIO,chain}$ correspond to the parallel complexity and to the sequential complexity, respectively, which are imposed by both the BBHT-QSA chain and the initialization process. Additionally, the parallel and sequential complexities imposed by these processes is on the order of $O(N_{OPF}\sqrt{N})$. Subsequently, let us consider the worst-case scenario for the backward-oriented BBHT-QSA process, where two BBHT-QSA search processes are activated, yielding an unsuccessful output from the first, whilst the second succeeds in identifying a potentially optimal route. This process is activated $(N_{OPF} - K + 1)$ times, resulting in a parallel complexity and sequential complexity equal to:

$$\begin{aligned} L^{P}_{NDQIO,BW} &= \tfrac{1}{K}\sum_{i=K}^{N_{OPF}} \left(2L_{BBHT}\right), \\ &= \tfrac{2(N_{OPF}-K+1)}{K} L_{BBHT}, \end{aligned} \tag{5.51}$$

$$\begin{aligned} L^{S}_{NDQIO,BW} &= \sum_{i=K}^{N_{OPF}} \left(2iL_{BBHT}\right), \\ &= \left(N^2_{OPF} - K^2 + K + N_{OPF}\right) L_{BBHT}, \end{aligned} \tag{5.52}$$

respectively. Therefore, since the BBHT-QSA complexity is on the order of $O(\sqrt{N})$ in terms of oracle queries, parallel and sequential complexities of the backward BBHT-QSA process is on the order of $O(N_{OPF}\sqrt{N})$ and $O(N^2_{OPF}\sqrt{N})$, respectively. As for the SR process, the execution time and power consumption imposed have been derived in Eqs. (5.22) and (5.23), respectively. Hence, the overall parallel complexity imposed by the NDQIO algorithm is derived by adding together Eqs. (5.22), (5.49) and (5.51), resulting in:

$$\begin{aligned} L^{P}_{NDQIO} = &\tfrac{1}{2K}N^2_{OPF} + \tfrac{1}{K}\left(L_{DHA} + 2L_{BBHT} - \tfrac{1}{2}\right)N_{OPF} + \\ &+(1-K)\left[\tfrac{2}{K}L_{BBHT} + \tfrac{1}{2}\right]. \end{aligned} \tag{5.53}$$

Similarly, the overall sequential complexity is derived as a result of the addition of Eqs. (5.23), (5.50) and (5.52), yielding:

$$\begin{aligned} L^{S}_{NDQIO} = &\left(\tfrac{1}{2} + L_{BBHT}\right)N^2_{OPF} + \left(L_{DHA} + L_{BBHT} - \tfrac{1}{2}\right)N_{OPF} + \\ &+(1-K)\left(L_{DHA} + L_{BBHT} + \tfrac{1}{2}K\right). \end{aligned} \tag{5.54}$$

Observe in Eqs. (5.53) and (5.54) that the amount of the overall parallel complexity and that of the sequential complexity imposed by the NDQIO algorithm are on the order of $O(N_{OPF}\sqrt{N})$ and of $O(N^2_{OPF}\sqrt{N})$, respectively, as opposed to those imposed by the NDQO, which are both on the order of $O(N_{OPF}N)$. Hence, a further investigation on the order of the number N_{OPF} of optimal routes in terms of the total number N of legitimate should be carried out. For this reason, let us assume that all the QD processes impose the maximum possible complexity, i.e. we set $L_{DHA} = L^{\max}_{DHA}$ and $L_{BBHT} = L^{\max}_{BBHT}$, as defined in Eqs. (5.35) as well as (4.16) and investigate the number of Pareto-optimal routes for which the NDQIO algorithm succeds in outperforming the NDQO one, where we have:

$$L_{NDQO} > L^{P}_{NDQIO}, \tag{5.55}$$

$$L_{NDQO} > L^{S}_{NDQIO}, \tag{5.56}$$

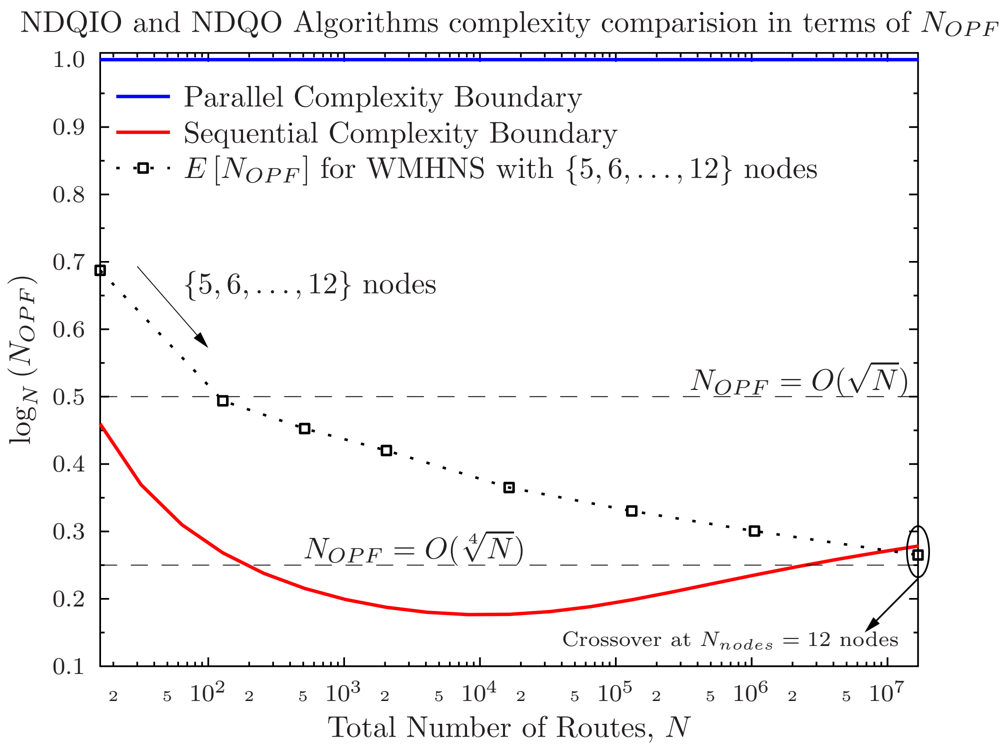


Figure 5.9: Normalized execution time and power consumption boundaries, where the NDQIO algorithm outperforms the NDQO one. The average number N_{OPF} of optimal routes based on our simulation setup is portrayed with the black squares WMHNs consisting of 5 until 12 nodes. The results have been averaged over 10^8 runs. Note that the curve of the average number $E[N_{OPF}]$ of OPF routes corresponds exclusively to the weak Pareto optimality problem of Eq. (2.7) associated with the UV of Eq. (2.5) relying on the system model of Table 2.1.

for the parallel complexity and the sequential complexity, respectively. The solution of Eq. (5.55) for the parallel complexity is portrayed in Fig. 5.9 using a blue line. Observe that the boundary is constant and equal to unity, demonstrating that the condition of Eq. (5.55) is satisfied for every N_{OPF} in the range of $1 \leq N_{OPF} \leq N$. Hence, the NDQIO algorithm outperforms the NDQO in terms of their parallel complexity regardless of the number N of the WMHN nodes and of the number N_{OPF} of Pareto-optimal routes, which is verified in Fig. 5.8(a). On the other hand, the corresponding sequential complexity boundary is shown in Fig. 5.9 by the red line. Hence, the latter is then compared to the average number $E\left[N_{OPF}\right]$ of Pareto-optimal routes, which were exported from our WMHN Pareto-optimality routing problem of Eq. (2.7) - associated with the UV of Eq. (2.5) relying on the system model of Table 2.1 - for WMHNs having between five and twelve nodes, in order to ascertain whether they are lower than the respective bound. In fact, observe in Fig. 5.9 that the average number of optimal routes lies above the sequential complexity bound for WMHNs having up to 11 nodes. However, there is a crossover in Fig. 5.9 between the sequential complexity bound and the average number of OPF routes for the 12 nodes, indicating that the NDQIO algorithm will eventually outperform the NDQO in terms of the sequential complexity as well for WMHNs having 12 nodes or more.

5.5.2 NDQIO Computational Accuracy Performance

Having characterized the NDQIO algorithm in terms of its complexity imposed and its power consumption, let us now examine the algorithm's performance in terms of its Average Pareto Distance $E[P_d]$ and the Average Pareto Completion Ratio $E[C]$, as they were defined in Eqs. (4.17) and (4.20), respectively. The latter metric corresponds to the specific portion of the TOPF identified by the respective optimization method.

Table 5.1: Number of individuals per generation and number of generations of the NSGA-II and the MO-ACO algorithm, based on Table 2.2.

Number of Nodes	Number of generations and of individuals per generation	
	Parallel Complexity Matching	Sequential Complexity Matching
6-node WMHNs	12	19
7-node WMHNs	19	29
8-node WMHNs	30	49

Let us now describe the evaluation process used for the NDQIO algorithm. When using a similar approach to that involved for the NDQO algorithm as described in Section 4.6.2, the iterative process does not necessarily impose the same number of CFEs, due to the stochastic nature of the BBHT-QSA [80]. Hence the evaluation process will be invoked each time a route is appended to the OPF. This event occurs right after the initialization process and after the completion of the iterative process, i.e. right after Steps 5.1.6 and 5.1.33 , respectively. However, since the total number of CFEs required by both of the DHAs of the initialization process, the BBHT-QSAs of the iterative process are rather random processes, the evaluation process will be activated at different complexity values. We will assume that between these evaluation processes the metrics remain constant, which results in a sum of step functions for each simulation. We can then extract a continuous distribution for these metrics versus the number of CFEs by performing an averaging operation in each respective domain. Additionally, for the evaluation of the accuracy in terms of the parallel complexity we have matched the complexity both of the NSGA-II and of the MO-ACO algorithm to that of NDQO algorithm' maximum complexity observed from the simulations. By contrast, in the sequential domain case-study we we have matched the complexity both of the NSGA-II and of the MO-ACO algorithm to that of NDQIO algorithm' maximum sequential complexity observed from the simulations. These specific parameters of the NSGA-II and the MO-ACO algorithm are shown in Table 5.1.

These metrics are shown in Figs. 5.10, 5.11 and 5.12 for 6-node, 7-node and 8-node WMHNs, respectively. As far as the average Pareto distance $E[P_d]$ is concerned, observe in Figs. 5.10(a,b), 5.11(a,b) and 5.12(a,b) that the NDQO algorithm performs optimally for 282, 502 and 929 CFEs for 6-node, 7-node and 8-node WMHNs, respectively, in both parallel and sequential complexity terms. It also exhibits an almost constant average Pareto distance $E[P_d]$, which is on the order of 10^{-8}. By contrast, the NDQIO algorithm exhibits an initial average Pareto distance $E[P_d]$, which is on the order of 10^{-4} and is then reduced

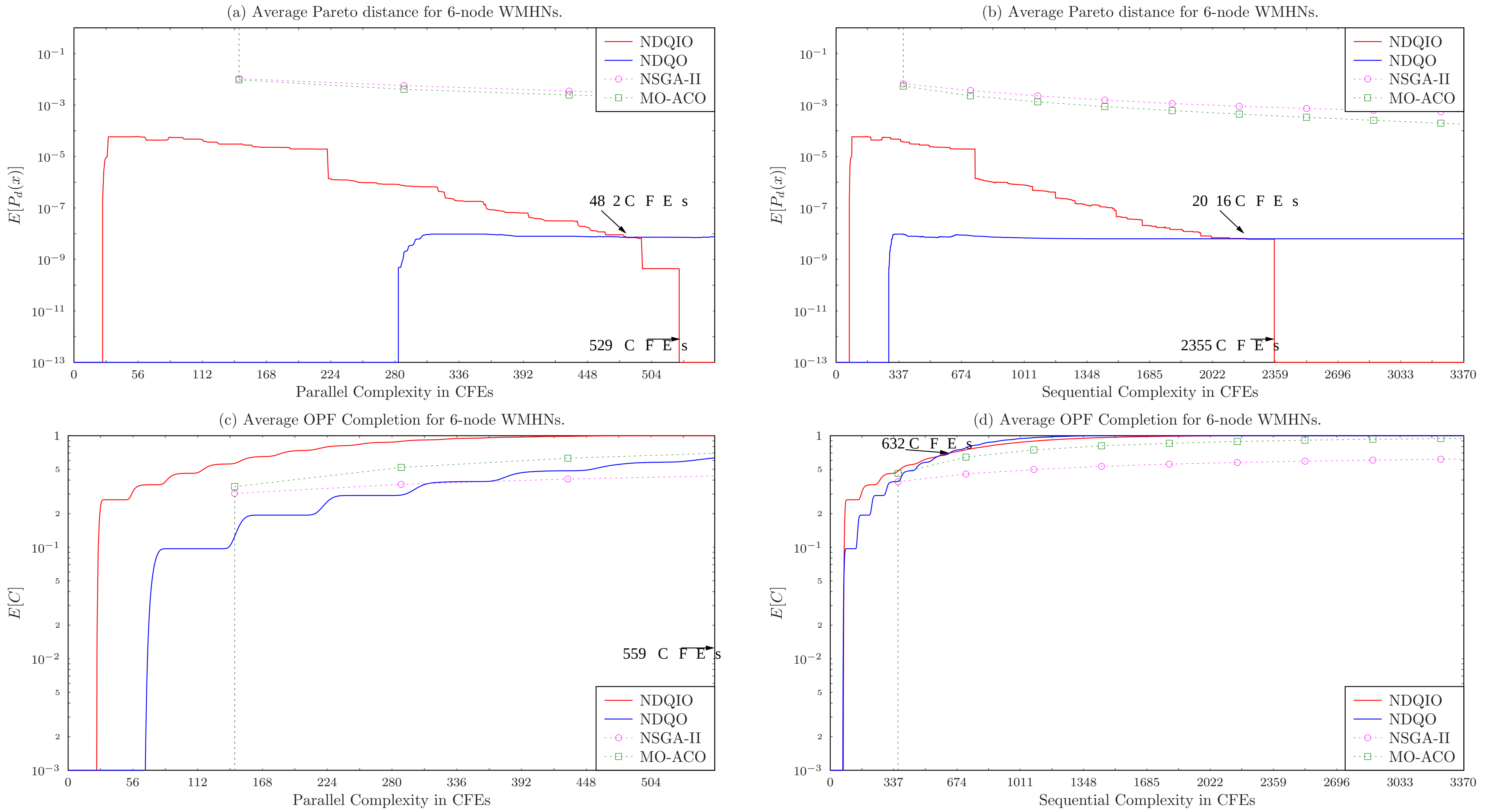


Figure 5.10: Perfomance comparison between the NDQIO and the NDQO algorithms and for 6-node WMHNs in terms of the Average Pareto Distance $E[P_d]$ (a,b) and Optimal Pareto Front Completion Ratio $E[C]$ (c,d) in vesus the parallel complexity (a,c) and the sequential complexity (b,d). The results have been averaged over 10^8 runs and they correspond to the weak Pareto optimality problem of Eq. (2.7) associated with the UV of Eq. (2.5) relying on the system model of Table 2.1. The parameters of the NSGA-II and the MO-ACO algorithm are presented in Tables 5.1 and 2.2.

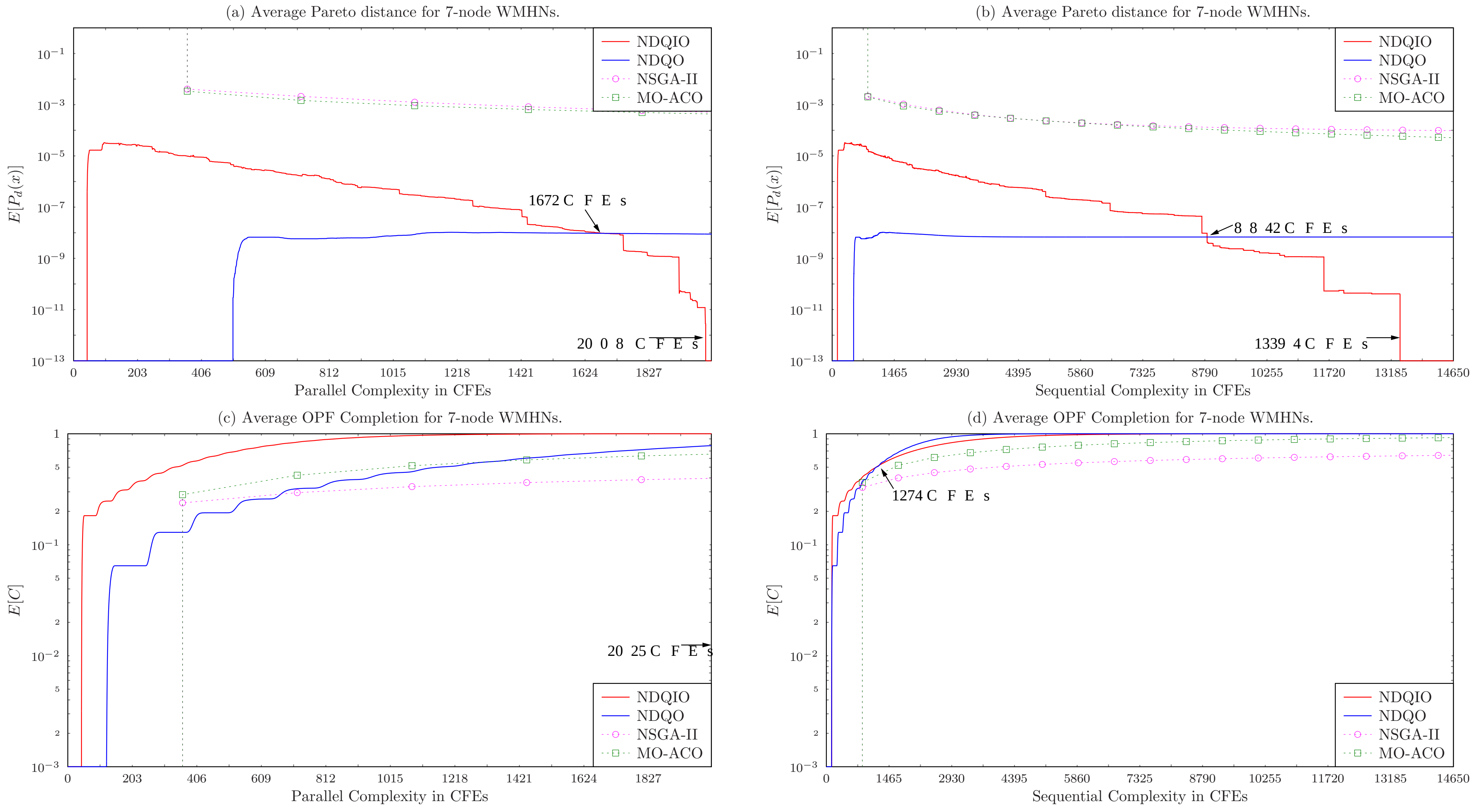


Figure 5.11: Perfomance comparison between the NDQIO and the NDQO algorithms and for 7-node WMHNs in terms of the Average Pareto Distance $E[P_d]$ (a,b) and Optimal Pareto Front Completion Ratio $E[C]$ (c,d) in vesus the parallel complexity (a,c) and the sequential complexity (b,d). The results have been averaged over 10^8 runs and they correspond to the weak Pareto optimality problem of Eq. (2.7) associated with the UV of Eq. (2.5) relying on the system model of Table 2.1. The parameters of the NSGA-II and the MO-ACO algorithm are presented in Tables 5.1 and 2.2.

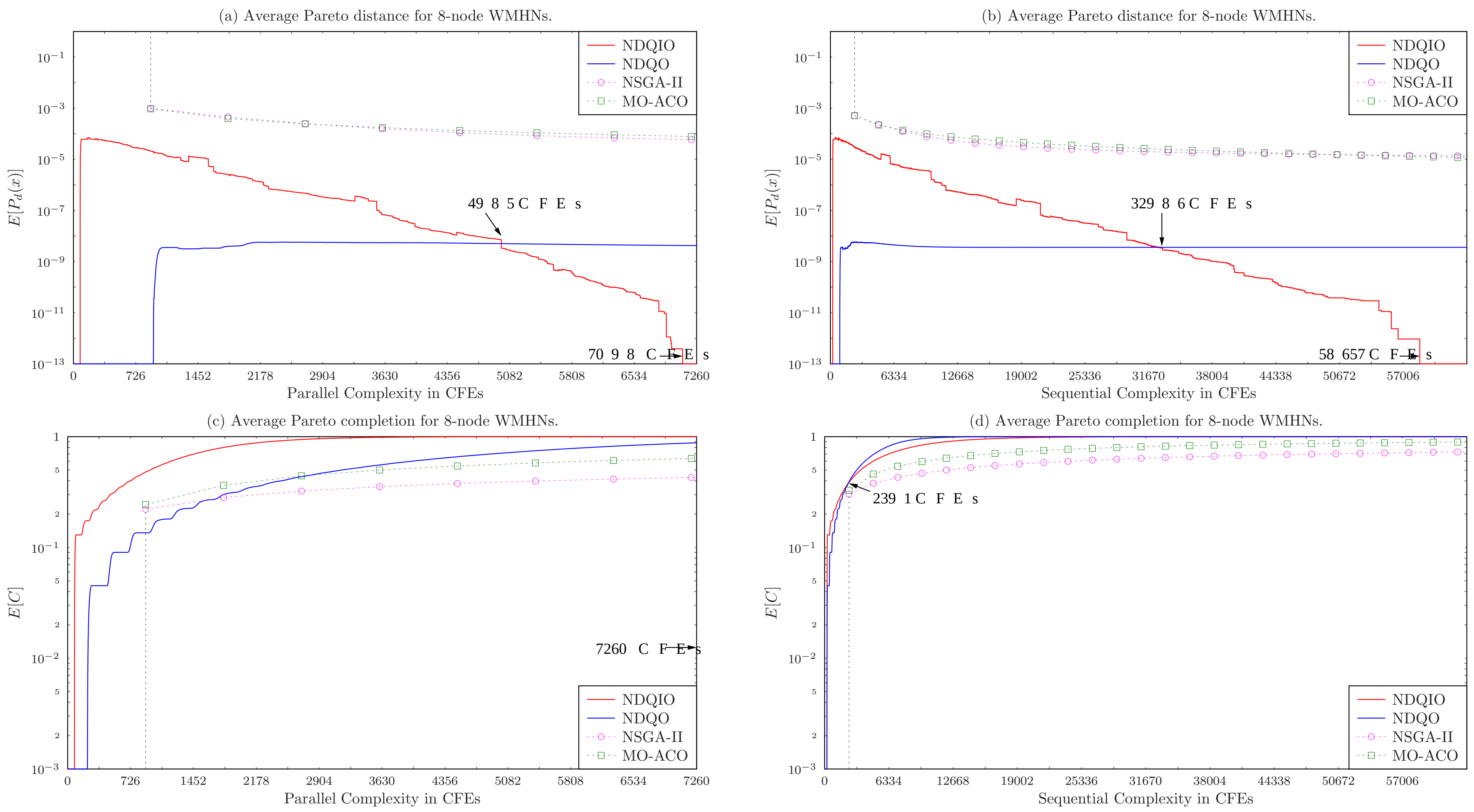


Figure 5.12: Perfomance comparison between the NDQIO and the NDQO algorithms and for 8-node WMHNs in terms of the Average Pareto Distance $E[P_d]$ (a,b) and Optimal Pareto Front Completion Ratio $E[C]$ (c,d) in vesus the parallel complexity (a,c) and the sequential complexity (b,d). The results have been averaged over 10^8 runs and they correspond to the weak Pareto optimality problem of Eq. (2.7) associated with the UV of Eq. (2.5) relying on the system model of Table 2.1. The parameters of the NSGA-II and the MO-ACO algorithm are presented in Tables 5.1 and 2.2.

as the number of the iterative NDQIO steps increases. To elaborate further, as the NDQIO algorithm invests in more CFEs, i.e. more iterative steps, the number of identified OPF routes increases and as the self-repair process is invoked at the end of each iterative step, the probability of identifying an optimal route that dominates a suboptimal one erroneously included in the OPF increases. Consequently, the average Pareto distance $E[P_d]$ drops as the number of CFEs increases, as portrayed in Fig. 5.11(a,b). More specifically for 7-node node WMHNs, observe in Fig. 5.11(a) that the NDQIO algorithm outperforms the NDQO algorithm after about 1672 CFEs in the parallel complexity domain and then after a total of 2008 CFEs the $E[P_d]$ becomes equal to zero, providing our algorithm with optimal performance in terms of this metric. The same holds for the sequential complexity, where the NDQIO algorithm outperforms the NDQO one after about 8753 CFEs and then after a total of 13394 CFEs the $E[P_d]$ becomes equal to zero, as portrayed in Fig. 5.11(b). For 6-node and 8 WMHNs, observe in Figs. 5.10(a,b) that the NDQIO algorithm begins to outperform the NDQO algorithm after 482 and 2016 CFEs in the parallel and sequential complexity domains, respectively, while it exhibits an infinitesimally low Pareto distance after 482 and 2016 CFEs, respectively. A similar trend is observed in 8-node WMHNs as well, where the NDQIO algorithm begins to outperform the NDQO algorithm after 4985 and 32 986 CFEs in the parallel and sequential complexity domains, respectively, while it exhibits an infinitesimally low Parato distance after 7098 and 58 657 CFEs, respectively, as portrayed in Figs. 5.12(a,b). As far as the NSGA-II and the MO-ACO algorithm are concerned, observe in Figs. 5.10(b), 5.11(b) and 5.12(b) that their associated Pareto distance decreases by an order of magnitude, when their complexity is matched to the maximum sequential complexity observed by the NDQIO algorithm, when compared to Figs. 5.10(a), 5.11(a) and 5.12(a), where their complexity is matched to the maximum observed by the NDQO algorithm. Despite this substantial increase, both the NDQIO and the NDQO algorithms exhibit a Pareto distance that is lower by a factor of several orders of magnitude. Hence, both the the NDQIO and the NDQO algorithms provide a better Pareto distance versus complexity trade-off, when compared to their evolutionary benchmarkers.

Moving on to the NDQIO algorithm's performance appraised in terms of the average Pareto Completion Ratio $E[C]$, this is portrayed in Figs. 5.10(c,d), 5.11(c,d) and 5.12(c,d) for 6-node, 7-node and 8-node WMHNs, respectively. As far as the parallel complexity is concerned, observe in Figs. 5.10(c), 5.11(c) and 5.12(c) that the NDQIO algorithm's completion probability converges to unity after 559, 2025 and 7260 CFEs, as opposed the 1583, 5575 and 25852 CFEs imposed by the NDQO algorithm, for 6-node, 7-node and 8-node WMHNs, respectively. This yields a further parallel complexity reduction of about 64.68%, 63.68% and 71.91%, respectively. By contrast, in the sequential complexity domain, the NDQIO algorithm achieves a completion probability of unity after 3372, 14651 and 63338 CFEs, respectively. This imposes an additional sequential complexity of 113.01%, 125.71% and 145%, respectively, compared to the NDQO algorithm. Moreover, observe in Figs. 5.10(d), 5.11(d) and 5.12(d) that the NDQIO requires fewer CFEs to produce the first OPF routes in terms of the minimum sequential complexity than the NDQO algorithm. This is a benefit of using the DHA in the initialization process, which is capable of identifying as many OPF routes as the number of the optimization objectives, hence imposing an overall parallel and sequential complexity equal to a single BBHT-QSA chain in the NDQO, which is capable of identifying a single OPF route. However, as the number of OPF routes identified increases, the sequential of the BBHT-QSA seeking new potential OPF routes increases. Explicitly, the NDQO algorithm becomes more efficient after about 632, 1274 and 2391 CFEs for 6-node, 7-node and 8-node WMHNs, respectively, as shown in Figs. 5.10(d), 5.11(d) and 5.12(d). By contrast, both the NSGA-II and the MO-ACO fail to converge to unity, despite the additional complexity invested due to their matching to the maximum sequential complexity observed by the NDQIO, as seen in Figs. 5.10(d), 5.11(d) and 5.12(d). Naturally, due to their evolutionary probabilistic structure they are unable to ensure the identification of the entire set of Pareto-optimal, hence yielding that they are less accurate for the complexity budgets considered than the NDQIO and the NDQO algorithms.

5.6 Chapter Summary

In this chapter, we have proposed a novel hardware parallelization framework for quantum processes, which offers some further complexity reduction in addition to that provided by QP. Based on this framework, we have developed a novel algorithm as an improvement to the existing NDQO algorithm. Due to the hardware parallelization, we have distinguished the complexity imposed by the novel algorithm in two distinct domains, namely the parallel complexity and the sequential complexity domains, and we have then analytically derived the upper and lower bound of complexity in both domains, which is on the order of $O(\sqrt{N})$ in both domains for the best-case scenario and on the order of $O(N\sqrt{N})$ and $O(N^2\sqrt{N})$ for the worst-case scenario in the execution time and power consumption domains, respectively.

Additionally, we have analytically characterized the average complexity as a function of the number N_{OPF} of Pareto-optimal routes, demonstrating that for the weak Pareto optimality of Eq. (2.7), which is associated with the UV of Eq. (2.5) and relies on the system model of Table 2.1, the NDQIO algorithm is capable of identifying the entire OPF, whilst providing a substantial parallel complexity reduction, when compared to the NDQO algorithm of Chapter 4. As for the NDQIO algorithm's sequential complexity, we have demonstrated that it will provide a sequential complexity reduction for WMHNs having more than 11 nodes as well, when compared to the NDQO algorithm. More specifically, for the WMHNs considered, namely those consisting of 5 to 9 nodes, the NDQIO algorithm exhibits an optimal performance, profiting from almost an order of magnitude of parallel complexity reduction compared to the NDQO algorithm, while imposing the same order of sequential complexity.

Chapter 6

Multi-Objective Routing and Load Balancing for Social Networks

6.1 Introduction

Based on a recent report conducted by *Shareholic*[1], 31.24% of the overall network traffic in 2014 has been generated by *Online Social Networks* (OSN), such as Facebook and Pinterest, while having an increase of about 10% compared to the previous year. Additionally, from a slightly different perspective, namely from that of the *Internet of Things* (IoT) [154], networked devices perpetually proliferate [155, 156] and they also tend to exhibit social behavior [157, 158]. Explicitly, the information flow among the IoT nodes follows a social pattern. For instance, a networked refrigerator would share no relationship with a networked TV. By contrast, the latter would share a social relationship with smart-phones or tablets.

Explicitly, the nodes' social behavior combined with their increased number results in a paradigm shift as to how the networks are designed and maintained [159], leading to encapsulating *Social Network Analysis* (SNA) [160] tools into the network's design. From this perspective, networks, which are often comprised by remote nodes having limited power, ought to configure their end-to-end links for satisfying diverse and often coflicting *Quality-of-Service* (QoS) criteria, such as the *Bit-Error-Ratio* (BER), the *Packet-Loss-Ratio* (PLR), total power dissipation or the overall delay. This requires the joint optimization of the aforementioned QoS criteria. As a further aspiration, Boldrini *et al.* pointed out in [161] that optimal multihop routing is more beneficial for socially aware networks than for the socially oblivious ones.

The authors of [162, 163, 164, 165, 166, 167, 168] advocated routing schemes, where both the specific QoS criteria and SNA-aided design are considered. To elaborate further, Bulut and Szymanski [165] proposed a routing scheme in the context of *Delay-Tolerant Networks* (DTN) for maximizing the associated routing efficiency by grouping the *Mobile Users* (MU)

[1]`https://blog.shareaholic.com/social-media-traffic-trends-01-2015/`

into clusters based on their contact history. Furthermore, Hui *et al.* [167] conceived a novel algorithm, namely the so-called BUBBLE algorithm, which exploits the social networking metrics of the centrality [169] routinely used in the community detection [170] for the sake of performing socially aware multi-cast routing in the context of DTNs. In a similar context, namely that of *Vehicular Social Networks* (VSN), Xia *et al.* [168] employed *Bee Colony Optimization* (BCO) in the form of the so-called BEEINFO packet forwarding scheme for the sake of maximizing the associated packet delivery ratio.

Apart from the above-mentioned routing schemes considering SNA-related metrics, some contributions utilize the structure of a twin-layer composite network [171], where the top layer characterizes the users' social relationships, hence it is often referred to as an *Online Social Network* (OSN). By contrast, the bottom one is constituted by the technological network. To elaborate further, in [172], each MU is assumed to communicate with its contacts with a probability that is inversely proportional to their respective geographic distance [173], while the technological network relies on a grid-based network. The same OSN layer has been deployed in [174] in conjunction with a mobile multi-cast network and a hybrid routing scheme has been proposed for the sake of improving the content dissemination among members of the same community. Additionally, epidemic routing [175] has been deployed [176, 177] in this specific cross-layer design, where the nodes allow their messages to "flood" their newly discovered contacts by mimicking the spread of a disease in a community. This scheme has a low complexity and a low delay, but it tends to use an excessive amount of resources, because multiple copies of the packets are allowed to flood the network.

Apart from optimal routing, the new socially aware design paradigm has to account for the nodes' *social selfishness* [178]. Explicitly, the nodes' social selfishness stems from their tendency to select specific routes for the sake of optimizing a specific utility, while being oblivious to the potential degradation of the overall network's performance inflicted by their particular choice [179, 180]. Naturally, the nodes' selfish route selection leads to the creation of bottlenecks in the network flow, especially for the case of nodes having a high centrality. This requires socially-aware load balancing [181, 182, 183].

Both the NDQO and the NDQIO algorithms presented in Chapters 4 and 5, respectively, provide us with some clear design guidelines for the sake of addressing the joint multi-objective routing and load balancing problem of socially-aware networks. Explicitly, the hybrid framework exploiting the synergy between the QP and the HP provides substantial complexity reduction by a factor of $O(K\sqrt{N})$ [2], where the factor K corresponds to the number of parallel independent quantum processes stemming from the HP, while N is the database size. Nevertheless, as Zalka [88] pointed out, Grover's QSA and inherently all the Grover-based QSAs, such as the BBHT-QSA, the DHA and the NDQIO algorithm, are optimal in terms of their complexity reduction, as long as the database entries are uncorrelated. Naturally, Zalka's proof of Grover's QSA optimality provides us with a further design consideration, namely the *database correlation exploitation*, as portrayed in Fig. 6.1. We note that the actual complexity reduction offered by the database correlation exploitation strictly depends on the optimization problem and, thus, its achievable complexity reduction

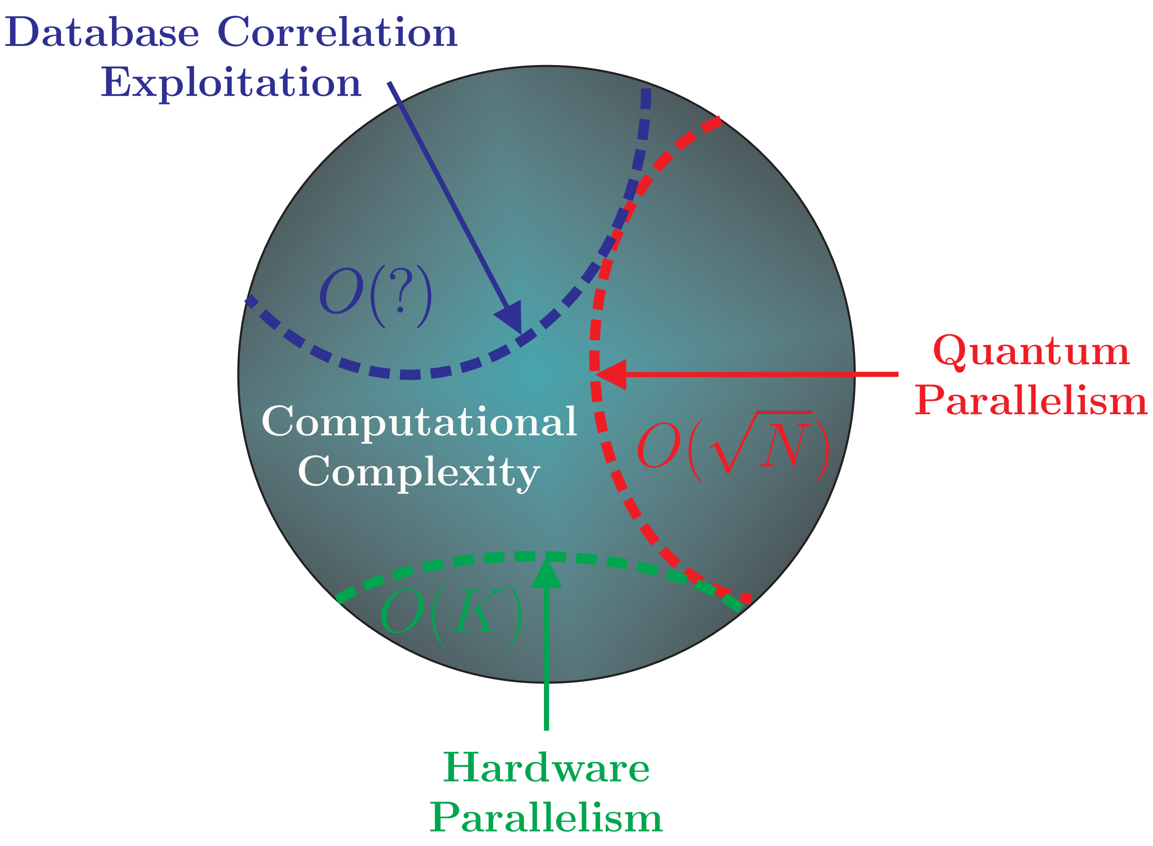


Figure 6.1: Eligible techniques of reducing the computational complexity. Explicitly, QP is capable of reducing the number of database calls from N on the order of $O(\sqrt{N})$, where N is the database length, while HP exhibits a complexity reduction on the order of $O(K)$, where K denotes the number of independent parallel processes. Finally, although the database correlation exploitation is problem-dependent and its complexity reduction cannot be readily quantified, it tends to rearrange the database into an uncorrelated one, substantially reducing the complexity imposed by both QP and HP, based on Zalka's [88] proof of Grover's QSA optimality.

order is denoted by $O(?)$ in Fig. 6.1. Nevertheless, we can view this method as a means of transforming the database into a series of shorter uncorrelated ones, thus, effectively reducing the database length N for pushing the complexity reduction offered by the hybrid HP and QP framework to its full potential.

In Chapters 4 and 5 we have developed a quantum-assisted framework for solving the Pareto-optimal routing problem in the context of WMHNs. More specifically, the system model of Eq. (2.7) associated with the UV of Eq. (2.5) and relying on the assumptions presented in Table 2.1 is oblivious of any network planning metrics, when balancing the tele-traffic load among the RNs, since it treats each of the potential pairs of SNs and DNs independently. Furthermore, optimizing the routes' integrity by using their expected BER as the optimization metric, which is recursively evaluated with the aid of Eq. (2.2), is rather impractical. Explicitly, near-capacity codes [184] are indeed capable of reducing the uncoded BER to infinitesimally low levels, as long as it does not exceed a specific threshold P_e^{th}. Therefore, the routes' integrity optimization is transformed into a connectivity problem, where a specific node is able to transmit its packet to another node, as long as this BER constraint is satisfied, hence ensuring a seamless transmission. In addition to this modification, for the sake of ensuring its practicality, we will consider a twin-layer network model, which consists of an *Online Social Network* (OSN) layer as well as a technological *Wireless Mesh Network* (WMN). Explicitly, this twin-layer network model is capable of encapsulating the nodes' social behavior, thus enabling the employment of SNA-aided joint cross-layer multiple-objective routing and load balancing optimization. We will elaborate on the specifics of the twin-layer model considered in Section 6.2. Subsequently, in Section 6.3, we will present our joint multiple-objective routing and load balancing scheme, namely the *Multi-Objective Decomposition Quantum Optimization* (MODQO) algorithm, which benefits both from the quantum-assisted framework of Chapters 4 and 5 as well as from the database correlation of the individual routes forming the Pareto-optimal route-combinations. Then, in Section 6.4 we will analytically characterize the complexity imposed by the MODQO algorithm and evaluate its performance with respect to the NSGA-II. Note that we have not employed the MO-ACO, since it is not directly applicable for the composite joint multiple-objective routing and load balancing problem presented in Section 6.2.3. Note that this chapter is based on our contribution in [3].

Let us now proceed by discussing the network model considered for our socially-aware application.

6.2 Network Specifications

We have considered a twin-layer network, which is shown in Figs. 6.2 and 6.3. To elaborate further, the network is comprised by a set of N_{MC} users, which from now on will be referred to as *Mesh Clients* (MCs) and by a set of N_{MR} wireless *Mesh Routers* (MRs). The latter form the backbone of a *Wireless Mesh Network* (WMN), which supports the communications among the MCs. The WMN constitutes the bottom layer of our network. Explicitly, the WMN layer is reminiscent of the network model considered in Chapters 4 and 5 relying on the assumptions of Table 2.1. On the other hand, the MCs are assumed to exhibit a specific social behavior and, hence, they form an OSN, which incorporates the upper layer of our network.

The locations of both the MCs and of the MRs are assumed to be random, obeying a uniform distribution within a (100×100) m^2 square block, which is the network's coverage area we considered for this scenario. We note that the chosen coverage area is just an example and a larger area of (1×1) km^2 - such as a university campus or an airport terminal - could have been readily considered; however, we have chosen this relatively small area for the sake of approaching the fully-interconnected network scenario, making the routing problem more challenging. Additionally, each of the MCs is exclusively served by its closest MR, as denoted by the gray arrows in Figs. 6.2 and 6.3.

As far as the packet dissemination process is concerned, we have assumed that the source and destination nodes belong exclusively to the set of MCs. Therefore, the MRs of the WMN layer can only act as intermediate relays forwarding the packets of the source MC to the destination MC. Furthermore, the communication between MCs is only feasible via MRs. For instance, let us consider the case, where MC_1 has to send a packet to MC_3. Despite the fact that in Fig. 6.2 MC_1 and MC_3 are pretty close to each other, their communication can only be realized through MR_1 and MR_3. Hence, the shortest route in terms of the number of hops that the packet can follow is the route $MC_1 \rightarrow MR_1 \rightarrow MR_3 \rightarrow MC_3$. We note that our twin-layer network parameters are summarized in Table 6.1.

Having defined the basic topology of the twin-layer network considered in our case study, let us now proceed with a brief description of the two layers comprising the network.

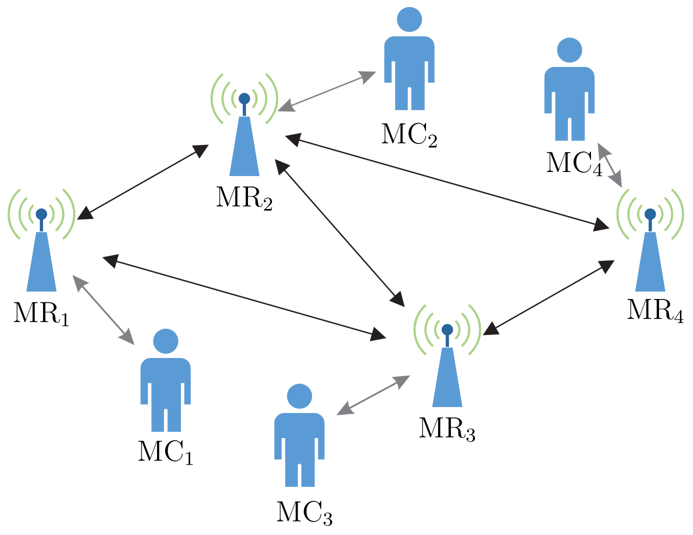


Figure 6.2: Exemplified topology with $N_{MC} = 4$ MCs and $N_{MR} = 4$ MRs for the twin-layer network considered. In the OSN layer the arrows among the MCs manifest their friendship status, whereas in the WMN layer the resective arrows correspond to links satisfying the QoS criteria. The gray-colored arrows denote the association of each MC with a specific MR.

