UNIVERSITY OF SOUTHAMPTON

FACULTY OF BUSINESS AND LAW

Southampton Business School

Routing Products or People: Single and Multiobjective Constrained Shortest Path and Related Problems

by

Yi Qu

Thesis for the degree of Doctor of Philosophy

February 2016

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF BUSINESS AND LAW
Southampton Business School

Doctor of Philosophy

ROUTING PRODUCTS OR PEOPLE: SINGLE AND MULTIOBJECTIVE CONSTRAINED SHORTEST PATH AND RELATED PROBLEMS

by Yi Qu

The aim of this thesis is to define, model and solve three research questions at the tactical and operational levels of decision making, the former arising in the context of intermodal service network design and the latter two in passenger transportation, by using integer programming, dynamic programming and heuristics. The first research question concerns intermodal freight transportation, which is concerned with the shipment of commodities from their origin to destination using combinations of transport modes. Traditional logistics models have concentrated on minimising transportation costs by appropriately determining the service network and the transportation routing. The first chapter considers an intermodal transportation problem with a detailed consideration of greenhouse gas emissions and intermodal transfers. Two mathematical models, one time-invariant and the other time-dependent, are described for the problem, which are both in the form of a non-linear integer programming formulation, but which are linearised. A hypothetical but realistic case study of the UK forms the test instances for our investigation, where uni-modal with multimodal transportation options are compared using a range of fixed costs. The second and third research questions concern the multiobjective shortest path problem (MSPP) and the constrained multiobjective shortest path problem (CMSPP), extensions of the classical shortest path problem, with a wide range of practical applications particularly in passenger transportation. The second and third research questions are studied in two different chapters. The first of these presents several labelling algorithms for the MSPP and the CMSPP. Extensive testing is performed on different types of networks, including randomly generated and grid networks. The results show that label correcting algorithms are more efficient than label setting algorithms for solving both the MSPP and the CMSPP. The second of these two chapters proposes two fast local search algorithms for the MSPP. Four performance indicators are used to evaluate the local search solutions. Computational results demonstrate that local search algorithms are faster than all heuristic methods for the MSPP presented in literature, and able to produce reasonably good-quality solutions.

中文摘要

Abstract in Chinese

本文利用最优化理论与方法, 结合整数规划,动态规划和启发式算法等经典算法求解平台与智能求解机制, 循序渐进的研究联合货运和客运服务网络设计问题。首先, 联合货物运输是指使用两种或两种以上的运输方式，完成货物从起点到终点运输任务的综合运输方式。传统的服务网络设计物流模型主要通过设计流量和路径分配致力于减少运输成本。本文第一部分在传统服务网络设计的基础上，考虑了碳排放和联合运输中转的成本，并构建了两个非线性整数规划模型，一个随时间变化，另一个不随时间变化。我们用包括英国的运货汽车，火车和轮船的实际运营网络对模型进行测试，结果显示联合运输比单一运输方式极大地减少了运输成本，同时，碳排放也大幅减少。其次，最短路径问题是具有广泛应用和影响的客运服务网络设计问题之一。本文第二部分和第三部分在经典的最短路径问题基础上，研究了多目标最短路径问题和有限制条件的多目标最短路径问题。本文第二部分对多目标最短路径问题和有限制条件的多目标最短路径问题，选择并提出了几种标号算法，并采用两种类型的网络数据，包括随机网络和格状网络，对几种算法的实际运行效率进行了对比试验。试验结果表明，标号改正算法优于标号设定算法。本文第三部分针对启发式算法在效率方面的不足，对多目标最短路径问题提出了两个有效率的局部搜索算法。通过四种性能评估方法及Pareto-最优解的比较，来对局部搜索算法进行检验分析。测试结果表明，局部搜索算法在收敛速度方面表现明显优于其他启发式算法，而且得到质量较高的解。

I would like to dedicate this thesis to my loving parents.

Table of Contents

List of Tables

List of Figures

Declaration of Authorship

I, Yi Qu, declare that the thesis entitled *Routing Products or People: Single and Multi-objective Constrained Shortest Path and Related Problems* and the work presented in the thesis are both my own, and have 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: (Qu et al., 2014)

Signed: Yi Qu

Date: January 2016

Acknowledgements

I would like to thank everyone who has helped me in the production of this thesis and has made my life in Southampton so rewarding and enjoyable.