6.2.1 OSN Layer

As we have mentioned in the previous subsection, the MCs exhibit social behavior increasing the probability of their communication with a specific set of other MCs. This set of MCs is often referred to in WSN terminology as *friends*. Hence, the MCs' friendship status can be modeled by the binary friendship matrix $\mathcal{F}_{MC}$ that defines the set of MCs being friends to a specific MC. Naturally, the friendship matrix $\mathcal{F}_{MC}$ is symmetric, with all the elements of its diagonal being equal to zero. Equivalently, since each MC is associated with a specific MR, a binary friendship matrix $\mathcal{F}_{MR}$ may be defined in the context of MR as a cross-layer metric. The elements $\mathcal{F}_{MR,ij}$ of the latter matrix indicate whether MR_i and MR_j are associated with a pair of MCs having a friendship relationship. We note that the $\mathcal{F}_{MR}$ matrix is also symmetric; however, its diagonal elements may not be strictly equal to zero, corresponding the scenario where two friendly MCs are associated with the same MR.

As for generating the $\mathcal{F}_{MC}$ matrix, we have utilized the well-studied social relationship of the *Karate Club* for the sake of practicality, which was proposed by Zachary [185] and is portrayed in Fig 6.4. To elaborate further, the MCs are randomly generated similarly

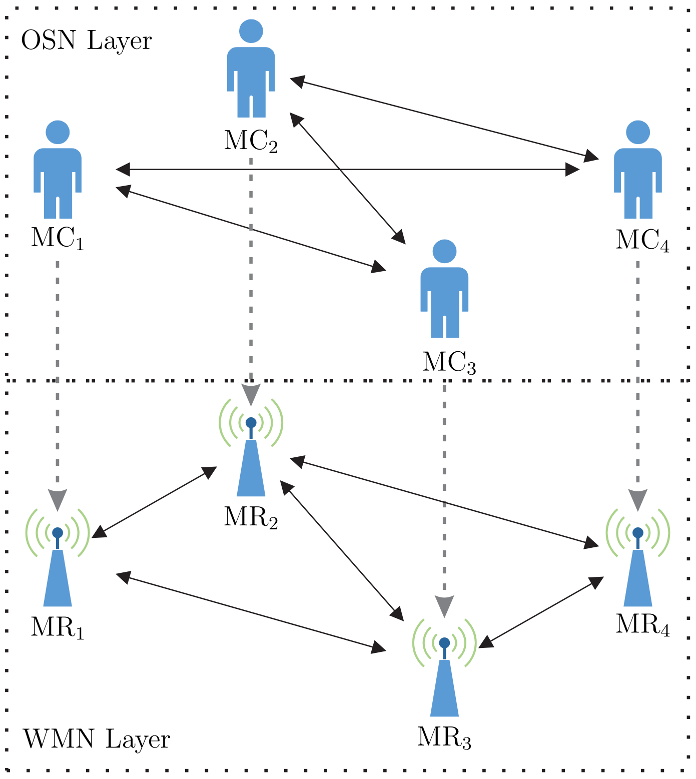


Figure 6.3: Presentation of the two layers of Fig. 6.2. In the OSN layer the arrows among the MCs manifest their friendship status, whereas in the WMN layer the resective arrows correspond to links satisfying the QoS criteria. The gray-colored arrows denote the association of each MC with a specific MR.

to the members of the *Karate Club*, while the sole constraint imposed is that of having a connected social graph for the sake of avoiding the potential isolation of certain MCs. In this way, packet dissemination emerging from a specific MC to the rest of the MCs is enabled, regardless of whether they have a friendship relationship by forwarding a packet in a friend-by-friend basis.

As mentioned in the introduction, the social behavior of the MCs provides us with the capability of employing SNA tools for analyzing the performance of socially-aware networks. In fact, the *betweenness centrality* B_{sin} metric has been proposed by Freeman in [186] for quantifying the information flow of each node MR$_k$ of the WMNs. In this context, each node has been considered to have a social friendship with the specific nodes it can reliably communicate with using a single hop. The betweenness centrality metric actually quantifies the usage of each node MR$_k$ as an intermediate relay. Explicitly, the *betweenness centrality* B_{sin} is defined as [186]:

$$B_{sin}(MR_k) = \sum_{\substack{i=1 \\ i \neq k}}^{M} \sum_{\substack{j=1 \\ j \neq i,k}}^{M} \frac{g_{MR_i,MR_j}(MR_k)}{g_{MR_i,MR_j}}, \tag{6.1}$$

where $g_{MR_i,MR_j}(MR_k)$ represents the number of times the node MR$_k$ is involved in the shortest routes - in terms of the number of hops - spanning from the node MR$_i$ to the node

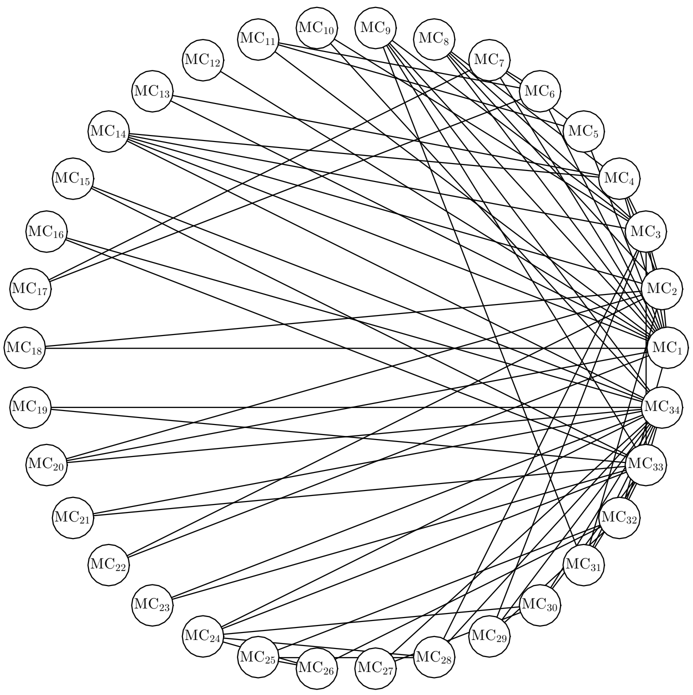


Figure 6.4: Complete karate club social relationship [185] used for determining the social relationship of the MCs in the OSN layer.

MR$_j$, while g_{MR_i,MR_j} denotes the number of the optimal routes and M corresponds to the total number of MRs. We note that the *normalized betweenness centrality* $\bar{B}_{sin}(MR_k)$, defined in Eq. (6.2), corresponds to the probability of MR$_k$ being used as a relay and it is formulated as:

$$\bar{B}_{sin}(MR_k) = B_{sin}(MR_k)/\sum_{i=1}^{M} B_{sin}(MR_i). \tag{6.2}$$

We can adapt the *betweenness centrality* to the context of twin-layer networks by defining the so-called *composite betweenness centrality.* For the latter, the friendship relationship is defined in a rather generic manner. To elaborate further, since the links between MCs that share a friendship are established on an exclusive basis, the *composite betweenness centrality* calculation is restricted to these specific routes. Therefore, the *composite betweenness centrality* $B_{com}(MR_k)$ is defined as [30]:

$$B_{com}(MR_k) = \sum_{i=1}^{N_{MC}} \sum_{\substack{j=1 \\ j \neq i}}^{N_{MC}} \frac{g_{MC_i,MC_j}(MR_k)}{g_{MC_i,MC_j}} \mathcal{F}_{MC_i,MC_j}, \tag{6.3}$$

where $\mathcal{F}_{MC_i,MC_j}$ denotes the friendship relationship between MC$_i$ as well as MC$_j$ and it is equal to the element of the i-th row and the j-th column of the $\mathcal{F}_{MC}$ matrix. Additionally, the term $g_{MC_i,MC_j}(MR_k)$ corresponds to the number of optimal routes between MC$_i$ and MC$_j$ involving MR$_k$ as an intermediate relay[2], while g_{MC_i,MC_j} is the total number of

[2]Note that if an MC is associated with an MR, the participation of this specific MR in the MC's routes is not taken into account.

optimal routes for the same source and destination pair. Equivalently to the *normalized betweenness centrality*, the *normalized composite betweenness centrality* $\bar{B}_{com}(MR_k)$ quantifies the probability of a specific MR$_k$ being used as an intermediate relay, in the context of the socially-aware network considered. The latter metric is defined as:

$$\bar{B}_{com}(MR_k) = B_{com}(MR_k)/\sum_{i=1}^{M} B_{com}(MR_i). \tag{6.4}$$

The vector $\bar{B}_{com}$ contains the probability distribution of the specific MRs being used as intermediate relays. Therefore, instead of optimizing a specific parameter, such as the sum-rate considered in [187], we can readily manipulate this distribution by selecting the appropriate routes. To guarantee fairness in terms of the forwarded tele-traffic load amongst the MRs, we will consider as the desired set of route-solutions, the specific set of routes having a normalized composite betweenness $\bar{B}_{com}$ that approaches the uniform distribution.

A direct approach of equally distributing the relayed load amongst the MRs would be to minimize the standard deviation $\sigma_{\bar{B}_{com}}$ of the normalized composite betweenness. Explicitly, minimizing $\sigma_{\bar{B}_{com}}$ yields a minimization in the discrepancy among the MRs' tele-traffic-load. Nevertheless, there exist cases, where none of the active routes utilizes intermediate MRs, which results in an all-zero normalized composite betweenness distribution, i.e. we have $\bar{B}_{com}(MR_k) = 0$, $\forall k \in \{1, \ldots, N_{MR}\}$. This kind of distribution yields a standard deviation equal to $\sigma_{\bar{B}_{com}} = 0$, which is optimal; however, these route-solutions often exhibit poor performance in terms of their power consumption or BER.

Therefore we will propose a novel metric, namely the *normalized entropy* $\bar{H}(\bar{B}_{com})$ of the normalized composite betweenness, which is defined as follows:

$$\bar{H}(\bar{B}_{com}) = \frac{H(\bar{B}_{com})}{\log_2(N_{MR})}, \tag{6.5}$$

where $H(\bar{B}_{com})$ corresponds to the Shannonian entropy, which is defined as follows [188]:

$$H(\bar{B}_{com}) = \sum_{k=1}^{N_{MR}} \bar{B}_{com}(MR_k) \log_2[\bar{B}_{com}(MR_k)]. \tag{6.6}$$

We note that the normalization factor of Eq. (6.5) is used to make the normalized entropy value independent of the number of MRs, N_{MR} and bounds its value to the range $[0, 1]$. Explicitly, the entropy of a distribution can be viewed as a metric of proximity of a specific distribution to the uniform one. This could be justified by the fact that the entropy of a specific distribution is inversely proportional to the *Kullback-Leibler divergence* $D_{KL}(\bar{B}_{com}||U)$ [189], where U denotes the uniform distribution of N_{MR} events. This is formally expressed as follows [190]:

$$H(\bar{B}_{com}) = \log_2(N_{MR}) - D_{KL}(\bar{B}_{com}||U), \tag{6.7}$$

where the Kullback-Leibler divergence $D_{KL}(\bar{B}_{com}||U)$ is defined as [190]:

$$D_{KL}(\bar{B}_{com}||U) = \sum_{k=1}^{N_{MR}} \bar{B}_{com}(MR_k) \log_2 \left[\frac{\bar{B}_{com}(MR_k)}{U(MR_k)} \right]. \tag{6.8}$$

Explicitly, it has been proven by Hobson [191] that the Kullback-Leibler divergence constitutes an appropriate metric of the difference between two different distributions. Hence, based on Eq. (6.8) the value of the divergence for the distributions $\bar{B}_{com}$ and U is bound to the range $[0, \log_2(N_{MR})]$. The upper bound of this region denotes complete divergence of the examined distributions, while its lower bound yields a perfect convergence of the two distributions. This can equivalently be translated into normalized entropy $\bar{H}(\bar{B}_{com})$ terms, where the perfect matching of the normalized composite betweenness distribution and the uniform one is achieved, when we have $\bar{H}(\bar{B}_{com}) = 1$, whereas they are uncorrelated when $\bar{H}(\bar{B}_{com}) = 0$. This metric circumvents the problem of the all-zero normalized composite betweenness distribution, since in this case its normalized entropy is equal to $\bar{H}(\bar{B}_{com}) = 0$. Therefore, efficient load balancing relies upon the maximization of the normalized entropy, leading to the optimization problem in terms of the active routes S formulated as:

$$\underset{\forall S}{\text{argmax}} \quad \bar{H}\left[\bar{B}_{com}(S)\right]. \tag{6.9}$$

Observe that the optimization problem of Eq. (6.9) is unconstrained, and hence it does not take into account any other QoS criteria, such as the network delay or power consumption. This results in the route-solutions defined by Eq. (6.9) that either exhibit excessive delay or excessive power consumption or cannot be established at all owning to a maximum transmit power violation. In fact, the aforementioned QoS criteria, which stem from the WMN layer, are presented in the next subsection. From a cross-layer optimization perspective, they will be encapsulated in Eq. (6.9) in the form of a set of constraints, for the sake of guaranteeing an optimal performance in terms of the QoS criteria considered.

6.2.2 WMN Layer

As mentioned at the beginning of this section, the WMN layer is constituted by that specific set of the MRs, which facilitate communications among the MCs by forwarding the respective packets, as portrayed in the bottom layer of Fig. 6.3. Their locations are random, which is typical for an ad hoc deployment, but then are considered to be stationary. By contrast the MCs are mobile. Additionally, a rather strong *Line-of-Sight* (LoS) component [11] is assumed to be encountered by each MR to MR link and, thus, only the link's path-loss is taken into account. The path-loss $L_{i,j}$ for a link between MR$_i$ and MR$_j$ is calculated using the classic *Path-Loss Model* of Eq. (2.3), which is formally expressed as [126]:

$$L_{i,j} \equiv \frac{P^t_{i,j}}{P^r_{i,j}} = L_0 \left(\frac{d_{i,j}}{d_0} \right)^{\alpha}, \tag{6.10}$$

where α corresponds to the *path-loss exponent.* Explicitly we set $\alpha = 3$, where $d_{i,j}$ is the Euclidean distance between MR$_i$ and MR$_j$, while L_0 denotes the reference path-loss at the reference distance $d_0 = 1$ m and $P^r_{i,j}$ and $P^t_{i,j}$ denote the transmitted and received power, respectively. The reference path-loss L_0 is quantified using the free-space path-loss formula [11] of:

$$L_0 = \left(\frac{4\pi d_0 f_c}{c}\right)^2, \tag{6.11}$$

where f_c is the carrier frequency, which is set to $f_c = 2.4$ GHz, complying with the IEEE 802.11b/g protocol [192] and c is the speed of light.

As far as the forwarding scheme is concerned, we have utilized the *Decode-and-Forward* (DF) scheme [12] in the same fashion as in the network model considered in Chapters 4 and 5 relying on the assumptions of Table 2.1, due to the scheme's capability of encapsulating the routing information into the packet header. In this context, the modulation scheme adopted was QPSK [126]. As for the transmission environment, the links among the MRs are subjected to only *Additive White Gaussian Noise* (AWGN), while the links between the MCs and their associated MRs are established for transmission over *Rayleigh Fading* channels [11]. Additionally, we have adopted an adaptive power control scheme, where each link, either between two MRs or between MCs and their associated MRs, can be successfully established as long as the link's *Bit Error Ratio* (BER) is lower than a certain threshold P^{th}_e. This constraint is imposed for the sake of guaranteeing that the packets are successfully recovered from the intermediate MRs, hence mitigating the need for retransmission. In our scenario, we have set this BER threshold to $P^{th}_e = 10^{-2}$, which corresponds to the uncoded BER of each link, because powerful state-of-the-art channel coding schemes a capable of further reducing the BER to infinitesimally low values [126]. Therefore, at each link we will attempt to match the link BER value to that of its threshold, hence minimizing the potential interference experienced by the rest of the nodes owing to excessive interferences. This yields an equivalent *Signal to Noise plus Interference Ratio* (SINR) threshold $\gamma^{th}_{i,j}$, which is equal to:

$$\gamma^{th}_{i,j} = \frac{P^r_{i,j}}{N_0 + I_{\max}} = \begin{cases} \frac{2(1-2P^{th}_e)^2}{1-(1-2P^{th}_e)^2} & i \text{ or } j \text{ are MCs,} \\ Q^{-1}\left(P^{th}_e\right) & \text{otherwise,} \end{cases} \tag{6.12}$$

where the function $Q^{-1}(\cdot)$ corresponds to the inverse of the Q-function, N_0 is the thermal noise power and $I_{\max}$ is the maximum tolerable interference power level. The thermal noise power is set to $N_0 = -114$ dBm, corresponding to a bandwidth of $W = 1$ MHz. Therefore, based on Eqs. (6.10) and (6.12) the transmit power $P^{t,req}_{i,j}$ required for satisfying the BER threshold is equal to:

$$P^{t,req}_{i,j} = L_{i,j}(N_0 + I_{\max})\gamma^{th}_{i,j}. \tag{6.13}$$

We have imposed a further constraint regarding the actual transmitters' maximum power level $P^{t,act}_{i,j}$. In fact, it is considered to be upper-bounded to $P^{t,act}_{\max} = 20$ dBm, which is a typical value for the IEEE 802.11b/g protocol. Based on this constraint, we can define

Table 6.1: Twin-Layer Network Parameters

Description	Parameter
Coverage Area	(100×100) m^2 Square Block
Number of MRs	$N_{MR} = \{5, 6, 7, 8, 9, 10\}$ MRs
Number of MCs	$N_{MC} = \{2, 4, 8, 16\}$ MCs
Social Relationship	Karate Club Members [185]
Max. Trans. Power	$P_{\max}^{t,act} = 20$ dBm
Carrier Frequency	$f_c = 2.4$ GHz
Trans. Bandwidth	$W = 10$ MHz
AWGN PSD	$N_0 = -174$ dBm/Hz
Reference Distance	$d_0 = 1$ m
Path-loss Exponent	$\alpha = 3$
Modulation	QPSK
BER Threshold	$P_e^{th} = 0.01$, uncoded
Max. Tol. Interfernce	$I_{\max} = \{-83.96, -83.06, -82.25,$ $-81.61, -81.01, -80.25\}$ dBm
Avg. MR Degree	$\{3.28,\ 3.81,\ 4.26, 4.72, 5.08,$ $5.32\}$ MRs

the *adjacency matrix* A as follows:

$$a_{i,j} = u(P_{i,j}^{t,req} - P_{\max}^{t,act}), \tag{6.14}$$

where $a_{i,j}$ corresponds to the element of the matrix A located at the i-th row and the j-th column, while $u(\cdot)$ is the Heaviside function defined in [193]. Therefore, the actual transmitted power $P_{i,j}^{t,act}$ required for establishing the link between the nodes i and j is equal to:

$$P_{i,j}^{t,act} = P_{i,j}^{t,req}/a_{i,j}. \tag{6.15}$$

Based on Eq. (6.15), the cost in terms of power for the link spanning from the i-th node to the j-th one will be equal to the power required for achieving a BER equal to the threshold value should the required power be less or equal to the maximum transmit power value. Otherwise, the cost is set to $+\infty$, implying that the link cannot be established.

As far as the maximum tolerable interference power level I_{max} is concerned, it is defined as the maximum interference level that allows the MRs to establish at least a single link with the rest of the WMN with a probability of 99%. Therefore, its value can be determined from the CDF of the connectivity of MRs versus the value of I_{max}, which is shown in Fig. 6.5, while the corresponding average number of potential connections for each MR, which is termed as their average *degree*, are presented in Fig. 6.6. Explicitly, observe in

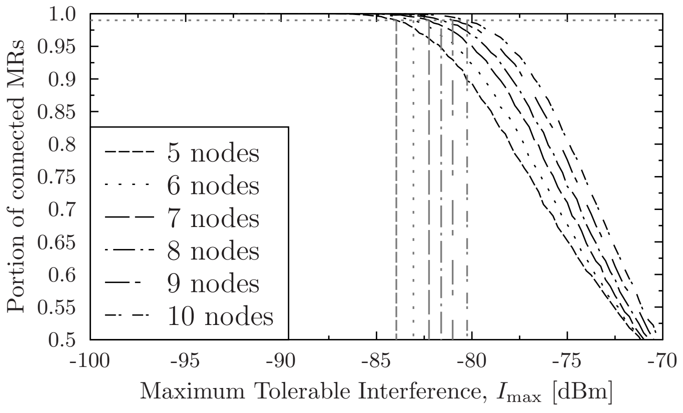


Figure 6.5: CDF of the Maximum tolerable interference level $I_{\max}$ for WMNs consisting from 5 until 10 MRs. The $I_{\max}$ power levels correponding to 99% of the MRs being connected to the WMN and their respective average degrees are shown in the last two rows of Table 6.1. The gray lines in (a) correspond to the respective $I_{\max}$ values. The results have been averaged over 10^6 runs.

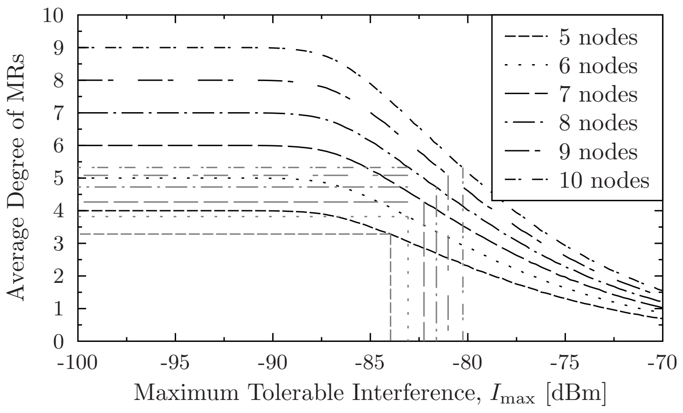


Figure 6.6: Average MR degree versus $I_{\max}$ for WMNs consisting from 5 until 10 MRs. The $I_{\max}$ power levels correponding to 99% of the MRs being connected to the WMN and their respective average degrees are shown in the last two rows of Table 6.1. The gray lines in (a) correspond to the respective $I_{\max}$ values. The results have been averaged over 10^6 runs.

Fig. 6.5 that the $I_{\max}$ value decreases as the WMN becomes more densely populated by MRs due to the inherent decrease in the minimum distance between the MRs. As for their degree corresponding to $I_{\max}$, it increases as the number of MRs increases ; however, the WMN becomes more sparsely connected than the fully-connected case scenario. The latter justifies the employment of a routing scheme, since a heuristic method has to be employed for identifying the realizable routes, i.e. routes consisting of links that can be established, based on the adjacency matrix A.

Having defined the physical layer parameters of the WMN layer, let us now proceed by defining our multiple-objective optimization problem. Firstly, let us consider the set of N_r active routes $S = \left[x^{(1)}, \dots, x^{(i)}, \dots, x^{(N_r)}\right]$, where $x^{(i)}$ is the i-th active route and corresponds to a unique pair of MCs. Note that the active routes are a subset of the feasible routes, which are determined by the MCs' social relationship matrix. To elaborate further, the set of active routes contains one-way routes from all the possible pairs of MCs. This is justified by our assumption of having coupled uplink and downlink. For instance, let us assume that MC$_i$ and MC$_j$ have a social relationship; in the active routes' set S^{act} only the route from MC$_i$ to MC$_j$ or vice versa is present, whilst its reverse will be active in the next timeslot. By contrast, both routes from MC$_i$ to MC$_j$ and from MC$_j$ to MC$_i$ are considered feasible. Since each MC is associated with a unique MR, each active route is defined as follows:

$$x^{(i)} = [MC_k, MR_l, \dots, MR_m, MC_n], \tag{6.16}$$

where the i-th active route corresponds to a transmission from MC_k to MC_n, while the source and destination MCs are associated with MR_l and MR_m, respectively. As our first objective, we will consider the average route delay $\bar{D}$, which is quantified as follows:

$$\bar{D}(S) = \sum_{i=1}^{N_r} \frac{D^{(i)}(S)}{N_r}, \tag{6.17}$$

where S is the set of the active routes and $D^{(i)}(S)$ corresponds to the delay of the i-th active route. For the sake of simplicity, we have chosen to quantify the latter as the number of hops incorporated by the route $x^{(i)}$. Hence, the route delay $D^{(i)}(S)$ is formulated as follows:

$$D^{(i)}(S) = \sum_{j=1}^{\left|x^{(i)}\right|-1} \left(a^{-1}_{x^{(i)}_j,\, x^{(i)}_{j+1}} \right), \tag{6.18}$$

where the factor $\left|x^{(i)}\right|$ denotes the number of nodes involved by the route $x^{(i)}$, while $x^{(i)}_j$ corresponds to the route's j-th node. Observe in Eq. (6.18) that the sum of the inverse of the adjacency matrix elements guarantees that all the route's links can be established. Otherwise, the route delay will be set to $D^{(i)}(S) = +\infty$, hence classifying the route $x^{(i)}$ as infeasible. We note that the nodes' specific buffer packet length could be readily encapsulated in Eq. (6.18) in order to account for delays the imposed by buffered packets.

Apart from the average delay $\bar{D}$, we have also considered the routes' average power consumption $\bar{P}$, which is quantified as follows:

$$\bar{P}(S) = \sum_{i=1}^{N_r} \frac{P^{(i)}(S)}{N_r}, \tag{6.19}$$

where $P^{(i)}(S)$ corresponds to the power consumption of the route $x^{(i)}$, which is in turn formulated based on Eq. (6.15) as follows:

$$P^{(i)}(S) = \sum_{j=1}^{\left|x^{(i)}\right|-1} P^{t,act}_{x^{(i)}_j,\, x^{(i)}_{j+1}}. \tag{6.20}$$

Observe in Eq. (6.20) that the adjacency matrix elements $a^{-1}_{x^{(i)}_j,\, x^{(i)}_{j+1}}$ are taken into account in Eq. (6.20) with the aid of Eq. (6.15). Explicitly, a route comprised by links that cannot guarantee satisfying the BER threshold will require a power set to $P^{(i)}(S) = +\infty$, based on Eq. (6.20). This is in line with the route's delay, which is at the same time set to $+\infty$. Therefore, we have encapsulated the BER constraint in both of our optimization objectives, which we will refer to as *Utility Functions* (UF). Based on these UFs let us now define the optimization *Utility Vector* (UV), which we will use for jointly optimizing both the average delay and the average power consumption, as follows:

$$\mathbf{f}(x^{(1)}, \ldots, x^{(N_r)}) \equiv \mathbf{f}(S) = [D(S), P(S)]\,. \tag{6.21}$$

Observe that the UV of Eq. (6.21) consists of two optimization components, namely the average delay and the average power consumption, which is in contrast to the UV of Eq. (2.5), consisting of three components, additionally considering the route's BER as well. Explicitly, the routes' BER has also been considered in Eq. (6.21) in the form of an optimization constraint, which exclusively opts for routes exhibiting an infinitesimally low BER, i.e. routes having near-perfect integrity.

6.2.3 Multiple-Objective Optimization Model

Based on Definitions 3 and 4, the OPF is composed by route-solutions having UFs, which cannot be further optimized individually without degrading the fitness of the rest of the UFs, as it can be observed in Fig. 2.4. As far as our specific application is concerned, when considering weak OPFs relying on Definition 3, there exist route-solutions classified as Pareto-optimal which may have the same metric, say in terms of their average delay $\bar{D}$, yet exhibiting a different performance in terms of their average power consumption $\bar{P}$. Nevertheless, the route-solution that exhibits lower average power consumption seems to outperform the other one, since it jointly minimizes both UFs. This specific caveat is rectified by the employment of strong Pareto-optimality, which is in contrast to the weak Pareto-optimality considered in Chapters 4 and 5, since the route-solution associated with lower average power consumption would dominate the other route-solution. Hence the latter will not be included in the respective OPF. Based on this observation, we will utilize the concept of strong Pareto-optimality for the sake of constraining the load balancing problem, which is formulated in Eq. (6.9). Consequently, our optimization problem is formulated as follows:

$$\begin{aligned} S^{\text{OPF}} = \; & \underset{\forall S_i \in S_{legit}}{\text{argmax}} \quad \bar{H}(\bar{B}_{com}(S_i)), \\ & \text{subject to} \quad \nexists j : \; \mathbf{f}(S_j) \succeq \mathbf{f}(S_i), \end{aligned} \tag{6.22}$$

where S^{OPF} represents the optimal active route allocation based on our constrained optimization problem and S_{legit} corresponds to the set containing all the potential sets of active routes, which are strictly comprised by individual Hamiltonian routes, i.e. by routes that visit each of the MRs at most once. In a nutshell, the optimization problem of Eq. (6.22) attempts to distribute the intermediate relay tele-traffic load among the MRs as close as possible to the ideal uniformly distributed load, whist ensuring that the associated network performance is Pareto-optimal in terms of its average delay and power consumption. Explicitly, the optimization problem of Eq. (6.22) is a four-component problem, jointly taking into account the routes' BER, their average delay, their average power consumption and the route-combinations' normalized entropy of their associated normalized betweeness. The routes' integrity quantified in terms of their BER is considered with the aid of the adjacency matrix A, defined in Eq. (6.14), hence solely considering valid routes, namely those having a lower BER than the BER threshold P_e^{th}.

Let us now characterize the Pareto optimality problem of Eq. (6.22) in terms of its complexity. For the sake of verifying as to whether a single set of active routes satisfies the Pareto optimality constraint of Eq. (6.22), we have to perform precisely N Pareto dominance comparisons, where N corresponds to the total number of Hamiltonian routes, as defined in Eq. (2.8). Let us now assume that the total number of pairs of source and destination MCs is exactly $N_r = |S|$. Then the total number N_{tot} of active routes-sets is given by:

$$N_{tot} = N^{N_r} = \left[\sum_{i=0}^{N_{MR}-2} \frac{(N_{MR}-2)!}{(N_{MR}-2-i)!} \right]^{N_r}. \tag{6.23}$$

Since it can be observed in Eq. (6.23) that the total number of the active routes-set N_{tot} increases exponentially with the number of MRs N_{MR}, our constrained optimization problem defined in Eq. (6.22) is classified as NP-hard. Consequently, sophisticated quantum-assisted methods are required for tackling the escalating complexity, hence rendering the optimization problem of Eq. (6.22) tractable. Let us now proceed by presenting a 12-node tutorial example, which we will solve using the exhaustive search method, for the sake of providing further insights into the operation of our socially-aware network model considered.

6.2.4 A 12-Node Tutorial Example using Exhaustive Search

In the context of this tutorial, let us consider the twin-layer network comprised by $N_{\text{MR}} = 7$ MRs and $N_{\text{MC}} = 5$ MCs, which relies on the topology portrayed in Fig. 6.7, where the association of each of the MCs to their respective MR is represented by the dashed lines connecting each of the MCs to their closest MR. We may consider this topology as a random snapshot of our general network topology, where the MCs are mobile, whilst the MRs are considered to be static. Note that this joint routing and load balancing optimization process has to be repeated at a frequency depending on the MCs' speeds. Nevertheless, we have not investigated this aspect in the context of this treatise.

The transmission power matrix $P^{t,act}$ defined in Eq. (6.15) for the exemplified topology of Fig. 6.7 is shown in Table 6.2, where we can observe that the transmission power levels of several links are set to infinity, indicating that the specific links are infeasible. We note that the elements of the transmission power matrix $P^{t,act}$ in Table 6.2 are quantified in dBm. Naturally, the transmission power of the links among the MCs is set to infinity, since they cannot directly communicate with each other, only through their associated MRs. Additionally, observe in Table 6.2 that for the links established between a specific MC and the MRs, there exists only a single link having a finite transmission power, owing to

Table 6.2: Transmission power matrix $P^{t,act}$ quantified in dBm for the exemplified topology of Fig. 6.7.

src\dst	MC$_1$	MC$_2$	MC$_3$	MC$_4$	MR$_5$	MR$_1$	MR$_2$	MR$_3$	MR$_4$	MR$_5$	MR$_6$	MR$_7$
MC$_1$	Inf	Inf	Inf	Inf	Inf	Inf	Inf	5.89	Inf	Inf	Inf	Inf
MC$_2$	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	-6.89	Inf
MC$_3$	Inf	Inf	Inf	Inf	Inf	Inf	5.15	Inf	Inf	Inf	Inf	Inf
MC$_4$	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	-7.97
MC$_5$	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	10.96	Inf	Inf
MR$_1$	Inf	Inf	Inf	Inf	Inf	Inf	11.59	14.87	12.31	-5.35	14.57	13.71
MR$_2$	Inf	Inf	5.15	Inf	Inf	11.59	Inf	Inf	Inf	7.77	Inf	Inf
MR$_3$	5.89	Inf	Inf	Inf	Inf	14.87	Inf	Inf	13.53	16.08	2.48	17.24
MR$_4$	Inf	Inf	Inf	Inf	Inf	12.31	Inf	13.53	Inf	15.27	7.64	-0.78
MR$_4$	Inf	Inf	Inf	Inf	10.96	-5.35	7.77	16.08	15.27	Inf	16.53	16.39
MR$_6$	Inf	-6.89	Inf	Inf	Inf	14.57	Inf	2.48	7.64	16.53	Inf	13.12
MR$_7$	Inf	Inf	Inf	-7.97	Inf	13.71	Inf	17.24	-0.78	16.39	13.12	Inf

the constraint that a specific MC can only connect to the rest of the network through its closest MR. As far as the links among the MRs are concerned, recall that according to Table 6.1 the links requiring a transmission power of infinity are unable to satisfying the BER threshold of $P_e^{th} = 10^{-2}$ at the maximum transmission power $P_{\max}^{t,act} = 20$ dBm.

Having presented the specifics of the tutorial example WMN layer, let us now proceed by elaborating on the details of its OSN layer. As we have mentioned in Subsec. 6.2.1, the MCs exhibit an identical social relationship to that of the members of a Karate Club, which is portrayed Fig. 6.4. Based on this relationship, we have randomly picked 5 Karate Club members, whose social relationship can be encapsulated in the following MC friendship matrix $\mathcal{F}_{\text{MC}}$:

$$\mathcal{F}_{\text{MC}} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}, \tag{6.24}$$

where we can observe that the MC$_1$ shares a social relationship with all the remaining MCs, MC$_2$ shares a social relationship with MC$_1$ and MC$_5$, MC$_3$ and MC$_4$ share a social relationship solely with MC$_1$, while MC$_5$ has a social relationship with both MC$_1$ as well as MC$_2$. This social relationship of the MCs is visually portrayed in Fig. 6.8. Based on the friendship relationship $\mathcal{F}_{\text{MC}}$ of Eq. (6.24), we will consider in this tutorial the following set

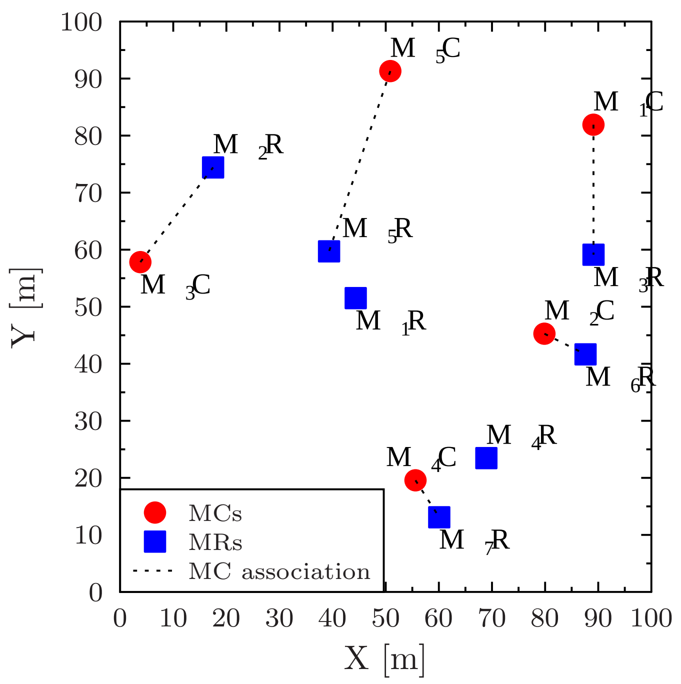


Figure 6.7: Exemplified twin-layer network topology with $N_{\text{MC}} = 5$ MCs and $N_{\text{MR}} = 7$ MRs for a coverage area of (100×100) m^2 square block. The association of a specific MC with a specific MR is annotated using the dashed lines. The presence of a central inteligence cluster head node is assumed, albeit not portrayed in this figure.

S^{act} of active source and destination pairs:

$$S^{\text{act}} = \left\{ \begin{array}{c} \text{MC}_2 \to \text{MC}_1 \\ \text{MC}_1 \to \text{MC}_3 \\ \text{MC}_4 \to \text{MC}_1 \\ \text{MC}_5 \to \text{MC}_1 \\ \text{MC}_2 \to \text{MC}_5 \end{array} \right\}. \tag{6.25}$$

In our scenario, there exists $N = 326$ Hamiltonian routes between each specific source and destination MCs, based on Eq. (2.8), while there exist about $N_{tot} = N^5 \simeq 3.682 \cdot 10^{12}$ legitimate route-combinations in total, based on Eqs. (6.23) and (6.53). The exhaustive search method relies upon evaluating each of the legitimate route-combinations for the sake of checking as to whether they are strongly Pareto-optimal, hence satisfying the constraint of Eq. (6.22). Subsequently, if a specific route-combination is identified as being strongly Pareto-optimal, the value of the normalized entropy of its associated normalized composite betweeness is evaluated as well, aiming for identifying the specific route-combination that maximizes this utility. Therefore, the exhaustive search has to carry out N_{tot}^2 weak Pareto-dominance checks just for identifying the strongly Pareto-optimal route-combinations, plus N_{OPF} single-objective comparisons for determining the maximum normalized entropy route-combination, where N_{OPF} is the number of strongly Pareto-optimal

Table 6.3: Pareto-optimal route-combinations identified by the exhaustive search for the socially-aware network of Fig. 6.7 corresponding to the OPF seen in Fig. 6.9. The individual routes $S^{\text{OPF}}_{\text{MC},i,j}$ in the second column are defined in Table 6.5.

ID	Route-Combination, S	$\bar{D}(S)$	$\bar{P}(S)$	$\bar{H}[\bar{B}_{com}(S)]$	$\max\{\bar{H}[\bar{B}_{com}(S)]\}$	$\operatorname{argmax}\{\bar{H}[\bar{B}_{com}(S)]\}$
S^{OPF}_{1}	$S^{\text{OPF}}_{\text{MC},1,1}, S^{\text{OPF}}_{\text{MC},2,1}, S^{\text{OPF}}_{\text{MC},3,1}, S^{\text{OPF}}_{\text{MC},4,1}, S^{\text{OPF}}_{\text{MC},5,1}$	16.62	3.20	0.000	0.000	S^{OPF}_{1}
S^{OPF}_{2}	$S^{\text{OPF}}_{\text{MC},1,1}, S^{\text{OPF}}_{\text{MC},2,3}, S^{\text{OPF}}_{\text{MC},3,1}, S^{\text{OPF}}_{\text{MC},4,1}, S^{\text{OPF}}_{\text{MC},5,1}$	16.00	3.40	0.000	0.000	S^{OPF}_{1}
S^{OPF}_{3}	$S^{\text{OPF}}_{\text{MC},1,1}, S^{\text{OPF}}_{\text{MC},2,3}, S^{\text{OPF}}_{\text{MC},3,1}, S^{\text{OPF}}_{\text{MC},4,1}, S^{\text{OPF}}_{\text{MC},5,1}$	15.63	3.60	0.327	0.327	S^{OPF}_{3}
S^{OPF}_{4}	$S^{\text{OPF}}_{\text{MC},1,1}, S^{\text{OPF}}_{\text{MC},2,2}, S^{\text{OPF}}_{\text{MC},3,3}, S^{\text{OPF}}_{\text{MC},4,1}, S^{\text{OPF}}_{\text{MC},5,1}$	15.29	3.80	0.534	0.534	S^{OPF}_{4}
S^{OPF}_{5}	$S^{\text{OPF}}_{\text{MC},1,1}, S^{\text{OPF}}_{\text{MC},2,2}, S^{\text{OPF}}_{\text{MC},3,3}, S^{\text{OPF}}_{\text{MC},4,1}, S^{\text{OPF}}_{\text{MC},5,4}$	15.04	4.00	0.488	0.534	S^{OPF}_{4}
S^{OPF}_{6}	$S^{\text{OPF}}_{\text{MC},1,1}, S^{\text{OPF}}_{\text{MC},2,2}, S^{\text{OPF}}_{\text{MC},3,3}, S^{\text{OPF}}_{\text{MC},4,4}, S^{\text{OPF}}_{\text{MC},5,4}$	14.81	4.20	0.638	0.638	S^{OPF}_{6}
S^{OPF}_{7}	$S^{\text{OPF}}_{\text{MC},1,1}, S^{\text{OPF}}_{\text{MC},2,2}, S^{\text{OPF}}_{\text{MC},3,3}, S^{\text{OPF}}_{\text{MC},4,4}, S^{\text{OPF}}_{\text{MC},5,2}$	14.64	4.60	0.679	0.679	S^{OPF}_{7}
S^{OPF}_{8}	$S^{\text{OPF}}_{\text{MC},1,1}, S^{\text{OPF}}_{\text{MC},2,2}, S^{\text{OPF}}_{\text{MC},3,3}, S^{\text{OPF}}_{\text{MC},4,2}, S^{\text{OPF}}_{\text{MC},5,2}$	14.45	5.00	0.675	0.679	S^{OPF}_{7}
S^{OPF}_{9}	$S^{\text{OPF}}_{\text{MC},1,1}, S^{\text{OPF}}_{\text{MC},2,2}, S^{\text{OPF}}_{\text{MC},3,2}, S^{\text{OPF}}_{\text{MC},4,4}, S^{\text{OPF}}_{\text{MC},5,4}$	14.64	4.40	0.656	0.679	S^{OPF}_{7}
S^{OPF}_{10}	$S^{\text{OPF}}_{\text{MC},1,1}, S^{\text{OPF}}_{\text{MC},2,2}, S^{\text{OPF}}_{\text{MC},3,2}, S^{\text{OPF}}_{\text{MC},4,2}, S^{\text{OPF}}_{\text{MC},5,4}$	14.46	4.80	0.674	0.679	S^{OPF}_{7}
S^{OPF}_{11}	$S^{\text{OPF}}_{\text{MC},1,1}, S^{\text{OPF}}_{\text{MC},2,2}, S^{\text{OPF}}_{\text{MC},3,2}, S^{\text{OPF}}_{\text{MC},4,2}, S^{\text{OPF}}_{\text{MC},5,2}$	14.26	5.20	0.665	0.679	S^{OPF}_{7}

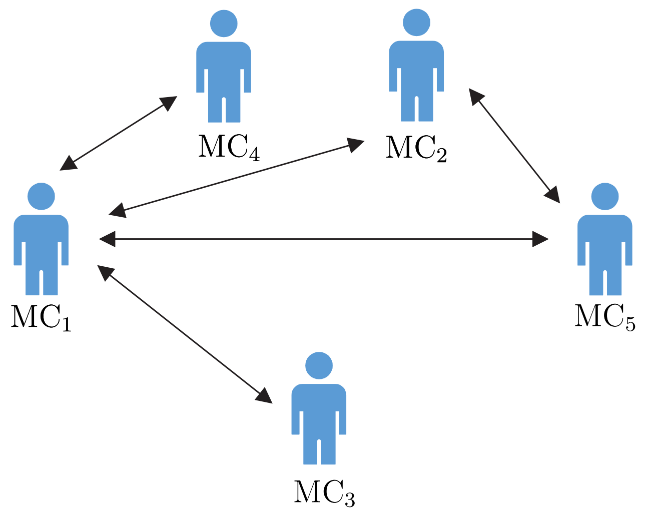


Figure 6.8: Visual representation of the MCs' social relationship, based on Eq. (6.24).

route-combinations.

The strongly Pareto-optimal routes exported by the exhaustive search are portrayed with the aid of the square markers in Fig. 6.9 along with the route-combinations belonging to the 100 lowest-rank PFs, which are represented by the dot markers. We note that we have opted for including only the 100 lowest-rank PFs in Fig. 6.9 for the sake of simplicity. Still referring to the same figure, the specific route-combination that maximizes the normalized entropy of its associated normalized composite betweenness is denoted by the diamond marker. Additionally, a more detailed presentation of the strongly Pareto-optimal route-combinations is included in Table 6.3, where the Pareto-optimal route-combinations S^{OPF_i} are sorted according to their order of being identified as being as Pareto-optimal by the exhaustive method. We note that the database of the route-combinations is constructed by combining the databases combining the individual routes' databases, which are in turn constructed by storing the respective routes in lexicographical ordering using *Lehmer encoding* [1]. Hence, we are able to observe in Table 6.3 the evolution of the maxi-

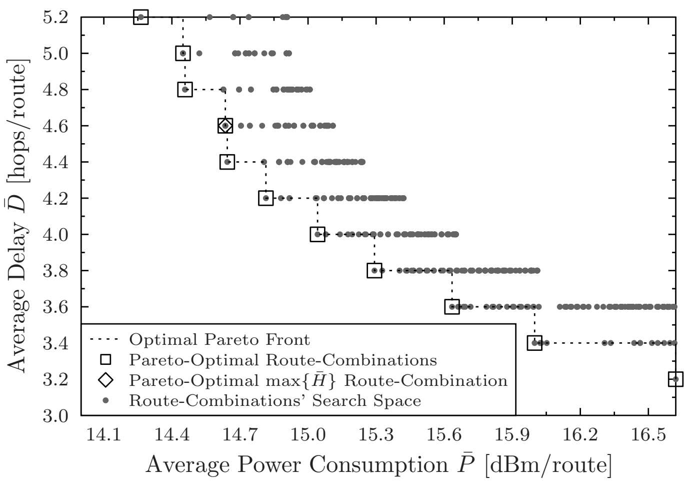


Figure 6.9: Solution space of the route-combinations of the socially-aware network of Fig. 6.7 in terms of their average power consumtpion $\bar{P}$,quantified in dBm per route, and their average delay $\bar{D}$, quantified in number of hops per route. For the sake of simplicity, we have opted for only portraying the 100 lowest-rank PFs.

mum value of the normalized entropy observed by the exhaustive search, which converges to its maximum value for the seventh route-combination S^{OPF_7} of Table 6.3. Observe in this table that although the maximum value is observed after identifying seven Pareto-optimal route-combinations, the exhaustive search has to identify the entire set of Pareto-optimal route-combinations to classify this value as the maximum observed.

Before delving into the presentation of our proposed algorithm, we provide a brief introduction to the existing quantum search and optimization algorithms, which will constitute the building blocks of our novel algorithm.

6.3 Design Methodology

Based on the parallel complexity of the NDQIO algorithm quantified in terms of the number of dominance comparisons, which is defined in Eq. (5.53) invoking the NDQIO algorithm for the optimization problem of Eq. (6.22) impose a parallel complexity, which is on the order of $O(N_{\text{OPF}}N^{N_r/2})$, where N corresponds to the total number of Hamiltonian routes from a specific pair of source and destination MCs, defined in Eq. (2.8). Naturally, this is significantly lower than $O(N^{2N_r})$ imposed by the exhaustive search. This complexity reduction, albeit substantial, may not be sufficient for near-real-time applications, when the nodes' locations rapidly change over time. In fact, Zalka [88] has proven that Grover's QSA is optimal in terms of the number of CFEs imposed by the algorithm. Since Grover's QSA has been the most popular technique in the family quantum amplitude amplification algorithms [80, 81, 112], which includes the NDQIO algorithm, we cannot achieve a complexity reduction more than that of a factor on the order of $O(\sqrt{N})$.

Having said this, all the QSA-based algorithms are totally oblivious of the optimization problem's structure, and hence they are incapable of exploiting the correlation of the elements in a database. Consequently, our design objective is twofold: on the one hand, we have to transform our composite optimization problem into a series of independent sub-problems, which exhibit a potentially uncorrelated search space; on the other hand, we have to develop a reduced-complexity quantum-assisted process for merging the results of the respective sub-processes, whilst minimizing the potential complexity overhead of the merging process. Naturally, this approach confines the initial search space considered, hence yielding a substantial complexity. Additionally, note that we have opted for designing our novel quantum-assisted algorithm with the goal of minimizing the parallel complexity. Recall from Section 5.2 that this specific type of computational complexity may be deemed to be commensurate with the algorithms' normalized time execution. Therefore, we minimize the algorithm's normalized execution time by minimizing its parallel complexity. This is of utter importance in a dynamic network, such as the socially-aware network relying on the assumptions of Table 6.1, where the algorithm's input parameters fluctuate. Explicitly, an extensive normalized time execution reduces the algorithm's heuristic accuracy owing to outdated input parameters. Despite this design consideration, we will characterize our novel algorithm's performance in terms of its sequential complexity as well, which is deemed to be commensurate with the algorithms' normalized power consumption.

Having defined our algorithmic design targets in broad terms, let us now proceed with a tutorial example using the exhaustive search for the sake of a better understanding of the the joint routing and load balancing optimization problem, defined in Eq. (6.22).

6.3.1 Weak Pareto Dominance Operator

Before delving into the aforementioned transformation specifics, we will introduce the unitary operator U_{g_w}, which carries out a single weak Pareto dominance comparison and will be used as the Oracle gate of the Grover operator $\mathcal{G}$ deployed in our proposed algorithm. Due to the universality of the quantum gate-based computation [62], we have to derive a binary function for implementing the weak Pareto dominance operator. For this reason, let us define the comparison functions $f_k^{\bullet}(x,i)$ in terms of the k-th objective as follows:

$$f_k^{\bullet}(x,i) = \begin{cases} 1, & f_k(x) \bullet f_k(i), \\ 0, & \text{otherwise}, \end{cases} \tag{6.26}$$

where the operator $\bullet$ is the generic comparison operator corresponding to the operators $\leq$, $=$, $<$ etc. Note that the set of $f_k^{\bullet}(x,i)$ functions is the generalization of the binary less comparison function $f_k(x,i)$ defined in Eq. (4.2). This comparison function is implemented by the quantum unitary operator $U_{f_k^{\bullet}}$ defined as follows:

$$|x\rangle\,|i\rangle\,|t\rangle \xrightarrow{U_{f_k^{\bullet}}} |x\rangle\,|i\rangle\,|t \oplus f_k^{\bullet}(x,i)\rangle\,, \tag{6.27}$$

where the quantum registers $|x\rangle$, $|i\rangle$, $|t\rangle$ are often referred to as *Quantum Index Register* (QIR), *Quantum Control Register* (QCR) and *Oracle Workspace* (OW), respectively [1]. The application of $U_{f_k^{\bullet}}$ results in entangling the states of the aforementioned registers.

We may readily create the binary expression of the weak dominance comparison using the generic comparison functions of Eq. (6.26). Based on Definition 1, the x-th route will be dominated by the reference route associated with the i-th index, provided that we have $f_k^{\leq}(x,1) = 1$, $\forall k \in \{1,\ldots,K\}$, while at the same time we have $\exists k \in \{1,\ldots,K\}$ so that $f_k^{<}(x,1) = 1$ is satisfied. Explicitly, the second requirement is that we have to exclude the specific route-solutions validated by the first requirement but have their UFs equal to the respective of the i-th route-solution, i.e. we have $f_k^{=}(x,i) = 1$, $\forall k \in \{1,\ldots,K\}$. Consequently, the weak dominance comparison function $g_w(x,i)$ is defined as follows:

$$g_w(x,i) = \bigcap_{k=1}^{K} f_k^{\leq}(x,i) \oplus \bigcap_{k=1}^{K} f_k^{=}(x,i) \equiv \begin{cases} 1, \ \mathbf{f}(x) \succeq \mathbf{f}(i) \\ 0, \ \text{otherwise.} \end{cases} \tag{6.28}$$

Having expressed the weak Pareto dominance comparison function $g_w(x,i)$ as a function of the $f_k^{\bullet}(x,i)$ function, we may readily employ the $U_{f_k^{\bullet}}$ operators for constructing the quantum circuit of the U_{g_w} operator, which is presented in Fig. 6.10. Observe in this figure that a series of CNOT gates [62] are used for the sake of entangling both the input QIR and input QCR to the respective local quantum registers of each of the $\{U_{f_k}^{\leq}\}_{k=1}^{K}$ and $\{U_{f_k}^{=}\}_{k=1}^{K}$ operators. Explicitly, the route-solutions' indices are stored in form of their superposition in the input QIR, while the index of the reference route-solution is stored in the QCR. As for the $\{U_{f_k}^{\leq}\}_{k=1}^{K}$ and $\{U_{f_k}^{=}\}_{k=1}^{K}$ operators, they implement the comparison functions defined in Eq. (6.26) with respect to the k-th UF. Subsequently, all the local OW registers states of the $\{U_{f_k}^{\leq}\}_{k=1}^{K}$ operators are combined by the *Toffoli* gate T [62, 2], which performs an *exclusive-OR* (XOR) between the input OW register $|t\rangle_3$ and the intersection product of the states of all the local OW registers. Therefore, after the first Toffoli gate the composite quantum system state is formulated as:

$$|x\rangle_1 |i\rangle_2 |t''\rangle_3 = |x\rangle_1 |i\rangle_2 \left| t \oplus \bigcap_{k=1}^{K} f^{\leq}(x,i) \right\rangle_3 . \tag{6.29}$$

A second Toffoli gate is then used for the sake of combining the local OW registers of the $\{U_{f_k}^{=}\}_{k=1}^{K}$ operators. Hence, the resultant composite quantum system state is equal to:

$$\begin{aligned} |x\rangle_1 |i\rangle_2 |t'\rangle_3 = & \ |x\rangle_1 |i\rangle_2 \left| t \oplus \bigcap_{k=1}^{K} f^{\leq}(x,i) \oplus \bigcap_{k=1}^{K} f^{=}(x,i) \right\rangle_3 , \\ = & \ |x\rangle_1 |i\rangle_2 |t \oplus g_w(x,i)\rangle_3 . \end{aligned} \tag{6.30}$$

Explictly, Eq. (6.30) proves that the circuit of Fig. 6.10 implements the weak dominance comparison function $g_w(x,i)$.

As for the circuit's complexity quantified in terms of the number of CFEs, we will still consider a single CFE as the complexity imposed by the strong Pareto dominance operator of Fig. 4.1, which consists of a series of $U_{f_k^{<}}$ operators serially connected, for the sake of continuity. Therefore, assuming that both the CNOT and the Toffoli gates have an

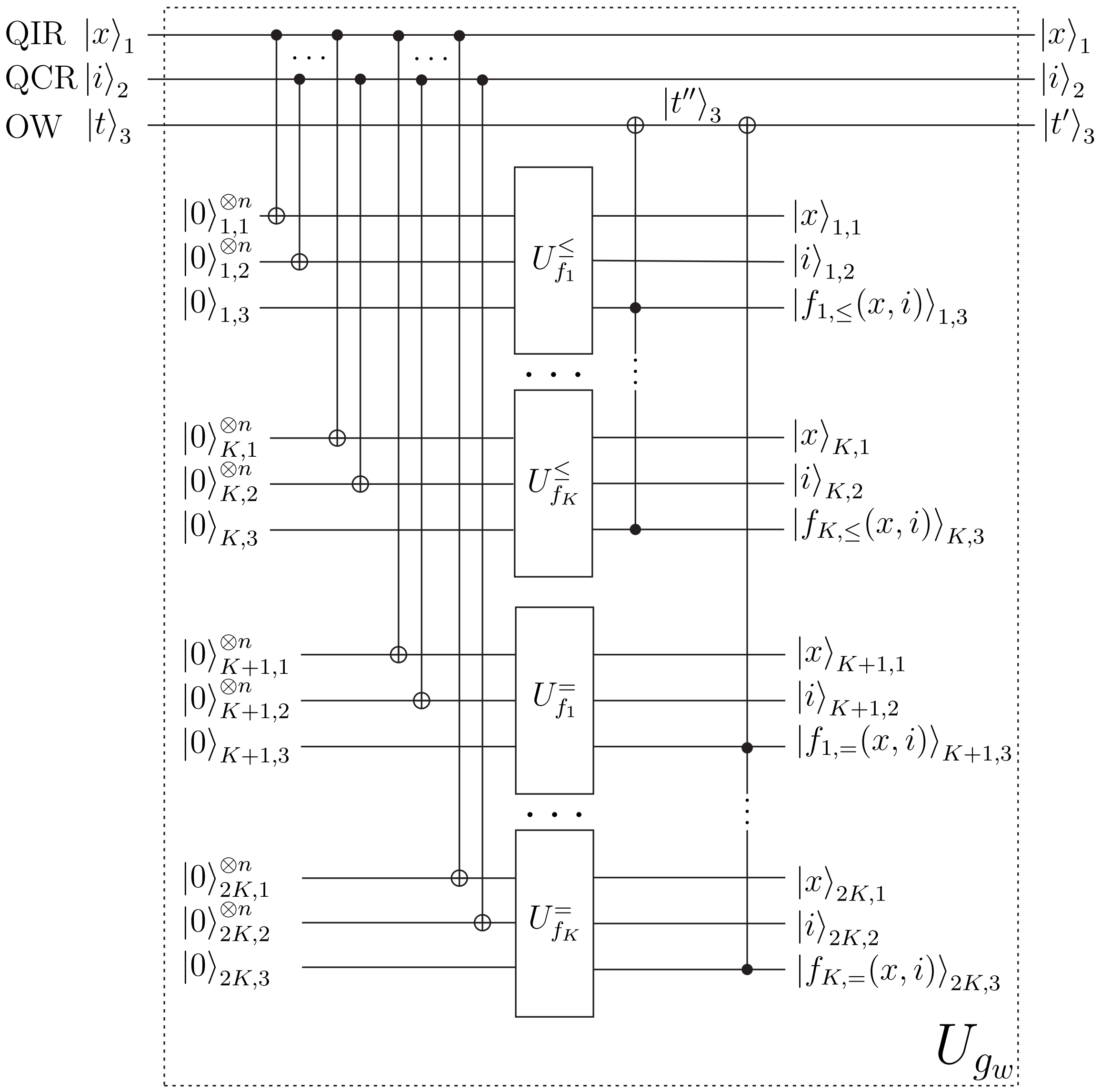


Figure 6.10: Quantum circuit of the unitary operator U_{g_w} implementing the weak Pareto dominance comparison.

instant response as in Section 5.2 and that the $U_{f_k^\bullet}$ comparison operators impose identical complexity, the parallel complexity imposed by the quantum circuit of Fig. 6.10 is equal to $1/K$ CFEs. This is justified by the parallel activation of the $U_{f_k^\bullet}$ operator through the employment of the synergistic framework between the QP and the HP, which was introduced in Section 5.2 and it is comprised by the series of CNOT gates and the Toffoli gates at the $U_{f_k^\bullet}$ comparison operators' input and output, respectively. Finally, the weak Pareto dominance comparison imposes 2 CFEs in terms of its sequential complexity. This is justified by the fact that the number of $U_{f_k^\bullet}$ comparison operators is doubled compared to the comparison operators used in the strong Pareto dominance comparison circuits presented in Figs. 4.1 and 5.4.

At this point, let us emphasize that in the context of this treatise the proposed U_{g_w} operator will be used as the Oracle gate in the NDQIO algorithm's sub-processes, so that the algorithm becomes capable of identifying the OPF formed by strongly Pareto-optimal route-solutions.

6.3.2 Multi-Objective Decomposition Quantum Optimization

As we mentioned in the introduction of this section, our design objective for the proposed algorithm is to transform the problem to a series of sub-problems having databases exhibiting the minimum amount of correlation among their elements. By a close inspection of Eqs. (6.17) and (6.19), which correspond to the average route delay and power consumption, respectively, we can conclude that both UFs considered in the constraint of Eq. (6.22) share the same generic form of:

$$f_k(S) = f_k(x^{(1)}, x^{(2)}, \dots, x^{(N_r)}) = \sum_{n=1}^{N_r} a_{k,n} f_k(x^{(n)}), \tag{6.31}$$

where S denotes the set of N_r active routes, $x^{(n)}$ corresponds to the n-th active route and $a_{k,n}$ is a constant, which may be different for each UF but obeys the constraint $a_{k,i} > 0$. In our scenario, we have $a_{k,i} = N_r^{-1}$ for both UFs and $\forall i \in \{1, \dots, N_r\}$. This specific form of the UFs can be exploited in the context of Pareto-optimality problems for reducing the search space, based on Proposition 1.

Proposition 1. *Let us assume having N_r independent strong Pareto-optimality problems, each associated with the UVs $\mathbf{f}_n(x^{(n)}) = \left[f_1(x^{(n)}), \dots, f_K(x^{(n)})\right]$, where K corresponds to the number of UFs, and that their OPF solutions form the sets $\left\{S_n^{OPF}\right\}_{n=1}^{N_r}$. Let us furthermore consider the composite Pareto-optimality problem of their convex combination, which is associated with the utility vector $\mathbf{f}(S) = \mathbf{f}(x^{(1)}, \dots, x^{(N_r)}) = \left[\sum_{n=1}^{N_r} a_{1,n} f_1(x^{(n)}), \dots, \sum_{n=1}^{N_r} a_{K,n} f_K(x^{(n)})\right]$, with $a_{k,n} > 0$ $\forall n \in \{1, \dots, N_r\}$ and $\forall k \in \{1, \dots, K\}$, and that the OPF solutions of this problem form the set S^{OPF}. The set S^{OPF} is a subset of the union of the sets $\left\{S_n^{OPF}\right\}_{n=1}^{N_r}$, i.e. we have:*

$$S^{OPF} \subseteq \bigcup_{n=1}^{N_r} S_n^{OPF}. \tag{6.32}$$

Proof. Let us assume that the solution $S = [x^{(1)}, \dots, x^{(j)}, \dots, x^{(N_r)}]$ of the composite Pareto-optimality problem is Pareto-optimal and that the independent solutions $\{x^{(n)}\}$ are Pareto-optimal in their respective independent problems except for the solution $x^{(j)}$, which is suboptimal in its respective independent problem. Hence, there exits a solution $x'^{(j)}$ such that $\mathbf{f}_j(x'^{(j)}) \succeq \mathbf{f}_j(x^{(j)})$, i.e. we have:

$$f_k(x^{(j)}) \geq f_k(x'^{(j)}), \ \forall n \in \{1, \dots, K\}. \tag{6.33}$$

Therefore, if we multiply Eq. (6.33) by the factor $a_{k,j}$ and add the terms $a_{k,n} f_k(x^{(n)})$

associated with $n \neq j$, we will have $\forall k \in \{1, \ldots, K\}$:

$$\sum_{n=1}^{N_r} a_{k,n} f_k(x^{(n)}) \geq \sum_{\substack{n=1 \\ n \neq j}}^{N_r} a_{k,n} f_k(x^{(n)}) + a_{k,j} f_k(x'^{(j)}), \tag{6.34}$$

which can be written in the following compact form:

$$f_k(S) \geq f_k(S'), \ \forall k \in \{1, \ldots, K\} \tag{6.35}$$

where $S' = [x^{(1)}, \ldots, x'^{(j)}, \ldots, x^{(N_r)}]$. Additionally, since $x'^{(j)}$ strongly dominates the solution $x^{(j)}$, we have that $\exists k' \in \{1, \ldots, K\}$ such that:

$$f_{k'}(x^{(j)}) > f_{k'}(x'^{(j)}). \tag{6.36}$$

If we now multiply Eq. (6.36) by the factor $a_{k',j}$ and add the terms $a_{k',n} f_{k'}(x^{(n)})$ with $n \neq j$, we will have for this specific k':

$$\sum_{n=1}^{N_r} a_{k',n} f_{k'}(x^{(n)}) > \sum_{\substack{n=1 \\ n \neq j}}^{N_r} a_{k',n} f_{k'}(x^{(n)}) + a_{k',j} f_{k'}(x'^{(j)}), \tag{6.37}$$

which can be written in the following compact form:

$$\exists k' \in \{1, \ldots, K\} : f_{k'}(S) > f_{k'}(S'). \tag{6.38}$$

Hence, observe that Eqs. (6.35) and (6.38) encapsulate the two critical conditions so that $\mathbf{f}(S') \succeq \mathbf{f}(S)$, based on Definition 1, yielding that the initial assumption of having $S \in S^{\text{OPF}}$ is invalid. Hence, we have $S = [x^{(1)}, \ldots, x^{(N_r)}] \in S^{\text{OPF}}$ only if $x^{(n)} \in S_n^{\text{OPF}}$, $\forall n \in \{1, ..., N_r\}$. Therefore, all members of S^{OPF} are contained in the union of the sets $\{S_n^{\text{OPF}}\}_{n=1}^{N_r}$, hence proving the claim of Eq. (6.32). $\square$

We note that the inverse of Proposition 1 does not apply, since there may exist solutions, which are composed by Pareto-optimal solutions in all the respective independent problems, but are suboptimal in the composite problem. Despite this limitation, Proposition 1 provides us with useful insight into a potential transformation of our search space for sake of reducing the total number N_{tot} of sets of active routes considered, which was defined in Eq. (6.23).

Explicitly, we do not have to consider all the possible sets of active routes S; instead, we only have to identify the routes belonging to the union of the OPFs of all the active pairs of source and destination MCs. This actually constitutes a *divide and conquer* approach, since the sub-problems created for finding the OPF of a specific pair of source and destination MCs are independent of each other, yielding a reduction in the total complexity required by the exhaustive search, which is on the order of $O(N_rN^2)$, down from $O(N^{2N_r})$. Nevertheless, the solutions identified are not Pareto-optimal as yet, as we know based on Proposition 1; we still need a process for identifying the composite OPF S^{OPF} from the union of the OPFs of the independent sub-problems $\{S_n^{\text{OPF}}\}_{n=1}^{N_r}$, which yields an additional overhead on the order of $O(\bar{N}_{\text{OPF}}^{2N_r})$ in terms of complexity imposed by the exhaustive search, where $\bar{N}_{\text{OPF}}$ corresponds to average number of Pareto-optimal routes for each of the sub-problems. Therefore, the total complexity imposed by the exhaustive search using this transformation method is on the order of $O(N_rN^2 + \bar{N}_{\text{OPF}}^{2N_r})$, which is far less than $O(N^{2N_r})$, assuming that $O(\bar{N}_{\text{OPF}}) \ll O(N)$.

Explicitly, the exhaustive search method is far from efficient, despite the use of the search space transformation method, which was analyzed in the previous paragraph. For this reason, we will exploit the hybrid hardware and quantum parallelism offered by the NDQIO algorithm for further reducing the complexity of both the search space transformation step and the merging step, which will be referred to from now on as the *inner step* and *outer step*, respectively. The flowchart of our proposed algorithm, namely the *Multi-Objective Decomposition Quantum Optimization* (MODQO) algorithm, is shown in Fig. 6.11, where it can be observed that the execution of the algorithm is comprised by four distinct sub-processes or blocks.

To elaborate further, since our routing algorithm requires a centralized quantum-computer the presence of a cluster head is assumed, which monitors and controls the dissemination of the packets throughout the network. For the sake of performing optimal joint routing and load balancing, the cluster head needs to gather all the necessary data for constructing $\mathcal{F}_{\text{MR}}$, $\mathcal{F}_{\text{MC}}$, Z_{MR}, Z_{MC} and I_{MC}, corresponding to the MRs' friendship matrix, the MCs' friendship matrix, the MR locations, the MC locations and the MC to MR association vector, respectively. This information is essential for accurately evaluating the routes' UVs and, in the context of this treatise, we will assume perfect estimation of the aforementioned parameters at the cluster head. This process is shown in Block 1 of Fig. 6.11.

Subsequently, the cluster head performs the inner step optimization, as described by Block 2 of Fig. 6.11, where the routing table $S_{\text{MC}}^{\text{OPF}}$ containing the Pareto-optimal routes of all the active source and destination MC pairs, which are identified independently through a NDQIO sub-process, based on Alg. 6.1. The routes contained in $S_{\text{MC}}^{\text{OPF}}$ are then combined through several iterations for producing the S^{OPF} set, which contains the Pareto-optimal route-combinations in terms of the network's average delay and average power consumption. This process constitutes the outer step of the MODQO algorithm and it is denoted by Block 3 of Fig. 6.11 and it is detailed in Alg. 6.2. Finally, the MODQO algorithm outputs the identified OPF routing table S^{OPF} along with the solution $S_{\text{opt}} \in S^{\text{OPF}}$ that maximizes the normalized entropy of the normalized composite betweenness distribution. We note that we do not have to invoke a new search process for the latter; instead, during the last iteration of Alg. 6.2, the normalized entropy value of the normalized composite betweenness

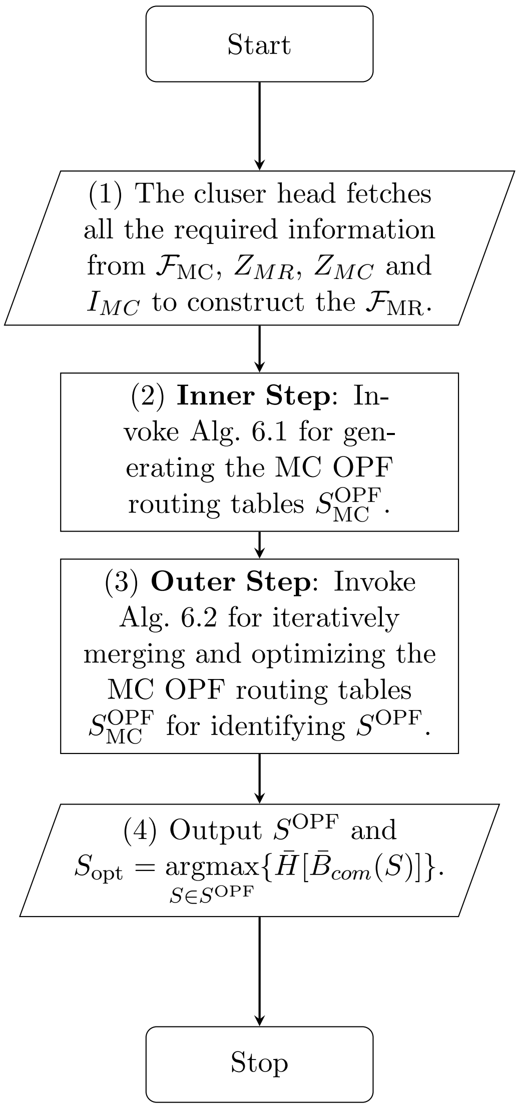

Start
(1) The cluser head fetches all the required information from $\mathcal{F}_{\mathrm{MC}}$, Z_{MR}, Z_{MC} and I_{MC} to construct the $\mathcal{F}_{\mathrm{MR}}$.
(2) **Inner Step**: Invoke Alg. 6.1 for generating the MC OPF routing tables $S_{\mathrm{MC}}^{\mathrm{OPF}}$.
(3) **Outer Step**: Invoke Alg. 6.2 for iteratively merging and optimizing the MC OPF routing tables $S_{\mathrm{MC}}^{\mathrm{OPF}}$ for identifying S^{OPF}.
(4) Output S^{OPF} and $S_{\mathrm{opt}} = \underset{S \in S^{\mathrm{OPF}}}{\mathrm{argmax}}\{\bar{H}[\bar{B}_{com}(S)]\}$.
Stop

Figure 6.11: Multi-Objective Decomposition Quantum Optimization (MODQO) algorithm flowchart. We note that the in parentheses numbers at the begining of each block correspond to the identification number of each block.

distribution of each S^{OPF} element is evaluated as long as the specific set of active routes is identified as Pareto-optimal. It is then compared to the maximum hitherto observed value and this value is updated, should the observed value be greater than the maximum value observed so far. This process imposes a further overhead in terms of complexity equal to the number $\left|S^{\mathrm{OPF}}\right|$ of elements comprising the OPF.

Having presented an overview of the MODQO algorithm let us now provide some further discussions regarding the inner (Block 2) and the outer (Block 3) steps in Subsections 6.3.3 and 6.3.4, respectively.

6.3.3 Building the MC Routing Tables

During the inner step, our design objective is to construct the routing table $S_{\mathrm{MC}}^{\mathrm{OPF}}$ for all the active pairs of source and destination MCs. The formal statement of the inner process introduced in Block 2 of Fig. 6.11 is presented in Alg. 6.1. We note that the n-th element $S^{\text{OPF}}_{\text{MC},n}$ of the routing table $S^{\text{OPF}}_{\text{MC}}$, which appears in Alg. 6.1, contains the Pareto-optimal routes of the n-th pair of source and destination MCs, while $S^{\text{OPF}}_{\text{MC},n,k}$ corresponds to the k-th identified Pareto-optimal route of $S^{\text{OPF}}_{\text{MC},n}$.

Algorithm 6.1 Inner Step of the Multi-Objective Decomposition Quantum Optimization (MODQO) algorithm, introduced in Block 2 of Fig. 6.11.

```
1: # Building MR Routing Table $S^{\text{OPF}}_{\text{MR}}$:
2: Set $S^{\text{OPF}}_{\text{MR}} \leftarrow [\,]$.
3: for $i = 0$ to $N_{\text{MR}}$ do
4:    for $j = i + 1$ to $N_{\text{MR}}$ do
5:       if $\mathcal{F}_{\text{MR},i,j} \neq 0$ then
6:          Invoke the NDQIO process of Alg. 5.1 with source node the MR$_i$ and destination
            node the MR$_j$ and store the OPF routes to $S^{\text{OPF}}_{\text{MR},i,j}$ and the reciprocal of these
            routes to $S^{\text{OPF}}_{\text{MR},j,i}$.
7:       end if
8:    end for
9: end for
10: # Building MC Routing Table $S^{\text{OPF}}_{\text{MC}}$:
11: Set $S^{\text{OPF}}_{\text{MC}} \leftarrow [\,]$ and $n \leftarrow 0$.
12: for $i = 0$ to $N_{\text{MC}}$ do
13:    for $j = i + 1$ to $N_{\text{MC}}$ do
14:       if $\mathcal{F}_{\text{MC},i,j} \neq 0$ then
15:          Set $n \leftarrow n + 1$, $l \leftarrow I_{\text{MC},i}$ and $m \leftarrow I_{\text{MC},j}$.
16:          if $l \neq m$ then
17:             for $k = 1$ to $\left|S^{\text{OPF}}_{\text{MR},i,j}\right|$ do
18:                Set $S^{\text{OPF}}_{\text{MC},n,k} = [\text{MC}_i, S^{\text{OPF}}_{\text{MR},l,m,k}, \text{MC}_j]$.
19:             end for
20:          else
21:             Set $S^{\text{OPF}}_{\text{MC},n} \leftarrow [\text{MC}_i, \text{MR}_l, \text{MC}_j]$.
22:          end if
23:       end if
24:    end for
25: end for
26: Export the $S^{\text{OPF}}_{\text{MC}}$ and exit.
```

Based on the assumption that the MRs operate in full-duplex with the aid of a sufficient number of orthogonal spreading codes and channels both in the frequency domain, we are ready to independently address each of the routing sub-problems for a specific pair of source and destination MCs. Explicitly, this specific assumption provides us with the upper bounds of the twin-layer network's achievable performance. Additionally, we can achieve a further reduction in the complexity imposed by the inner step, if we exploit the fact that each of the MCs is assigned to its closest MR. Based on this allocation scheme, different source MCs assigned to the same MR have identical sets of Pareto-optimal routes leading to the specific destination MCs that are also assigned to the same MR. Therefore, we can directly perform the routing optimization directly among the MRs based on their friendship matrix $\mathcal{F}_{\text{MR}}$, hence constructing the respective routing table $S^{\text{OPF}}_{\text{MR}}$. This process is performed in Steps 6.1.3-9, where the NDQIO process of [2, Alg. 4] is activated in Step 6.1.6, as long as the source MR_i and destination MR_j share a social relationship, i.e. we have $\mathcal{F}_{\mathrm{MR},i,j} \neq 0$. In this way, we have managed to reduce the number of NDQIO activations to $||\mathcal{F}_{\mathrm{MR}}||_2/2$ from $||\mathcal{F}_{\mathrm{MC}}||_2/2$, where the $||\cdot||_2$ operator corresponds to the power-2 norm of a matrix [193].

After successfully constructing the MR routing table $S_{\mathrm{MC}}^{\mathrm{OPF}}$, the cluster can readily construct the respective MC routing table $S_{\mathrm{MC}}^{\mathrm{OPF}}$ from $S_{\mathrm{MC}}^{\mathrm{OPF}}$, the active MC association vector I_{MC} to MRs and the MC friendship $\mathcal{F}_{\mathrm{MC}}$ in Steps 6.1.11-25. The exception of the source and destination MCs that are associated with the same MR is examined in Step 6.1.16. Naturally, should the source and destination MCs be associated with different MRs, the MODQO algorithm will rely on the MR routing table to construct the Pareto-optimal set of MC routes, as shown in Step 6.1.18. Otherwise, the two MCs are directly connected through their associated MR without the inclusion of intermediate MRs based on Step 6.1.21, since there exists no other valid Hamiltonian route that does not traverse the specific MR twice.

Finally, we note that should half-duplex and a finite number of either orthogonal codes or orthogonal channels be considered, our MODQO algorithm will still be applicable; however, a further constraint has to be imposed regarding the priority of each active route. This leads to a modified inner step, where the MC routing table is directly constructed, rather than exported from the respective MR routing table $S_{\mathrm{MR}}^{\mathrm{OPF}}$.

6.3.4 Merging the MC Routes

Having identified the Pareto-optimal routes $S_{\mathrm{MC},i}^{\mathrm{OPF}}$ for each individual active pair of source and destination MCs, we now have to combine the routes for identifying the network's Pareto-optimal sets of routes in terms of the network's average delay and its average power dissipation. Naturally, the conceptually simplest method of combining the individual routes would be to consider them jointly. Therefore, assuming N_r active pairs of source and destination MCs having on average $\bar{N}_{\mathrm{OPF}}$ Pareto-optimal routes, the resultant complexity imposed by the exhaustive search is equal to:

$$L_{\mathrm{ES}}^{\mathrm{outer}} = \bar{N}_{\mathrm{OPF}}^{2N_r}. \tag{6.39}$$

Observe in Eq. (6.39) that the complexity imposed by the exhaustive search increases exponentially, as the number of active routes increases. By contrast, when an NDQIO process is utilized, the resultant complexity quantified in terms of the number of CFEs will be on the order of $O(\bar{N}_{\mathrm{OPF}}^{N_r})$, owing to the complexity reduction offered by the QP, based on Eqs. (4.15), (5.35) and (5.53). Explicitly, this problem is NP-hard, since it belongs to the class of *Multi-Objective Knapsack Problems* [194]. Hence, for the sake of efficiently identifying the Pareto-optimal route-combinations, a heuristic approach has to be adopted.

In fact, the solution space formed by the combinations of the independent of the sub-problems Pareto-optimal routes can be portrayed as the *irregular trellis* diagram [195] of Fig. 6.12. Explicitly, an irregular trellis diagram is utilized, since the trellis paths can reach different total number of states at the n-th stage, which is denoted by $\left|S_{\mathrm{MC},n}^{\mathrm{OPF}}\right|$. Still

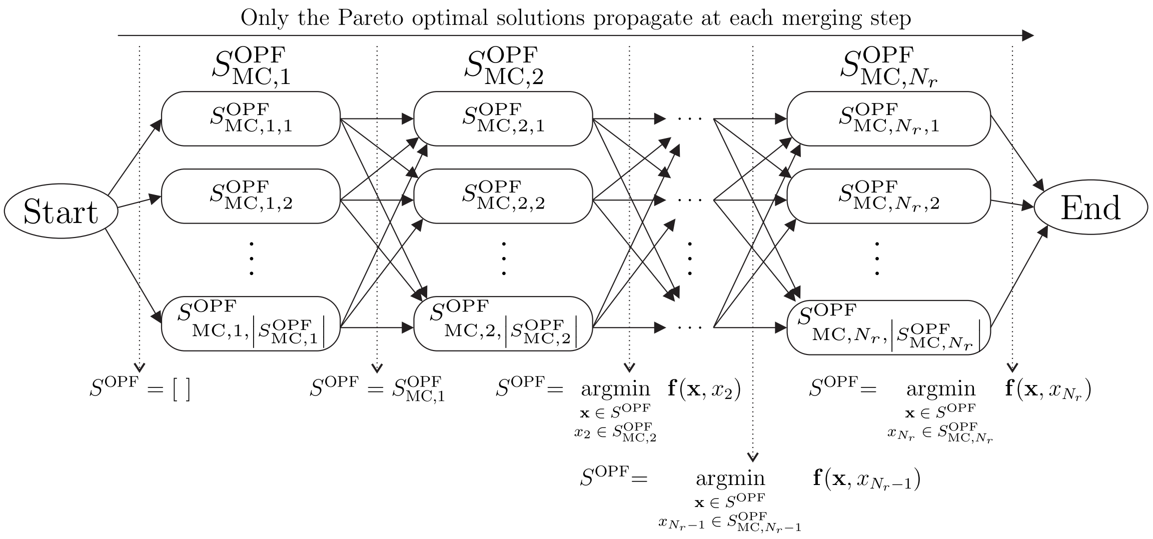

Only the Pareto optimal solutions propagate at each merging step
$S^{\text{OPF}}_{\text{MC},1}$
$S^{\text{OPF}}_{\text{MC},2}$
$S^{\text{OPF}}_{\text{MC},N_r}$
Start
End
$S^{\text{OPF}}_{\text{MC},1,1}$
$S^{\text{OPF}}_{\text{MC},1,2}$
$S^{\text{OPF}}_{\text{MC},1,\left|S^{\text{OPF}}_{\text{MC},1}\right|}$
$S^{\text{OPF}}_{\text{MC},2,1}$
$S^{\text{OPF}}_{\text{MC},2,2}$
$S^{\text{OPF}}_{\text{MC},2,\left|S^{\text{OPF}}_{\text{MC},2}\right|}$
$S^{\text{OPF}}_{\text{MC},N_r,1}$
$S^{\text{OPF}}_{\text{MC},N_r,2}$
$S^{\text{OPF}}_{\text{MC},N_r,\left|S^{\text{OPF}}_{\text{MC},N_r}\right|}$
$S^{\text{OPF}} = [\,]$
$S^{\text{OPF}} = S^{\text{OPF}}_{\text{MC},1}$
$S^{\text{OPF}} = \underset{\substack{\mathbf{x} \in S^{\text{OPF}} \\ x_2 \in S^{\text{OPF}}_{\text{MC},2}}}{\text{argmin}} \; \mathbf{f}(\mathbf{x}, x_2)$
$S^{\text{OPF}} = \underset{\substack{\mathbf{x} \in S^{\text{OPF}} \\ x_{N_r-1} \in S^{\text{OPF}}_{\text{MC},N_r-1}}}{\text{argmin}} \; \mathbf{f}(\mathbf{x}, x_{N_r-1})$
$S^{\text{OPF}} = \underset{\substack{\mathbf{x} \in S^{\text{OPF}} \\ x_{N_r} \in S^{\text{OPF}}_{\text{MC},N_r}}}{\text{argmin}} \; \mathbf{f}(\mathbf{x}, x_{N_r})$

Figure 6.12: Trellis diagram of the merging process solely using the CM method.or the outer step optimization. We note that at each stage only $\left|S^{\text{OPF}}\right|$ routes continue propagating towards the End node, as denoted by the dotted arrows, while traversing precisely $\left|S^{\text{OPF}}_{\text{MC},i}\right|$ nodes at the i-th step.

referring to Fig. 6.12, a specific trellis path representing a particular route-combination is formed once a trellis transition traverses a specific trellis node, which correspods to a specific route. For instance, at the 3rd stage of Fig. 6.12 a trellis path visiting the trellis nodes $S^{\text{OPF}}_{\text{MC},1,2}$, $S^{\text{OPF}}_{\text{MC},2,1}$ and $S^{\text{OPF}}_{\text{MC},3,5}$ represents the particular route-combination formed by the second identified Pareto-optimal route of the first active pair of source and destination MCs, the first identified Pareto-optimal route of the second active pair of source and destination MCs and the fifth identified Pareto-optimal route of the third active pair of source and destination MCs.