First, I owe my greatest thanks to my main supervisor Prof Tolga Bektaş. His great encouragement gave me the inspiration and confidence to start pursuing my academic dream. His unwavering support and excellent guidance have enabled me to sustain every step of my Ph.D. journey. I am so fortunate and privileged to have had the opportunity of studying and conducting research with such a great mentor.

I would like to thank my other supervisor, Prof Julia Bennell. I appreciate her invaluable comments and instruction which guided me in the right direction and for her counsel when I needed advice or encouragement along the way. Special thanks go to Prof Gilbert Laporte, Prof Douglas Macbeth and Dr Yue Wu, who were willing to participate in my presentations and provided valued feedback.

I would like to thank Dr Andrea Raith and Prof. Matthias Ehrgott for making their instances available for testing in my thesis and paper.

I would also like to acknowledge the use of the IRIDIS High Performance Computing Facility, and the associated support services at the University of Southampton, in the completion of this work.

So many people have helped me in my Ph.D. study. I would like to thank them all, particularly those in Southampton.

Finally, my deepest gratitude goes to my parents, Mr Jiwen Qu and Ms Dianping Pan for their encouragement, support and love, at all times. They have raised me through all those hard years and have done everything possible to ensure that I receive their unconditional love and the very best education. Thanks for always being there and loving me more and more each passing day.

Chapter 1

Introduction

1.1 Background

The transportation industry is changing dramatically due to technological advances and the constant need to find an integrated and more efficient transportation system for increasing volumes of freight and passengers across the globe, the volume of which has grown rapidly in the last four decades. In 1971, domestic freight and passenger transport totalled just 134 billion tonne-kilometres and 418 billion passenger-kilometres, while by 2010 it had expanded to more than 222 billion tonne-kilometres and 771 billion passenger-kilometres, respectively (Department for Transport, 2014). Intermodal transportation is concerned with the shipment of goods and people using combinations of transport modes, particularly over longer distances and across international borders, which has played a significant role in the transportation industry.

An intermodal freight transportation includes ocean and coastal routes, inland waterways, railways, roads, and airways. In relation to the carriage of goods, intermodal transportation is the shipment of commodities from one point to another using combinations of at least two different transportation modes (e.g., truck to train to barge to ocean-going vessel) (Bektaş and Crainic, 2007). Commodities must be loaded, unloaded and transferred at an intermodal terminal. An illustration of an intermodal freight transportation chain is shown in Figure 1.1.

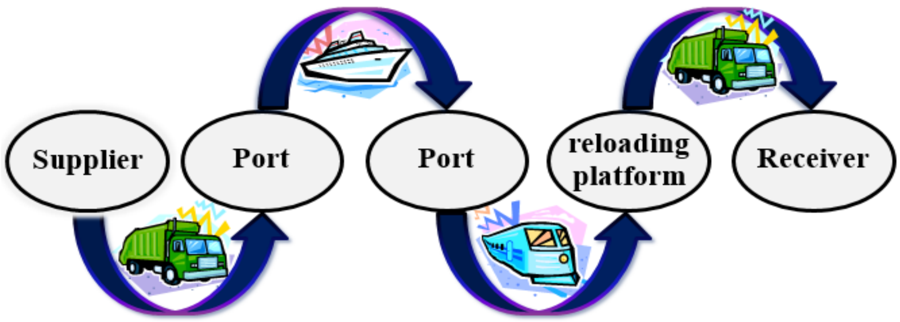


Fig. 1.1 An illustration of an intermodal freight transportation chain

In Figure 1.1, containers leave the supplier's facilities by truck to a port, from which they are reloaded onto a ship to be transported to another port. Containers arriving into the latter port are then transferred onto a train, and sent to another loading platform from which goods are carried by truck to their final destinations. The load of freight must be overseen by a manager from its origin to its destination.

Intermodal passenger transportation, on the other hand, involves using two or more transport modes for passengers on a journey. Its systems often involve rapid transit modes, including rail, bus, taxi, ferry, air, bicycle and even walking. In this setting, each passenger is an independent decision-maker. A real-life illustration of an intermodal passenger transportation network in Southampton is shown in Figure 1.2.