Owing to the irregular trellis structure of the outer problem, we may readily employ the classic *Viterbi Algorithm* [196]. However, we have to modify it so that it becomes applicable for multi-objective *Cost Functions* (CF), since it has been initially designed for single-objective CFs in the context of decoding *Forward Error Correction* (FEC) schemes [97]. Explicitly, we have proven in Proposition 1 that a route-combination is potentially Pareto-optimal, when all of the routes comprising the route-combination are Pareto-optimal in their respective individual sub-problems. Naturally, due to the specific form of the utility functions of the UV defined in Eq. (6.31) it is possible to group n sub-problems into a smaller composite sub-problem $\mathbf{x}$, which is defined as follows:

$$\mathbf{x} = \left[x^{(1)}, \ldots, x^{(n)}\right], \tag{6.40}$$

while its k-th respective UF is given by:

$$f_k(\mathbf{x}) = \sum_{i=1}^{n} a_{k,n} f_k(x^{(i)}). \tag{6.41}$$

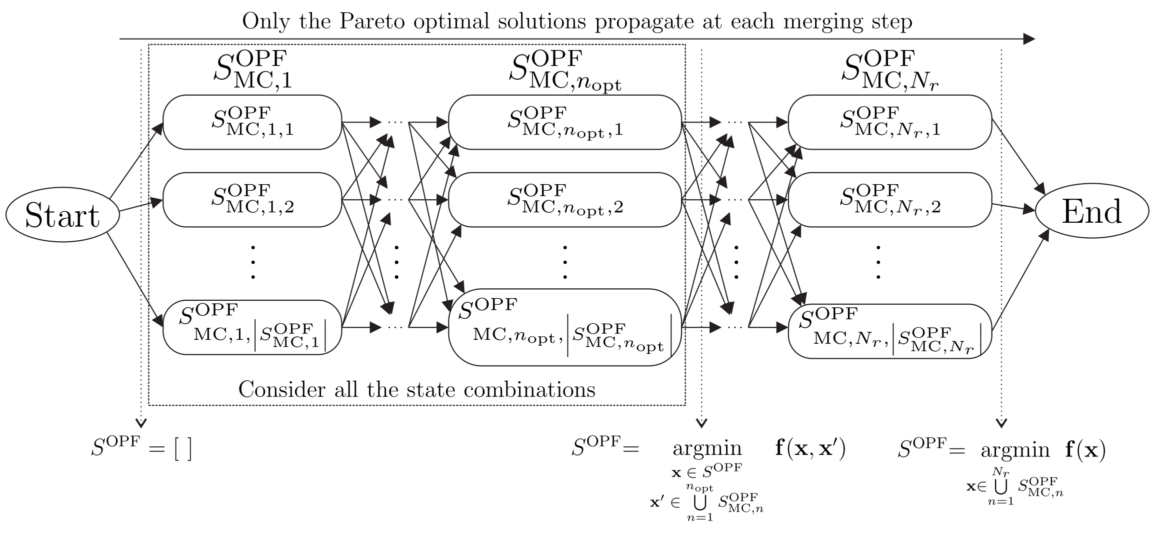

Only the Pareto optimal solutions propagate at each merging step
$S^{\text{OPF}}_{\text{MC},1}$
$S^{\text{OPF}}_{\text{MC},n_{\text{opt}}}$
$S^{\text{OPF}}_{\text{MC},N_r}$
Start
End
Consider all the state combinations
$S^{\text{OPF}} = [\,]$
$S^{\text{OPF}} = \operatorname{argmin}_{\mathbf{x} \in S^{\text{OPF}},\ \mathbf{x}' \in \bigcup_{n=1}^{n_{\text{opt}}} S^{\text{OPF}}_{\text{MC},n}} \mathbf{f}(\mathbf{x},\mathbf{x}')$
$S^{\text{OPF}} = \operatorname{argmin}_{\mathbf{x} \in \bigcup_{n=1}^{N_r} S^{\text{OPF}}_{\text{MC},n}} \mathbf{f}(\mathbf{x})$

Figure 6.13: Trellis diagram of the merging process solely using the QM method for the outer step optimization. In the same fashion as in Fig. 6.12, only $|S^{\text{OPF}}|$ routes continue propagating at the next trellis stage, as denoted by the dotted arrows, while traversing precisely $|S^{\text{OPF}}_{\text{MC},i}|$ nodes at the i-th step.

Based on Eq. (6.40) the composite problem's solution S is now defined as follows:

$$S = \left[\mathbf{x}, x^{(n+1)}, \ldots, x^{(N_r)}\right], \tag{6.42}$$

and based on Eq. (6.41), the route-combination S has a k-th UF that attains the form of Eq. (6.31). This is the critical condition for Proposition 1 to be valid and, hence, we have:

$$S^{\text{OPF}} \subseteq S^{\text{OPF}}_{(n)} \cup \left(\bigcup_{i=n+1}^{N_r} S^{\text{OPF}}_{\text{MC},i} \right), \ \forall n, N_r \in \mathbb{N}^*, \tag{6.43}$$

where we have $n \leq N_r$ and $S^{\text{OPF}}_{(n)}$ corresponds to the OPF of the Pareto-optimality routing sub-problem with route solutions having the form $\mathbf{x} = \left[x^{(1)}, \ldots, x^{(n)}\right]$. Naturally, if we have $n = N_r$, Eq. (6.43) is reduced to $S^{\text{OPF}} = S^{\text{OPF}}_{(N_r)}$, while if we set $N_r = n + 1$, Eq. (6.43) is transformed into the following recursive closed form:

$$S^{\text{OPF}}_{(n+1)} \subseteq S^{\text{OPF}}_{(n)} \cup S^{\text{OPF}}_{\text{MC},n+1}, \tag{6.44}$$

where the equality of the sets is satisfied, as long as the set $S^{\text{OPF}}_{\text{MC},n+1}$ consists of a single Pareto-optimal route, i.e. we have $\left|S^{\text{OPF}}_{\text{MC},n+1}\right| = 1$.

Explicitly, Eq. (6.44) provides us with a reduced-complexity optimal merging method for the sake of iteratively combining the OPF $S^{\text{OPF}}_{\text{MC},n}$ of each stage to form the network's overall OPF S^{OPF}. To elaborate further, at the n-th horizontal step or stage of Fig. 6.12 we only have to consider the arrival of hitherto Pareto-optimal route-combinations for the previous $(n-1)$ stages, i.e. the route-combinations belonging to the set $S^{\text{OPF}}_{(n-1)}$. Subsequently, these routes visit each of the nodes of the n-th stage, thus constructing the set $S^{\text{OPF}}_{(n-1)} \cup S^{\text{OPF}}_{\text{MC},n}$ of route-combinations. Since the OPF of the n-th stage route-combinations $S^{\text{OPF}}_{(n-1)}$ is a subset of $S^{\text{OPF}}_{(n-1)} \cup S^{\text{OPF}}_{\text{MC},n}$ we will have to identify the new OPF for the sake of discarding the sub-optimal routes and, thus, confining the search space for the sake of reducing the complexity imposed without degrading the associated accuracy. As far as the initialization of the OPF of route-combinations is concerned, it is set to $S_{(0)}^{\text{OPF}} = [\,]$, which represents an empty set. The aforementioned iterative process is repeated until the route-combinations have encapsulated routes from all the active pairs of source and destination MCs. The intermediate products of this iterative process are portrayed in Fig. 6.12 with the aid of dotted arrows; these arrows demonstrate the routes that horizontally propagate at the next stage.

In a nutshell, this iterative process resembles the classic Viterbi Algorithm [196, 197], since they both attempt to reduce the search space using a series of iterations. The most notable difference in our approach compared to the classic Viterbi Algorithm is that the number of propagating routes in our scenario is variable and it depends on the number $\left|S_{(n)}^{\text{OPF}}\right|$ of the Pareto-optimal route-combinations at the n-th stage. Therefore, the complexity of this method quantified in both domains, which will be referred to from now on as *Classical Merging* (CM) method, is bounded by:

$$L_{\text{CM,min}}^{\text{outer}} = (N_r - 1)\bar{N}_{\text{OPF}}^4 = O(N_r \bar{N}_{\text{OPF}}^4), \tag{6.45}$$

$$L_{\text{CM,max}}^{\text{outer}} = \sum_{n=2}^{N_r} \bar{N}_{\text{OPF}}^{2n} = \frac{\bar{N}_{\text{OPF}}^{2(N_r+2)} - \bar{N}_{\text{OPF}}^4}{\bar{N}_{\text{OPF}}^2 - 1} = O(\bar{N}_{\text{OPF}}^{2(N_r+1)}), \tag{6.46}$$

where $\bar{N}_{\text{OPF}}$ corresponds to the average number of OPF in the independent sub-problems, while $L_{\text{CM,min}}^{\text{outer}}$ and $L_{\text{CM,max}}^{\text{outer}}$ are the lower and upper bounds of the CM method. The lower bound of Eq. (6.45) corresponds to the best-case scenario, where the number of route-combinations that propagate across the irregular trellis stages is equal to the number of the respective states at the specifc trellis stage, i.e. equal to $\left|S_{\text{MC,n}}^{\text{OPF}}\right|$ for the n-th stage, while the upper bound of Eq. (6.46) is encountered in the worst-case scenario, where all the possible route-combinations are Pareto-optimal. Observe in Eqs. (6.45) and (6.46) that although the CM method imposes a substantially lower amount of complexity in the best-case scenario than the exhaustive search, in the worst-case scenario the CM method actually imposes a higher complexity, owing to the design assumption of having sub-optimal route-combinations that will be eliminated during the intermediate stages. In the next section we will provide the critical condition for the CM to outperform the exhaustive search.

Furthermore, we can readily empower the CM method with the quantum-aided framework of Section 5.2, which was invoked in the inner step, for the sake of achieving a further reduction in the parallel complexity imposed by the outer step. A naive approach would be to invoke the NDQIO algorithm at each stage; however, this approach would only be efficient for a high number of route-combinations, whilst its potential application would impose a higher parallel complexity than the convetional CM method. For the sake of circumventing this problem, we have introduced an additional degree of freedom for the NDQIO process. In particular, we have designed the NDQIO sub-process to be capable of jointly considering multiple stages of the irregular trellis, set to an optimal number of stages n_{opt}, as it is portrayed in Fig. 6.13. We note that from this point on this process will be referred to as the *Quantum Merging* (QM) process. Observe in Fig. 6.13 that the NDQIO process jointly considers the route-combinations formed by the OPF sets $\left\{S_{\text{MC},n}^{\text{OPF}}\right\}_{n=1}^{n_{\text{opt}}}$, as marked by the dashed-line-bordered rectangle.

Based on this, we have to define the selection criterion regarding the value of the parameter n_{opt}, which would allow the NDQIO sub-process to impose a lower parallel complexity than the CM method. For this reason, let us first define a metric for quantifying the complexity imposed by the CM. Explicitly, for accurately quantifying the complexity imposed by the CM method, the number of Pareto-optimal route-combinations has to be known prior to the optimization. Since the optimization process is incapable of estimating the number of Pareto-optimal route-combinations at each stage without actually solving the respective Pareto-optimality sub-problems, we will attempt to approximate the complexity imposed by the CM method using its lower bound. Hence, we set the number of Pareto-optimal route-combinations at each stage equal to the number of nodes of that stage. Based on this assumption, the reference complexity $L_{\text{CM}}^{\text{ref}}$ imposed by the CM method quantified in terms of the trellis stage index n is equal to:

$$L_{\text{CM}}^{\text{ref}}(n) = \sum_{n=2}^{\left|S_{\text{MC}}^{\text{OPF}}\right|} \left|S_{\text{MC},n-1}^{\text{OPF}}\right|^2 \left|S_{\text{MC},n}^{\text{OPF}}\right|^2. \tag{6.47}$$

Based on the same assumption and Eq. (5.53), we are capable of deriving the upper and lower bounds of the parallel complexity imposed by the NDQIO algorithm, when n stages are considered jointly. Explicitly, the aforementioned bounds of parallel complexity quantified in terms of the number of CFEs are equal to:

$$\begin{aligned} L_{\text{QM,min}}^{\text{ref}}(n) = \; & (2K)^{-1}\left|S_{\text{MC},n}^{\text{OPF}}\right|^2 + K^{-1}\left\{L_{\text{DHA}}^{\min}[\hat{N}(n)] + \right. \\ & \left. +2L_{\text{BBHT}}^{\min}[\hat{N}(n)] - \tfrac{1}{2}\right\}\left|S_{\text{MC},n}^{\text{OPF}}\right| + \\ & +(1-K)\left\{\tfrac{2}{K}L_{\text{BBHT}}^{\min}[\hat{N}(n)] + \tfrac{1}{2}\right\}, \end{aligned} \tag{6.48}$$

$$\begin{aligned} L_{\text{QM,max}}^{\text{ref}}(n) = \; & (2K)^{-1}\left|S_{\text{MC},n}^{\text{OPF}}\right|^2 + K^{-1}\left\{L_{\text{DHA}}^{\max}[\hat{N}(n)] + \right. \\ & \left. +2L_{\text{BBHT}}^{\max}[\hat{N}(n)] - \tfrac{1}{2}\right\}\left|S_{\text{MC},n}^{\text{OPF}}\right| + \\ & +(1-K)\left\{\tfrac{2}{K}L_{\text{BBHT}}^{\max}[\hat{N}(n)] + \tfrac{1}{2}\right\}, \end{aligned} \tag{6.49}$$

where $L_{\text{BBHT}}^{\min}[\hat{N}(n)]$ and $L_{\text{BBHT}}^{\max}[\hat{N}(n)]$ correspond to the lower and upper bounds of the complexity imposed by the BBHT-QSA for a database of $\hat{N}(n)$ elements, as defined in Eqs. (4.15) and (4.16), respectively, whereas $L_{\text{DHA}}^{\min}[\hat{N}(n)]$ and $L_{\text{DHA}}^{\max}[\hat{N}(n)]$ are the respective complexity bounds of the DHA, which are defined in Eqs. (5.26) and (5.35), respectively. Additionally, recall that the parameter K represents the number of utility functions considered, which is set equal to $K = 2$ utility functions in our scenario. We note that the database length $\hat{N}(n)$ is quantified in terms of the number n of joint stages considered by the NDQIO algorithm as follows:

$$\hat{N}(n) = \prod_{i=1}^{n} |S_{\mathrm{MC},i}^{\mathrm{OPF}}|. \tag{6.50}$$

We note that we are using the upper and lower bounds of complexity for the NDQIO algorithm due to the stochastic nature of both the DHA and of the BBHT-QSA. Having derived these bounds, we may now assess the criteria for an efficient deployment of the NDQIO algorithm. Our ultimate design target is to determine at each iteration the optimal number n_{opt} of joint stages for ensuring that the NDQIO algorithm outperforms the CM method, while imposing the lowest possible complexity. Therefore, the optimal value n_{opt} occurs at the specific trellis stage, where the lower bound of the complexity reduction with respect to the CM is maximized. This can be formulated as follows:

$$n_{\mathrm{opt}} = \underset{n\in\{1,\ldots,|S_{\mathrm{MC}}^{\mathrm{OPF}}|\}}{\arg\max} \left\{ \frac{L_{\mathrm{CM}}^{\mathrm{ref}}(n)}{L_{\mathrm{QM,max}}^{\mathrm{ref}}(n)} \right\}. \tag{6.51}$$

Nonetheless, we should also ensure that the NDQIO process achieves a beneficial complexity reduction compared to the partial CM method, for this optimal value of n_{opt}. Explicitly, this can be verified by comparing the lower bound $L_{\mathrm{QM,max}}^{\mathrm{ref}}(n_{\mathrm{opt}})$ of the complexity imposed by the NDQIO algorithm to the respective value $L_{\mathrm{CM}}^{\mathrm{ref}}(n_{\mathrm{opt}})$ of the CM method: should the NDQIO lower bound for the specific value of n_{opt} derived by Eq. (6.51) be lower than that of the CM method, then we may conclude that the NDQIO process is potentially capable of outperforming the CM method. Otherwise, it is clear that the CM method should be invoked. At this point, we would like to point out that we have opted for utilizing the lower bound of the NDQIO's average complexity, since it tends to be closer to average case, when compared to its upper bound counterpart, as presented in [2].

Algorithm 6.2 Outer Step of the Multi-Objective Decomposition Quantum Optimization (MODQO) algorithm, introduced in Block 3 of Fig. 6.11.

```
1: Sort $S_{\mathrm{MC}}^{\mathrm{OPF}}$ in terms of their number of OPF routes in ascending order.
2: Initialize $S^{\mathrm{OPF}} \leftarrow S_{\mathrm{MC},1}^{\mathrm{OPF}}$ and remove $S_{\mathrm{MC},1}^{\mathrm{OPF}}$ from $S_{\mathrm{MC}}^{\mathrm{OPF}}$.
3: repeat
4:   Calculate $L_{\mathrm{CM}}(n)$, $L_{\mathrm{QM,min}}^{\mathrm{ref}}(n)$ and $L_{\mathrm{QM,max}}^{\mathrm{ref}}(n)$, defined in (6.47), (6.48) and (6.49)
     respectively, $\forall n \in \{1,\ldots,|S_{\mathrm{MC}}^{\mathrm{OPF}}|\}$.
5:   Evaluate $n_{\mathrm{opt}}$ from Eq (6.51).
6:   if $L_{\mathrm{QM\,min}}^{\mathrm{ref}}(n_{\mathrm{opt}}) < L_{\mathrm{CM}}^{\mathrm{ref}}(n_{\mathrm{opt}})$ then
7:     Consider jointly the solutions formed by merging $S^{\mathrm{OPF}}$ and $\{S_{\mathrm{MC},n}^{\mathrm{OPF}}\}_{n=1}^{n_{\mathrm{opt}}}$ and acti-
       vate a NDQIO process of Alg. 5.1 storing the identified OPF to $S^{\mathrm{OPF}}$ and removing
       the elements $\{S_{\mathrm{MC},n}^{\mathrm{OPF}}\}_{n=1}^{n_{\mathrm{opt}}}$ from $S_{\mathrm{MC}}^{\mathrm{OPF}}$.
8:   else
9:     Consider jointly the solutions formed by merging $S^{\mathrm{OPF}}$ and $S_{\mathrm{MC},1}^{\mathrm{OPF}}$ and store the
       identified OPF by exhaustive search to $S^{\mathrm{OPF}}$, removing $S_{\mathrm{MC},1}^{\mathrm{OPF}}$ from $S_{\mathrm{MC}}^{\mathrm{OPF}}$.
10:  end if
11: until $|S_{\mathrm{MC}}^{\mathrm{OPF}}| > 0$.
12: Export the $S^{\mathrm{OPF}}$ and exit.
```

Based on this framework, which relies on a synergistic hybrid of classical and quantum optimization, let us now describe the outer step of the MODQO algorithm, which is formally presented in Alg 6.2. Prior to constructing the irregular trellis structure of Fig. 6.13, we have to sort the MC Pareto-optimal routing tables $\{S_{\text{MC},n}^{\text{OPF}}\}_{n=1}^{N_r}$ based on their number $\left|S_{\text{MC},n}^{\text{OPF}}\right|$ of the independent Pareto-optimal routes, as it is shown in Step 6.2.1. This can be justified by the fact that the number of route-combinations will eventually increase, as more stages of the trellis diagram are examined. Naturally, this sorting is consistent with the design assumption that the number of Pareto-optimal routes at the end of each stage is equal to the number of the specific states at that trellis stage. Subsequently, the set of Pareto-optimal route-combinations S^{OPF} is initialized to the first set $S_{\text{MC},1}^{\text{OPF}}$ of Pareto-optimal routes in the MC routing table, which is then removed from $S_{\text{MC}}^{\text{OPF}}$ in Step 6.2.2 followed by the iterative process of Steps 6.2.3-11. This iterative process is invoked for the sake of determining as to whether a single NDQIO activation is capable of imposing a lower complexity than the CM method for the same number of trellis stages. Hence, the more efficient method is activated at each iteration. To elaborate further, the respective reference complexities are quantified in terms of the trellis stage index n and the its optimal value n_{opt} with respect to the maximum complexity reduction achieved by the QM over the CM method in Steps 6.2.4-5, respectively. Should the algorithm conclude that the QM method is potentially capable of operating at a lower complexity than the CM, n_{opt} stages are considered jointly with the Pareto-optimal route-combinations S^{OPF} gleaned from the previous iterations in Step 6.2.7; it then updates the OPF route-combinations S^{OPF} and removes the encapsulated stages from the MC routing table. Otherwise, a single CM iteration is applied, i.e. the first set $S_{\text{MC},1}^{\text{OPF}}$ of the MC routing table is jointly considered with the S^{OPF} gleaned from the previous iterations and Alg. 6.2 removes the set $S_{\text{MC},1}^{\text{OPF}}$ from the MC routing table after updating the OPF of route-combinations S^{OPF} in Step 6.2.9. We note that a single CM is applied, rather than n_{opt} for the sake of allowing the algorithm to iteratively calibrate itself for all the potential outcomes of all Pareto-optimal route-combinations.

For the sake of simplicity, we have not included the specific process, appearing in Block 4 of Fig. 6.11, which identifies the specific Pareto-optimal route-combination S_{opt} exhibiting the maximum value of the normalized entropy of the normalized composite betweenness. In fact, this specific process can be incorporated during the last iteration of Alg. 6.2, where the normalized entropy $\bar{H}[\bar{B}_{com}(S)]$ of the normalized composite betweenness is evaluated, when a route-combination is identified as being Pareto-optimal. This value of $\bar{H}[\bar{B}_{com}(S)]$ is then compared to the maximum observed $\bar{H}_{\max}$ value, which is updated depending on the comparison outcome. In this way, we can reach the optimal route-combination S_{opt} with respect to the optimization problem of Eq. (6.22) by imposing a modest overhead of $\left|S^{\text{OPF}}\right|$ CFEs.

6.3.5 A 12-Node Tutorial Example using MODQO

Having fully described the MODQO algorithm's the inner and the outer sub-processes, let us now elaborate on its function with the aid of a low-paced tutorial example. We note that the following tutorial assumes knowledge of the NDQIO algorithm. Therefore, the readers new to this subject should refer to the tutorial section of [2]. Additionally, we will assume an identical twin-layer network to that of the tutorial using the Exhaustive Search presented in Sec. A.4.

Based on the friendship matrix $\mathcal{F}_{\text{MC}}$ of Eq. (6.24), the MCs' locations Z_{MC}, the MRs' locations Z_{MR} and the MCs to MRs association vector I_{MC}, which are shown in Fig. 6.7, the cluster head invokes the MODQO algorithm seen in Fig. 6.11. Firstly, based on Block 1 of the MODQO flowchart seen in Fig. 6.11 the MR friendship matrix $\mathcal{F}_{\text{MR}}$ is constructed, whose elements indicate which pairs of MRs are associated with MCs that share a social relationship. In our scenario, the MR friendship matrix $\mathcal{F}_{\text{MR}}$ is equal to:

$$\mathcal{F}_{\text{MR}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{6.52}$$

where we can observe that the MR_1 and MR_4 share no social relationship with the rest of the MRs, since they are not associated with any of the MCs. To elaborate further, MR_2 and MR_7 shares a social relationship only with MC_3, MR_3 in turn shares a friendship relationship with MR_2, MR_5, MR_6 and MR_7, since it is associated with the social by-minded MC_1, while MR_5 and MR_6 share a social relationship both with each other and with MR_3. The MRs' social relationship is visually portrayed in Fig. 6.14.

After the construction of the MR friendship matrix $\mathcal{F}_{\text{MR}}$, the MODQO algorithm proceeds with its inner step, according to Block 2 of the MOQDO flowchart seen in Fig. 6.11. In this step, the MODQO algorithm initializes the MR routing table to an empty matrix based on Step 6.1.2. It then acts on the upper triangular part of the $\mathcal{F}_{\text{MR}}$ matrix by activating a NDQIO sub-process in Steps 6.1.3-9 for the sake of identifying the entire set of Pareto-optimal routes between MR_i and MR_j as long as $\mathcal{F}_{\text{MR},i,j} = 1$, i.e. MR_i and MR_j have a social relationship. We note that the NDQIO sub-process activates the weak Pareto dominance operator U_{g_w} portrayed in Fig. 6.10. Based on Eq. (6.52) for our scenario, the NDQIO sub-process will be activated $||\mathcal{F}_{\text{MR}}||_2/2 = 5$ times, namely for the elements $\mathcal{F}_{\text{MR},2,3}$, $\mathcal{F}_{\text{MR},3,5}$, $\mathcal{F}_{\text{MR},3,6}$, $\mathcal{F}_{\text{MR},3,7}$ and $\mathcal{F}_{\text{MR},5,6}$, thus exporting the sets of Pareto-optimal routes $S^{\text{OPF}}_{\text{MR},2,3}$, $S^{\text{OPF}}_{\text{MR},3,5}$, $S^{\text{OPF}}_{\text{MR},3,6}$, $S^{\text{OPF}}_{\text{MR},3,7}$ and $S^{\text{OPF}}_{\text{MR},5,6}$, respectively. The aforementioned OPFs constituting the MR routing table along with their associated utility functions, namely their delay $D(x)$ quantified in terms of the number of established hops and their power consumption $P(x)$ quantified in dBm, are presented in Table 6.4. Recall that the notation $S^{\text{OPF}}_{\text{MR},i,j,k}$ of Table 6.4 represents the k-th Pareto-optimal route identified

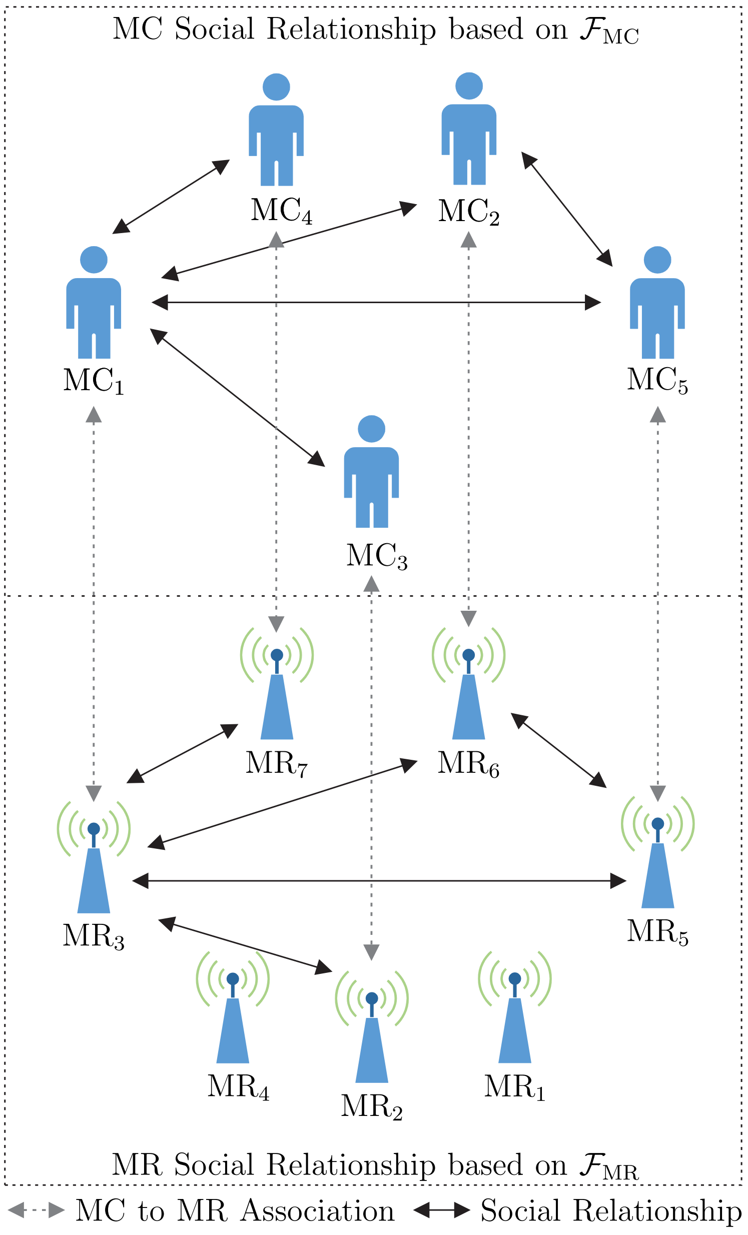

MC Social Relationship based on $\mathcal{F}_{\text{MC}}$
MC$_4$
MC$_2$
MC$_1$
MC$_5$
MC$_3$
MR$_7$
MR$_6$
MR$_3$
MR$_5$
MR$_4$
MR$_2$
MR$_1$
MR Social Relationship based on $\mathcal{F}_{\text{MR}}$
MC to MR Association
Social Relationship

Figure 6.14: Visual representation of the MRs' social relationship, based on Eq. (6.24). Note that the gray arrows indicate the association between a specific MC and a specific MR.

by the NDQIO sub-process for producing the OPF $S^{\text{OPF}}_{\text{MR},i,j}$, which in turn corresponds to the set of Pareto-optimal routes spanning from MR$_i$ to MR$_j$ and vice versa.

Subsequently, the inner process of the MODQO algorithm builds up the respective MC routing table $S^{\text{OPF}}_{\text{MC}}$ based on the MR routing table $S^{\text{OPF}}_{\text{MR}}$ of Table 6.4, on the MC to MR association vector I_{MC} and on the MC friendship matrix $\mathcal{F}_{\text{MR}}$, which in our scenario is quantified in Eq. (6.24). We note that in the context of this current treatise, we have assumed that only the routes spanning from a specific MC$_i$ to another specific MC$_j$ are active during a transmission period, but not the reverse of these routes. The latter are considered to be activated during the next transmission period, when the first ones are inactive. In our scenario, the set S^{act} of active source and destination pairs considered is

Table 6.4: MR routing table $S_{\text{MR}}^{\text{OPF}}$ exported by Alg. 6.1.

ID	Route, x	$D(x)$	$P(x)$
	$S_{\text{MR},2,3}^{\text{OPF}}$		
$S_{\text{MR},2,3,1}^{\text{OPF}}$	$\text{MR}_2 \to \text{MR}_1 \to \text{MR}_3$	2.00	16.55
$S_{\text{MR},2,3,2}^{\text{OPF}}$	$\text{MR}_2 \to \text{MR}_5 \to \text{MR}_1 \to \text{MR}_4 \to \text{MR}_6 \to \text{MR}_3$	5.00	14.90
$S_{\text{MR},2,3,3}^{\text{OPF}}$	$\text{MR}_2 \to \text{MR}_5 \to \text{MR}_1 \to \text{MR}_6 \to \text{MR}_3$	4.00	15.65
$S_{\text{MR},2,3,4}^{\text{OPF}}$	$\text{MR}_2 \to \text{MR}_5 \to \text{MR}_1 \to \text{MR}_3$	3.00	15.68
	$S_{\text{MR},3,5}^{\text{OPF}}$		
$S_{\text{MR},3,5,1}^{\text{OPF}}$	$\text{MR}_3 \to \text{MR}_5$	1.00	16.08
$S_{\text{MR},3,5,2}^{\text{OPF}}$	$\text{MR}_3 \to \text{MR}_6 \to \text{MR}_4 \to \text{MR}_1 \to \text{MR}_5$	4.00	13.96
$S_{\text{MR},3,5,3}^{\text{OPF}}$	$\text{MR}_3 \to \text{MR}_6 \to \text{MR}_1 \to \text{MR}_5$	3.00	14.88
$S_{\text{MR},3,5,4}^{\text{OPF}}$	$\text{MR}_3 \to \text{MR}_1 \to \text{MR}_5$	2.00	14.92
	$S_{\text{MR},3,6}^{\text{OPF}}$		
$S_{\text{MR},3,6,1}^{\text{OPF}}$	$\text{MR}_3 \to \text{MR}_6$	1.00	2.48
	$S_{\text{MR},3,7}^{\text{OPF}}$		
$S_{\text{MR},3,7,1}^{\text{OPF}}$	$\text{MR}_3 \to \text{MR}_7$	1.00	17.24
$S_{\text{MR},3,7,2}^{\text{OPF}}$	$\text{MR}_3 \to \text{MR}_6 \to \text{MR}_4 \to \text{MR}_7$	3.00	9.25
$S_{\text{MR},3,7,3}^{\text{OPF}}$	$\text{MR}_3 \to \text{MR}_6 \to \text{MR}_7$	2.00	13.48
	$S_{\text{MR},5,6}^{\text{OPF}}$		
$S_{\text{MR},5,6,1}^{\text{OPF}}$	$\text{MR}_5 \to \text{MR}_6$	1.00	16.53
$S_{\text{MR},5,6,2}^{\text{OPF}}$	$\text{MR}_5 \to \text{MR}_1 \to \text{MR}_4 \to \text{MR}_6$	3.00	13.64
$S_{\text{MR},5,6,3}^{\text{OPF}}$	$\text{MR}_5 \to \text{MR}_1 \to \text{MR}_6$	2.00	14.62

equal to:

$$S^{\text{act}} = \left\{ \begin{array}{c} \text{MC}_2 \to \text{MC}_1 \\ \text{MC}_1 \to \text{MC}_3 \\ \text{MC}_4 \to \text{MC}_1 \\ \text{MC}_5 \to \text{MC}_1 \\ \text{MC}_2 \to \text{MC}_5 \end{array} \right\}, \tag{6.53}$$

where we can observe that the number N_r of active pairs of source and destination MCs is equal to:

$$N_r = \frac{||\mathcal{F}_{\text{MC}}||_2}{2} = \left|S^{\text{act}}\right| = 5. \tag{6.54}$$

Based on the active set S^{act} defined in Eq (6.53) and on the association vector I_{MC}, which is portrayed with the aid of the dashed lines in Fig. 6.7, the MODQO algorithm

Table 6.5: MC routing table $S_{\mathrm{MC}}^{\mathrm{OPF}}$ after Step 6.2.1.

ID	Route, x	$D(x)$	$P(x)$
	$S_{\mathrm{MC},1}^{\mathrm{OPF}}$		
$S_{\mathrm{MC},1,1}^{\mathrm{OPF}}$	$\mathrm{MC}_2 \rightarrow \mathrm{MR}_6 \rightarrow \mathrm{MR}_3 \rightarrow \mathrm{MC}_1$	3.00	7.68
	$S_{\mathrm{MC},2}^{\mathrm{OPF}}$		
$S_{\mathrm{MC},2,1}^{\mathrm{OPF}}$	$\mathrm{MC}_4 \rightarrow \mathrm{MR}_7 \rightarrow \mathrm{MR}_3 \rightarrow \mathrm{MC}_1$	3.00	17.56
$S_{\mathrm{MC},2,2}^{\mathrm{OPF}}$	$\mathrm{MC}_4 \rightarrow \mathrm{MR}_7 \rightarrow \mathrm{MR}_4 \rightarrow \mathrm{MR}_6 \rightarrow \mathrm{MR}_3 \rightarrow \mathrm{MC}_1$	5.00	10.95
$S_{\mathrm{MC},2,3}^{\mathrm{OPF}}$	$\mathrm{MC}_4 \rightarrow \mathrm{MR}_7 \rightarrow \mathrm{MR}_6 \rightarrow \mathrm{MR}_3 \rightarrow \mathrm{MC}_1$	4.00	14.20
	$S_{\mathrm{MC},3}^{\mathrm{OPF}}$		
$S_{\mathrm{MC},3,1}^{\mathrm{OPF}}$	$\mathrm{MC}_2 \rightarrow \mathrm{MR}_6 \rightarrow \mathrm{MR}_5 \rightarrow \mathrm{MC}_5$	3.00	17.61
$S_{\mathrm{MC},3,2}^{\mathrm{OPF}}$	$\mathrm{MC}_2 \rightarrow \mathrm{MR}_6 \rightarrow \mathrm{MR}_4 \rightarrow \mathrm{MR}_1 \rightarrow \mathrm{MR}_5 \rightarrow \mathrm{MC}_5$	5.00	15.54
$S_{\mathrm{MC},3,3}^{\mathrm{OPF}}$	$\mathrm{MC}_2 \rightarrow \mathrm{MR}_6 \rightarrow \mathrm{MR}_1 \rightarrow \mathrm{MR}_5 \rightarrow \mathrm{MC}_5$	4.00	16.20
	$S_{\mathrm{MC},4}^{\mathrm{OPF}}$		
$S_{\mathrm{MC},4,1}^{\mathrm{OPF}}$	$\mathrm{MC}_1 \rightarrow \mathrm{MR}_3 \rightarrow \mathrm{MR}_1 \rightarrow \mathrm{MR}_2 \rightarrow \mathrm{MC}_3$	4.00	17.19
$S_{\mathrm{MC},4,2}^{\mathrm{OPF}}$	$\mathrm{MC}_1 \rightarrow \mathrm{MR}_3 \rightarrow \mathrm{MR}_6 \rightarrow \mathrm{MR}_4 \rightarrow \mathrm{MR}_1 \rightarrow \mathrm{MR}_5 \rightarrow \mathrm{MR}_2 \rightarrow \mathrm{MC}_3$	7.00	15.80
$S_{\mathrm{MC},4,3}^{\mathrm{OPF}}$	$\mathrm{MC}_1 \rightarrow \mathrm{MR}_3 \rightarrow \mathrm{MR}_6 \rightarrow \mathrm{MR}_1 \rightarrow \mathrm{MR}_5 \rightarrow \mathrm{MR}_2 \rightarrow \mathrm{MC}_3$	6.00	16.42
$S_{\mathrm{MC},4,4}^{\mathrm{OPF}}$	$\mathrm{MC}_1 \rightarrow \mathrm{MR}_3 \rightarrow \mathrm{MR}_1 \rightarrow \mathrm{MR}_5 \rightarrow \mathrm{MR}_2 \rightarrow \mathrm{MC}_3$	5.00	16.45
	$S_{\mathrm{MC},5}^{\mathrm{OPF}}$		
$S_{\mathrm{MC},5,1}^{\mathrm{OPF}}$	$\mathrm{MC}_5 \rightarrow \mathrm{MR}_5 \rightarrow \mathrm{MR}_3 \rightarrow \mathrm{MC}_1$	3.00	17.55
$S_{\mathrm{MC},5,2}^{\mathrm{OPF}}$	$\mathrm{MC}_5 \rightarrow \mathrm{MR}_5 \rightarrow \mathrm{MR}_1 \rightarrow \mathrm{MR}_4 \rightarrow \mathrm{MR}_6 \rightarrow \mathrm{MR}_3 \rightarrow \mathrm{MC}_1$	6.00	16.16
$S_{\mathrm{MC},5,3}^{\mathrm{OPF}}$	$\mathrm{MC}_5 \rightarrow \mathrm{MR}_5 \rightarrow \mathrm{MR}_1 \rightarrow \mathrm{MR}_6 \rightarrow \mathrm{MR}_3 \rightarrow \mathrm{MC}_1$	5.00	16.73
$S_{\mathrm{MC},5,4}^{\mathrm{OPF}}$	$\mathrm{MC}_5 \rightarrow \mathrm{MR}_5 \rightarrow \mathrm{MR}_1 \rightarrow \mathrm{MR}_3 \rightarrow \mathrm{MC}_1$	4.00	16.76

attempts to construct the respective MC routing table $S_{\mathrm{MC}}^{\mathrm{OPF}}$ in Steps 6.1.11-25. In particular, for the routes from MC_2 to MC_1 associated with MR_6 and MR_3, respectively, the set $S_{\mathrm{MC},3,6}^{\mathrm{OPF}}$ consisting of a single route will be utilized, with the order of its nodes reversed, and the source and destination MCs are appended at the start and the end of the Pareto-optimal route, as stated in Step 6.1.18. The same process is repeated until the Pareto-optimal routes of the entire set of active source and destination pairs have been constructed forming the MR routing table $S_{\mathrm{MC}}^{\mathrm{OPF}}$.

The construction of the MC routing table $S_{\mathrm{MC}}^{\mathrm{OPF}}$ denotes the end of the MODQO inner step presented in Alg. 6.1. Then, the MODQO algorithm invokes its outer step, which is formally defined in Alg. 6.2. Initially, the MODQO outer step sorts the elements $\{S_{\mathrm{MC},n}^{\mathrm{OPF}}\}_{n=1}^{5}$ of the MC routing in ascending order according to their number $\left|S_{\mathrm{MC},n}^{\mathrm{OPF}}\right|$ of the Pareto-optimal routes in Step 6.2.1. The sorted MC routing table associated with our scenario along with the routes' delay and power consumption is presented in Table 6.5, where the notation $S^{\mathrm{OPF}}_{\mathrm{MC},i,j}$ is used, denoting the j-th Pareto-optimal route of the set $S^{\mathrm{OPF}}_{\mathrm{MC},i}$, which in turn corresponds to the Pareto-optimal routes' set for the i-th active pair of source and destination MCs. The MODQO algorithm then initializes the set S^{OPF} of Pareto-optimal route-combinations to the first set of element in the MC routing table, i.e. we have $S^{\mathrm{OPF}} = S^{\mathrm{OPF}}_{\mathrm{MC},1}$, and it then removes this element from the MC routing table, according to Step 6.2.2.

Subsequently, observe for the MODQO outer step's iterative process of Steps 6.2.3-11 that the route-combinations' OPF S^{OPF} contains routes from all the active pairs of source and destination MCs, as encapsulated by Step 6.2.11. During the first iteration, the MODQO outer procedure assesses the maximum achievable complexity reduction offered by the NDQIO algorithm by jointly considering multiple stages. After the initialization the MC routing table consists of four elements, namely the elements $\{S^{\mathrm{OPF}}_{\mathrm{MC},n}\}_{n=2}^{5}$ of Table 6.5. According to Step 6.2.4, the reference complexity $L^{\mathrm{ref}}_{\mathrm{CM}}(n)$ quantified in terms of the number of CFEs as a function of n encapsulated stages imposed by the CM method based on Eq. (6.47) is equal to:

$$L^{\mathrm{ref}}_{\mathrm{CM}}(n) = \begin{cases} 9, & n = 1, \\ 90, & n = 2, \\ 234, & n = 3, \\ 490, & n = 4, \end{cases} \tag{6.55}$$

while the upper bound of the reference complexity $L^{\mathrm{ref}}_{\mathrm{QM,max}}(n)$ quantified in terms of the number of CFEs as a function of n encapsulated stages imposed by the QM method based on Eq. (6.49) is equal to:

$$L^{\mathrm{ref}}_{\mathrm{QM,max}}(n) = \begin{cases} 59, & n = 1, \\ 405, & n = 2, \\ 777, & n = 3, \\ 2016, & n = 4. \end{cases} \tag{6.56}$$

Therefore, based on Eqs. (6.55) and (6.56) the lower bound of the complexity reduction offered by the NDQIO with respect to the CM method as a function of n encapsulated stages is equal to:

$$\frac{L^{\mathrm{ref}}_{\mathrm{CM}}(n)}{L^{\mathrm{ref}}_{\mathrm{QM,max}}(n)} = \begin{cases} 0.1525, & n = 1, \\ 0.2222, & n = 2, \\ \mathbf{0.3012}, & n = 3, \\ 0.2431, & n = 4. \end{cases} \tag{6.57}$$

Therefore, it is clear from Eq. (6.57) that the lower bound of the complexity reduction offered by the NDQIO algorithm with respect to the CM method is maximized for $n_{\text{opt}} = 3$ joint stages.

Having exported the optimal number n_{opt} of the joint stages to be considered by the NDQIO algorithm, the MODQO outer step has to assess as to whether the NDQIO algorithm is indeed capable of outperforming the CM method for this specific optimal value. This action is undertaken by Step 6.2.6, where the lower bound of the complexity imposed by the NDQIO algorithm is compared to that of the CM method. More particularly, for our scenario we have:

$$L^{\text{ref}}_{\text{QM,min}}(n_{\text{opt}}) = 123 < 234 = L^{\text{ref}}_{\text{CM}}(n_{\text{opt}}). \tag{6.58}$$

Therefore, based on Eq. (6.58) the MODQO outer step concludes that NDQIO may potentially impose a lower number of CFEs than the CM method, and activates it for $n_{\text{opt}} = 3$ stages, according to Step 6.2.7, thus invoking the process portrayed in Fig. 6.13 and the stages examined are then removed from the MC routing table. At the end of the NDQIO process the MODQO outer step reaches the end of the fourth stage in Fig. 6.13 and, thus, the set S^{OPF} of the route-combinations is updated to the OPF exported by the NDQIO process, which in turn constitutes the set[3] $S^{\text{OPF}}_{(4)}$ of the surviving route-combinations of the fourth stage in the irregular trellis of Fig. 6.13. These eight route-combinations along with their respective average delay and their respective average power consumption are presented in Table 6.6, where the route-combinations are represented using the IDs from the first column of Table 6.5 containing the MC routing table. For instance, the first surviving route-combination $S^{\text{OPF}}_{(4),1}$ of the fourth stage is translated using Table 6.5 as follows:

$$S^{\text{OPF}}_{(4),1} = \left\{ \begin{array}{c} \text{MC}_2 \rightarrow \text{MR}_6 \rightarrow \text{MR}_3 \rightarrow \text{MC}_1, \\ \text{MC}_4 \rightarrow \text{MR}_7 \rightarrow \text{MR}_3 \rightarrow \text{MC}_1, \\ \text{MC}_2 \rightarrow \text{MR}_6 \rightarrow \text{MR}_5 \rightarrow \text{MC}_5, \\ \text{MC}_1 \rightarrow \text{MR}_3 \rightarrow \text{MR}_1 \rightarrow \text{MR}_2 \rightarrow \text{MC}_3 \end{array} \right\}. \tag{6.59}$$

Subsequently, the eight surviving route-combinations contained in the set $S^{\text{OPF}}_{(4)}$ continue their propagation to the final stage of the irregular trellis diagram of Fig. 6.13. A new iteration of the MODQO outer step is invoked and the reference complexities quantified in terms of the number of CFEs are evaluated during Step 6.2.4, as follows:

$$L^{\text{ref}}_{\text{CM}}(n) = 1024,\ n = 1, \tag{6.60}$$
$$L^{\text{ref}}_{\text{QM,max}}(n) = 1563\ n = 1, \tag{6.61}$$
$$L^{\text{ref}}_{\text{QM,min}}(n) = 291\ n = 1, \tag{6.62}$$

where the maximum complexity reduction offered by the NDQIO algorithm with respect to

[3]In general, the notation $S^{\text{OPF}}_{(n)}$ corresponds to the surviving route-combinations of the n-th trellis stage.

Table 6.6: Surviving route-combinations S of the outer step.

ID	Pareto-optimal Route-combinations	$\bar{P}(S)$	$\bar{D}(S)$	$\bar{H}[\bar{B}_{com}(S)]$	$\max\{\bar{H}[\bar{B}_{com}(S)]\}$	$\text{argmax}\{\bar{H}[\bar{B}_{com}(S)]\}$
Initialization $S^{\text{OPF}}_{(1)}$						
$S^{\text{OPF}}_{(1),1}$	$S^{\text{OPF}}_{\text{MC},1,1}$	7.68	3.00	N/A		
Iteration 1: NDQIO encapsulates 3 joint stages						
$S^{\text{OPF}}_{(4),1}$	$S^{\text{OPF}}_{\text{MC},1,1}$, $S^{\text{OPF}}_{\text{MC},2,1}$, $S^{\text{OPF}}_{\text{MC},3,1}$, $S^{\text{OPF}}_{\text{MC},4,1}$	16.35	3.25	N/A		
$S^{\text{OPF}}_{(4),2}$	$S^{\text{OPF}}_{\text{MC},1,1}$, $S^{\text{OPF}}_{\text{MC},2,2}$, $S^{\text{OPF}}_{\text{MC},3,2}$, $S^{\text{OPF}}_{\text{MC},4,2}$	13.63	5.00			
$S^{\text{OPF}}_{(4),3}$	$S^{\text{OPF}}_{\text{MC},1,1}$, $S^{\text{OPF}}_{\text{MC},2,2}$, $S^{\text{OPF}}_{\text{MC},3,3}$, $S^{\text{OPF}}_{\text{MC},4,1}$	14.48	4.00			
$S^{\text{OPF}}_{(4),4}$	$S^{\text{OPF}}_{\text{MC},1,1}$, $S^{\text{OPF}}_{\text{MC},2,2}$, $S^{\text{OPF}}_{\text{MC},3,3}$, $S^{\text{OPF}}_{\text{MC},4,4}$	14.15	4.25			
$S^{\text{OPF}}_{(4),5}$	$S^{\text{OPF}}_{\text{MC},1,1}$, $S^{\text{OPF}}_{\text{MC},2,2}$, $S^{\text{OPF}}_{\text{MC},3,3}$, $S^{\text{OPF}}_{\text{MC},4,2}$	13.89	4.75			
$S^{\text{OPF}}_{(4),6}$	$S^{\text{OPF}}_{\text{MC},1,1}$, $S^{\text{OPF}}_{\text{MC},2,3}$, $S^{\text{OPF}}_{\text{MC},3,1}$, $S^{\text{OPF}}_{\text{MC},4,1}$	15.51	3.50			
$S^{\text{OPF}}_{(4),7}$	$S^{\text{OPF}}_{\text{MC},1,1}$, $S^{\text{OPF}}_{\text{MC},2,2}$, $S^{\text{OPF}}_{\text{MC},3,2}$, $S^{\text{OPF}}_{\text{MC},4,4}$	13.90	4.50			
$S^{\text{OPF}}_{(4),8}$	$S^{\text{OPF}}_{\text{MC},1,1}$, $S^{\text{OPF}}_{\text{MC},2,3}$, $S^{\text{OPF}}_{\text{MC},3,3}$, $S^{\text{OPF}}_{\text{MC},4,1}$	14.99	3.75			
Final Iteration: NDQIO encapsulates 1 stage						
$S^{\text{OPF}}_{(5),1}$	$S^{\text{OPF}}_{\text{MC},1,1}$, $S^{\text{OPF}}_{\text{MC},2,1}$, $S^{\text{OPF}}_{\text{MC},3,1}$, $S^{\text{OPF}}_{\text{MC},4,1}$, $S^{\text{OPF}}_{\text{MC},5,1}$	16.62	3.20	0.000	0.000	$S^{\text{OPF}}_{(5),1}$
$S^{\text{OPF}}_{(5),2}$	$S^{\text{OPF}}_{\text{MC},1,1}$, $S^{\text{OPF}}_{\text{MC},2,2}$, $S^{\text{OPF}}_{\text{MC},3,2}$, $S^{\text{OPF}}_{\text{MC},4,2}$, $S^{\text{OPF}}_{\text{MC},5,2}$	14.26	5.20	0.665	0.665	$S^{\text{OPF}}_{(5),2}$
$S^{\text{OPF}}_{(5),3}$	$S^{\text{OPF}}_{\text{MC},1,1}$, $S^{\text{OPF}}_{\text{MC},2,2}$, $S^{\text{OPF}}_{\text{MC},3,3}$, $S^{\text{OPF}}_{\text{MC},4,1}$, $S^{\text{OPF}}_{\text{MC},5,4}$	15.04	4.00	0.488	0.665	$S^{\text{OPF}}_{(5),2}$
$S^{\text{OPF}}_{(5),4}$	$S^{\text{OPF}}_{\text{MC},1,1}$, $S^{\text{OPF}}_{\text{MC},2,2}$, $S^{\text{OPF}}_{\text{MC},3,2}$, $S^{\text{OPF}}_{\text{MC},4,4}$, $S^{\text{OPF}}_{\text{MC},5,4}$	14.64	4.40	0.656	0.665	$S^{\text{OPF}}_{(5),2}$
$S^{\text{OPF}}_{(5),5}$	$S^{\text{OPF}}_{\text{MC},1,1}$, $S^{\text{OPF}}_{\text{MC},2,3}$, $S^{\text{OPF}}_{\text{MC},3,1}$, $S^{\text{OPF}}_{\text{MC},4,1}$, $S^{\text{OPF}}_{\text{MC},5,1}$	16.00	3.40	0.000	0.665	$S^{\text{OPF}}_{(5),2}$
$S^{\text{OPF}}_{(5),6}$	$S^{\text{OPF}}_{\text{MC},1,1}$, $S^{\text{OPF}}_{\text{MC},2,2}$, $S^{\text{OPF}}_{\text{MC},3,3}$, $S^{\text{OPF}}_{\text{MC},4,4}$, $S^{\text{OPF}}_{\text{MC},5,2}$	14.64	4.60	**0.679**	0.679	$S^{\text{OPF}}_{(5),6}$
$S^{\text{OPF}}_{(5),7}$	$S^{\text{OPF}}_{\text{MC},1,1}$, $S^{\text{OPF}}_{\text{MC},2,3}$, $S^{\text{OPF}}_{\text{MC},3,1}$, $S^{\text{OPF}}_{\text{MC},4,1}$, $S^{\text{OPF}}_{\text{MC},5,1}$	15.63	3.60	0.327	0.679	$S^{\text{OPF}}_{(5),6}$
$S^{\text{OPF}}_{(5),8}$	$S^{\text{OPF}}_{\text{MC},1,1}$, $S^{\text{OPF}}_{\text{MC},2,2}$, $S^{\text{OPF}}_{\text{MC},3,3}$, $S^{\text{OPF}}_{\text{MC},4,1}$, $S^{\text{OPF}}_{\text{MC},5,1}$	15.29	3.80	0.534	0.679	$S^{\text{OPF}}_{(5),6}$
$S^{\text{OPF}}_{(5),9}$	$S^{\text{OPF}}_{\text{MC},1,1}$, $S^{\text{OPF}}_{\text{MC},2,2}$, $S^{\text{OPF}}_{\text{MC},3,2}$, $S^{\text{OPF}}_{\text{MC},4,2}$, $S^{\text{OPF}}_{\text{MC},5,4}$	14.46	4.80	0.674	0.679	$S^{\text{OPF}}_{(5),6}$
$S^{\text{OPF}}_{(5),10}$	$S^{\text{OPF}}_{\text{MC},1,1}$, $S^{\text{OPF}}_{\text{MC},2,2}$, $S^{\text{OPF}}_{\text{MC},3,3}$, $S^{\text{OPF}}_{\text{MC},4,4}$, $S^{\text{OPF}}_{\text{MC},5,4}$	14.81	4.20	0.638	0.679	$S^{\text{OPF}}_{(5),6}$
$S^{\text{OPF}}_{(5),11}$	$S^{\text{OPF}}_{\text{MC},1,1}$, $S^{\text{OPF}}_{\text{MC},2,2}$, $S^{\text{OPF}}_{\text{MC},3,3}$, $S^{\text{OPF}}_{\text{MC},4,2}$, $S^{\text{OPF}}_{\text{MC},5,2}$	14.45	5.00	0.675	0.679	$S^{\text{OPF}}_{(5),6}$

the CM method is achieved for $n_{\text{opt}} = 1$, in the absence of any other stages. Additionally, since the lower bound of the NDQIO reference complexity $L^{\text{ref}}_{\text{QM,min}}$ is lower than that of the CM method, a NDQIO process is activated for encapsulating the final stage into the eight surviving route-combinations and, thus, updating the OPF of the route combinations to $S^{\text{OPF}} = S^{\text{OPF}}_{(5)}$. Naturally, since we have $n_{\text{opt}} = \left|S^{\text{OPF}}_{\text{MC}}\right| = 1$, the outer step concludes that this is the final stage. Hence, the normalized entropy of the normalized composite betweenness $\bar{H}[\bar{B}_{com}(S)]$ is calculated after the identification of a Pareto-optimal route-combination by the NDQIO algorithm and a record of the route-combination exhibiting the highest $\bar{H}[\bar{B}_{com}(S)]$ value is kept. For this reason, the normalized entropy of the normalized composite betweenness values of the Pareto-optimal route-combinations are exclusively included in the last three columns of Table 6.6, where we can observe that the Pareto-optimal route-combination is the $S^{\text{OPF}}_{(5),6}$ one, which is translated using Table 6.5 as follows:

$$S^{\text{OPF}}_{(5),6} = \left\{ \begin{array}{c} \text{MC}_2 \to \text{MR}_6 \to \text{MR}_3 \to \text{MC}_1, \\ \text{MC}_4 \to \text{MR}_7 \to \text{MR}_4 \to \text{MR}_6 \to \text{MR}_3 \to \text{MC}_1 \\ \text{MC}_2 \to \text{MR}_6 \to \text{MR}_1 \to \text{MR}_5 \to \text{MC}_5 \\ \text{MC}_1 \to \text{MR}_3 \to \text{MR}_1 \to \text{MR}_5 \to \text{MR}_2 \to \text{MC}_3 \\ \text{MC}_5 \to \text{MR}_5 \to \text{MR}_1 \to \text{MR}_4 \to \text{MR}_6 \to \text{MR}_3 \to \text{MC}_1 \end{array} \right\}. \tag{6.63}$$

Finally, the MOQDO algorithm outputs the specific route-combination $S^{\text{OPF}}_{(5),6}$ that exhibits the highest load balancing metric along with the entire set S^{OPF} of Pareto-optimal route-combinations, as described in Block 4 of the MODQO flowchart seen in Fig. 6.11. By a close inspection, the eight routes of the last stage are identical to those of Table 6.3 exported by the exhaustive search.

6.4 Accuracy versus Complexity Discussions

Having provided a detailed description of the MODQO algorithm in the previous section, let us now assess it performance in terms of both its complexity imposed and its accuracy in terms of the optimization metrics considered.

6.4.1 Complexity

As it can be observed in the MODQO flowchart of Fig. 6.11, the MODQO algorithm's operation is constituted by two distinct steps, namely the *inner* and *outer steps*. Therefore, we have to characterize the complexity associated with each step independently for the sake of characterizing the total complexity imposed by the MODQO algorithm. Let us commence with the characterization of the MODQO inner step complexity.

6.4.1.1 Inner Step

As far as the inner step is concerned, the NDQIO algorithm, defined in [2, Alg. 4], is activated by the cluster head in Step 6.1.6 precisely $||\mathcal{F}_{MR}||_2/2$ times, namely once per two friendly MRs. This is justified by the fact that we can utilize the Pareto-optimal routes spanning from MR$_i$ to MR$_j$ for constructing the Pareto-optimal routes emerging from MR$_j$ to MR$_i$ by inverting the sequence of the nodes at the cost of no additional CFEs. Therefore, the total number N_{NDQIO} of the NDQIO process activations is equal to:

$$N_{\text{NDQIO}} = \frac{||\mathcal{F}_{MR}||_2}{2}, \tag{6.64}$$

while the respective lower and upper bounds are equal to:

$$N_{\text{NDQIO}}^{\min} = 1, \tag{6.65}$$

$$N_{\text{NDQIO}}^{\max} = \frac{N_{\text{MR}}(N_{\text{MR}}-1)}{2}, \tag{6.66}$$

where the lower bound $N_{\text{NDQIO}}^{\min}$ corresponds to the best-case scenario, and the MCs are associated with only two different MR. By contrast, the upper bound $N_{\text{NDQIO}}^{\max}$ occurs in the worst-case scenario, where each of the MRs shares a friendship relationship with all the rest of the MRs. Therefore, the complexity $L_{\text{MODQO}}^{\text{inner}}$ quantified in terms of the number of CFEs imposed by the MODQO inner step is equal to:

$$L_{\text{MODQO}}^{\text{inner}} = \frac{||\mathcal{F}_{MR}||_2 \, L_{\text{NDQIO}}}{2}, \tag{6.67}$$

where the complexity L_{NDQIO} imposed by a single NDQIO algorithm activation has been quantified in Eqs. (5.53) and (5.54) for its parallel and sequential complexities, respectively, in terms of the number of CFEs. We note that Eq. (6.67) corresponds to the general case of the complexity imposed by the MODQO algorithm's inner step. Explicitly, we can readily derive the lower and upper bounds of the parallel complexity using Eqs. (6.65) and (6.66) as follows:

$$L_{\text{MODQO}}^{P,\text{inner},\min}(N_{\text{OPF}}) = L_{\text{NDQIO}}^{P,\min}(N_{\text{OPF}}) = O(N_{\text{OPF}}\sqrt{N}), \tag{6.68}$$

$$\begin{aligned} L_{\text{MODQO}}^{P,\text{inner},\max}(N_{\text{OPF}}) &= \tfrac{N_{\text{MR}}(N_{\text{MR}}-1)}{2} L_{\text{NDQIO}}^{P,\max}(N_{\text{OPF}}), \\ &= O(N_{\text{MR}}^2 N_{\text{OPF}}\sqrt{N}), \end{aligned} \tag{6.69}$$

where N_{OPF} corresponds to the number of Pareto-optimal route-solutions of each independent sub-problem and N is the total number of legitimate routes between two MRs as defined in Eq. (2.8). As for the sequential complexity's lower and upper bounds characterizing the MODQO algorithm's inner step, they can be derived using Eqs. (6.65) and (6.66), respectively, as well as Eqs. (5.54) and (6.67) as follows:

$$L_{\text{MODQO}}^{S,\text{inner},\min}(N_{\text{OPF}}) = 2L_{\text{NDQIO}}^{S,\min}(N_{\text{OPF}}) = O(N_{\text{OPF}}^2\sqrt{N}), \tag{6.70}$$

$$\begin{aligned} L_{\text{MODQO}}^{S,\text{inner},\max}(N_{\text{OPF}}) = & \ N_{\text{MR}}(N_{\text{MR}}-1)L_{\text{NDQIO}}^{S,\max}(N_{\text{OPF}}), \\ = & \ O(N_{\text{MR}}^2 N_{\text{OPF}}^2 \sqrt{N}). \end{aligned} \tag{6.71}$$

We note that the minimum complexity imposed by the NDQIO algorithm in both domains can be derived by considering that both the BBHT-QSA and the DHA sub-processes impose the minimum possible amount of complexity, namely $L_{\text{BBHT}}^{\min}$ and $L_{\text{DHA}}^{\min}$ defined in Eqs. (4.15) and (5.26), respectively. Equivalently, the NDQIO algorithm's maximum complexity in both domains is derived by considering that both the BBHT-QSA and the DHA sub-processes impose the maximum amount of complexity, namely $L_{\text{BBHT}}^{\max}$ and $L_{\text{DHA}}^{\max}$ defined in Eqs. (4.16) and (5.35), respectively.

Observe in Eq. (6.75) that the upper bound of the complexity imposed by the MODQO algorithm becomes independent of the number N_{MC} of MCs, since it solely depends on the number N_{MR} of MRs. Naturally, as the number of MCs increases the number of friendly MRs per MR also increases up to a maximum of the total number of MRs.

Additionally, it is possible to derive the upper and lower bounds of the number N_{OPF} of Pareto-optimal routes specifically for our scenario. Explicitly, the lower bound corresponds to the best-case scenario, where a single Pareto-optimal route exists for all the possible pairs of source and destination MCs. By contrast, the upper bound of the number N_{OPF} of Pareto-optimal routes corresponds to the worst-case scenario, where a single Pareto-optimal route exists for all the possible values of the delay due to the utilization of the weak Pareto dominance operator of Definition 1. Since the delay has been quantified in terms of the number of hops, its maximum value is encountered when all the MRs participate in the route establishment, i.e. we have $D_{\max}(x) = N_{\text{MR}} + 2$ hops, while its minimum value corresponds to the case where there are no intermediate MR involved in the route construction, i.e. we have $D_{\min}(x) = 3$ hops. This range provides us with at most $D_{\max}(x) - D_{\min}(x) = N_{\text{MR}} - 1$ possible delay values. Therefore the upper and lower bounds of the number of Pareto-optimal routes are quantified as follows:

$$N_{\text{OPF}}^{\min} = 1, \tag{6.72}$$

$$N_{\text{OPF}}^{\max} = N_{\text{MR}} - 1. \tag{6.73}$$

Consequently, we may readily derive the strict upper and lower bounds of the MODQO inner step parallel complexity by substituting the bounds of Eqs. (6.72) and (6.73) into Eqs. (6.68) and (6.69), respectively. Hence, the resultant strict MODQO inner step lower bound of its parallel complexity is equal to:

$$L_{\text{MODQO}}^{P,\text{inner},\min} = L_{\text{NDQIO}}^{P,\min}(1) = O(\sqrt{N}), \tag{6.74}$$

while the respective strict upper bound of its parallel complexity is equal to:

$$\begin{aligned} L^{P,\text{inner},\max}_{\text{MODQO}} &= \tfrac{N_{\text{MR}}(N_{\text{MR}}-1)}{2} L^{P,\max}_{\text{NDQIO}}(N_{\text{MR}}-1), \\ &= O(N^3_{\text{MR}}\sqrt{N}). \end{aligned} \tag{6.75}$$

Additionally, we may derive strict bounds of the MODQO inner step sequential complexity by substituting the bounds of Eqs. (6.72) and (6.73) into Eqs. (6.70) and (6.71), respectively. Consequently, the resultant strict MODQO inner step lower bound of its parallel complexity is equal to:

$$L^{S,\text{inner},\min}_{\text{MODQO}} = 2L^{S,\min}_{\text{NDQIO}}(1) = O(\sqrt{N}), \tag{6.76}$$

while the respective strict upper bound of its parallel complexity is equal to:

$$\begin{aligned} L^{S,\text{inner},\max}_{\text{MODQO}} &= N_{\text{MR}}(N_{\text{MR}}-1) L^{S,\max}_{\text{NDQIO}}(N_{\text{MR}}-1), \\ &= O(N^4_{\text{MR}}\sqrt{N}). \end{aligned} \tag{6.77}$$

Based on this analysis, let us now proceed by presenting the average MODQO inner step complexity in both domains quantified as a function of the number of CFEs for twin-layer networks consisting of $N_{\text{MR}} = \{5, 6, \dots, 10\}$ MRs and $N_{\text{MR}} = \{2, 4, 8, 16\}$ MCs, which are shown in Figs. 6.15(a) and 6.15(b) for the parallel and sequential complexities, respectively. In these figures, the MODQO algorithm's average inner step parallel and sequential complexities are compared both to their lower and upper bounds. Recall that as they were quantified in Eqs. (6.74) and (6.75) for the parallel complexity and in Eqs. (6.76) and (6.77) for the sequential complexity. They will also be compared to that average parallel and sequential complexities imposed by the exhaustive search. We note that both the exhaustive search and the MODQO algorithm have been deployed and their complexity was averaged over identical twin-layer network structures. Observe in Figs. 6.15(a,b) that the MODQO algorithm's average complexity increases exponentially with the number N_{MR} of MRs in both domains. Nevertheless, both the parallel and the sequential complexities increase with a substantially lower gradient than the respective average complexities imposed by the exhaustive search. More specifically, for the parallel complexity presented in Fig. 6.15(a), the MODQO inner step achieves a significant complexity reduction even for twin-layer networks comprised by as few as $N_{\text{MR}} = 5$ MRs, where the MODQO algorithm imposes a parallel complexity, which is four times lower than that of the exhaustive search, while the parallel complexity imposed by the MODQO algorithm's inner step becomes about six orders of magnitude lower than that of the exhaustive search for twin-layer networks consisting of $N_{\text{MR}} = 10$ MRs. A similar trend is also observed for the sequential complexity in Fig. 6.15(b), where the MODQO algorithm imposes a twice lower sequential complexity than that of the exhaustive search for twin-layer networks associated with $N_{\text{MR}} = 5$ MRs, while this sequential complexity reduction reaches to several orders of magnitude for twin-layer networks associated with $N_{\text{MR}} = 10$ MRs.

Additionally, both the parallel and the sequential complexities quantified in terms of

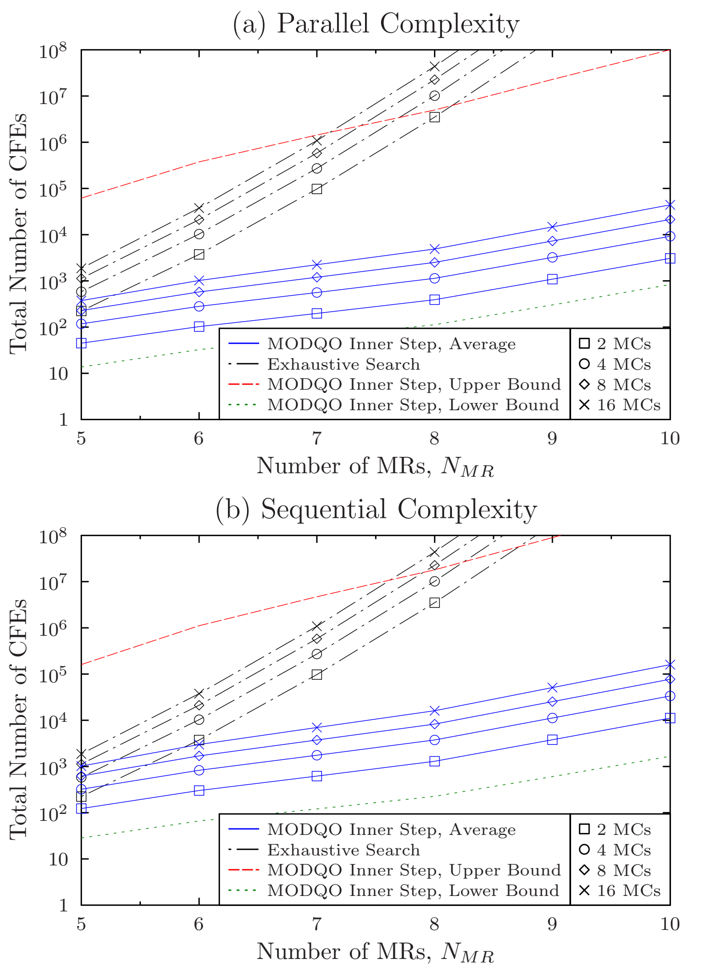


Figure 6.15: Average MODQO inner step (a) parallel and (b) sequential complexities quantified as a function of the number of CFEs for twin-layer networks consisting of $N_{\text{MR}} = \{5, 6, \ldots, 10\}$ MRs and $N_{\text{MC}} = \{2, 4, 8, 16\}$ MCs. The MODQO inner step complexity is compared to that of the exhaustive search and to its respective upper and lower bounds in both domains. Note that the lower and upper bounds of the parallel complexity are defined in Eqs. (6.74) and (6.75), respectively, whereas the respective bounds of the sequential complexity are defined in Eqs. (6.76) and (6.77). The average complexity results have been averaged over 10^8 runs for twin-layer networks based on the optimization problem of Eq. (6.22) relying on the UV defined in Eq. (6.21) and on the assumptions of Table 6.1.

the number of CFEs imposed both by the MODQO inner step and by the exhaustive search increase almost proportionally to number of MCs as the the number N_{MC} of MCs increases. Naturally, as N_{MC} increases, the average number of friends of each of the MRs increases, since the associated number of active pairs of source and destination MCs increases. This results, in turn, in an increase in the number of the NDQIO process activations, hence yielding an increase in the MODQO's inner step complexity in both domains. Nevertheless, this rate of increase slows down as the network becomes more densely populated by MCs, slowly tending to the upper bounds of Eqs. (6.75) and (6.77) for the parallel and sequential complexities, respectively, where all the MRs share a social relationship with all the rest of the MRs.

6.4.1.2 Outer Step

As far as the complexity imposed by the MODQO outer step is concerned, we first quantify that of the CM method, which constitutes the upper bound of our proposed QM process. For this reason, let us make the additional assumption that the number of Pareto-optimal route-combinations increases by a factor ρ after each state. This can be formally formulated as follows:

$$\left|S_{(n)}^{OPF}\right| = \rho^{n-1}\bar{N}_{\text{OPF}}, \tag{6.78}$$

where $\bar{N}_{\text{OPF}}$ corresponds to the average number of Pareto-optimal routes on the independent sub-problems, while $S_{(n)}^{\text{OPF}}$ is the set of Pareto-optimal route-combinations after n stages of the irregular trellis diagram. We note that it is possible to fully characterize $\bar{N}_{\text{OPF}}$ with the aid of statistical analysis of offline data. Thus, we can estimate the complexity imposed. Based on Eq. (6.78), we can quantify the complexity in terms of the number of CFEs imposed by the CM method in both domains as follows:

$$\begin{aligned} L_{\text{CM}}^{\text{outer}} &= 2\sum_{n=2}^{N_r} \rho^{2(n-1)}\bar{N}_{\text{OPF}}^4 = \bar{N}_{\text{OPF}}^4 \frac{\rho^{2N_r}-\rho^2}{\rho^2-1}, \\ &= O(\rho^{2N_r}\bar{N}_{\text{OPF}}^4), \end{aligned} \tag{6.79}$$

where the complexity $L_{\text{CM}}^{\text{outer}}$ corresponds both to the parallel complexity and to sequential complexity imposed by the CM method, since no parallelization technique is used. Explicitly, the upper bound of the factor ρ, for which the CM method is capable offering beneficial complexity reduction compared to the exhaustive search, is given by:

$$\rho < \bar{N}_{\text{OPF}}^{\frac{2N_r-2}{2N_r-1}}. \tag{6.80}$$

Consequently, the CM method is capable of offering a complexity reduction, as long as the number of Pareto-optimal route-combinations increases by a factor that is slightly less than $\bar{N}_{\text{OPF}}$, based on Eq. (6.80).

Having characterized the CM method in terms of its parallel and sequential complexities, let us now proceed with the characterization of our proposed QM method. To begin with, we will assess the complexity reduction achieved by the QM method compared to that of the CM one at a single iteration of the QM method. For this reason, we ought to investigate the dynamics behind the selection of the number n_{opt} of joint stages. Note that we will initially focus on the parallel complexity, since it constitutes the sole criterion for the selection of the number of joint stages in the irregular trellis of Fig. 6.13. In general, the parallel complexity quantified in terms of the number of CFEs imposed by the NDQIO process during a single iteration, where the NDQIO initiates from the n-th trellis stage and considers n_{opt} joint stages, is equal to:

$$\begin{aligned}L_{\text{QM}}^{P,\text{outer}}(n,n_{\text{opt}}) = \; & (2K)^{-1}\rho^{2(n+n_{\text{opt}}-1)}\bar{N}_{\text{OPF}}^{2}+\\ & K^{-1}\left[L_{\text{DHA}}(\rho^{n-1}\bar{N}_{\text{OPF}}^{n_{\text{opt}}+1})+2L_{\text{BBHT}}(\rho^{n-1}\bar{N}_{\text{OPF}}^{n_{\text{opt}}+1})-\tfrac{1}{2}\right]\rho^{n+n_{\text{opt}}-1}\bar{N}_{\text{OPF}}+\\ & +(1-K)\left[\tfrac{2}{K}L_{\text{BBHT}}(\rho^{n-1}\bar{N}_{\text{OPF}}^{n_{\text{opt}}+1})+\tfrac{1}{2}\right],\end{aligned} \tag{6.81}$$

with its order being equal to:

$$L_{\text{QM}}^{P,\text{outer}}(n,n_{\text{opt}}) = \begin{cases} O(\rho^{2n}\bar{N}_{\text{OPF}}^{2}), & n_{\text{opt}}=1,\\ O\left[\rho^{(3n+2n_{\text{opt}}-3)/2}\bar{N}_{\text{OPF}}^{(n_{\text{opt}}+3)/2}\right], & n_{\text{opt}}>1.\end{cases} \tag{6.82}$$

As for the respective sequential complexity, it can be expressed using Eq. (5.54) as follows:

$$\begin{aligned}L_{\text{QM}}^{S,\text{outer}}(n,n_{\text{opt}}) = \; & 2\Big\{\left[\tfrac{1}{2}+L_{\text{BBHT}}(\rho^{n-1}\bar{N}_{\text{OPF}}^{n_{\text{opt}}+1})\right]\rho^{2(n+n_{\text{opt}}-1)}\bar{N}_{\text{OPF}}^{2}+\\ & +\left[L_{\text{DHA}}(\rho^{n-1}\bar{N}_{\text{OPF}}^{n_{\text{opt}}+1})+L_{\text{BBHT}}(\rho^{n-1}\bar{N}_{\text{OPF}}^{n_{\text{opt}}+1})-\tfrac{1}{2}\right]\rho^{n+n_{\text{opt}}-1}\bar{N}_{\text{OPF}}+\\ & +(1-K)\left[L_{\text{DHA}}(\rho^{n-1}\bar{N}_{\text{OPF}}^{n_{\text{opt}}+1})+L_{\text{BBHT}}(\rho^{n-1}\bar{N}_{\text{OPF}}^{n_{\text{opt}}+1})+\tfrac{1}{2}K\right]\Big\},\end{aligned} \tag{6.83}$$

with its order being equal to:

$$L_{\text{QM}}^{S,\text{outer}}(n,n_{\text{opt}}) = O\left[\rho^{(5n+4n_{\text{opt}}-5)/2}\bar{N}_{\text{OPF}}^{(n_{\text{opt}}+5)/2}\right]. \tag{6.84}$$

Furthermore, the respective complexity imposed by the CM method, based on Eq. (6.79), is equal to:

$$\begin{aligned}L_{\text{CM}}^{\text{outer}}(n,n_{\text{opt}}) & = 2\sum_{n'=n}^{n+n_{\text{opt}}-1}\rho^{2(n'-1)}\bar{N}_{\text{OPF}}^{4}\\ & = 2\bar{N}_{\text{OPF}}^{4}\frac{\rho^{2(n+n_{\text{opt}})}-\rho^{2n}}{\rho-1},\\ & = O(\rho^{2(n+n_{\text{opt}})}\bar{N}_{\text{OPF}}^{4}).\end{aligned} \tag{6.85}$$

Explicitly, we have to estimate the value of the optimal number n_{opt} of stages to be considered by the QM method. Naturally, we can derive the orders of the respective reference complexities by setting $\rho=1$ in Eqs. (6.85) and (6.82), yielding:

$$L_{\text{QM}}^{\text{ref}}(n,n_{\text{opt}}) = O\left[\bar{N}_{\text{OPF}}^{(n_{\text{opt}}+3)/2}\right], \tag{6.86}$$

$$L_{\text{CM}}^{\text{ref}}(n,n_{\text{opt}}) = O(\bar{N}_{\text{OPF}}^{4}), \tag{6.87}$$

where it is clear that the CM method's order of reference complexity is constant, whilst the QM method's complexity increases as the number n_{opt} of joint stages increases. In fact, after $n_{\text{opt}}=5$ stages the order of the QM method reference complexity becomes equal to that of the CM method, implying that the NDQIO process will no longer offer any complexity reduction. Based on this dynamic, we can conclude that the optimal value of the number of joint stages considered by the NDQIO process, which provides the maximum possible complexity reduction is $n_{\text{opt}} = 1$ stage. Therefore, the total parallel complexity $L^{P,\text{outer}}_{\text{MODQO}}$ and imposed by the QM method of Alg. 6.2 is equal to:

$$\begin{aligned} L^{P,\text{outer}}_{\text{MODQO}} &= \sum_{n=1}^{N_r} L^{P,\text{outer}}_{\text{QM}}(n,1), \\ &= \sum_{n=1}^{N_r} O(\rho^{2n}\bar{N}^2_{\text{OPF}}), \\ &= O\left[\bar{N}^2_{\text{OPF}}\frac{\rho^{2(Nr+1)}-\rho^4}{\rho^2-1}\right] \\ &= O\left[\rho^{2(Nr+1)}\bar{N}^2_{\text{OPF}}\right], \end{aligned} \tag{6.88}$$

while the respective sequential complexity $L^{S,\text{outer}}_{\text{MODQO}}$ is equal to:

$$\begin{aligned} L^{S,\text{outer}}_{\text{MODQO}} &= \sum_{n=1}^{N_r} L^{S,\text{outer}}_{\text{QM}}(n,1), \\ &= \sum_{n=1}^{N_r} O(\rho^{(5n-1)/2}\bar{N}^3_{\text{OPF}}), \\ &= O\left[\bar{N}^3_{\text{OPF}}\rho^{-\frac{1}{2}}\frac{\rho^{5(Nr+1)/2}-\rho^2}{\rho^{\frac{5}{2}}-1}\right] \\ &= O\left[\rho^{5(Nr+1)/2}\bar{N}^3_{\text{OPF}}\right]. \end{aligned} \tag{6.89}$$

Hence, based on Eqs. (6.79) and (6.88) the QM method achieves a parallel complexity reduction, which is on the the order of $O\left(\rho^{-2}\bar{N}^2_{\text{OPF}}\right)$, when compared to the CM method, as long as we have $\rho < \bar{N}_{\text{OPF}}$. We note that if we have $\rho = \bar{N}_{\text{OPF}}$, then the QM method will match the complexity order of the CM method. Furthermore, the QM method offers a sequential complexity reduction factor, which is on the order of $O\left[\rho^{(N_r-3)/2}\bar{N}_{\text{OPF}}\right]$, when compared to the CM method, based on Eqs. (6.79) and (6.89). More specifically, the QM gradually outperforms the CM method in terms of its sequential complexities, as long as the following condition is satisfied:

$$\rho < \bar{N}^{2/(N_r-3)}_{\text{OPF}}. \tag{6.90}$$

Let us now compare the QM method to the case, where we invoke the NDQIO algorithm by jointly considering all the trellis stages. Explicitly, a single NDQIO iteration invoked for all the stages would result in a parallel complexity that is equal to:

$$\begin{aligned} L^{P,\text{outer}}_{\text{NDQIO}} &= L^{P,\text{outer}}_{\text{QM}}(1,N_r), \\ &= O\left[\rho^{N_r}\bar{N}^{(N_r+3)/2}_{\text{OPF}}\right], \end{aligned} \tag{6.91}$$

while the respective sequential complexity becomes:

$$\begin{aligned} L_{\text{NDQIO}}^{S,\text{outer}} &= L_{\text{QM}}^{S,\text{outer}}(1, N_r), \\ &= O\left[\rho^{2N_r} \bar{N}_{\text{OPF}}^{(N_r+5)/2}\right], \end{aligned} \tag{6.92}$$

by setting $n = 1$ and $n_{\text{opt}} = N_r$ in Eqs. (6.82) and (6.84), respectively. Observe in Eq. (6.91) that the QM method offers a parallel complexity reduction on order of $O(N_{\text{OPF}}^{Nr/2}\rho^{-Nr})$, when compared to that of a single NDQIO activation that jointly considers all the stages. Consequently, the QM method outperforms the NDQIO algorithm that considers jointly all the trellis stages in terms of their parallel complexities, as long as we strictly have the following asymptotic bound for the surviving route-combinations' growth factor ρ:

$$\rho < \sqrt{N_{\text{OPF}}}. \tag{6.93}$$

Additionally, based on Eqs. (6.92) and (6.89), the QM method offers a sequential complexity reduction factor on the order of $O\left[\rho^{-(N_r+5)/2} \bar{N}_{\text{OPF}}^{(N_r-1)/2}\right]$, while the respective asymptotic bound is quantified as follows:

$$\rho < N_{\text{OPF}}^{\frac{N_r-1}{N_r+5}}. \tag{6.94}$$

For the sake of demonstrating the benefits of the QM method both against the single NDQIO algorithm activation for all the stages and against the CM method, we have to statistically characterize the factor ρ for our scenario. Explicitly, the average number $\bar{N}_{\text{OPF}}$ of Pareto-optimal routes in the MC routing quantified as a function of the number N_{MR} of MRs is shown in Fig. 6.16. Observe in this figure that the average number lies far below the NDQIO parallel asymptotic bound of Eq. (6.93) and it is inversely proportional both to the number of MCs and to the number of MRs. This is justified by the fact that the number of Pareto-optimal route-combinations increases at a lower rate compared to the number of active routes, i.e. compared to the number of stages in the irregular trellis diagram of Figs. 6.12 and 6.13, both of which increase proportionally to the number of MCs. Additionally, an increase in the number of MRs results in an increase in the value of $\bar{N}_{\text{OPF}}$ and, thus, the number of states per trellis stage; however, the number of Pareto-optimal route-combinations tends to grow slower, hence, reducing the order of the factor ρ with respect to $\bar{N}_{\text{OPF}}$ as the number of MRs increase.

As far as the NDQIO sequential asymptotic bound is concerned, observe in Fig. 6.16 that the bound increases, as the MCs proliferate owing to the inclusion of more active routes. Explicitly, the bound of Eq. (6.94) is monotonically increasing in terms of the number N_r of active routes, approaching unity as N_r approaches infinity. Observe in the same figure that for a sufficient number of MCs, namely for $N_{\text{MC}} \geq 8$, the OPF growth factor ρ is lower than the respective bound, as seen in Fig. 6.16. However, below this value some anomalies can be observed. This specific trend implies that the QM method offers some sequential complexity reduction, when compared to the NDQIO algorithm. Hence, we expect the QM method's sequential complexity reduction to increase as the number

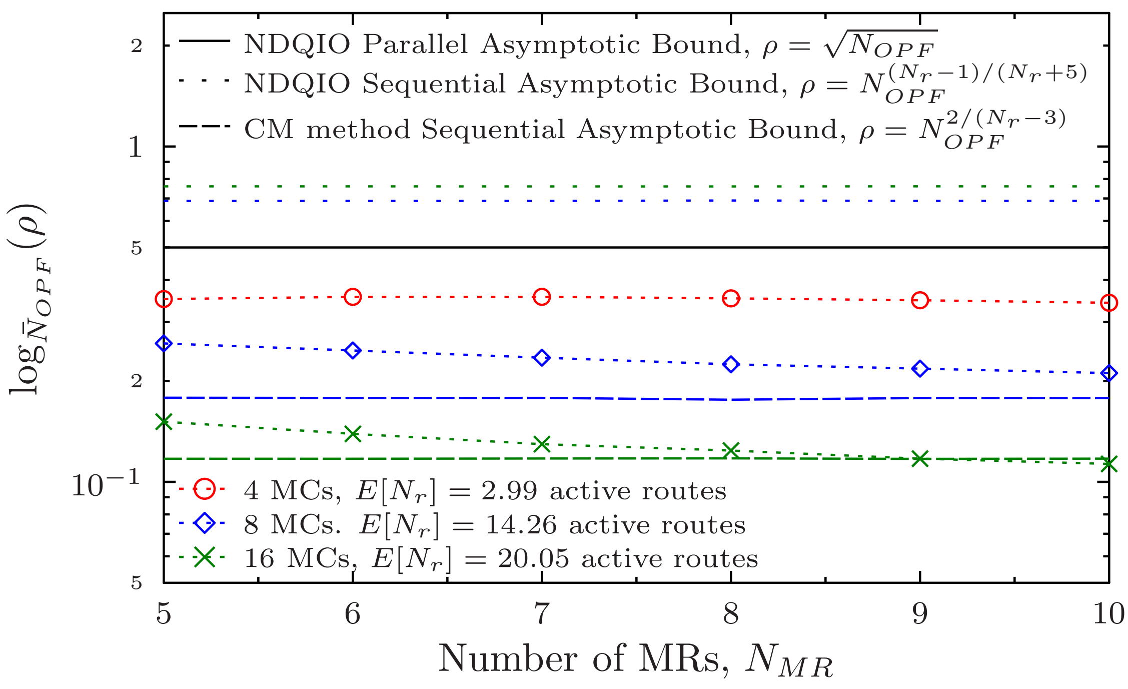


Figure 6.16: Order of the surviving route-combinations' growth factor ρ versus the average number $\bar{N}_{\text{OPF}}$ of Pareto-optimal routes in the MC routing table for twin-layer networks comprised by $N_{\text{MC}} = \{4, 8, 16\}$ MCs and $N_{\text{MR}} = \{5, \ldots, 10\}$ MRs. The order is compared to the NDQIO parallel asymptotic bound, the NDQIO sequential asymptotic bound as well as the CM method sequential asymptotic bound, defined in Eqs. (6.93), (6.94) and (6.90), respectively. Note that both the NDQIO and the CM method sequential asymptotic bounds depend on the number of MCs and thus the respective color code of the bottom legend is used. The network model relies on the optimization problem of Eq. (6.22) and on the assumptions of Table 6.1. The results have been averaged over 10^8 runs.

of MCs increases, since the bound tends to approach unity. At the same time the OPF growth factor ρ decays even further, as it can be verified in Fig. 6.16, resulting in an increase of the respective gap. As for the CM method's sequential complexity bound, observe in the same figure that for twin-layer networks having $N_{\text{MC}} = 8$ MCs, the argument of the OPF growth factor ρ lies above this specific bound but the OPF growth factor ρ tends to asymptotically approach the bound as the number N_{MR} of MRs increases. Additionally, for twin-layer networks having $N_{\text{MC}} = 16$ MCs, observe in Fig. 6.16 that for twin-layer networks having less than 9 MRs the OPF growth factor ρ lies slightly above the respective bound. However, a crossover occurs for twin-layer networks having $N_{\text{MR}} = 9$ MRs, while for twin-layer networks associated with $N_{\text{MR}} = 10$ MRs the argument of the OPF growth factor ρ lies below the bound, indicating that the QM method offers a sequential complexity reduction over the CM method.