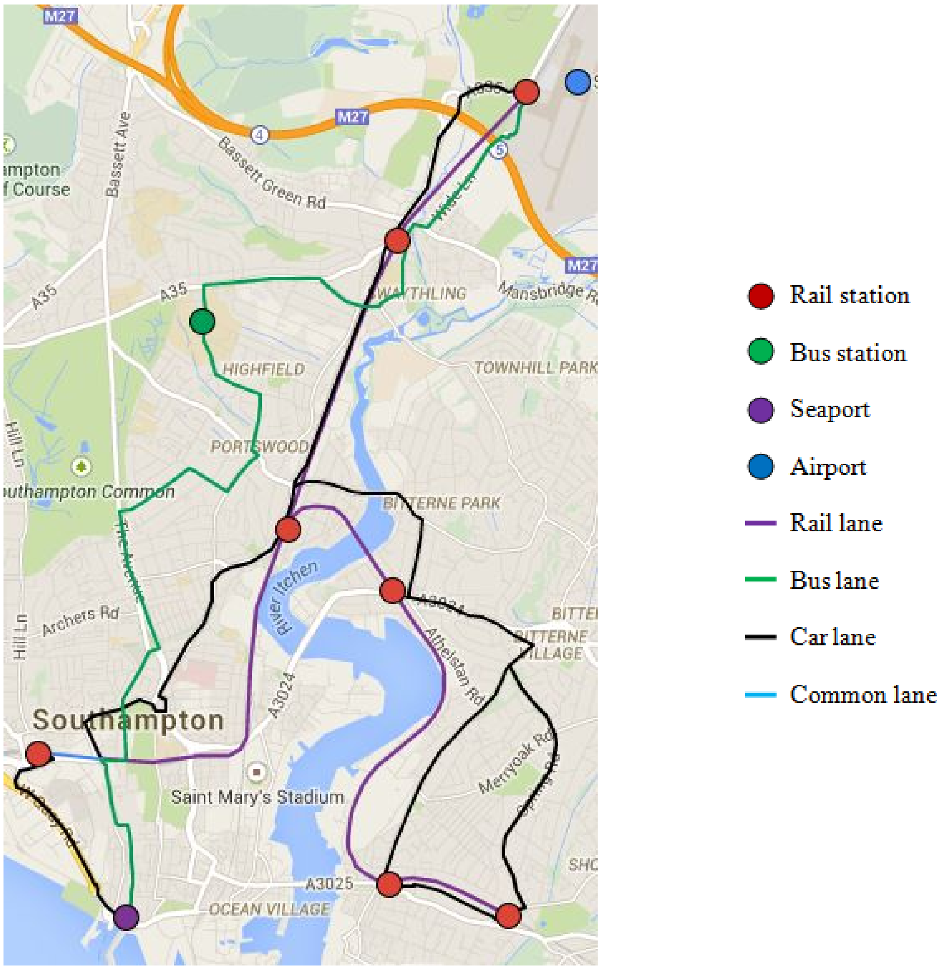


Fig. 1.2 An illustration of an intermodal passenger transportation network

In Figure 1.2, a passenger travels from Southampton airport (shown by the blue dot) to the seaport (indicated by the purple dot). The passenger could travel by taxi, which is faster but more expensive, or use other public transport, such as the train service from Southampton airport to Southampton central railway station, from which they will have alternative modes to go to the seaport. The passenger either selects a bus service, which costs a small amount of money and takes a few minutes, or walks to the seaport, which costs nothing but takes longer.

Increasing freight and passenger transportation brings with it concerns about air quality and climate change. Freight and passenger transportation is largely driven by fossil fuel combustion, mostly diesel fuel, resulting in emissions of greenhouse gases (GHG), such as carbon dioxide (CO_2), nitrogen oxide (NO_X), sulphur oxide (SO_X), particulate matters and air toxics. GHG emissions are not only harmful to the health of humans, but also have harmful impacts on the environment. Examples of the latter include increased drought, heavier downpours and flooding, a rise in sea levels and harm to water resources, agriculture, wildlife and ecosystems. Global emissions of CO_2 as the primary GHG increased by 3% in 2011, reaching an all-time high of 34 billion tonnes in 2011 (Oliver et al., 2012). The proportion of total GHG emissions in the UK attributable to transport has increased from 18% in 1990 to 26% in 2012. In 2012, 21% of UK domestic GHG emissions were from transport at 118MtCO_2e. Transport contributes 26% of the total GHG emissions when both domestic and international emissions are included (Department for Transport, 2014). An efficient, safe, flexible and environmentally friendly way of operating transportation networks is of vital importance.