Moving on to the upper and lower bounds of the complexity imposed by the MODQO algorithm's outer step, we will consider two extreme scenarios. In the best-case scenario, we assume that the number of Pareto-optimal route-combinations at the n-th stage is equal to the average number of Pareto-optimal routes per stage, hence, we have $\rho = 1$. We have assumed furthermore that all the quantum processes impose the minimum possible complexity in terms of CFEs. As for the lower bound of the number N_r of active routes, since we have assumed that all of the MCs share a social relationship with at least another MC, we have:

$$N_r^{\min} = N_{\text{MC}}. \tag{6.95}$$

Therefore, the lower bound of the parallel complexity imposed by a single iteration using the QM method as a function of the average number $\bar{N}_{\text{OPF}}$ of Pareto-optimal routes may be expressed using Eq. (6.81) as follows:

$$\begin{aligned} L^{P,\text{outer}}_{\text{QM},\min}(\bar{N}_{\text{OPF}}) = & \ (2K)^{-1}\bar{N}^2_{\text{OPF}} + K^{-1}\left\{L^{\min}_{\text{DHA}}(\bar{N}^2_{\text{OPF}}) + \right. \\ & \left. +2L^{\min}_{\text{BBHT}}(\bar{N}^2_{\text{OPF}}) - \tfrac{1}{2}\right\}\bar{N}_{\text{OPF}} + \\ & +(1-K)\left\{\tfrac{2}{K}L^{\min}_{\text{BBHT}}(\bar{N}^2_{\text{OPF}}) + \tfrac{1}{2}\right\}, \\ = & \ O(\bar{N}^2_{\text{OPF}}). \end{aligned} \tag{6.96}$$

Equivalently, the lower bound of the sequential complexity imposed by a single iteration using the QM method can be expressed using Eqs. (6.83) as follows:

$$\begin{aligned} L^{S,\text{outer}}_{\text{QM},\min}(\bar{N}_{\text{OPF}}) = & \ 2\left\{\left[\tfrac{1}{2} + L^{\min}_{\text{BBHT}}(\bar{N}^2_{\text{OPF}})\right]\bar{N}^2_{\text{OPF}} + \right. \\ & + \left[L^{\min}_{\text{DHA}}(\bar{N}^2_{\text{OPF}}) + L^{\min}_{\text{BBHT}}(\bar{N}^2_{\text{OPF}}) - \tfrac{1}{2}\right]\bar{N}_{\text{OPF}} + \\ & \left. + (1-K)\left[L^{\min}_{\text{DHA}}(\bar{N}^2_{\text{OPF}}) + L^{\min}_{\text{BBHT}}(\bar{N}^2_{\text{OPF}}) + \tfrac{1}{2}K\right]\right\}, \\ = & \ O(\bar{N}^3_{\text{OPF}}). \end{aligned} \tag{6.97}$$

Hence, based on Eq. (6.96) the lower bound of the MODQO outer step's parallel complexity, quantified in terms of the number of CFEs and as a function of the average number $\bar{N}_{\text{OPF}}$ of Pareto-optimal routes, is equal to:

$$\begin{aligned} L^{P,\text{outer}}_{\text{MODQO},\min}(\bar{N}_{\text{OPF}}) = & \sum_{n=2}^{N_r^{\min}} L^{P,\text{outer}}_{\text{QM},\min}(\bar{N}_{\text{OPF}}), \\ = & \ (N_{\text{MC}}-1)L^{P,\text{outer}}_{\text{QM},\min}(\bar{N}_{\text{OPF}}), \\ = & \ O(N_{\text{MC}}\bar{N}^2_{\text{OPF}}), \end{aligned} \tag{6.98}$$

while the respective the lower bound of the MODQO outer step's parallel complexity is expressed using Eq. (6.104) as follows:

$$\begin{aligned} L^{S,\text{outer}}_{\text{MODQO},\min}(\bar{N}_{\text{OPF}}) = & \sum_{n=2}^{N_r^{\min}} L^{S,\text{outer}}_{\text{QM},\min}(\bar{N}_{\text{OPF}}), \\ = & \ (N_{\text{MC}}-1)L^{S,\text{outer}}_{\text{QM},\min}(\bar{N}_{\text{OPF}}), \\ = & \ O(N_{\text{MC}}\bar{N}^3_{\text{OPF}}). \end{aligned} \tag{6.99}$$

Consequently, observe that based on Eq. (6.72) the lower bound of the MODQO outer step's parallel and sequential complexities occurs when the average number of Pareto-optimal routes is strictly equal to $N_{\text{OPF}} = 1$ for all the routes. However, in this particular case no CFEs are required for constructing the single Pareto-optimal route-combination, since there is only a single possible route-combination, yielding:

$$L^{\{S,P\},\text{outer}}_{\text{MODQO,min}} = 0. \tag{6.100}$$

We note that the same lower bound is valid for the CM method as well, yielding:

$$L^{\text{outer}}_{\text{CM,min}} = 0. \tag{6.101}$$

Subsequently, let us now derive the strict upper bound of the MODQO outer step complexity. For this reason, we will consider the worst case scenario, where all the potential route-combinations identified by the MODQO inner step are Pareto-optimal. In this scenario, the surviving route-combinations' growth factor is set to $\rho = \bar{N}_{\text{OPF}}$. Additionally, the maximum number $N_r^{\max}$ of the active pairs of source and destination MCs occurs in the case, where all the MCs share a friendship relationship with each other. Nevertheless, as the number N_{MC} of MCs increases, so does the probability of two MCs being associated with the same MR. This results in a single Pareto-optimal route and, thus, in a single state in the respective stage of the irregular trellis diagram, where no CFEs are required for encapsulating it in the set of Pareto-optimal route-combinations. Therefore, the upper bound of the number of active pair of source and destination MCs that require at least a single CFE for their processing and for their inclusion in the Pareto-optimal route combinations is derived as follows:

$$N_r^{\max} = \frac{1}{2}\left[(N_{\text{MC}}(N_{\text{MC}} - 1) - \left\lfloor \frac{N_{\text{MC}}}{N_{\text{MR}}} \right\rfloor\right] = O(N_{\text{MC}}^2). \tag{6.102}$$

As for the upper bound $L^{P,\text{outer}}_{\text{QM,max}}(n, \bar{N}_{\text{OPF}})$ of the parallel complexity imposed by a single iteration using the QM method as a function of the average number $\bar{N}_{\text{OPF}}$ of Pareto-optimal routes at the n-th stage of the irregular trellis diagram, it is derived a follows:

$$\begin{aligned} L^{P,\text{outer}}_{\text{QM,max}}(n, \bar{N}_{\text{OPF}}) = \quad & (2K)^{-1}\bar{N}^{2n}_{\text{OPF}} + K^{-1}\left[L^{\max}_{\text{DHA}}(\bar{N}^{n}_{\text{OPF}}) + \right. \\ & \left. + 2L^{\max}_{\text{BBHT}}(\bar{N}^{n}_{\text{OPF}}) - \tfrac{1}{2}\right]\bar{N}^{n}_{\text{OPF}} + \\ & + (1-K)\left[\tfrac{2}{K}L^{\max}_{\text{BBHT}}(\bar{N}^{n}_{\text{OPF}}) + \tfrac{1}{2}\right], \\ = \quad & O(\bar{N}^{2n}_{\text{OPF}}), \end{aligned} \tag{6.103}$$

while the respective upper bound of the sequential complexity is expressed as:

$$\begin{aligned} L^{S,\text{outer}}_{\text{QM,max}}(n,\bar{N}_{\text{OPF}}) = {} & 2\left\{\left[\tfrac{1}{2}+L^{\min}_{\text{BBHT}}(\bar{N}^n_{\text{OPF}})\right]\bar{N}^{2n}_{\text{OPF}}+\right. \\ & +\left[L^{\min}_{\text{DHA}}(\bar{N}^n_{\text{OPF}})+L^{\min}_{\text{BBHT}}(\bar{N}^n_{\text{OPF}})-\tfrac{1}{2}\right]\bar{N}^n_{\text{OPF}}+ \\ & \left.+(1-K)\left[L^{\min}_{\text{DHA}}(\bar{N}^n_{\text{OPF}})+L^{\min}_{\text{BBHT}}(\bar{N}^n_{\text{OPF}})+\tfrac{1}{2}K\right]\right\}, \\ = {} & O\left(\bar{N}^{5n/2}_{\text{OPF}}\right). \end{aligned} \tag{6.104}$$

Based on Eq. (6.103), the upper bound $L^{P,outer}_{\text{MODQO,max}}(\bar{N}_{\text{OPF}})$ of the MODQO outer step's parallel complexity, which is quantified in terms of the number of CFEs and as a function of the average number $\bar{N}_{\text{OPF}}$ of the Pareto-optimal routes, is formulated as follows:

$$\begin{aligned} L^{P,\text{outer}}_{\text{MODQO,max}}(\bar{N}_{\text{OPF}}) = {} & \sum_{n=2}^{N_r^{\max}} L^{P,\text{outer}}_{\text{QM,min}}(n,\bar{N}_{\text{OPF}}), \\ = {} & O\left[\frac{\bar{N}^{2(N_r^{\max}+1)}_{\text{OPF}}-\bar{N}^4_{\text{OPF}}}{\bar{N}^2_{\text{OPF}}-1}\right], \\ = {} & O(\bar{N}^{2N^2_{\text{MC}}}_{\text{OPF}}), \end{aligned} \tag{6.105}$$

whereas the respective upper bound of its sequential complexity is expressed with the aid of Eq. (6.104) as:

$$\begin{aligned} L^{S,\text{outer}}_{\text{MODQO,max}}(\bar{N}_{\text{OPF}}) = {} & \sum_{n=2}^{N_r^{\max}} L^{S,\text{outer}}_{\text{QM,min}}(n,\bar{N}_{\text{OPF}}), \\ = {} & O\left[\frac{\bar{N}^{5(N_r^{\max}+1)/2}_{\text{OPF}}-\bar{N}^5_{\text{OPF}}}{\bar{N}^{5/2}_{\text{OPF}}-1}\right], \\ = {} & O(\bar{N}^{5N^2_{\text{MC}}/2}_{\text{OPF}}). \end{aligned} \tag{6.106}$$

Consequently, the upper bounds $L^{\{P,S\},\text{outer}}_{\text{MODQO,max}}$ of the MODQO outer step's complexity are encountered, when the average number of Pareto-optimal routes is strictly equal to $N_{\text{OPF}} = N_{\text{MR}} - 1$ for all the routes. Then, based on Eq. (6.73), we have:

$$L^{P,\text{outer}}_{\text{MODQO,max}} = O\left(N^{2N^2_{\text{MC}}}_{\text{MR}}\right), \tag{6.107}$$

$$L^{S,\text{outer}}_{\text{MODQO,max}} = O\left(N^{5N^2_{\text{MC}}/2}_{\text{MR}}\right). \tag{6.108}$$

The respective upper bound of the CM method is derived by setting $\rho = \bar{N}_{\text{OPF}} = N_{MR} - 1$ in Eq. (6.79), yielding:

$$L^{\text{outer}}_{\text{CM,max}} = O\left(N^{2N^2_{\text{MC}}}_{\text{MR}}\right). \tag{6.109}$$

Consequently, both the QM method and the CM method impose the same order of parallel complexity in the worst-case scenario, matching the exhaustive search complexity. Explicitly, in the worst-case scenario there will be no complexity reduction for either of these methods. By contrast, the QM method imposes a complexity order, which is a factor of $N^{N^2_{\text{MC}}/2}_{\text{MR}}$ higher than that of the QM method. We note though that this specific scenario hardly occurs in the light of the ρ parameter trend shown in Fig 6.16, which decreases as the number N_{MC} of MCs increases.

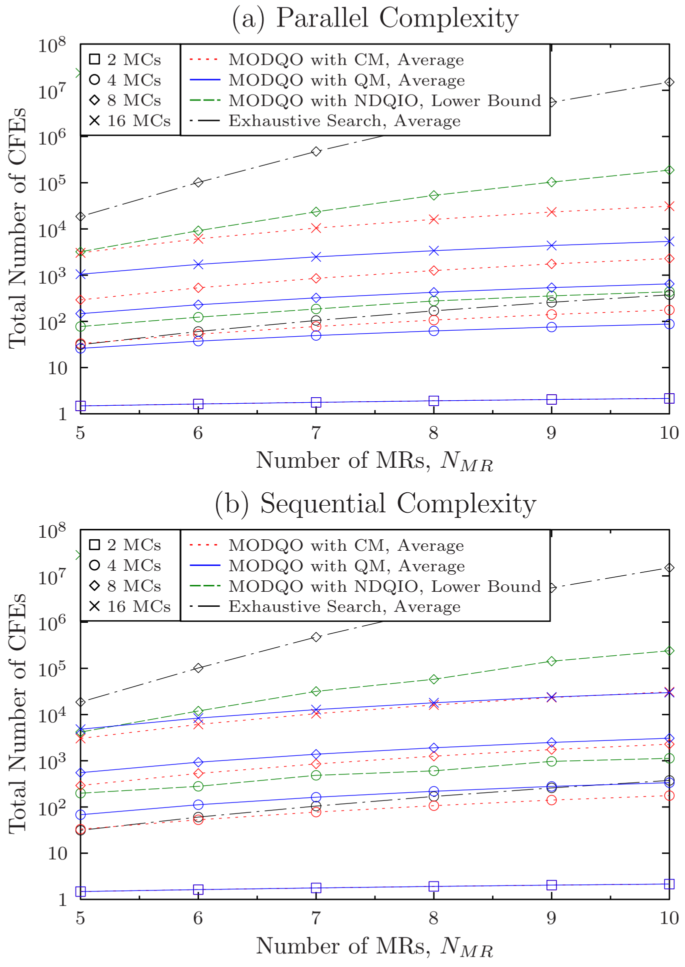


Figure 6.17: Average MODQO outer step (a) parallel and (b) sequential complexities quantified as a function of the number of CFEs for twin-layer networks consisting of $N_{\text{MR}} = \{5, 6, \ldots, 10\}$ MRs and $N_{\text{MC}} = \{2, 4, 8, 16\}$ MCs. The MODQO outer step complexity is compared to that of the exhaustive search, that of the CM method and to the lower bound of the NDQIO algorithm invoked jointly for all the available stages. The mean complexity results have been averaged over 10^8 runs for twin-layer networks based on the optimization problem of Eq. (6.22) relying on the UV defined in Eq. (6.21) and on the assumptions of Table 6.1.

The average parallel and sequential complexities quantified in terms of the number of CFEs of the MODQO outer step are shown in Figs. 6.17(a) and 6.17(b), respectively, for twin-layer networks consisting of 5 to 10 MRs and of 2, 4, 8, and 16 MCs. The MODQO outer step complexities, which are represented by the solid lines in Figs. 6.17(a,b) and denoted as "MODQO with QM", are compared both to the respective average complexities of the CM method, corresponding to the dotted dotted lines and referred to as "MODQO with CM", as well as to the respective lower bounds of the NDQIO process considering all the trellis stages jointly, which is marked by dashed lines and finally to the respective average complexities imposed by the exhaustive search, which are represented by the dashed and dotted lines. In general, observe that the outer step parallel and sequential complexities of all the methods examined increases exponentially, as the number of MCs increases, whereas the respective gradient is substantially reduced, when the number of MRs increases. This is justified by the fact that the average number of Pareto-optimal routes increases almost linearly with the number of MRs, while the total number N_r of active source and destination MCs, which is the exponent of the complexity function, increases almost quadratically with the number of MCs, as it may be inferred based on Eqs. (6.39), (6.79), (6.88) and (6.91).

For networks having 2 MCs, we can observe in Figs. 6.17(a,b) that all the methods considered impose identical parallel and sequential complexities, since a single stage is encountered and no merging takes place. Explicitly, this specific complexity is directly determined by the number of Pareto-optimal routes between the two MCs, since the algorithm has to identify the particular route that exhibits the highest value in terms of the normalized entropy of its associated composite betweenness. Naturally, the merging process imposes non-negligible complexity for networks having 3 or more MCs.

Let us now consider the parallel complexity of the aforementioned algorithms. For the scenario of $N_{\text{MC}} = 4$ MCs, we can clearly observe in Fig. 6.17(a) that the proposed MODQO outer step design, namely the one that invokes the QM method, imposes a lower number of CFEs that its counterparts. More specifically, for networks consisting of $N_{\text{MR}} = 10$ MRs the MODQO outer step relying on the QM method imposes about half the complexity imposed by the CM method and less than a third of the exhaustive search as well as a third of the NDQIO algorithm's lower bound. We note that for this N_{MC} value, the exhaustive search imposes a lower complexity than the minimum required by the NDQIO, which is indeed expected owing to the rather low total number of route combinations, hence not allowing the QP to excel. This trend is reversed however for networks with a higher number of MCs. More specifically for $N_{\text{MC}} = 8$, the MODQO outer step operates at a 2 times to 3.5 times lower complexity than that of the CM method for larger networks having 5 and 10 MRs, respectively, as seen in Fig 6.17(a). More dramatically, it imposes a complexity that is two and four orders of magnitude lower than the lower bound of the NDQIO algorithm and than the average exhaustive search complexity. Finally, for networks having $N_{\text{MC}} = 16$ MCs, the MODQO outer step imposes almost 6 times lower number of CFEs than that of the CM method for $N_{\text{MR}} = 10$ MRs and several orders of magnitude less than the exhaustive search and the lower bound of the NDQIO algorithm.

In a nutshell, we expect the parallel complexity reduction offered by the QM method of the MODQO outer step to increase even further, as the number of MCs increases. Recall that this complexity reduction offered by the QM method of the MODQO outer step is on the order of $O\left(\rho^{-2}\bar{N}_{\text{OPF}}^{2}\right)$ when compared to the CM method, based on Eqs. (6.79) and (6.88). Explicitly, as the numbers of MRs and MCs increase, the average number $\bar{N}_{\text{OPF}}$ of Pareto-optimal routes increases, while the surviving route-combinations' growth factor ρ decreases with respect to $\bar{N}_{\text{OPF}}$, as we demonstrated in Fig. 6.16, which drives the associated complexity reduction to even higher levels.

As for the sequential complexity of the aforementioned algorithms, observe in Fig. 6.17(b) that for the scenario of $N_{\text{MC}} = 4$ MCs the proposed MODQO outer step offers a complexity reduction by a factor of two against the NDQIO algorithm's lower bound. Explicitly, this complexity reduction is improved as the number of MCs increases, as seen in Fig. 6.17(b). Explicitly, this ever reduced sequential complexity demonstrates the effect of the database correlation, which our proposed MODQO outer step exploits. By contrast, for the scenario of $N_{\text{MC}} = 4$ MCs the exhaustive search imposes almost half the complexity of the MODQO outer step for networks having 5 MRs, while this discrepancy seems to decrease as the number of MRs increases, ultimately imposing almost the same sequential complexity as the proposed MODQO outer step for networks having 10 MRs. However, this trend is reversed for networks having more than 4 MCs, since the MODQO outer step offers a sequential complexity reduction of several orders of magnitude. Additionally, observe in Fig. 6.17(b) that for the scenario of $N_{\text{MC}} = 4$ MCs the MODQO outer step imposes almost twice the complexity of the CM method for networks having 5 to 10 MRs. This gap between the proposed MODQO outer step and the CM method tends to subside for networks associated with a higher number of MCs. More specifically for networks having $N_{\text{MC}} = 16$ MCs, the proposed MODQO outer step begins to offer a beneficial complexity reduction compared to the CM method for networks having more than 8 MRs. This trend is consistent with that of Fig. 6.16, where the growth factor ρ remains below the bound of Eq. (6.94). Based on this specific trend, we expect the MODQO outer step to offer an increasing complexity reduction compared to the CM method, as both the MCs and the MRs proliferate.

6.4.1.3 Total Complexity

Having characterized both the inner and the outer steps of the MODQO algorithm, let us now provide some further insights into the total complexity trends quantified in terms of the number of CFEs. Explicitly, the MODQO algorithm's total complexity as a function of the average number $\bar{N}_{\text{OPF}}$ of Pareto-optimal routes and of the surviving route-combinations' growth factor ρ is derived as follows:

$$L_{\text{MODQO}}^{\{P,S\},\text{tot}} = L_{\text{MODQO}}^{\{P,S\},\text{inner}} + L_{\text{MODQO}}^{\{P,S\},\text{outer}} + \rho^{Nr-1}\bar{N}_{\text{OPF}}, \tag{6.110}$$

where $L_{\text{MODQO}}^{\text{inner}}$ and $L_{\text{MODQO}}^{\text{outer}}$ correspond to the complexity imposed by the MODQO algorithm's inner and outer steps, respectively, while the last factor $\rho^{Nr-1}\bar{N}_{\text{OPF}}$ is equal to the number of Pareto-optimal route-combinations at the termination of the QM procedure. This specific factor accounts for the selection of the route-combination S exhibiting the highest value of normalized entropy for its normalized composite betweenness $\bar{H}[\bar{B}_{com}(S)]$ at the very last iteration of the QM method. Consequently, we may readily derive the lower and upper bounds of the total parallel complexity associated with the best- and worst-case scenarios, respectively, as follows:

$$L_{\text{MODQO,min}}^{P,\text{tot}} = O(\sqrt{N}), \tag{6.111}$$

$$L_{\text{MODQO,max}}^{P,\text{tot}} = O(N_{\text{MR}}^{3}\sqrt{N} + N_{\text{MR}}^{2N_{\text{MC}}^{2}}), \tag{6.112}$$

where Eq. (6.111) is derived by substituting Eqs. (6.74) and (6.100) into Eq. (6.110), while Eq. (6.112) is derived by substituting Eqs. (6.75) and (6.100) into Eq. (6.110). Equivalently, the respective lower and upper bounds of the total sequential complexity are given by:

$$L^{P,\text{tot}}_{\text{MODQO,min}} = O(\sqrt{N}), \tag{6.113}$$
$$L^{P,\text{tot}}_{\text{MODQO,max}} = O(N^4_{\text{MR}}\sqrt{N} + N^{5N^2_{\text{MC}}/2}_{\text{MR}}), \tag{6.114}$$

where Eq. (6.113) is derived by substituting Eqs. (6.76) and (6.100) into Eq. (6.110), while Eq. (6.114) is derived by substituting Eqs. (6.77) and (6.100) into Eq. (6.110). Therefore, a significant complexity reduction in both domains is achieved even for the worst-case scenario as opposed to the naive exhaustive search, which would check every legitimate route-combination, constituted by all the possible Hamiltonian routes, and would impose a complexity on the order of $O(N^{2N^2_{\text{MC}}})$ with $O(N) \gg O(N_{\text{MR}})$. Additionally, the MODQO-CM method, which incorporates first the MODQO inner step and then the CM method as its outer step, exhibits the same upper and lower bounds of parallel as those of the MODQO algorithm, based on Eqs. (6.46) and (6.45). Nevertheless, since the MODQO algorithm's lower bound of parallel complexity is based on no complexity being imposed by its outer step, the MODQO algorithm's total parallel complexity is upper bounded by that of the MODQO-CM algorithm.

Both the MODQO and the MODQO-CM algorithms' average parallel complexities quantified in terms of their imposed number of CFEs are presented in Fig. 6.18(a) for networks consisting of 5 to 10 MRs and of 2, 4, 8, and 16 MCs. The parallel complexities of these two algorithms are compared to that of the exhaustive search, which carries out two separate exhaustive search procedures, namely one for the inner and one for the outer step as represented by the black dashed and dotted lines. We note that in Fig. 6.18(a) the MODQO algorithm is labeled as "MODQO using QM" and its average total complexity portrayed with a blue solid line, whilst the MODQO-CM algorithm is labelled "MODQO using CM" and its average total complexity is portrayed by a dotted line. For networks having $N_{\text{MC}} = 2$ MCs, there will be a single active source and destination MC pair, hence the MODQO and the MODQO-CM algorithms impose an identical number of CFEs, which is more than four orders of magnitude below the exhaustive search procedure's average total complexity. Indeed, this complexity advantage increases even further as the number of MRs increases. This is justified by the fact that their respective outer step requires no CFEs to identify the Pareto-optimal route-combinations in the presence of a single trellis stage. As the number of MCs increases, the complexity reduction offered by the MODQO algorithm with respect to the MODQO-CM increases. More specifically, observe in Fig. 6.18(a) that for networks having $N_{\text{MC}} = 4$ MCs, our proposed MODQO algorithm imposes 3% fewer CFEs than the MODQO-CM. Explicitly, in this case the order of both algorithms' complexities is governed by that of the inner step, which is one order of magnitude higher than that of the algorithms' outer steps, as seen in Figs. 6.15 and 6.17(a).

This effect can be observed for networks comprised by a higher number of MCs as well, yielding an interesting trade-off. Explicitly, for networks having $N_{\text{MC}} = 16$ MCs, observe in Fig. 6.18(a) that the MODQO algorithm offers an almost constant complexity reduction factor of 2.5 compared to the MODQO-CM alg. for N_{MR} values up to 8 MRs.

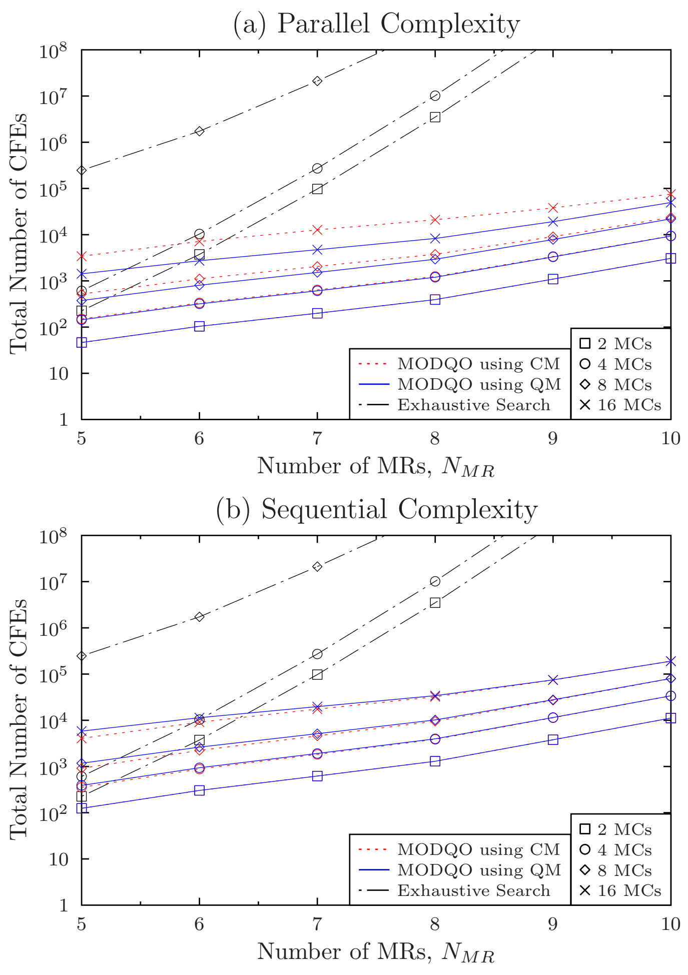


Figure 6.18: Average MODQO outer step (a) parallel and (b) sequential complexities quantified as a function of the number of CFEs for twin-layer networks consisting of $N_{\mathrm{MR}} = \{5, 6, \ldots, 10\}$ MRs and $N_{\mathrm{MR}} = \{2, 4, 8, 16\}$ MCs. The MODQO average complexity is compared to that of the exhaustive search and that of the MODQO using the CM method. The mean complexity results have been averaged over 10^8 runs for twin-layer networks based on the optimization problem of Eq. (6.22) relying on the UV defined in Eq. (6.21) and on the assumptions of Table 6.1.

However, for larger networks the complexity reduction is gradually eroded due to the steep rise in the inner step's complexity, which dominates the total complexity. This drives the MODQO total complexity to an asymptotic convergence woth that of the MODQO-CM, as the number of MRs increases. However, based on Fig. 6.18(b), in practical scenarios, where the total number N_{MC} of MCs is significantly higher than N_{MR}, the MODQO algorithm will outperform the MODQO-CM in terms of the required number of CFEs and the complexity reduction offered by the MODQO will increase as the number of MCs increases, which is owing to a better exploitation of the QP.

Finally, as far as the sequential complexity is concerned, observe in Fig. 6.18(b) that for the examined networks associated with 5 to 10 MRs and with 2, 4 , 8 and 16 MR, the MODQO algorithm imposes roughly the same sequential complexity as the MODQO-CM. This is justified by the fact that their inner step complexity, which is identical for both algorithms, is the predominant factor of the total sequential complexity. More specifically, for networks having less than 16 MCs the MODQO-CM method offers a slight complexity reduction, while for the scenario of 16 MCs this trend changes, since the MODQO algorithm imposes a slight complexity reduction for networks having more than 9 MRs. However, since we have demonstrated in Section 6.4.1.1 that the inner step complexity is upper bounded, we can surmise that the MODQO algorithm is capable of offering a significant sequential complexity reduction for a sufficiently high number MRs and MCs.

6.4.2 Accuracy

Having fully characterized the MODQO algorithm in terms of its complexity, let us now proceed by assessing its accuracy. Explicitly, we have analytically proven that the search space transformation relying on Proposition 1 attains a full-search-based accuracy and it has been demonstrated in Section 5.5.2 that the NDQIO algorithm approaches a full-search-based accuracy as well. Consequently, we can surmise that the MODQO algorithm also exhibits a full-search-based accuracy. Therefore, instead of comparing the MODQO algorithm's accuracy to that of the naive exhaustive search, we will use as a benchmark algorithm the state-of-the-art multi-objective evolutionary algorithm, namely the NSGA-II [124, 29], which we discussed in Section 2.4. Note that we have opted out of comparing the MODQO to the MO-ACO implementation presented in Section 2.5, since our intention was to directly address the composite problem without decomposing it. Explicitly, our forthcoming case study will examine as to whether the MODQO algorithm strikes an efficient accuracy versus complexity trade-off, when compared to the NSGA-II. For this reason, we will compare these algorithms' accuracy to each other, when both operate at the same complexity, quantified in terms of CFEs.

We note that we have adapted the NSGA-II presented in Section 2.4 so that it can benefit from the search space transformation of Proposition 1. To elaborate further, we have assumed that each of the individuals is constituted by multiple chromosomes, each corresponding to a Hamiltonian route for a specific pair of source and destination MCs. Furthermore, during the mating process independent crossover and mutation operations are performed for each of the chromosomes. For the sake of simplicity, we have assumed that the number N_{pop} of individuals per generation is equal to the number N_{gen} of generetions, i.e. we have:

$$N_{\text{pop}} = N_{\text{G}}, \tag{6.115}$$

yielding a total complexity in terms of the number of CFEs, which is equal to:

$$L_{\text{NSGA-II}} = N_{\text{pop}}^3, \tag{6.116}$$

since the non-dominated sort requires precisely N_{pop}^2 CFEs for N_{pop} generations. The simulation parameters considered for the NSGA-II are presented in Table 6.7, where the number N_{pop} of individuals was set to match the maximal total parallel complexity of the MODQO algorithm observed throughout the simulations characterized in Fig. 6.18. Note that we have opted for rounding up the number of individuals N_{pop} to the next number divisible by 4, since we have to produce two mating pools, each having $N_{\text{pop}}/4$ individuals.

The rest of the parameters values have been optimized through extensive simulations and they were found to differ from the optimal values of Table 2.2 corresponding to the single source and single destination scenario.

Table 6.7: NSGA-II Simulation Parameters

Parameter	Value
Number of individuals, N_{pop}	$\{8, 8, 8, 12, 16, 20\}$ for $N_{\text{MC}} = 2$ MCs
	$\{12, 12, 12, 16, 16, 20\}$ for $N_{\text{MC}} = 4$ MCs
	$\{16, 16, 20, 20, 28, 28\}$ for $N_{\text{MC}} = 8$ MCs
	$\{24, 28, 28, 36, 36, 36\}$ for $N_{\text{MC}} = 16$ MCs
Crossover Probability, P_c	0.8
Mutation Probability, P_m	0.1

6.4.2.1 Accuracy Comparison

Ideally we would have to compare the MODQO algorithm and the NSGA-II in terms of the average of the identified OPF formed by the Pareto-optimal route combinations as far as the average network delay and the average network power consumption are concerned. However, the visualization of the OPF would complicate the representation of the results rendering the related trends rather opaque. For the sake of simplifying the presentation of the results, we will assess both algorithms' performance by providing the simulation results for the Pareto-optimal solutions having four distinct characteristics. First, we have to assess the networks' limits in terms of the WMN QoS criteria considered, namely the average minimum network delay $\min\{\bar{D}(S)\}$ and the average minimum network power consumption $\min\{\bar{P}(S)\}$. In addition to these metrics, we will provide the Pareto-optimal solutions of the maximum normalized entropy of the normalized composite betweenness $\max\{\bar{H}[\bar{B}_{com}(S)]\}$ and compare it to the one exhibiting the minimum standard deviation of the normalized composite betweenness $\min\{\sigma_{\bar{B}_{com}}\}$ for the sake of assessing the proposed load balancing strategy.

Let us now proceed by jointly assessing the MODQO performance in terms of the QoS criteria considered for the WMN layer, namely the average network delay performance $\bar{D}$ per route quantified in terms of the number of established hops and average network power consumption $\bar{P}$ per route in dBm. The aforementioned metrics are portrayed in Figs. 6.19 and 6.20 for networks having 5 to 10 MRs for 2, 4, 8 and 16 MCs. As far as the delay is concerned, observe in Fig. 6.19 that the minimum delay achieved by the MODQO algorithm slightly increases, as the number N_{MR} of MRs increases, regardless of the number of MCs considered. This is justified by the fact that as the number of MRs increases, the probability of two specific MCs being associated with the same MR decreases, hence reducing the probability of establishing a connection with the minimum possible delay of two hops. On the other hand, the minimum power consumption portrayed in Fig. 6.20 is governed by a pair of conflicting dynamics. To elaborate further, as the number N_{MR}

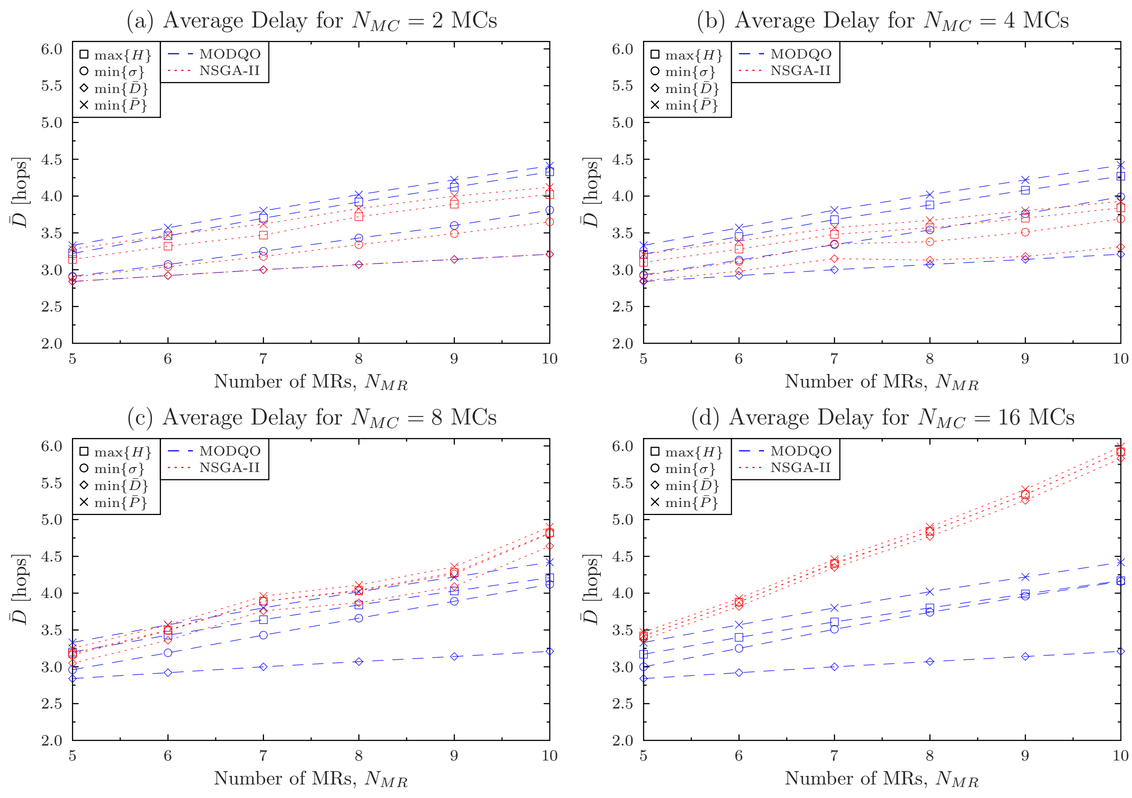


Figure 6.19: Average network delay performance $\bar{D}$ per route in terms of the number of established hops for both the MODQO algorithm and the NSGA-II for networks having from 5 to 10 MRs and associated with (a) $N_{\text{MC}} = 2$ MCs, (b) $N_{\text{MC}} = 4$ MCs, (c) $N_{\text{MC}} = 8$ MCs and (d) $N_{\text{MC}} = 16$ MCs. The NSGA-II initialization parameters are presented in Table 6.7. The results have been averaged over 10^8 runs for twin-layer networks based on the optimization problem of Eq. (6.22) relying on the UV defined in Eq. (6.21) and on the assumptions of Table 6.1.

of MRs the distances between the MRs decrease, hence the shorter links require a lower power, while if the probability of two MCs being associated with the same MR decreases, this virtually increases the average distance among the MCs quantified in terms of the number of hops. The latter is justified by the fact that the MCs tend to be associated with their closest MRs; however, this does not necessarily imply that the MR association is optimal in terms of the routes' power consumption, since an MR that is closest to the source MC can potentially be located further away from the destination MC, hence increasing in the average power consumption.

As for the load balancing metrics, namely the maximum normalized entropy of the normalized composite betweenness distribution and minimum standard deviation of the specific distribution, which are denoted by max$\{\bar{H}\}$ and min$\{\sigma\}$, respectively, observe in Figs. 6.19 and 6.20 that they both lie between the two extreme strategies, regardless of the number of MCs considered as well. Additionally, we can observe in Fig. 6.19 that the specific min$\{\sigma\}$-strategy that minimizes the standard deviation of the composite betweenness distribution seems to be biased towards the minimum-delay solution. This trend is more distinguishable for lower number of MCs, namely for 4 and 8 MCs of Figs. 6.19(a,b), respectively. It is justified by the fact that this strategy considers to be optimal that specific route-combination, which utilizes no intermediate MRs in the construction of all the

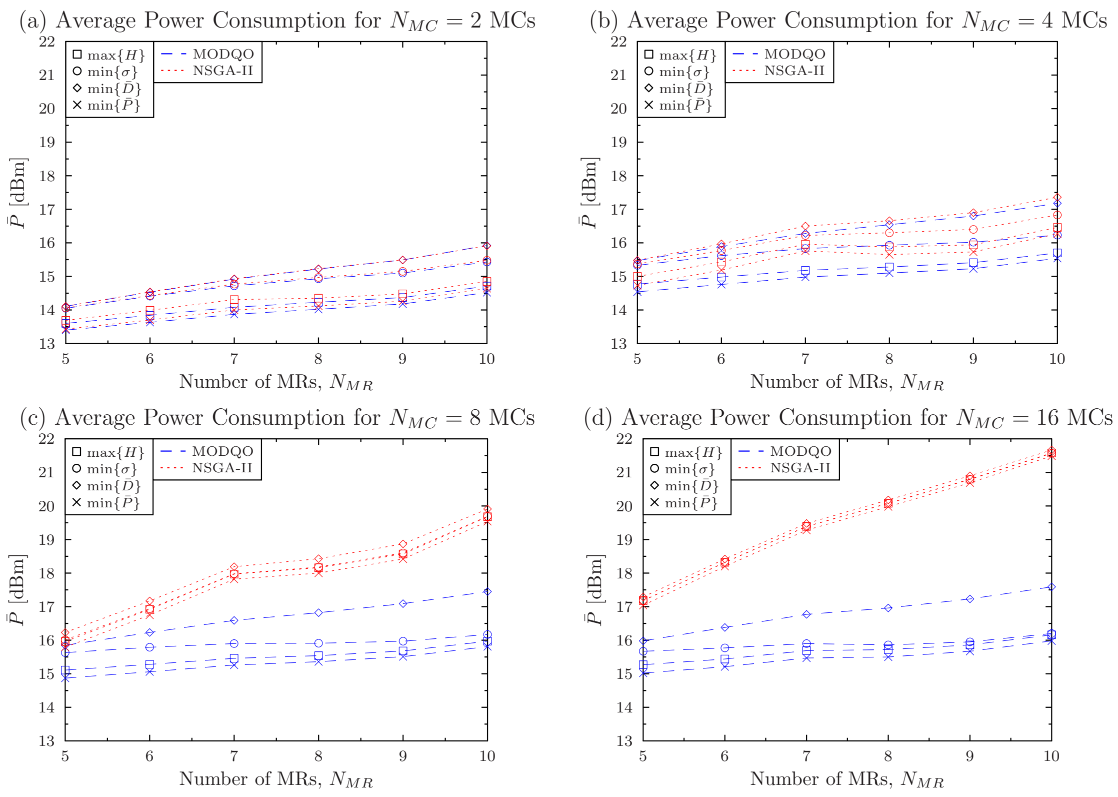


Figure 6.20: Average network power consumption $\bar{P}$ per route in dBm for both the MODQO algorithm and the NSGA-II for networks having from 5 to 10 MRs and and associated with (a) $N_{\mathrm{MC}} = 2$ MCs, (b) $N_{\mathrm{MC}} = 4$ MCs, (c) $N_{\mathrm{MC}} = 8$ MCs and (d) $N_{\mathrm{MC}} = 16$ MCs. The NSGA-II initialization parameters are presented in Table 6.7. The results have been averaged over 10^8 runs for twin-layer networks based on the optimization problem of Eq. (6.22) relying on the UV defined in Eq. (6.21) and on the assumptions of Table 6.1.

routes. By contrast, while in the absence of such a route it will identify as optimal the same route-combination as that of the particular strategy aiming for maximizing the normalized entropy, yielding max$\{\bar{H}\}$. This explains the trend that the min$\{\sigma\}$-strategy exhibits lower average delay in Fig. 6.19 and a higher average power consumption in Fig. 6.20 than those of the max$\{\bar{H}\}$-strategy for networks having less than 8 MRs. By contrast, for a higher number of MRs the performance of the min$\{\sigma\}$-strategy asymptotically converges to that of the max$\{\bar{H}\}$-strategy. Explicitly, as the number N_{MR} of MRs increases, the probability of forming a Pareto-optimal route-combination without the involvement of intermediate MRs decreases, since the MCs tend to become more distant in terms of the number of hops, as the network becomes populated by more MRs.