Following the categorisation of Crainic and Laporte (1997), decision makers, in designing and operating a transportation network, are faced with planning problems associated with three different time spans. This includes those at the strategic, tactical and operational level of planning. Long-term, strategic planning involves the highest level of management. Decisions at this planning level affect the design of the physical infrastructure network. Strategic planning, therefore, usually requires large capital investments over a long period of time, e.g., between one and five years. Tactical planning helps to improve the performance of the whole transportation system by ensuring an efficient allocation of existing resources over the medium-term, such as six months to a year. Finally, short-term operational planning is performed by local management. At this level, day-to-day decisions are made in a highly dynamic environment where the time factor plays an important role. A detailed literature review of the three planning levels in freight and passenger transportation will be presented in Chapter 2.

The service network design problem is a key tactical problem in intermodal transportation. Service network design decisions generally relate to the routes on which services will run, the types and capacity of the services offered, as well as the frequency of the schedule of each route (Crainic, 2000). The performance of the service network design is evaluated by the trade-offs between total operating costs and service quality. In solving freight transportation network design problems, a great amount of effort has been dedicated to the variant of the problem where there are no limitations on the transportation capacity. Furthermore, traditional service network design models have concentrated on minimising operational costs alone. There is an increasing need for new planning models to capture environmental measures, and in particular greenhouse emissions.

Operational decisions relate to the specific processes that take place at a local level of an organisation, including the scheduling of services, vehicle distribution, routing and repositioning. As one of the most important problems, finding the shortest path between the origin and the destination is broadly used in many areas, such as urban traffic planning, routing of telecommunications messages, robot navigation and texture mapping. In relation to path planning in transportation, a typical problem is often better represented by considering multiple, as opposed to single, objectives, using indicators such as time, distance, or greenhouse gas emissions, which must also be taken into account simultaneously. The multiobjective shortest path problem (MSPP) is a multiobjective combinational optimisation problem (Ehrgott and Gandibleux, 2000), which consists of finding all efficient solutions in a given network that minimise all criteria under consideration. The MSPP, and in particular the bi-objective shortest path problem, has been widely studied previously. However, there is limited work on this class of problems with three or more objectives. To the best of our knowledge, the multiobjective shortest path problem, either with single or multiple resource constraints (CMSPP), has not yet been studied.

1.2 Aims and Objectives

The aim of this thesis is to define, model and solve three research questions under two main and interdependent research topics at the tactical and operational levels of decision making. The former arises in the context of intermodal service network design and the latter in passenger transportation. The problem will be addressed by using integer programming, dynamic programming and heuristic algorithms. This will be achieved through the following three objectives:

1. First, we will propose new models for the tactical service network design problem using wider objectives that account for greenhouse gas emissions. A formulation will be described, which, to the best of our knowledge, is the first to explicitly include both intermodal transfer costs and greenhouse gas emissions in the objective. Furthermore, a time-space network design model will be presented better to capture the time-related aspects of network

design problems for multimodal multicommodity freight transportation. The formulation will be used to solve realistic case studies to derive insights on the trade-offs between various measures of performance, including costs and emissions.

2. For the operational level design problem, we will study the multiobjective shortest path problems with or without constraints, and describe labelling algorithms for the MSPP with three or more objectives. We will also perform a computational comparison of two label setting algorithms and one label correcting algorithm for the MSPP.

3. In many applications, fast algorithms are required for the MSPP. There is little work in the literature on solving the MSPP using heuristics. Furthermore, the existing literature does not report any computational comparisons between exact and heuristic algorithms for the MSPP with three or more objectives. To fill this gap, we will propose local search algorithms to solve the MSPP and present computational results, including computational comparisons between local search algorithms and the labelling algorithms developed in Objective 2.

1.3 Organisation of the Thesis

The rest of the thesis is organised as follows. Chapter 2 reviews various problems and issues in a transportation system according to the three levels of planning, i.e., strategic, tactical and operational. Chapter 3, entitled "Multimode Multicommodity Intermodal Service Network Design" presents time-invariant and time-dependent service network design formulations and the associated computational results. Chapter 4, entitled "Labelling Algorithms for Multiobjective Shortest Path Problems", describes a number of labelling algorithms for the MSPP, and the CMSPP, some of which are new. Chapter 4 also presents computational results carried out on randomly generated and grid network instances. Chapter 5, entitled "Local Search for Multiobjective Shortest Path Problems", introduces two local search algorithms for the MSPP and presents comparisons between the local search and exact algorithms. Finally, conclusions will be offered in Chapter 6, including limitations of the study, and suggestions for future research. Since the thesis follows a three-paper approach, an overview of the three research papers and the corresponding chapters are described in Table 1.1.

Table 1.1 Overview of Chapters 3, 4 and 5

	Chapter 3	Chapter 4	Chapter 5
Level of decision making	Operational Tactical	Operational	Operational
Number of objectives	1 and 2	≥ 2	≥ 2
Type of objectives	Environmental	Any	Any
Solution algorithm	Exact	Exact	Heuristic
Problem type	Freight	Passenger	Passenger
Number of commodities	Multiple	Single	Single

1.4 Research Outputs

This thesis has resulted in the following outputs:

1. Research papers

    - Qu, Y., Bektaş, T., and Bennell, J. (2014). Sustainability SI: Multimode multicommodity network design model for intermodal freight transportation with transfer and emission costs. *Networks and Spatial Economics*. (in press).

2. Conference and workshop presentations

    - Qu, Y., Bektaş, T., and Bennell, J. 2012. Green service network design for intermodal freight transportation. September, 2012. OR54 Annual Conference, Edinburgh, UK.
    - Qu, Y., Bektaş, T., and Bennell, J. 2013. Minimizing greenhouse gas emissions in intermodal freight transportation through mathematical modelling. July, 2013. 26th European Conference on Operational Research, Rome, Italy.
    - Qu, Y., Bektaş, T., and Bennell, J. 2013. Green time-space service network design for intermodal freight transportation. September, 2013. 10th International Workshop on Cutting, Packing and Related Topics, Lake Constance, Germany.
    - Qu, Y., Bektaş, T., and Bennell, J. 2014. A computational study of labelling algorithms for multiobjective shortest path problems with or without resource constraints. September, 2014. OR56 Annual Conference, Surrey, UK.

Chapter 2

Literature Review

2.1 Introduction

Transportation systems are complex structures that involve substantial human and material resources and which consequently display intricate relationships and trade-offs between the various decision and management policies that affect their respective components. The existing literature on transportation systems has previously thoroughly documented many of the issues arising.

One of the most relevant studies in this field is that of Crainic and Laporte (1997), which uses a convenient and effective research framework that classifies general transportation planning problems into three levels according to the planning horizon, namely strategic, tactical and operational levels. This chapter will present an overview of the studies in relation to the three levels of planning, relevant to the aim and objectives of this dissertation. In particular, Sections 2.2, 2.3 and 2.4 will discuss the concepts and models within strategic, tactical, and operational level problems, respectively. Section 2.5 will then identify some of the gaps and limitations of existing studies, which the present study will aim to fill.

2.2 Strategic Level Problems

Strategic (long-term) planning is policy driven and involves the highest level of management. Examples of strategic level decisions include those related to the planning of infrastructure, workforce, equipment and facility location. A number of models of strategic planning are discussed in Crainic and Laporte (1997), including location models, network design models and regional multimodal planning models. Other types of problems at strategic level identified in Macharis and Bontekoning (2004) include cooperation between drayage companies, determination of truck and chassis fleet size, and terminal design. The following section will mainly focus on the location problems and network design problems. The relevant literature will be discussed in the following sections.

2.2.1 Location Problems

Location problems involve the sitting of one or several facilities, usually on a set of potential nodes of a given network, in order to facilitate the movement of goods or the provision of services along the network (Drezner and Hamacher, 2002). Location models are often categorised into three types, as follows (Crainic and Laporte, 1997):

1. Covering models: This problem is either to find the locations of the minimum number of facilities to cover the demands of a given set of customers, or to find a set of nodes that maximise the total demand that is covered.

2. Centre models: This problem is to find the locations of p facilities on a graph, which minimise the maximal distance between a node and a facility.

3. Median models: This problem is to find the locations of p facilities on a graph, which minimise the sum of distances between the nodes and their nearest located facility.