Additionally, we can observe both in Fig. 6.19 and in 6.20 that the proposed maximum-entropy strategy yielding max$\{\bar{H}\}$ exhibits an average delay that is about 0.3 hops lower than that of the strategy minimizing the average power consumption, namely min$\{\bar{P}\}$, for networks having $N_{\mathrm{MR}} = 5$ MRs associated with 8 and 16 MCs. This performance-discrepancy widens, as the number of MRs increases, reaching a reduction of 0.5 hops for networks having $N_{\mathrm{MR}} = 10$ MRs. Naturally, this reduction comes at a cost of about 0.3 dB in terms of the average power consumption, as observed in Fig. 6.20. Explicitly, this delay reduction exhibits an underlying trade-off among the max$\{\bar{H}\}$, the min$\{\bar{P}\}$ and the $\min\{\bar{D}\}$ strategies: the normalized composite entropy asymptotically converges to the uniform distribution, as and when more MRs become involved as intermediate relays and reaches its minimum divergence for the route-combination of the $\max\{\bar{H}\}$ strategy. From this point onwards, an increase in the number of MRs results in the central MRs becoming bottlenecks, hence driving the normalized composite entropy further away from the uniform distribution. However, for networks associated with 2 and 4 MCs, observe in Figs 6.19(a,b) that this delay-gap remains almost constant, as the number of MRs increases and it is equal to about 0.2 and 0.3 hops, respectively. For these relatively low numbers of MCs, the total number of route-combinations is rather low, offering less flexibility in the selection of the $\max\{\bar{H}\}$ route-combination.

As for the NSGA-II performance we can observe that it fails to converge to the Pareto-optimal route-combinations of the MODQO algorithm for networks having 16 MCs. More specifically, based on Fig. 6.19(d) and 6.20(d), we can clearly observe that the route-combinations of all four strategies identified by the NSGA-II are dominated by the respective ones identified by the MODQO algorithm for networks having more than $N_{\text{MR}} = 7$ MRs. More specifically for networks having $N_{\text{MR}} = 10$ MRs, the MODQO algorithm achieves a power-reduction of at least 4 dB and a delay-reduction of at least 1.5 hops at the same number of CFEs, as seen in Fig 6.19(d). For networks having less than 16 MCs, the NSGA-II becomes “less sub-optimal” as the number of MCs decreases. Therefore, we may conclude that our proposed MODQO algorithm exhibits a better performance versus complexity trade-off associated with identifying the Pareto-optimal solutions, when both the number of MRs and that of MCs increases.

Subsequently, the evaluation to the networks’ load balancing performance is characterized in Figs. 6.21 and 6.22 in terms of the normalized entropy of the normalized composite betweenness distribution and the distribution’s standard deviation, respectively. In a nutshell, we can observe that a more efficient load balancing is performed, as the number of MRs and that of the MCs increase, owing to the increasing number of Pareto-optimal combinations. The specific strategy minimizing the average network delay constitutes an exception. On the one hand it exhibits the lowest value of $\bar{H}(\bar{B}_{com})$ yielding that its respective route-combinations’ $\bar{B}_{com}$ distribution deviates more substantially from the uniform distribution, based on Fig 6.21. On the other hand, as we can observe in Fig. 6.22, this specific strategy exhibits a lower standard deviation for the $\bar{B}_{com}$ distribution in networks having $N_{\text{MR}} = 5$ MRs, owing to the inclusion of direct routes relying on no intermediate relays, which in turn exhibit zero standard deviation. However, as the number N_{MR} of MRs increases, the probability of these specific routes being identified as Pareto-optimal decreases, yielding an increase in the associated standard deviation, which then obeys similar trends to the rest of the strategies. This standard deviation trend is observed in Figs. 6.22(c,d) for the $\min\{\sigma\}$ strategy, where the standard deviation is seen to increase for networks having up to 7 and 8 MRs associated with 16 and 8 MCs, respectively. It then decreases, as the number of MRs increases further. By contrast, in networks associated with 2 and 4 MCs this metric increases with the number of MRs, according to Figs. 6.22(c,d). We note that the respective values of $\bar{H}(\bar{B}_{com})$ seen in Fig. 6.21 for the route-combinations

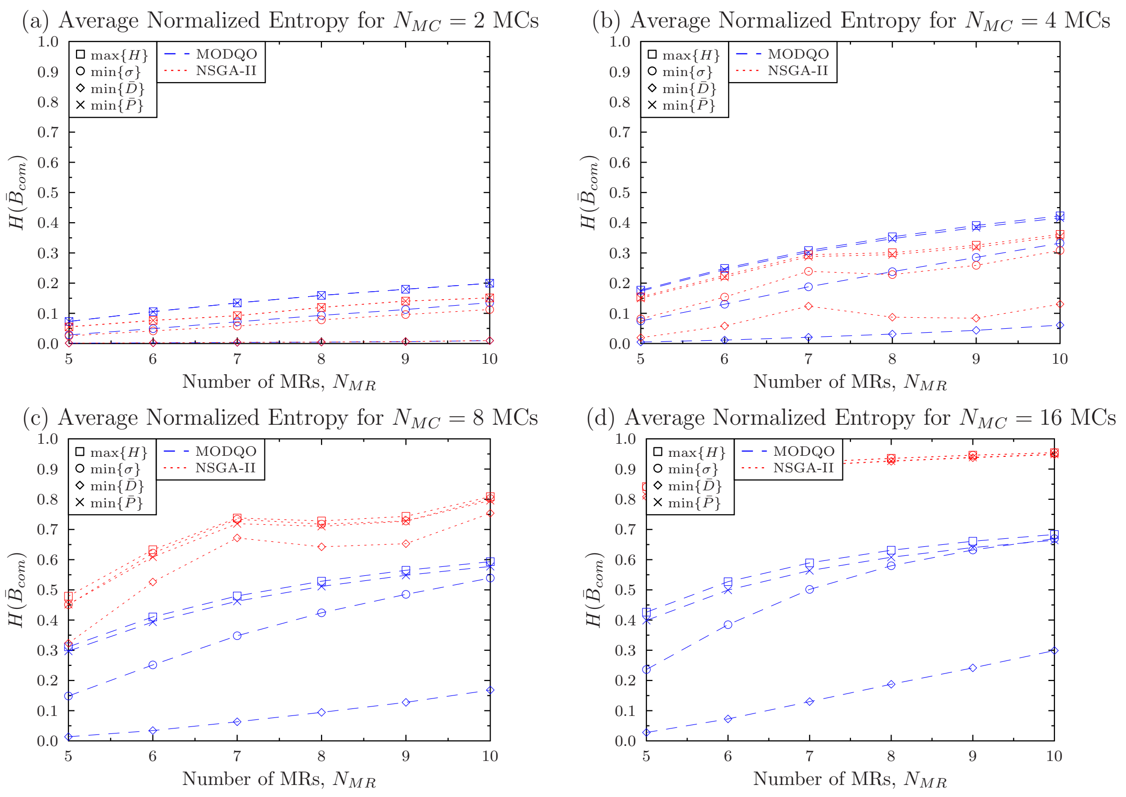


Figure 6.21: Average normalized entropy of the normalized composite betweenness $\bar{H}[\bar{B}_{com}(S)]$ for both the MODQO algorithm and the NSGA-II for networks having from 5 to 10 MRs and associated with (a) $N_{\text{MC}} = 2$ MCs, (b) $N_{\text{MC}} = 4$ MCs, (c) $N_{\text{MC}} = 8$ MCs and (d) $N_{\text{MC}} = 16$ MCs. The NSGA-II initialization parameters are presented in Table 6.7. The results have been averaged over 10^8 runs for twin-layer networks based on the optimization problem of Eq. (6.22) relying on the UV defined in Eq. (6.21) and on the assumptions of Table 6.1.

selected by this strategy are lower than those of both the max$\{\bar{H}\}$ and max$\{\bar{P}\}$ strategies. This exhibits a poorer resemblance to the uniform distribution of the min$\{\sigma\}$ as well as to the max$\{\bar{H}\}$ and max$\{\bar{P}\}$ strategies.

Finally, as far as the NSGA-II algorithm is concerned, we can observe two different trends in Figs. 6.21 and 6.22. For networks associated with 2 and 4 MCs, where the NSGA-II approximates more accurately the Pareto-optimal route combinations, they exhibit a a poorer load balancing capability than the MODQO algorithm's max$\{\bar{H}\}$ strategy, based on Figs 6.21(a,b) and 6.22(a,b). By contrast, for networks having 8 and 16 MCs they exhibit a far better load balancing performance than the MODQO algorithm for all the strategies examined. This is justified by the fact that the route-combinations exported by the NSGA-II do not comply with the constraint of Eq. (6.22), since the Pareto-optimal route-combinations identified the NSGA-II are sub-optimal in comparison to the respective ones identified by the MODQO algorithm. This results in an excessive involvement of MRs for the sake of approximating the uniform distribution, which leads to both an excessive delay and an excessive power consumption. Based on this fact, we can infer that load balancing tends to degrade both the average network delay and the average power consumption. Naturally, based on the problem formulation in Eq. (6.22), load balancing is imposed as a secondary optimization objective, whilst the constraint of Eq. (6.22) con-

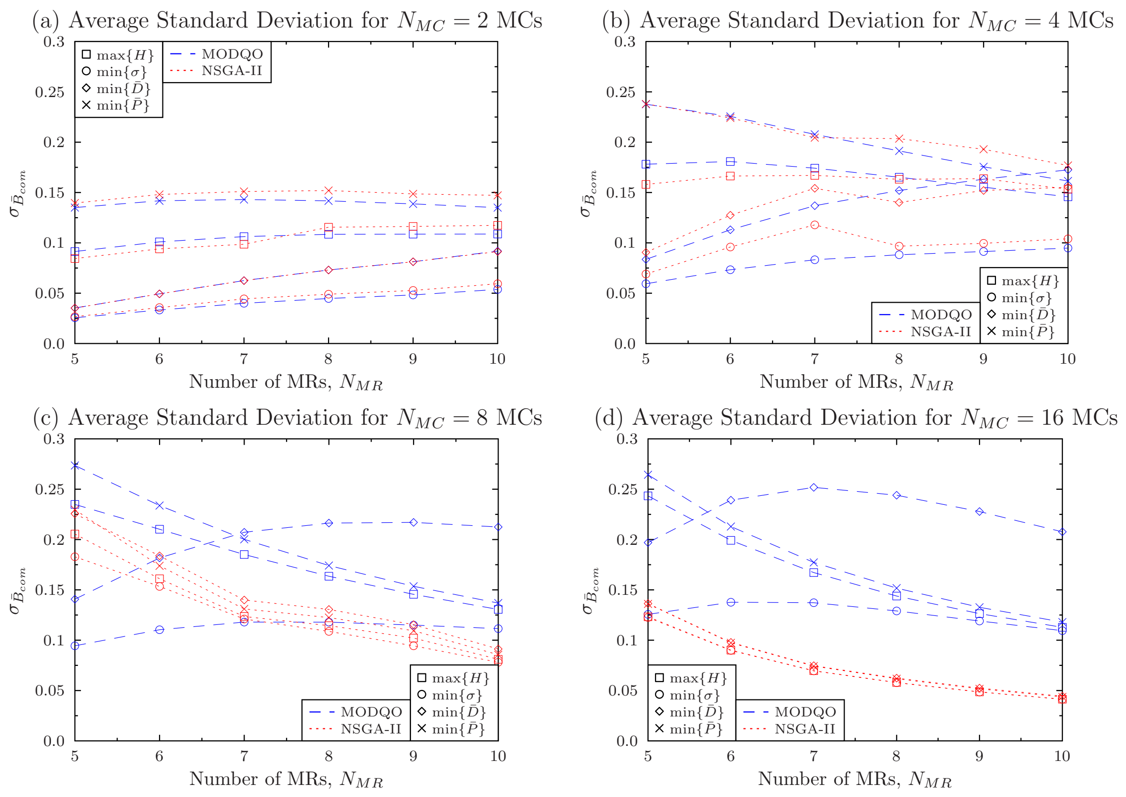


Figure 6.22: Average standard deviation of the normalized composite betweenness $\sigma_{\bar{B}_{com}}$(b) for both the MODQO algorithm and the NSGA-II for networks having from 5 to 10 MRs and associated with (a) $N_{\text{MC}} = 2$ MCs, (b) $N_{\text{MC}} = 4$ MCs, (c) $N_{\text{MC}} = 8$ MCs and (d) $N_{\text{MC}} = 16$ MCs.. The NSGA-II initialization parameters are presented in Table 6.7. The results have been averaged over 10^8 runs for twin-layer networks based on the optimization problem of Eq. (6.22) relying on the UV defined in Eq. (6.21) and on the assumptions of Table 6.1.

stitutes the primary optimization criterion, since it forces the optimization to additionally perform load balancing, while explicitly considering Pareto-optimal route-combinations.

6.5 Chapter Summary

In this chapter, we have proposed an optimal quantum-assisted algorithm, namely the MODQO algorithm, for addressing the joint multi-objective routing and load balancing problem in socially-aware networks. The MODQO algorithm benefits from both a framework exploiting the synergies between the QP and HP, which is inherited by the NDQIO algorithm as well as from the novel database transformation framework advocated. The latter succeeds in transforming the strongly correlated database into a series of weakly correlated ones, where the QP and HP synergistic framework exploits the optimality of Grover's QSA [88]. Additionally, we have analytically proven that this transformation has no negative impact on the MODQO accuracy. Furthermore, we have introduced a novel socially-aware metric for characterizing the load balancing, namely the normalized entropy of the normalized composite betweenness distribution. We have also demonstrated that it succeeds in mitigating the biasing towards the minimum delay solution incurred by the employment of the standard deviation of the respective distribution. Furthermore, we have characterized the computational complexity in terms of the number of CFEs imposed by the MODQO algorithm. In fact, the parallel complexity is on the order of $O(\sqrt{N})$ and $O(N_{\text{MR}}^{2N_{\text{MC}}^2})$ for networks having N_{MR} MRs and N_{MC} MCs in the best- and the worst-case scenarios, respectively. Additionally, the respective upper and lower bounds of the sequential complexity are on the order of $O(\sqrt{N})$ and $O(N_{\text{MR}}^{5N_{\text{MC}}^2/2})$ in the best- and the worst-case scenarios, respectively. Explicitly, we have achieved a significant complexity reduction compared to the exhaustive search, which is on the order of $O(N^{2N_{\text{MC}}^2})$, with $N \gg N_{\text{MR}}$ being the total number of Hamiltonian routes between a pair of specific users. Additionally, we demonstrated using extensive simulations that the average complexity of the MODQO algorithm is multiple orders of magnitude lower than that of the exhaustive search. Finally, we have compared the MODQO algorithm's accuracy to that of the NSGA-II [124, 29], which constitutes the state-of-the-art for socially-oblivious networks. More specifically for a scenario where the network is sufficiently densely populated by MCs, i.e. we have $N_{\text{MC}} = 16$ MCs and have demonstrated that our proposed MODQO algorithm is capable of improving both the delay and the power consumption by about 2 hops and 4 dB, respectively, for networks having 10 routers, when compared to the NSGA-II. This trend suggests that the MODQO algorithm exhibits a better complexity versus accuracy trade-off than the NSGA-II.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this treatise we have developed a variety of quantum-assisted algorithms, namely the *Non-Dominated Quantum Optimization* (NDQO) algorithm [1], the *Non-Dominated Quantum Iterative Optimization* (NDQIO) algorithm [2] and the *Multi-Objective Decomposition Quantum Optimization* (MODQO) algorithm [3] for the multi-objective optimization of routing in *Heterogeneous Networks* (HetNets) [14]. In fact, the aforementioned algorithms constitute an extension of the well-established quantum optimization framework [79,80,81] to Pareto optimality problems, where the concept of *Quantum Parallelism* (QP) [62] is exploited for the sake of achieving near-full-search-based search accuracy at the expense of a substantially reduced complexity. In a nutshell, our contributions, which are illustrated in Fig. 7.1, constitute an extension of the existing quantum-assisted optimization framework of Fig. 3.12 to multi-objective optimization problems relying on the concept of Pareto optimality [47].

As a first case-study for the multi-objective optimization of HetNets, we have considered the multi-objective routing problem of *Wireless Multi-Hop Networks* (WMHNs) [25], where a single source node transmits its message to a single destination node, while utilizing a cloud of mobile relays, as detailed in Section 2.2. For this specific application, we initially proposed in Chapter 4 the NDQO algorithm, which achieves a substantial complexity reduction compared to the *Brute-Force* (BF) search by using a sophisticated quantum-assisted process, namely the so-called *Boyer-Brassard-Høyer-Tapp Quantum Search Algorithm* (BBHT-QSA) *chains* introduced in Section 4.3 for determining as to whether a specific route is Pareto optimal, while being capable of approaching the Pareto optimal routes in case of a sub-optimal route. We have also demonstrated with the aid of Figs. 4.6, 4.7 and 4.8 that the NDQO algorithm exhibits a near-optimal performance approaching that of the full-search-based method, while imposing a complexity on the order of $O(N)$ and $O(N\sqrt{N})$ in the best- and the worst-case scenario, respectively. This is significantly lower than the complexity of $O(N^2)$ imposed by the BF method. Furthermore, in Section 4.6.2 we have compared the NDQO algorithm's accuracy to those of a pair of popular

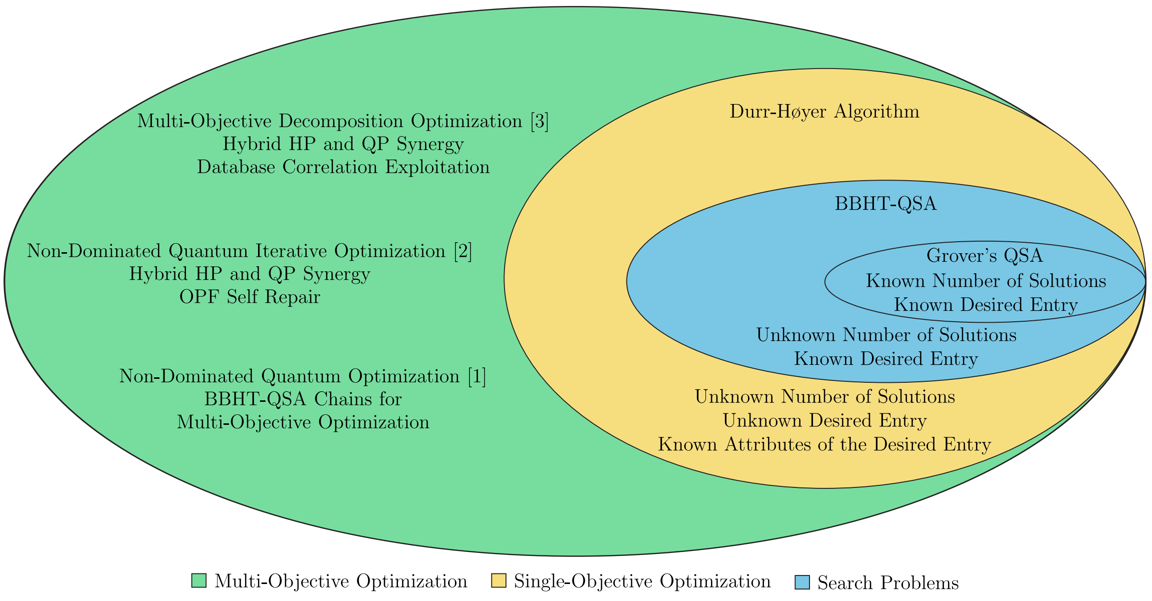

Multi-Objective Decomposition Optimization [3]
Hybrid HP and QP Synergy
Database Correlation Exploitation
Non-Dominated Quantum Iterative Optimization [2]
Hybrid HP and QP Synergy
OPF Self Repair
Non-Dominated Quantum Optimization [1]
BBHT-QSA Chains for
Multi-Objective Optimization
Durr-Høyer Algorithm
BBHT-QSA
Grover's QSA
Known Number of Solutions
Known Desired Entry
Unknown Number of Solutions
Known Desired Entry
Unknown Number of Solutions
Unknown Desired Entry
Known Attributes of the Desired Entry
🟩 Multi-Objective Optimization
🟨 Single-Objective Optimization
🟦 Search Problems

Figure 7.1: Extension of Fig. 3.12 based on our contributions in this report.

multi-objective evolutionary algorithms, namely to that of the *Non-dominate Sort Genetic Algorithm II* (NSGA-II) [124, 29] and that of the *Multi-Objective Ant Colony Optimization* (MO-ACO) [52], which have been discussed in Sections 2.4 and 2.5, respectively. We have demonstrated with the aid of Figs. 4.6, 4.7 and 4.8 that the NDQO algorithm is capable of ensuring the identification of the entire set of Pareto-optimal solutions, while exhibiting an almost negligible error-floor, which is several orders of magnitude lower than that of the NSGA-II and of the MO-ACO algorithm, while imposing the same complexity. However, when the number of mobile relays is high, the multi-objective routing problem cannot be realistically solved by the NDQO algorithm, even when considering its best-case scenario.

This observation is as our main motivation in Chapter 5, where we have introduced an improved algorithm, namely the NDQIO algorithm. Explicitly, this specific algorithm is capable of exploiting the hybrid synergies between *Hardware Parallelism* (HP) and *Quantum Parallelism* (QP) based on the framework we developed in Section 5.2. Note that due to the presence of HP we have classified the complexity imposed by the NDQIO algorithm into two separate domains, namely the parallel and the sequential complexities, as detailed in Section 5.2. Based on this framework, in Section 5.3.2 we have introduced an element of elitism into our new algorithm, enabling the NDQIO algorithm to terminate its operation, once it concludes that all the Pareto-optimal routes have been identified. This element of elitism allows the NDQIO algorithm to a substantial parallel complexity reduction when compared to the NDQO algorithm. Explicitly, the NDQIO algorithm imposes a parallel complexity on the order of $O(\sqrt{N})$. In addition to the aforementioned meritorious attributes, we have also employed a process for pushing the NDQO algorithm's error-floor to infinitesimally low levels by protecting the *Optimal Pareto Front* (OPF) against the inclusion of sub-optimal routes by relying on the identification of the entire set of Pareto-optimal routes. This specific process is termed as the *OPF Self-Repair* (OPF-SR) process and it is discussed in Section 5.3.3. Nevertheless, the sequential complexity of the worst-case scenario is increased by the above particular set of improvements, leading to a sequential complexity on order of $O(N^2\sqrt{N})$. Furthermore, we have demonstrated in Section 5.5 that the NDQIO algorithm exhibits a full-search-based accuracy associated with an infinitesimally low error-floor, while imposing a parallel complexity of about an order of magnitude lower than that of the NDQO algorithm for 9-node WMHNs. This is attained at the cost of twice the sequential complexity, when compared to that of the NDQO algorithm. Explicitly, based on Fig. 5.9 this parallel complexity reduction is deemed to be further increased in more densely populated WMHNs. Based on the same figure, the NDQIO algorithm's sequential complexity approaches that of the NDQO algorithm, as the relays proliferate, whilst the NDQIO algorithm is expected to exhibit a beneficial sequential complexity reduction for WMHNs associated with more than 12 nodes.

As discussed in Section 5.2, the parallel and sequential complexities may be deemed to be commensurate with the normalized execution time and the normalized power consumption of the algorithm. Therefore, a normalized execution time versus normalized power consumption trade-off emerges. To elaborate further, the NDQIO algorithm is more appropriate for applications, where the relays are moving at a high vehicular speed, yielding a swift change in the network specification parameters and hence having a low normalized execution time is crucial for maintaining optimal routing. In this case, the NDQO algorithm would fail to export the Pareto-optimal solutions, since the real network parameters would diverge from the specific ones considered in the optimization problem. On the other hand, there are WMHNs applications, where the number of nodes is less than 12 and they are stationary or slowly moving. In this specific scenario, the WMHN parameters do not drastically fluctuate over time and thus the NDQO algorithm's employment is more appropriate, since it would contribute to increasing the WMHN's lifetime, due to the reduced normalized computational power requirement compared to that of the NDQIO algorithm.

In Chapter 6, we have opted for including an additional objective into the HetNet's optimization, namely that of load balancing. For this specific reason, we adapted our network model so that it considers multiple source and destination nodes as well as exploiting an element the element of *social awareness* [159], as discussed in Section 6.2. Therefore, in this chapter we have introduced a novel quantum-assisted algorithm, namely the MODQO algorithm [3], for the sake of addressing the joint multi-objective routing and load balancing problem in the context socially-aware networks. The MODQO algorithm relies both on the hybrid synergy between the QP and the HP and on a novel framework that we developed in Section 6.3.4 for transforming the composite search space into a series of less-complex spaces. Consequently, the MODQO algorithm also benefits from a third complexity reduction source, namely that of the database correlation exploitation. Explicitly, we have shown in Proposition 1 that the Pareto-optimal route-combinations are exclusively comprised by individually Pareto-optimal routes. Consequently, we only have to consider the set individually Pareto-optimal routes for identifying the Pareto-optimal combinations, as portrayed in Figs. 6.12 and 6.13. This further improves the complexity reduction offered by the QP [88], since the series of smaller search spaces appear to have a significantly reduced correlation. Additionally, we have developed a novel quantum-assisted heuristic method for iteratively identifying the route-combinations formed by the individual Pareto-optimal routes of these smaller search spaces, which was termed in Section 6.3.2 as MODQO algorithm's *outer step* and its is discussed in Section 6.3.4. Explicitly, we have demonstrated with the aid of Fig. 6.17 that the aforementioned quantum-assisted iterative process offers a substantial complexity reduction, when compared to employing the NDQIO algorithm for the longer-dimensional composite problem. *This specific finding is of utter importance, since it provides us with specific guidelines as to when the employment of full-search-based quantum algorithms is beneficial. Naturally, their employment becomes more beneficial, when the database considered becomes more uncorrelated, as observed in Fig. 6.17.*

Additionally, we have introduced a novel socially-aware load balancing metric in the context of the joint multi-objective routing and load balancing problem, namely the *normalized entropy of the normalized composite betweeness* defined in Eq. (6.5). Based on Figs. 6.19 and 6.20, we have demonstrated that this metric succeeds in mitigating the bias towards the minimum-delay solution, which is imposed by using the standard deviation of the normalized composite betweeness. Finally, we have compared the MODQO algorithm's performance to that of the NSGA-II. Based on Figs. 6.19–6.22, we have concluded that when the NSGA-II identifies sub-optimal route-combinations in terms of their average delay and power consumption, the load balancing performance quantified in terms of both metrics considered becomes better than that of the MODQO algorithm. On the other hand, when the NSGA-II approaches the Pareto-optimal route combinations, its load balancing performance becomes worse than that of the MODQO algorithm. This specific trend unveils a trade-off, namely the load balancing versus optimal routing trade-off. Explicitly, the network's load balancing performance improves, when the routing becomes more and more sub-optimal, since including more intermediate nodes drives the load distribution closer to uniform distribution.

7.2 Future Work

Apart from the aforementioned quantum-assisted multi-objective algorithms, which we designed in the context of routing or load balancing, there are various others, such as the *Quantum Genetic Optimization Algorithm* (QGOA) [89] and the *Quantum Search Heuristics* (QSH) [198] algorithm. Therefore, we have in our possession a variety of quantum algorithms to address our main problem, which is that of conceiving a quantum-assisted holistic network optimizer for achieving near-capacity performance in HetNets. However, in our quest for this optimizer we have to address several problems besides the routing and the load balancing problems. Therefore, our future work will be focused on the following issues:

(a) In Chapter 6 we investigated how to exploit the underlying correlations in the database for the formation of the Pareto-optimal routes. The Pareto-optimal routes' formation also exploit the database correlation. In fact, several studies [199, 200] transformed the single-component routing optimization problem into a dynamic programming problem by creating a trellis diagram. This specific structure is in line with MODQO algorithm's outer step described in Section 6.3.4, hence it could be readily exploited for the sake of extending the aforementioned single-component framework to a multi-objective technique by using the concept of Pareto optimality.

(b) A rather promising application for Pareto optimality problems can be identified in the field of *user-centric networking* [22,201,202]. Explicitly, the cluster head may readily assume a multi-component *Utility Vector* (UV) comprised by each user's QoS criteria. Consequently, our NDQO and NDQIO algorithms [1,2] become readily applicable in these problems.

(c) Our network models used in Sections 2.2 and 6.2 may be enriched by introducing the principle of network coding [203,204,205] into our HetNet paradigm and design a quantum-assisted scheduler in the same fashion, as our quantum-assisted MODQO and NDQIO algorithms [3].

(d) Apart from network coding, the network model may be revisited in the context of *cooperative energy harvesting* [206]. To elaborate further, an *energy buffer* [207, 208] as well as a *memory buffer* [209, 210] could be considered at every node for supporting opportunistic forwarding. In this specific scenario, our quantum-assisted multi-objective algorithms may be readily invoked for the sake of investigating the achievable rate versus energy efficiency trade-off [211].

(e) As an extension to our joint multi-objective routing and load balancing, we may employ our multi-objective framework constituted by the MODQO, the NDQIO and the NDQO algorithms for addressing the problem of *proactive caching* [212,213,214,30]. In proactive caching the packets are buffered in the nodes by carefully considering their popularity for the sake of reducing both the delay and the power consumption, which is reminiscent of our multi-objective routing problem. Additionally, this specific case study could be undertaken with the aid of *machine learning* [215]. In fact, Kapoor *et al.* [216] have recently proposed a model for quantum perceptrons, which may constitute beneficial building blocks for quantum-aided neural networks. Therefore, it would be worth investigating as to whether our quantum-assisted solutions adapted to this context.

(f) Our quantum-assisted framework operates under the assumption of error-free quantum circuits. In fact, we have only investigated the effect of imperfect quantum circuits [73] in the context of Grover's algorithm. Therefore, as an extension we could invoke *Quantum Error Correction* (QEC) codes [68,70,72] for the BBHT-QSA and the DHA leading to powerful multi-objective algorithms.

(g) Finally, our advocated quantum assisted solutions could be adapted to localization-aided networking problems [217,218,219,220,221,222].

Appendix A

Routing Problem Transformation into Binary Combinatorial Search using Lehmer Encoding and Decoding Processes

A.1 Introduction to Lehmer Coding

It has been argued in Section 3.2 that quantum systems comprised by qubits have an inherently binary nature. Consequently, the multiple-objective routing problem has to be transformed into an equivalent binary combinatorial problem. This transformation becomes feasible, when each of the routes is mapped into an individual binary index. However, an appropriate transformation pattern has to be defined, so that the inverse transformation is also feasible. The latter is of utter importance, since the superimposed binary states have to be directly mapped into their respective routes using unitary transformations.

Since we have imposed the constraint of having Hamiltonian routes, implying that each node is to be traversed at most once, each route can be viewed as a permutation of the RNs involved, with the source and destination nodes being appended at the beginning and the end of the route, respectively. For this reason, we will rely on *Lehmer Coding* [137] as a benefit of its capability of encoding a permutation into its respective index in the factoradic[1] computational basis. Explicitly, we can then map the respective indices in the factoradic computational basis to binary indices. This enables the employment of unitary operators U_f, portrayed in Fig. 3.1, which decode the routes' indices in an online fashion in parallel by exploiting the QP and then evaluate each of the components of UV considered in Eq. (2.5). Note that all the unitary operators U_{f_k}, defined in # in the next chapters,

[1]The factoradic number system is a mixed radix numeral system used for describing permutations, where the i-th digit from the right has the base i, implying that this specific digit must be strictly less than i and that its value is multiplied by $(i1)!$.

incorporate *Lehmer Decoding* for the sake of determining from each route's binary index the directed set of nodes creating the route.

Having described our motivation for employing Lehmer code, let us proceed with its detailed description. Assuming a random permutation $\sigma = [\sigma_0, ..., \sigma_{n-1}]$ of n elements, its *Lehmer Encoding* $L_e(\sigma)$ is defined as follows [137]:

$$L_e(\sigma) = \left[L_e(\sigma)_0, ..., L_e(\sigma)_{n-1}\right], \tag{A.1}$$

where each element of $L_e(\sigma)$ is defined as:

$$L_e(\sigma)_i = \#\{j > i \ : \ \sigma_j < \sigma_i\}. \tag{A.2}$$

Explicitly, the operator $\#\{\cdot\}$ corresponds to the number of elements that satisfy the statement inside the curly brackets, as defined in Definition 5. Hence, the term $L_e(\sigma)_i$ corresponds to the number of elements belonging to the permutation σ that are placed at the right of σ_i and at the same time they are less than σ_i. Naturally, each element $L_e(\sigma)_i$ is bounded to the range $\{0, 1, .., n-i\}$, since the number of elements to the right of σ_i is equal to $n - i$.

A.2 Lehmer Encoding Example

Let us now elaborate on the specifics of Lehmer encoding with the aid of a brief tutorial example. Therefore, let us consider the route SN→RN$_5$→RN$_2$→RN$_3$→RN$_1$→RN$_4$→DN of an 8-node WMHN. Since the SN and the DN are not taken into consideration, the respective permutation σ in terms of the RNs' zero-initialized indices is equal to:

$$\sigma = [4, 1, 2, 0, 3]. \tag{A.3}$$

In order to calculate the encoding, Eq. (A.2) is utilized. Hence, starting from the far left element of the permutation vector σ of Eq. (A.3) and moving to right, the Lehmer-encoding $L_e(\sigma)_i$ is equal to the number of elements σ_j associated with $j > i$, i.e. with the elements to the right σ_i, that are less than σ_i. Subsequently, the elements σ_j with $j > i$ that are greater or equal than $L_e(\sigma)_i$ are reduced by 1. The detailed process of calculating the Lehmer-encoded vector $L_e(\sigma)$ for our tutorial example is shown in Table A.1, while a summarized version is following:

$$\begin{array}{ccccc} \mathbf{4} & 1 & 2 & 0 & 3 \\ 4 & \mathbf{1} & 1 & 0 & 2 \\ 4 & 1 & \mathbf{1} & 0 & 1 \\ 4 & 1 & 1 & \mathbf{0} & 0 \\ 4 & 1 & 1 & 0 & \mathbf{0} \end{array} \cdot \tag{A.4}$$

Hence, the Lehmer-encoded vector $L_e(\sigma)$ is equal to:

$$L_e(\sigma) = [4, 1, 1, 0, 0]. \tag{A.5}$$

Table A.1: Detailed steps of the Lehmer encoding process for the example of (A.3).

Step						Notes
1.1	**4**	1	2	0	3	Fetch the first element.
1.2	**4**	1	2	0	3	Calculate the number of the smaller elements to the right; hence, $L(\sigma)_1 = 4$.
1.3	**4**	1	2	0	3	Reduce by one the elements to the right that are greater or equal; in this case no element is reduced.
1.4	$\boxed{\mathbf{4}}$	1	2	0	3	Set the inspected element equal to $L(\sigma)_1 = 4$.
2.1	4	**1**	2	0	3	Fetch the second element.
2.2	4	**1**	2	0	3	Calculate the number of the smaller elements to the right; hence, $L(\sigma)_2 = 1$.
2.3	4	**1**	$\boxed{1}$	0	$\boxed{2}$	Reduce by one the elements to the right that are greater or equal; the elements 2 and 3 will be reduced by one.
2.4	4	$\boxed{\mathbf{1}}$	1	0	2	Set the inspected element equal to $L(\sigma)_2 = 1$.
3.1	4	1	**1**	0	2	Fetch the third element.
3.2	4	1	**1**	0	2	Calculate the number of the smaller elements to the right; hence, $L(\sigma)_3 = 1$.
3.3	4	1	**1**	0	$\boxed{1}$	Reduce by one the elements to the right that are greater or equal; the element 2 will be reduced by one only.
3.4	4	1	$\boxed{\mathbf{1}}$	0	1	Set the inspected element equal to $L(\sigma)_3 = 1$.
4.1	4	1	1	**0**	1	Fetch the fourth element.
4.2	4	1	1	**0**	1	Calculate the number of the smaller elements to the right; hence, $L(\sigma)_4 = 0$.
4.3	4	1	1	**0**	$\boxed{0}$	Reduce by one the elements to the right that are greater or equal; the element 1 will be reduced by one only.
4.4	4	1	1	$\boxed{\mathbf{0}}$	0	Set the inspected element equal to $L(\sigma)_4 = 0$.
5.1	4	1	1	0	**0**	Fetch the fifth element.
5.2	4	1	1	0	**0**	Calculate the number of the smaller elements to the right; hence, $L(\sigma)_5 = 0$.
5.3	4	1	1	0	**0**	Reduce by one the elements to the right that are greater or equal; there a are no elements to the right and hence no element will be reduced.
5.4	4	1	1	0	$\boxed{\mathbf{0}}$	Set the inspected element equal to $L(\sigma)_4 = 0$.

A.3 Lehmer Decoding Example

As far as the decoding process is concerned, we have to simply reverse the encoding process. To elaborate further, starting from the far right element $L_e(\sigma)_i$ and moving to the left, the elements $L_e(\sigma)_j$ associated with $j < i$, i.e. iwth the elements located to the right of the examined element $L_e(\sigma)_i$, are increased by one, if the particular element $L_e(\sigma)_j$ is greater or equal than the examined element $L_e(\sigma)_i$. This process is repeated for all the elements until the left-most element is reached. Naturally, this process leads to the reconstruction of the index vector σ of the permutation. For instance, in our tutorial example the decoding process is carried out as follows:

$$\begin{array}{ccccc} 4 & 1 & 1 & 0 & \mathbf{0} \\ 4 & 1 & 1 & \mathbf{0} & 1 \\ 4 & 1 & \mathbf{1} & 0 & 2 \\ 4 & \mathbf{1} & 2 & 0 & 3 \\ \mathbf{4} & 1 & 2 & 0 & 3 \end{array} \cdot \tag{A.6}$$

Note that a more detailed version of Eq. (A.6) is presented in Table. A.2.

A.4 Applying Lehmer Coding to the Routing Problem

The Lehmer Coding process is of utter importance, since it succeeds in transforming each route into a factoradic index, which may then be converted into a decimal or a binary index. Therefore, the routing problem inherently obtains a binary combinatorial problem structure, facilitating the employment of quantum computing search and optimization methods. For instance, the respective binary index of the aforementioned tutorial example is equal to 1101000, which correspond to the number 104 in the decimal system. This specific index indicates the permutation index in a list, in which all possible permutations of a certain set of RNs have been stored in lexicographical order, where the zero index accounts for the permutation, whose elements are sorted in ascending order, i.e. $\sigma_i < \sigma_j$ with $i < j$.

Therefore, a method of generating all the possible prototype sets $\mathcal{S}$ of the RNs participating in the formation the route has to be designed for completing the transformation. This may be implemented by using binary vectors, having a length, which is equal to the total number of RNs in the network. The elements of these vectors having a value equal to unity indicate that the respective RNs are participating in the formation of the route. For instance, the binary vector of 111101 would result in a prototype set $\mathcal{S} = \{\text{RN}_1, \text{RN}_2, \text{RN}_3, \text{RN}_4, \text{RN}_6\}$ for an 8-node network .

Moreover, the total number of permutations, which a set $\mathcal{S}$ may produce is equal to $(|\mathcal{S}|)!$ [137], where the operation $|\cdot|$ denotes the number of elements comprising the set, i.e. its cardinality. This is equivalent to the number of unity-valued elements of the associated

Table A.2: Detailed steps of the Lehmer decoding process for the example of (A.5).

Step						Notes
1.1	4	1	1	0	**0**	Fetch the fifth element.
1.2	4	1	1	0	**0**	Increase by one the elements to the right that are greater or equal; there a are no elements to the right and hence no element will be increased.
2.1	4	1	1	**0**	0	Fetch the fourth element.
2.2	4	1	1	**0**	$\boxed{1}$	Increase by one the elements to the right that are greater or equal; only the element equal to zero will be increased.
3.1	4	1	**1**	0	1	Fetch the third element.
3.2	4	1	**1**	0	$\boxed{2}$	Increase by one the elements to the right that are greater or equal; only the elements equal to one will be increased.
4.1	4	**1**	1	0	2	Fetch the second element.
4.2	4	**1**	$\boxed{2}$	0	$\boxed{3}$	Increase by one the elements to the right that are greater or equal; the elements equal to one and two will be increased.
5.1	**4**	1	1	0	2	Fetch the first element.
5.2	**4**	1	2	0	3	Increase by one the elements to the right that are greater or equal; no elements to the right are greater or equal.

binary vectors. Hence, assuming a route index x, we are now capable of determining the route's binary vector by calculating the cumulative sums of the number of possible permutations stemming from the binary vectors representing decimal values in the range of $\{0, 1, \ldots, 2^{N_{\text{nodes}}-2}\}$, where N_{nodes} corresponds to the total number of nodes including the SN and the DN. Given this cumulative sum, we are now capable of exporting the range of decimal indices associated with a specific binary vector and thus it is possible to associate a specific route with its respective binary vector.

Having determined the prototype set $\mathcal{S}$, the permutation index has to be evaluated. In fact, it is equal to the difference between the route's decimal index x and the lower bound of the range of its associated binary vector. Then, the permutation index is transformed into the factoradic computational basis, hence evaluating the Lehmer-encoding vector $L_e(\sigma)$. Finally, the actual route is exported by applying Lehmer decoding to the factoradic vector $L_e(\sigma)$. For instance, assuming that the route having the decimal index of $x = 1101$ should be determined, the process of finding the respective binary vector is shown in Table A.3, where we can observe that the appropriate binary vector corresponding to the given route index is the vector [111101]. This vector corresponds to the prototype set $\mathcal{S} = \{\mathrm{RN}_1, \mathrm{RN}_2, \mathrm{RN}_3, \mathrm{RN}_4, \mathrm{RN}_6\}$. For the sake of obtaining the permutation, the cumulative sum of the previous vector represented by its equivalent decimal value should be subtracted from index x. The permutation index is then found to be equal to 104. The latter index, in turn, is transformed into the factoradic computational basis and its value is found to be equal to the vector $L_e(\sigma) = (4, 1, 1, 0, 0)$. Finally, Lehmer decoding is applied to $L_e(\sigma)$ and the route is found to be SN→RN$_5$→RN$_2$→RN$_3$→RN$_1$→RN$_4$→DN by appending the SN and DN at the beginning and the end of the exported route, respectively.

Table A.3: Calculation of the binary vector for the path with decimal index $x = 1101$.

Binary Vector	Cumulative Sum	Relationship to index i
000000	1	< 1101
000001	2	< 1101
...	...	< 1101
111100	997	< 1101
111101	**1117**	$>$ **1101**
...	...	< 1101

Bibliography

[1] D. Alanis, P. Botsinis, S. X. Ng, and L. Hanzo, "Quantum-Assisted Routing Optimization for Self-Organizing Networks," *IEEE Access*, vol. 2, pp. 614–632, 2014.

[2] D. Alanis, P. Botsinis, Z. Babar, S. X. Ng, and L. Hanzo, "Non-dominated quantum iterative routing optimization for wireless multihop networks," *IEEE Access*, vol. 3, pp. 1704–1728, 2015.

[3] D. Alanis, J. Hu, P. Botsinis, Z. Babar, S. X. Ng, and L. Hanzo, "Quantum-Assisted Joint Multi-Objective Routing and Load Balancing for Socially-Aware Networks," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2016.

[4] M. Weiser, "The computer for the 21st century," *Scientific american*, vol. 265, no. 3, pp. 94–104, 1991.

[5] L. Hanzo, H. Haas, S. Imre, D. O'Brien, M. Rupp, and L. Gyongyosi, "Wireless myths, realities, and futures: from 3g/4g to optical and quantum wireless," *Proceedings of the IEEE*, vol. 100, no. 13, pp. 1853–1888, 2012.

[6] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, and F. Gutierrez, "Millimeter Wave Mobile Communications for 5G Cellular: It Will Work!" *IEEE Access*, vol. 1, pp. 335–349, 2013.