Klose and Drexl (2005) presented a detailed review on location problems that covers the broader literature and several formulations. They classified location problems in a different way to Crainic and Laporte (1997), and reviewed continuous location models, network location models and mixed-integer programming models, summarising a variety of applications of location models.

For the most recent and comprehensive review on location problems, readers are referred to Laporte et al. (2015), which provides an exhaustive coverage of concepts and location models, detailing how the location models are applied to solve real-world problems, as well as giving an extended overview on the interaction between location and other related problems.

Hub location problems are extensions of classical facility locations problems, which have applications in both transportation and telecommunication systems. Hubs provide connections between many origins and destinations and serve as switching, sorting, and consolidation points for flows of freight and passengers in a hub-and-spoke type transportation network. In other words, hubs are used to increase the number of transportation links between the origin and destination nodes (Farahani et al., 2013). For instance, without a hub node, a fully connected network with n nodes has $n(n-1)$ origin-destination links. However, if a hub node is selected to connect all other nodes (also known as spokes) with each other, only $2(n-1)$ connections or links are needed to serve all origin-destination pairs. Readers are referred to Campbell et al. (2005a), Campbell et al. (2005b) and Hekmatfar and Farahani (2009) for surveys on the problem. Alumur and Kara (2008) presented a review on different types of such problems, including the p-hub median problem, the hub location problem with fixed costs, the p-hub centre problem, and hub covering problems. Alumur and Kara (2008)'s review covers the hub location literature until 2007, which is continued by Farahani et al. (2013) who provide a more recent coverage and classify papers in terms of models, solutions, performance measures and applications.

2.2.2 Network Design Problems

Network design problems are at the strategic level of decision making and are concerned with the configuration of the infrastructure network, including improving existing infrastructure, or establishing new roads, railways, sea links and freight terminals. They also consider the type, number and way in which equipment is used, and the type and capacity of load unit storage facilities, along with the operations carried out at the hubs.

Network design problems are generalisations of location formulations. They are defined on a network, which contains nodes and links, in which the directed and the undirected links are represented by arcs and edges, respectively. In the network design problem, some or all of the links between nodes of the network have to be established at a fixed "design" cost such that demands between a set of node pairs can be transported on the network.

Bruynooghe (1972) first considered adding new links or selecting existing links for improvement in the transport network with given demand from each origin to each destination, which resulted in the discrete network design problem (DNDP). Lower bounds were computed by relaxing integrability requirements and by underestimating the objective function with a continuous function. However, no computational results are reported.

A framework proposed by Balakrishnan et al. (1989) presented a family of dual-ascent algorithms for large-scale uncapacitated network design problems. In Balakrishnan's model, a network $G = (N, A)$ is constructed with the assumption that each commodity has a single origin and a single destination. The dual-ascent algorithms were applied to the network design problems on instances ranging in size from 20 nodes, 80 arcs and 380 commodities to 45 nodes, 500 arcs and 1980 commodities. Computational results showed that almost all cases were solved to near-optimality. The gap between their dual-based upper and lower bounds, expressed as a percentage of the lower bound, was less than 4%.

Holmberg and Yuan (2000) considered a fixed charge network design model for capacitated multicommodity network flow problems, which set limitations on the amount of flow that can pass on each arc. A mathematical formulation is presented as follows:

$$\text{minimise} \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^k x_{ij}^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \tag{2.1}$$

subject to

$$\sum_{j:(i,j)\in A} x_{ij}^k - \sum_{j:(j,i)\in A} x_{ji}^k = \begin{cases} r_k, & i = o(k) \\ -r_k, & i = d(k) \quad \forall i \in N, \forall k \in K \\ 0, & \text{otherwise}, \end{cases} \tag{2.2}$$

$$\sum_{k\in K} x_{ij}^k \leq u_{ij} y_{ij}, \qquad \forall (i,j) \in A \tag{2.3}$$

$$x_{ij}^k \leq l_{ij}^k y_{ij}, \qquad \forall (i,j) \in A, \forall k \in K \tag{2.4}$$

$$x_{ij}^k \geq 0, \qquad \forall (i,j) \in A, \forall k \in K \tag{2.5}$$

$$y_{ij} \in \{0,1\}, \quad \forall (i,j) \in A. \tag{2.6}$$

In this model,

$$l_{ij}^k = \min(r_k, u_{ij}), \tag{2.7}$$

where k denotes each commodity in a large set of commodities, K. For each k, r_k is the required amount of flow of commodity k to be sent from the point of its origin $o(k)$ to its destination $d(k)$. u_{ij} is the maximum capacity for each arc (i,j).

The model shown by (2.1) – (2.7) is in the form of a linear cost, capacitated, multicommodity network design formulation. The objective function (2.1) represents the total cost of the network, including the fixed costs of the selected links and the variable costs of flows. A bundle of flow constraint is shown by (2.2), used to meet the origin-destination demands. Constraint (2.3) models the arc capacity constraints. In particular, the constraints indicate that, for each commodity, the total flow on arc (i,j) must not exceed its capacity u_{ij}. If the arc (i,j) is not chosen in the network (i.e. $y_{ij} = 0$), then the flow on arc (i,j) is 0. This particular feature differentiates this model from that of the uncapacitated network design models. The linking constraints (2.4) are disaggregations of the capacity constraint (2.3), which is redundant, but is used to provide better bounds in the context of relaxations.

For further details about various models and algorithms for network design problems, readers are referred to Steenbrink (1974), Magnanti and Wong (1984), Yang and Bell (1998) and Gao et al. (2005).

2.3 Tactical Level Problems

According to Crainic and Laporte (1997), service network design is a key tactical problem in freight transportation. Service network design formulations are generally used to determine the routes on which service will be offered as well as the frequency of the schedule of each specific route, while satisfying the capacity requirements, customer satisfaction and minimising the total operating costs (Crainic, 2000). For a comprehensive review of service network design problems, we refer readers to Crainic (2000), and Crainic and Kim (2007).

Over the last 20 years, research on intermodal freight and passenger transportation service network design has steadily grown. Problems in intermodal transportation are more complex due to the inclusion of different transport modes, multiple decision makers and types of load units. An overview on intermodal transportation is provided in Bektaş and Crainic (2007),Crainic and Kim (2007), and SteadieSeifi et al. (2014). A variety of mathematical solutions, operational research models and methods, have been applied to generate and evaluate the transportation network. Macharis and Bontekoning (2004) and Janic and Bontekoning (2002) present the opportunities for operational research in the intermodal transportation research application field. They defined various operational research problems of the network operator, the terminal operator, the drayage operator and the intermodal operator, and reviewed the associated mathematical models that were used in this field up until 2002. A general description of current issues and challenges related to large-scale implementation of intermodal transportation systems in the United States and Europe is presented by Zografos and Regan (2004). One common expanding area is the consideration of wider objectives and issues in intermodal transportation, especially related to GHG emissions.

One of the earlier studies in this context is by Winebrake et al. (2008b), who present a geospatial intermodal freight transport model to help analyse the cost, time-of-delivery, energy, and environmental impacts of intermodal freight transport. Three case studies are also applied to exercise the model. However, they use a single criterion objective function (such as minimising cost, or time, or CO_2). Winebrake et al. (2008b) also only considered a single origin-destination commodity.

Bauer et al. (2009) were one of the first to explicitly consider GHG emissions as a primary objective and proposed a linear cost, multicommodity, capacitated network design formulation to minimise the amount of GHG emissions of intermodal transportation activities. They applied this model to a real-life rail freight service network design problem and presented the trade-offs between conflicting objectives of minimising time-related and environmental costs.

One other often ignored aspect when modelling intermodal transportation problems is terminal operations. Whilst research exists on intermodal terminals, e.g., location and assignment (Vidović et al., 2011) and optimal pricing and space allocation (Holguín-Veras and Jara-Dírz, 2006), a detailed consideration of terminals within the context of the service network design needs more attention. As pointed out by Bektaş and Crainic (2007), terminals are 'perhaps the most critical components of the entire intermodal transportation chain', and 'the efficiency of the latter highly depends on the speed and reliability of the operations performed in the former'. It is therefore important to capture terminal operations in a modelling framework. However, it is difficult to explicitly model the speed and reliability of terminal operations at a tactical level, as these measures require a treatment at an operational level.