[7] Z. Wei, X. Zhu, S. Sun, Y. Huang, A. Al-Tahmeesschi, and Y. Jiang, "Energy-Efficiency of Millimeter-Wave Full-Duplex Relaying Systems: Challenges and Solutions," *IEEE Access*, vol. 4, pp. 4848–4860, 2016.

[8] L. Zeng, D. C. O'Brien, H. L. Minh, G. E. Faulkner, K. Lee, D. Jung, Y. Oh, and E. T. Won, "High data rate multiple input multiple output (MIMO) optical wireless communications using white led lighting," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 9, pp. 1654–1662, December 2009.

[9] T. Komine and M. Nakagawa, "Fundamental analysis for visible-light communication system using LED lights," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 100–107, Feb 2004.

[10] H. Haas, L. Yin, Y. Wang, and C. Chen, "What is lifi?" *Journal of Lightwave Technology*, vol. 34, no. 6, pp. 1533–1544, March 2016.

[11] R. Steele and L. Hanzo, *Mobile radio communications: second and third generation cellular and WATM systems*, ser. Wiley - IEEE. J. Wiley, 1999. [Online]. Available: http://books.google.co.uk/books?id=pRUfAQAAIAAJ

[12] S. Yang, X. Xu, D. Alanis, S. X. Ng, and L. Hanzo, "Is the Low-Complexity Mobile Relay Aided FFR-DAS Capable of Outperforming the High-Complexity CoMP?" *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2015.

[13] W. Feng, Y. Chen, N. Ge, and J. Lu, "Optimal Energy-Efficient Power Allocation for Distributed Antenna Systems With Imperfect CSI," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 9, pp. 7759–7763, Sept 2016.

[14] A. Damnjanovic, J. Montojo, Y. Wei, T. Ji, T. Luo, M. Vajapeyam, T. Yoo, O. Song, and D. Malladi, "A survey on 3GPP heterogeneous networks," *IEEE Wireless Communications*, vol. 18, no. 3, pp. 10–21, 2011.

[15] J. Weitzen, M. Li, E. Anderland, and V. Eyuboglu, "Large-Scale Deployment of Residential Small Cells," *Proceedings of the IEEE*, vol. 101, no. 11, pp. 2367–2380, Nov 2013.

[16] D. Lee, H. Seo, B. Clerckx, E. Hardouin, D. Mazzarese, S. Nagata, and K. Sayana, "Coordinated multipoint transmission and reception in LTE-advanced: deployment scenarios and operational challenges," *IEEE Communications Magazine*, vol. 50, no. 2, pp. 148–155, February 2012.

[17] L. Hanzo, J. Blogh, and S. Ni, *3G, HSPA and FDD versus TDD Networking: Smart Antennas and Adaptive Modulation.* Wiley, 2008. [Online]. Available: https://books.google.co.uk/books?id=MmSsuQAACAAJ

[18] T. Nakamura, S. Nagata, A. Benjebbour, Y. Kishiyama, T. Hai, S. Xiaodong, Y. Ning, and L. Nan, "Trends in small cell enhancements in LTE advanced," *IEEE Communications Magazine*, vol. 51, no. 2, pp. 98–105, February 2013.

[19] H. Yetgin, K. T. K. Cheung, M. El-Hajjar, and L. Hanzo, "Cross-Layer Network Lifetime Maximization in Interference-Limited WSNs," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 8, pp. 3795–3803, Aug 2015.

[20] S. Pack, X. Shen, J. W. Mark, and J. Pan, "Mobility Management in Mobile Hotspots with Heterogeneous Multihop Wireless Links," *IEEE Communications Magazine*, vol. 45, no. 9, pp. 106–112, September 2007.

[21] D. Zhou, W. Song, P. Wang, and W. Zhuang, "Multipath TCP for user cooperation in LTE networks," *IEEE Network*, vol. 29, no. 1, pp. 18–24, Jan 2015.

[22] G. Aloi, M. D. Felice, V. Loscr, P. Pace, and G. Ruggeri, "Spontaneous smartphone networks as a user-centric solution for the future internet," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 26–33, December 2014.

[23] J. Wen, M. Sheng, X. Wang, J. Li, and H. Sun, "On the Capacity of Downlink Multi-Hop Heterogeneous Cellular Networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 8, pp. 4092–4103, Aug 2014.

[24] P. Mach, Z. Becvar, and T. Vanek, "In-band device-to-device communication in ofdma cellular networks: A survey and challenges," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 1885–1922, Fourthquarter 2015.

[25] B. Alawieh, Y. Zhang, C. Assi, and H. Mouftah, "Improving Spatial Reuse in Multihop Wireless Networks - A Survey," *IEEE Communications Surveys Tutorials*, vol. 11, no. 3, pp. 71–91, rd 2009.

[26] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[27] M. Gerla, "Ad hoc networks," in *Ad Hoc Networks.* Springer, 2005, pp. 1–22.

[28] S. Galli, A. Scaglione, and Z. Wang, "For the Grid and Through the Grid: The Role of Power Line Communications in the Smart Grid," *Proceedings of the IEEE*, vol. 99, no. 6, pp. 998–1027, June 2011.

[29] H. Yetgin, K. Cheung, and L. Hanzo, "Multi-objective routing optimization using evolutionary algorithms," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2012, pp. 3030–3034.

[30] J. Hu, L.-L. Yang, and L. Hanzo, "Cross-Layer Design for Wireless Mesh Networking Aided Content Sharing in Online Social Networks," *IEEE Transactions on Vehicular Technology*, 2015, submitted.

[31] J. Zuo, C. Dong, H. V. Nguyen, S. X. Ng, L. L. Yang, and L. Hanzo, "Cross-Layer Aided Energy-Efficient Opportunistic Routing in Ad Hoc Networks," *IEEE Transactions on Communications*, vol. 62, no. 2, pp. 522–535, February 2014.

[32] C. Luo, S. Guo, S. Guo, L. T. Yang, G. Min, and X. Xie, "Green Communication in Energy Renewable Wireless Mesh Networks: Routing, Rate Control, and Power Allocation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3211–3220, Dec 2014.

[33] M. Dehghan, M. Ghaderi, and D. Goeckel, "Minimum-energy cooperative routing in wireless networks with channel variations," *IEEE Transactions on Wireless Communications*, vol. 10, no. 11, pp. 3813–3823, November 2011.

[34] E. Dall'Anese and G. B. Giannakis, "Statistical routing for multihop wireless cognitive networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 10, pp. 1983–1993, November 2012.

[35] H. Yetgin, K. T. K. Cheung, M. El-Hajjar, and L. Hanzo, "Network-Lifetime Maximization of Wireless Sensor Networks," *IEEE Access*, vol. 3, pp. 2191–2226, 2015.

[36] A. Abdulla, H. Nishiyama, J. Yang, N. Ansari, and N. Kato, "HYMN: A Novel Hybrid Multi-Hop Routing Algorithm to Improve the Longevity of WSNs," *IEEE Transactions on Wireless Communications*, vol. 11, no. 7, pp. 2531–2541, 2012.

[37] L. Tan, Z. Zhu, F. Ge, and N. Xiong, "Utility Maximization Resource Allocation in Wireless Networks: Methods and Algorithms," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 7, pp. 1018–1034, July 2015.

[38] Y. Shi, Y. T. Hou, and H. Sherali, "Cross-Layer Optimization for Data Rate Utility Problem in UWB-based Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 6, pp. 764–777, June 2008.

[39] X. Zhu, L. Shen, and T.-S. Yum, "Hausdorff Clustering and Minimum Energy Routing for Wireless Sensor Networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 2, pp. 990–997, 2009.

[40] M. Chen, V. Leung, S. Mao, Y. Xiao, and I. Chlamtac, "Hybrid Geographic Routing for Flexible Energy – Delay Tradeoff," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 9, pp. 4976–4988, 2009.

[41] M. Al-Rabayah and R. Malaney, "A New Scalable Hybrid Routing Protocol for ANETs," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 6, pp. 2625–2635, July 2012.

[42] H. Huang, S. Guo, W. Liang, K. Li, B. Ye, and W. Zhuang, "Near-Optimal Routing Protection for In-Band Software-Defined Heterogeneous Networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 11, pp. 2918–2934, Nov 2016.

[43] J. Yao, S. Feng, X. Zhou, and Y. Liu, "Secure Routing in Multihop Wireless Ad-Hoc Networks With Decode-and-Forward Relaying," *IEEE Transactions on Communications*, vol. 64, no. 2, pp. 753–764, Feb 2016.

[44] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[45] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[46] N. Basalto, R. Bellotti, F. De Carlo, P. Facchi, E. Pantaleo, and S. Pascazio, "Hausdorff clustering," *Physical Review E*, vol. 78, no. 4, p. 046112, 2008.

[47] K. Deb, "Multi-objective optimization," in *Search Methodologies*, E. K. Burke and G. Kendall, Eds. Springer US, 2005, pp. 273–316.

[48] E. Masazade, R. Rajagopalan, P. Varshney, C. Mohan, G. Sendur, and M. Keskinoz, "A Multiobjective Optimization Approach to Obtain Decision Thresholds for Distributed Detection in Wireless Sensor Networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 2, pp. 444–457, 2010.

[49] M. Camelo, C. Omaa, and H. Castro, "QoS routing algorithm based on multi-objective optimization for Wireless Mesh Networks," in *2010 IEEE Latin-American Conference on Communications (LATINCOM)*, 2010, pp. 1–6.

[50] A. Perez, M. Labrador, and P. Wightman, "A multiobjective approach to the relay placement problem in WSNs," in *2011 IEEE Wireless Communications and Networking Conference (WCNC)*, 2011, pp. 475–480.

[51] F. Martins, E. Carrano, E. Wanner, R. H. C. Takahashi, and G. Mateus, "A Hybrid Multiobjective Evolutionary Approach for Improving the Performance of Wireless Sensor Networks," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 545–554, 2011.

[52] D. Pinto and B. Barán, "Solving multiobjective multicast routing problem with a new ant colony optimization approach," in *Proceedings of the 3rd international IFIP/ACM Latin American conference on Networking.* ACM, 2005, pp. 11–19.

[53] W. Stadler, "A survey of multicriteria optimization or the vector maximum problem, part I: 1776–1960," *Journal of Optimization Theory and Applications*, vol. 29, no. 1, pp. 1–52, 1979.

[54] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.

[55] M. Dorigo and G. Di Caro, "Ant colony optimization: a new meta-heuristic," in *Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99)*, vol. 2. IEEE, 1999.

[56] M. Golshahi, M. Mosleh, and M. Kheyrandish, "Implementing an ACO routing algorithm for AD-HOC networks," in *International Conference on Advanced Computer Theory and Engineering (ICACTE'08.).* IEEE, 2008, pp. 143–147.

[57] S. Chandra, U. Shrivastava, R. Vaish, S. Dixit, and M. Rana, "Improved-AntNet: ACO routing algorithm in practice," in *11th International Conference on Computer Modelling and Simulation (UKSIM'09).* IEEE, 2009, pp. 25–29.

[58] H. Wang, Z. Wu, X. Yang, and H. Liu, "A Novel ACO-Based Multicast Path Algorithm In Hypercube Networks," *Intelligent Automation & Soft Computing*, vol. 17, no. 5, pp. 541–549, 2011.

[59] L. Wang, Q. Niu, and M. Fei, "A novel quantum ant colony optimization algorithm," in *Bio-Inspired Computational Intelligence and Applications.* Springer, 2007, pp. 277–286.

[60] H. Jiang, M.-r. Wang, M. Liu, and J.-w. Yan, "A quantum-inspired ant-based routing algorithm for WSNs," in *IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD).* IEEE, 2012, pp. 609–615.

[61] Z. Fei, B. Li, S. Yang, C. Xing, H. Chen, and L. Hanzo, "A Survey of Multi-Objective Optimization in Wireless Sensor Networks: Metrics, Algorithms and Open Problems," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, 2016.

[62] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information.* Cambridge university press, 2010.

[63] M. M. Waldrop, "The chips are down for Moores law," *Nature News*, vol. 530, no. 7589, p. 144, 2016.

[64] S. Boixo, T. Albash, F. M. Spedalieri, N. Chancellor, and D. A. Lidar, "Experimental signature of programmable quantum annealing," *Nature communications*, vol. 4, 2013.

[65] M. Johnson, M. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. Berkley, J. Johansson, P. Bunyk *et al.*, "Quantum annealing with manufactured spins," *Nature*, vol. 473, no. 7346, pp. 194–198, 2011.

[66] S. Boixo, T. F. Rønnow, S. V. Isakov, Z. Wang, D. Wecker, D. A. Lidar, J. M. Martinis, and M. Troyer, "Evidence for quantum annealing with more than one hundred qubits," *Nature Physics*, vol. 10, no. 3, pp. 218–224, 2014.

[67] S. Imre and F. Balazs, *Quantum Computing and Communications: an engineering approach.* Wiley, 2005.

[68] Z. Babar, S. X. Ng, and L. Hanzo, "Near-Capacity Code Design for Entanglement-Assisted Classical Communication over Quantum Depolarizing Channels," *IEEE Transactions on Communications*, vol. 61, no. 12, pp. 4801–4807, December 2013.

[69] Z. Babar, S. Ng, and L. Hanzo, "Exit-chart aided near-capacity quantum turbo code design," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2014.

[70] Z. Babar, P. Botsinis, D. Alanis, S. X. Ng, and L. Hanzo, "The road from classical to quantum codes: A hashing bound approaching design procedure," *Access, IEEE*, vol. 3, pp. 146–176, 2015.

[71] ——, "Fifteen Years of Quantum LDPC Coding and Improved Decoding Strategies," *IEEE Access*, vol. 3, pp. 2492–2519, 2015.

[72] Z. Babar, L. Hanzo, H. Nguyen, P. Botsinis, D. Alanis, D. Chandra, S. X. Ng, and R. G. Maunder, "A Fully-Parallel Quantum Turbo Decoder," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2016.

[73] P. Botsinis, Z. Babar, D. Alanis, D. Chandra, H. Nguyen, S. X. Ng, and L. Hanzo, "Quantum error correction protects quantum search algorithms against decoherence," *Scientific Reports*, vol. 6, 2016.

[74] R. P. Feynman, "Simulating physics with computers," *International journal of theoretical physics*, vol. 21, no. 6, pp. 467–488, 1982.

[75] P. Benioff, "Quantum mechanical hamiltonian models of turing machines," *Journal of Statistical Physics*, vol. 29, no. 3, pp. 515–546, 1982. [Online]. Available: http://dx.doi.org/10.1007/BF01342185

[76] D. Deutsch, "Quantum theory, the Church-Turing principle and the universal quantum computer," *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 400, no. 1818, pp. 97–117, 1985.

[77] D. Deutsch and R. Jozsa, "Rapid solution of problems by quantum computation," *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, vol. 439, no. 1907, pp. 553–558, 1992.

[78] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM journal on computing*, vol. 26, no. 5, pp. 1484–1509, 1997.

[79] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing.* ACM, 1996, pp. 212–219.

[80] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, "Tight bounds on quantum searching," *arXiv preprint quant-ph/9605034*, 1996.

[81] C. Durr and P. Høyer, "A quantum algorithm for finding the minimum," *arXiv preprint quant-ph/9607014*, 1996.

[82] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, "Quantum algorithms revisited," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1969, pp. 339–354, 1998.

[83] G. Brassard, P. Høyer, and A. Tapp, "Quantum counting," in *Automata, Languages and Programming.* Springer, 1998, pp. 820–831.

[84] D. S. Abrams and S. Lloyd, "Quantum Algorithm Providing Exponential Speed Increase for Finding Eigenvalues and Eigenvectors," *Phys. Rev. Lett.*, vol. 83, pp. 5162–5165, Dec 1999. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.83.5162

[85] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, "Quantum amplitude amplification and estimation," *arXiv preprint quant-ph/0005055*, 2000.

[86] G. Brassard, F. Dupuis, S. Gambs, and A. Tapp, "An optimal quantum algorithm to approximate the mean and its application for approximating the median of a set of points over an arbitrary distance," *arXiv preprint arXiv:1106.4267*, 2011.

[87] P. Botsinis, S. X. Ng, and L. Hanzo, "Quantum Search Algorithms, Quantum Wireless, and a Low-Complexity Maximum Likelihood Iterative Quantum Multi-User Detector Design," *IEEE Access*, vol. 1, pp. 94–122, 2013.

[88] C. Zalka, "Grover's quantum searching algorithm is optimal," *Physical Review A*, vol. 60, no. 4, p. 2746, 1999.

[89] A. Malossini, E. Blanzieri, and T. Calarco, "Quantum Genetic Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 231–241, April 2008.

[90] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[91] G. Syswerda, "A study of reproduction in generational and steady state genetic algorithms," *Foundations of genetic algorithms*, vol. 2, pp. 94–101, 1991.

[92] J. Chiaverini, J. Britton, D. Leibfried, E. Knill, M. Barrett, R. Blakestad, W. Itano, J. Jost, C. Langer, R. Ozeri *et al.*, "Implementation of the semiclassical quantum Fourier transform in a scalable system," *Science*, vol. 308, no. 5724, pp. 997–1000, 2005.

[93] P. Botsinis, S. X. Ng, and L. Hanzo, "Fixed-Complexity Quantum-Assisted Multi-User Detection for CDMA and SDMA," *IEEE Transactions on Communications*, vol. 62, no. 3, pp. 990–1000, March 2014.

[94] P. Botsinis, D. Alanis, S. Ng, and L. Hanzo, "Low-Complexity Soft-Output Quantum-Assisted Multiuser Detection for Direct-Sequence Spreading and Slow Subcarrier-Hopping Aided SDMA-OFDM Systems," *IEEE Access*, vol. 2, pp. 451–472, 2014.

[95] P. Botsinis, D. Alanis, Z. Babar, S. X. Ng, and L. Hanzo, "Iterative quantum-assisted multi-user detection for multi-carrier interleave division multiple access systems," *IEEE Transactions on Communications*, vol. 63, no. 10, pp. 3713–3727, Oct 2015.

[96] ——, "Noncoherent Quantum Multiple Symbol Differential Detection for Wireless Systems," *IEEE Access*, vol. 3, pp. 569–598, 2015.

[97] L. Hanzo, Y. Akhtman, J. Akhtman, L. Wang, and M. Jiang, *MIMO-OFDM for LTE, WiFi and WiMAX: Coherent versus non-coherent and cooperative turbo transceivers.* John Wiley & Sons, 2010, vol. 9.

[98] S. Chen, X. Wang, and C. J. Harris, "Experiments with repeating weighted boosting search for optimization signal processing applications," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 4, pp. 682–693, Aug 2005.

[99] J. Kennedy and R. Eberhart, "A Discrete Binary Version of the Particle Swarm Algorithm," in *IEEE International Conference on Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation, 1997*, vol. 5, October 1997, pp. 4104–4108 vol.5.

[100] P. Botsinis, D. Alanis, Z. Babar, S. X. Ng, and L. Hanzo, "Joint Quantum-Assisted Channel Estimation and Data Detection," *IEEE Access*, vol. 4, pp. 7658–7681, 2016.

[101] M. Jiang and L. Hanzo, "Multiuser MIMO-OFDM for Next-Generation Wireless Systems," *Proceedings of the IEEE*, vol. 95, no. 7, pp. 1430–1469, 2007.

[102] J. Zhang, S. Chen, X. Mu, and L. Hanzo, "Evolutionary-Algorithm-Assisted Joint Channel Estimation and Turbo Multiuser Detection/Decoding for OFDM/SDMA," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 3, pp. 1204–1222, March 2014.

[103] W. Yao, S. Chen, and L. Hanzo, "Generalized MBER-Based Vector Precoding Design for Multiuser Transmission," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 2, pp. 739–745, February 2011.

[104] C. Masouros, M. Sellathurai, and T. Ratnarajah, "Vector Perturbation Based on Symbol Scaling for Limited Feedback MISO Downlinks," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 562–571, Feb 2014.

[105] P. Botsinis, D. Alanis, Z. Babar, H. V. Nguyen, D. Chandra, S. X. Ng, and L. Hanzo, "Quantum-Aided Multi-User Transmission in Non-Orthogonal Multiple Access Systems," *IEEE Access*, vol. 4, pp. 7402–7424, 2016.

[106] L. Dai, B. Wang, Y. Yuan, S. Han, C. l. I, and Z. Wang, "Non-orthogonal multiple access for 5g: solutions, challenges, opportunities, and future research trends," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 74–81, September 2015.

[107] G. Gan, C. Ma, and J. Wu, *Data clustering: theory, algorithms, and applications.* Siam, 2007, vol. 20.

[108] D. Yang, L. L. Yang, and L. Hanzo, "Performance of SDMA Systems Using Transmitter Preprocessing Based on Noisy Feedback of Vector-Quantized Channel Impulse Responses," in *IEEE Vehicular Technology Conference*, April 2007, pp. 2119–2123.

[109] T. J. Hastie, R. J. Tibshirani, and J. H. Friedman, *"The Elements of Statistical Learning : Data Mining, Inference, and Prediction".* Springer, 2009. [Online]. Available: http://opac.inria.fr/record=b1127878

[110] J. Lu, G. Wang, and P. Moulin, "Human Identity and Gender Recognition From Gait Sequences With Arbitrary Walking Directions," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 1, pp. 51–61, Jan 2014.

[111] D. S. Matovski, M. S. Nixon, S. Mahmoodi, and J. N. Carter, "The Effect of Time on Gait Recognition Performance," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 543–552, 2012.

[112] E. Aïmeur, G. Brassard, and S. Gambs, "Quantum speed-up for unsupervised learning," *Machine Learning*, vol. 90, no. 2, pp. 261–287, 2013.

[113] N. Wiebe, A. Kapoor, and K. Svore, "Quantum Algorithms for Nearest-Neighbor Methods for Supervised and Unsupervised Learning," *ArXiv e-prints*, Jan. 2014.

[114] S. Du, Y. Yan, and Y. Ma, "Quantum-Accelerated Fractal Image Compression: An Interdisciplinary Approach," *IEEE Signal Processing Letters*, vol. 22, no. 4, pp. 499–503, April 2015.

[115] A. Aquino, M. E. Gegundez-Arias, and D. Marin, "Detecting the Optic Disc Boundary in Digital Fundus Images Using Morphological, Edge Detection, and Feature Extraction Techniques," *IEEE Transactions on Medical Imaging*, vol. 29, no. 11, pp. 1860–1869, Nov 2010.

[116] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: bringing order to the web," 1999.

[117] S. Brin and L. Page, "Reprint of: The anatomy of a large-scale hypertextual web search engine," *Computer networks*, vol. 56, no. 18, pp. 3825–3833, 2012.

[118] G. Paparo and M. Martin-Delgado, "Google in a Quantum Network," *Scientific Reports*, vol. 2, p. 444, 2012.

[119] G. D. Paparo, M. Müller, F. Comellas, and M. A. Martin-Delgado, "Quantum Google in a Complex Network," *Scientific Reports*, vol. 3, p. 2773, 2013.

[120] S. Lloyd, S. Garnerone, and P. Zanardi, "Quantum algorithms for topological and geometric analysis of data," *Nature communications*, vol. 7, 2016.

[121] P. Scheiblechner, "On the complexity of deciding connectedness and computing betti numbers of a complex algebraic variety," *Journal of Complexity*, vol. 23, no. 3, pp. 359–379, 2007.

[122] C. Durr, M. Heiligman, P. Hoyer, and M. Mhalla, "Quantum Query Complexity of Some Graph Problems," *eprint arXiv:quant-ph/0401091*, Jan. 2004.

[123] F. Hoffmann, D. Medina, and A. Wolisz, "Joint Routing and Scheduling in Mobile Aeronautical Ad Hoc Networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 6, pp. 2700–2712, July 2013.

[124] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[125] C. Xu, B. Hu, L.-L. Yang, and L. Hanzo, "Ant-Colony-Based Multiuser Detection for Multifunctional-Antenna-Array-Assisted MC DS-CDMA Systems," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 1, pp. 658–663, Jan 2008.

[126] L. Hanzo, S. X. Ng, W. Webb, and T. Keller, *Quadrature amplitude modulation: From basics to adaptive trellis-coded, turbo-equalised and space-time coded OFDM, CDMA and MC-CDMA systems.* IEEE Press-John Wiley, 2004.

[127] M. Salem, A. Adinoyi, M. Rahman, H. Yanikomeroglu, D. Falconer, Y.-D. Kim, E. Kim, and Y.-C. Cheong, "An Overview of Radio Resource Management in

Relay-Enhanced OFDMA-Based Networks," *IEEE Communications Surveys Tutorials*, vol. 12, no. 3, pp. 422–438, 2010.

[128] U. K. Chakraborty, "A branching process model for genetic algorithms," *Information Processing Letters*, vol. 56, no. 5, pp. 281–292, 1995.

[129] I. Alaya, C. Solnon, and K. Ghedira, "Ant algorithm for the multi-dimensional knapsack problem," in *International Conference on Bioinspired Optimization Methods and their Applications (BIOMA 2004)*. Citeseer, 2004.

[130] M. Gravel, W. L. Price, and C. Gagné, "Scheduling continuous casting of aluminum using a multiple objective ant colony optimization metaheuristic," *European Journal of Operational Research*, vol. 143, no. 1, pp. 218–229, 2002.

[131] P. R. McMullen, "An ant colony optimization approach to addressing a JIT sequencing problem with multiple objectives," *Artificial Intelligence in Engineering*, vol. 15, no. 3, pp. 309–317, 2001.

[132] B. Barán and M. Schaerer, "A Multiobjective Ant Colony System for Vehicle Routing Problem with Time Windows," in *Applied Informatics*, 2003, pp. 97–102.

[133] B. Bullnheimer, R. F. Hartl, and C. Strauss, "An improved ant System algorithm for the Vehicle Routing Problem," *Annals of operations research*, vol. 89, pp. 319–328, 1999.

[134] K. Doerner, W. J. Gutjahr, R. F. Hartl, C. Strauss, and C. Stummer, "Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection," *Annals of Operations Research*, vol. 131, no. 1-4, pp. 79–99, 2004.

[135] K. Doerner, R. F. Hartl, and M. Reimann, "Are COMPETants more competent for problem solving? The case of a multiple objective transportation problem," 2001.

[136] T. Stützle and H. H. Hoos, "MAX–MIN ant system," *Future generation computer systems*, vol. 16, no. 8, pp. 889–914, 2000.

[137] D. H. Lehmer, "Teaching combinatorial tricks to a computer," in *Proc. Sympos. Appl. Math. Combinatorial Analysis*, vol. 10, 1960, pp. 179–193.

[138] W. Buttler, R. Hughes, P. Kwiat, S. Lamoreaux, G. Luther, G. Morgan, J. Nordholt, C. Peterson, and C. Simmons, "Practical free-space quantum key distribution over 1 km," *arXiv preprint quant-ph/9805071*, 1998.

[139] M. Razavi, "Multiple-Access Quantum Key Distribution Networks," *IEEE Transactions on Communications*, vol. 60, no. 10, pp. 3071–3079, October 2012.

[140] N. L. Piparo, M. Razavi, and C. Panayi, "Measurement-Device-Independent Quantum Key Distribution With Ensemble-Based Memories," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 21, no. 3, pp. 138–147, May 2015.

[141] S. Bahrani, M. Razavi, and J. A. Salehi, "Orthogonal Frequency-Division Multiplexed Quantum Key Distribution," *Journal of Lightwave Technology*, vol. 33, no. 23, pp. 4687–4698, Dec 2015.

[142] S. Aaronson, *Quantum computing since Democritus.* Cambridge University Press, 2013.

[143] M. Saeedi and M. Pedram, "Linear-depth quantum circuits for n-qubit Toffoli gates with no ancilla," *Physical Review A*, vol. 87, no. 6, p. 062318, 2013.

[144] Y. Han, D. Ancajas, K. Chakraborty, and S. Roy, "Exploring High-Throughput Computing Paradigm for Global Routing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 1, pp. 155–167, Jan 2014.

[145] S. Mu, X. Zhang, N. Zhang, J. Lu, Y. Deng, and S. Zhang, "IP routing processing with graphic processors," in *2010 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2010, pp. 93–98.

[146] J. Zhao, X. Zhang, X. Wang, Y. Deng, and X. Fu, "Exploiting graphics processors for high-performance IP lookup in software routers," in *2011 Proceedings IEEE INFOCOM*, April 2011, pp. 301–305.

[147] R. Mangharam and A. Saba, "Anytime Algorithms for GPU Architectures," in *2011 IEEE 32nd Real-Time Systems Symposium (RTSS)*, Nov 2011, pp. 47–56.

[148] A. Uchida, Y. Ito, and K. Nakano, "An Efficient GPU Implementation of Ant Colony Optimization for the Traveling Salesman Problem," in *2012 Third International Conference on Networking and Computing (ICNC)*, Dec 2012, pp. 94–102.

[149] U. Cekmez, M. Ozsiginan, and O. Sahingoz, "Adapting the GA approach to solve Traveling Salesman Problems on CUDA architecture," in *2013 IEEE 14th International Symposium on Computational Intelligence and Informatics (CINTI)*, Nov 2013, pp. 423–428.

[150] T. Mohsenin, D. Truong, and B. Baas, "A Low-Complexity Message-Passing Algorithm for Reduced Routing Congestion in LDPC Decoders," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 5, pp. 1048–1061, May 2010.

[151] J. Owens, M. Houston, D. Luebke, S. Green, J. Stone, and J. Phillips, "Gpu computing," *Proceedings of the IEEE*, vol. 96, no. 5, pp. 879–899, May 2008.

[152] J. D. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krüger, A. E. Lefohn, and T. J. Purcell, "A survey of general-purpose computation on graphics hardware," in *Computer graphics forum*, vol. 26, no. 1. Wiley Online Library, 2007, pp. 80–113.

[153] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, "Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels," *Phys. Rev. Lett.*, vol. 70, pp. 1895–1899, Mar 1993. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.70.1895

[154] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[155] D. Evans, "The internet of things: How the next evolution of the internet is changing everything," White Paper, CISCO, Tech. Rep., 2011.

[156] J. Fry, "The Intelligent Flexible Cloud," White Paper, ARM Ltd, Tech. Rep. DOC-9981, 2015. [Online]. Available: https://community.arm.com/docs/

[157] L. Atzori, A. Iera, and G. Morabito, "Siot: Giving a social structure to the internet of things," *IEEE Communications Letters*, vol. 15, no. 11, pp. 1193–1195, 2011.

[158] L. Atzori, A. Iera, G. Morabito, and M. Nitti, "The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization," *Computer Networks*, vol. 56, no. 16, pp. 3594–3608, 2012.

[159] K.-C. Chen, M. Chiang, and H. V. Poor, "From Technological Networks to Social Networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 548–572, September 2013.

[160] C.-Y. Lin, L. Wu, Z. Wen, H. Tong, V. Griffiths-Fisher, L. Shi, and D. Lubensky, "Social Network Analysis in Enterprise," *Proceedings of the IEEE*, vol. 100, no. 9, pp. 2759–2776, Sept 2012.

[161] C. Boldrini, M. Conti, and A. Passarella, "The Stability Region of the Delay in Pareto Opportunistic Networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 1, pp. 180–193, Jan 2015.

[162] J. Wu and Y. Wang, "Hypercube-Based Multipath Social Feature Routing in Human Contact Networks," *IEEE Transactions on Computers*, vol. 63, no. 2, pp. 383–396, Feb 2014.

[163] Z. Li and H. Shen, "SEDUM: Exploiting Social Networks in Utility–Based Distributed Routing for DTNs," *IEEE Transactions on Computers*, vol. 62, no. 1, pp. 83–97, Jan 2013.

[164] K. Chen and H. Shen, "SMART: Utilizing Distributed Social Map for Lightweight Routing in Delay-Tolerant Networks," *IEEE/ACM Transactions on Networking*, vol. 22, no. 5, pp. 1545–1558, Oct 2014.

[165] E. Bulut and B. K. Szymanski, "Exploiting friendship relations for efficient routing in mobile social networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 12, pp. 2254–2265, Dec 2012.

[166] L. Yao, Y. Man, Z. Huang, J. Deng, and X. Wang, "Secure routing based on social similarity in opportunistic networks," *IEEE Transactions on Wireless Communications*, vol. PP, no. 99, pp. 1–1, 2015.

[167] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE Rap: Social-Based Forwarding in Delay-Tolerant Networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1576–1589, Nov 2011.

[168] F. Xia, L. Liu, J. Li, A. Ahmed, L. Yang, and J. Ma, "BEEINFO: Interest-Based Forwarding Using Artificial Bee Colony for Socially Aware Networking," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 3, pp. 1188–1200, March 2015.

[169] M. E. J. Newman, "The Structure and Function of Complex Networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003. [Online]. Available: http://dx.doi.org/10.1137/S003614450342480

[170] P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft, "Distributed community detection in delay tolerant networks," in *Proceedings of 2Nd ACM/IEEE International Workshop on Mobility in the Evolving Internet Architecture*, ser. MobiArch '07. New York, NY, USA: ACM, 2007, pp. 7:1–7:8. [Online]. Available: http://doi.acm.org/10.1145/1366919.1366929

[171] E. Stai, V. Karyotis, and S. Papavassiliou, "Exploiting socio-physical network interactions via a utility-based framework for resource management in mobile social networks," *IEEE Wireless Communications*, vol. 21, no. 1, pp. 10–17, February 2014.

[172] B. Azimdoost, H. R. Sadjadpour, and J. J. Garcia-Luna-Aceves, "Capacity of Wireless Networks with Social Behavior," *IEEE Transactions on Wireless Communications*, vol. 12, no. 1, pp. 60–69, January 2013.

[173] D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins, "Geographic routing in social networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 33, pp. 11 623–11 628, 2005. [Online]. Available: http://www.pnas.org/content/102/33/11623.abstract

[174] J. Hu, L. L. Yang, H. V. Poor, and L. Hanzo, "Bridging the Social and Wireless Networking Divide: Information Dissemination in Integrated Cellular and Opportunistic Networks," *IEEE Access*, vol. 3, pp. 1809–1848, 2015.

[175] Z. Zhang, H. Wang, C. Wang, and H. Fang, "Modeling epidemics spreading on social contact networks," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 3, pp. 410–419, Sept 2015.

[176] H. Sun and C. Wu, "Epidemic forwarding in mobile social networks," in *2012 IEEE International Conference on Communications (ICC)*, June 2012, pp. 1421–1425.

[177] O. Yagan, D. Qian, J. Zhang, and D. Cochran, "Conjoining speeds up information diffusion in overlaying social-physical networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 6, pp. 1038–1048, June 2013.

[178] Y. Zhu, B. Xu, X. Shi, and Y. Wang, "A survey of social-based routing in delay tolerant networks: Positive and negative social effects," *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 387–401, First 2013.

[179] Y. Li, P. Hui, D. Jin, L. Su, and L. Zeng, "Evaluating the Impact of Social Selfishness on the Epidemic Routing in Delay Tolerant Networks," *IEEE Communications Letters*, vol. 14, no. 11, pp. 1026–1028, November 2010.

[180] Y. Li, G. Su, D. O. Wu, D. Jin, L. Su, and L. Zeng, "The impact of node selfishness on multicasting in delay tolerant networks," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 5, pp. 2224–2238, Jun 2011.

[181] Z. Wang, C. Wu, L. Sun, and S. Yang, "Peer-assisted social media streaming with social reciprocity," *IEEE Transactions on Network and Service Management*, vol. 10, no. 1, pp. 84–94, March 2013.

[182] G. Xue, Q. He, H. Zhu, T. He, and Y. Liu, "Sociality-aware access point selection in enterprise wireless lans," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 10, pp. 2069–2078, Oct 2013.

[183] A. Turk, R. O. Selvitopi, H. Ferhatosmanoglu, and C. Aykanat, "Temporal workload-aware replicated partitioning for social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 11, pp. 2832–2845, Nov 2014.

[184] L. Hanzo, T. Liew, and B. Yeap, *Turbo Coding, Turbo Equalisation and Space-Time Coding.* John Wiley & Sons, August 2002.

[185] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of anthropological research*, pp. 452–473, 1977.

[186] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, pp. 35–41, 1977.

[187] X. Li, R. Zhang, and L. Hanzo, "Cooperative Load Balancing in Hybrid Visible Light Communications and WiFi," *IEEE Transactions on Communications*, vol. 63, no. 4, pp. 1319–1329, April 2015.

[188] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, July 1948.

[189] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 03 1951. [Online]. Available: http://dx.doi.org/10.1214/aoms/1177729694

[190] T. van Erven and P. Harremos, "Rényi Divergence and Kullback-Leibler Divergence," *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 3797–3820, July 2014.

[191] A. Hobson, *Concepts in statistical mechanics.* CRC Press, 1971.

[192] "IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements Part Ii: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11g-2003 (Amendment to IEEE Std*

802.11, 1999 Edn. (Reaff 2003) as amended by IEEE Stds 802.11a-1999, 802.11b-1999, 802.11b-1999/Cor 1-2001, and 802.11d-2001), pp. i–67, 2003.

[193] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions: with formulas, graphs, and mathematical tables.* Courier Corporation, 1964, vol. 55.

[194] H. Ishibuchi, N. Akedo, and Y. Nojima, "Behavior of Multiobjective Evolutionary Algorithms on Many-Objective Knapsack Problems," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 264–283, April 2015.

[195] W. Zhang, M. F. Brejza, T. Wang, R. G. Maunder, and L. Hanzo, "Irregular Trellis for the Near-Capacity Unary Error Correction Coding of Symbol Values From an Infinite Set," *IEEE Transactions on Communications*, vol. 63, no. 12, pp. 5073–5088, Dec 2015.

[196] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, April 1967.

[197] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, March 1973.

[198] T. Hogg, "Quantum search heuristics," *Phys. Rev. A*, vol. 61, p. 052311, Apr 2000. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.61.052311

[199] Q. You, Y. Li, M. S. Rahman, and Z. Chen, "A near optimal routing scheme for multi-hop relay networks based on Viterbi algorithm," in *2012 IEEE International Conference on Communications (ICC)*, June 2012, pp. 4531–4536.

[200] Y. Wang, M. Z. Bocus, and J. P. Coon, "Dynamic Programming for Route Selection in Multihop Fixed Gain Amplify-and-Forward Relay Networks," *IEEE Communications Letters*, vol. 17, no. 5, pp. 932–935, May 2013.

[201] X. Li, F. Jin, R. Zhang, J. Wang, Z. Xu, and L. Hanzo, "Users First: User-Centric Cluster Formation for Interference-Mitigation in Visible-Light Networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 39–53, Jan 2016.

[202] S. Feng, X. Li, R. Zhang, M. Jiang, and L. Hanzo, "Hybrid Positioning Aided Amorphous-Cell Assisted User-Centric Visible Light Downlink Techniques," *IEEE Access*, vol. 4, pp. 2705–2713, 2016.

[203] J. L. Rebelatto, B. F. Uchoa-Filho, Y. Li, and B. Vucetic, "Adaptive Distributed Network-Channel Coding," *IEEE Transactions on Wireless Communications*, vol. 10, no. 9, pp. 2818–2822, September 2011.

[204] H. V. Nguyen, C. Xu, S. X. Ng, and L. Hanzo, "Non-coherent near-capacity network coding for cooperative multi-user communications," *IEEE Transactions on Communications*, vol. 60, no. 10, pp. 3059–3070, October 2012.

[205] H. V. Nguyen, S. X. Ng, and L. Hanzo, "Irregular convolution and unity-rate coded network-coding for cooperative multi-user communications," *IEEE Transactions on Wireless Communications*, vol. 12, no. 3, pp. 1231–1243, March 2013.

[206] H. Chen, Y. Li, J. L. Rebelatto, B. F. Ucha-Filho, and B. Vucetic, "Harvest-Then-Cooperate: Wireless-Powered Cooperative Communications," *IEEE Transactions on Signal Processing*, vol. 63, no. 7, pp. 1700–1711, April 2015.

[207] S. Gupta, R. Zhang, and L. Hanzo, "Throughput maximization for a buffer-aided successive relaying network employing energy harvesting," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, pp. 6758–6765, Aug 2016.

[208] ——, "Energy Harvesting Aided Device-to-Device Communication Underlaying the Cellular Downlink," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2016.

[209] C. Dong, L. L. Yang, and L. Hanzo, "Performance of Buffer-Aided Adaptive Modulation in Multihop Communications," *IEEE Transactions on Communications*, vol. 63, no. 10, pp. 3537–3552, Oct 2015.

[210] C. Dong, L. L. Yang, J. Zuo, S. X. Ng, and L. Hanzo, "Energy, Delay, and Outage Analysis of a Buffer-Aided Three-Node Network Relying on Opportunistic Routing," *IEEE Transactions on Communications*, vol. 63, no. 3, pp. 667–682, March 2015.

[211] Y. Chen, S. Zhang, S. Xu, and G. Y. Li, "Fundamental trade-offs on green wireless networks," *IEEE Communications Magazine*, vol. 49, no. 6, pp. 30–37, June 2011.

[212] M. A. Maddah-Ali and U. Niesen, "Fundamental Limits of Caching," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.

[213] J. Tadrous and A. Eryilmaz, "On Optimal Proactive Caching for Mobile Networks With Demand Uncertainties," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2715–2727, Oct 2016.

[214] B. Azimdoost, C. Westphal, and H. R. Sadjadpour, "Fundamental Limits on Throughput Capacity in Information-Centric Networks," *IEEE Transactions on Communications*, vol. 64, no. 12, pp. 5037–5049, Dec 2016.

[215] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. C. Chen, and L. Hanzo, "Machine Learning Paradigms for Next-Generation Wireless Networks," *IEEE Wireless Communications*, vol. PP, no. 99, pp. 2–9, December 2016.

[216] A. Kapoor, N. Wiebe, and K. Svore, "Quantum Perceptron Models," in *Advances in Neural Information Processing Systems*, 2016, pp. 3999–4007.

[217] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 427–450, Feb 2009.

[218] H. Wymeersch, S. Marano, W. M. Gifford, and M. Z. Win, "A Machine Learning Approach to Ranging Error Mitigation for UWB Localization," *IEEE Transactions on Communications*, vol. 60, no. 6, pp. 1719–1728, June 2012.

[219] J. Chen, W. Dai, Y. Shen, V. K. N. Lau, and M. Z. Win, “Power Management for Cooperative Localization: A Game Theoretical Approach,” *IEEE Transactions on Signal Processing*, vol. 64, no. 24, pp. 6517–6532, Dec 2016.

[220] T. Zhang, A. F. Molisch, Y. Shen, Q. Zhang, H. Feng, and M. Z. Win, “Joint Power and Bandwidth Allocation in Wireless Cooperative Localization Networks,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 10, pp. 6527–6540, Oct 2016.

[221] Y. Han, Y. Shen, X. P. Zhang, M. Z. Win, and H. Meng, “Performance Limits and Geometric Properties of Array Localization,” *IEEE Transactions on Information Theory*, vol. 62, no. 2, pp. 1054–1075, Feb 2016.

[222] K. Witrisal, P. Meissner, E. Leitinger, Y. Shen, C. Gustafson, F. Tufvesson, K. Haneda, D. Dardari, A. F. Molisch, A. Conti, and M. Z. Win, “High-Accuracy Localization for Assisted Living: 5G systems will turn multipath channels from foe to friend,” *IEEE Signal Processing Magazine*, vol. 33, no. 2, pp. 59–70, March 2016.

Subject Index

Author Index