2.4 Operational Level Problems

Operational (short-term) planning is performed by local management at a high level of detail and flexibility on planning activities. The vehicle routing problem (VRP) is one of the most studied among the operational level problems. The VRP aims at servicing a number of customers using fleet vehicles. Several variations of the VRP exist, such as the VRP with pickup and delivery (VRPPD), the VRP with time windows (VRPTW), the capacitated VRP with or without time windows (CVRP or CVRPTW), the VRP with multiple trips (VRPMT). For a thorough review on the models and algorithms proposed for the VRP, readers are referred to Cordeau et al. (2002), Laporte (2007), Cordeau et al. (2007) and Toth and Vigo (2014). Other main operational decisions include fleet management, which has been developed for optimising and simulating the operation of transport fleets in order to serve the customers' demands with the objective of cost efficiency (Bielli et al., 2011; Crainic and Laporte, 1998; Dejax and Crainic, 1987), and crew scheduling, which is used to assign crews to vehicles and convoys (Azadeh et al., 2013; Crainic and Laporte, 1997; Crainic and Roy, 1992).

An important operational level problem is the shortest path problem(SPP). In the remainder of this section, we focus on the multiobjective SPP(MSPP) as it is relevant to passenger transportation and the research of this thesis. Sections 2.4.1 and 2.4.2 present a review of the literature on the solution methods proposed for two main problems studied in Chapter 4, in particular the unconstrained MSPP and the constrained MSPP (CMSPP), respectively. We also briefly review some extensions of this problem in Section 2.4.3.

2.4.1 The Unconstrained MSPP

The single-objective shortest path problem is one of the most widely studied network optimisation problems (Dantzig, 1966; Dijkstra, 1959; Floyd, 1962; Ford, 1956). For a thorough review, readers are referred to Cherkassky et al. (1996) and Zhan and Noon (1998). However, it is widely recognised that many network optimisation problems cannot be described by only one single-objective function (Current and Marsh, 1993). The MSPP is widely used in many applications, such as robot path planning (Fujimura, 1996), route planning (Clímaco et al., 2003; Machuca et al., 2012), computer networks (Kerbache and Smith, 2000), and satellite scheduling (Gabrel and Vanderpooten, 2002). In this section, we will present both exact and heuristic algorithms for the unconstrained MSPP.

According to Clímaco and Pascoal (2012), three main categories of exact algorithms exist for solving the MSPP, namely a *posteriori* aggregation of preferences methods, interactive methods and a *priori* aggregation of preferences methods.

A *posteriori* aggregation of preferences methods generate all Pareto-optimal solutions, while other methods do not. We consider three classes of this group of methods, in particular those based on (i) labelling, (ii) ranking, and (iii) two phases.

Labelling algorithms include label setting and label correcting methods. The former class is characterised by setting one of the labels of a node as permanent at each iteration of the algorithm, whereas the latter class considers all labels as temporary until the algorithm terminates. Hansen (1980) generalised Dijkstra's algorithm and was one of the first to propose a label setting method for the bi-objective SPP (BSPP). His paper described 10 different BSPP as well as their solution procedures. Martins (1984) presented a generalisation of Hansen's algorithm for the MSPP, where a lexicographically smallest temporary label is chosen at each iteration. Tung and Chew (1992) also described a label setting algorithm for the MSPP where a different ordering method is used for selecting the label at each iteration. Machuca et al. (2012) presented a computational comparison of three label setting algorithms that accept heuristic information, namely NAMOA* (Mandow and Pérez, 2008), MOA* (Stewart and White, III, 1991) and Tung & Chew's algorithm (Tung and Chew, 1992), for the BSPP. Results show that the NAMOA* is the most effective among these three algorithms regarding the space and time requirements. Two labelling algorithms for the MSPP were presented in Paixão and Santos (2013). Their computational results showed that the label correcting technique is faster to obtain the efficient set. A first-in first-out principle was used in a label correcting method to select nodes for solving the BSPP in Brumbaugh-Smith and Shier (1989). Skriver and Andersen (2000) presented two improvements to the Brumbraugh-Smith and Shier's algorithm. Raith and Ehrgott (2009) presented a comprehensive computational comparison of solution strategies for the BSPP, including a standard label correcting and label setting method, a ranking method using a near shortest path approach, and a two-phase method, investigating different approaches to solving problems arising in phases 1 and 2. Grid networks, random networks and road networks were used to compare different strategies for solving the BSPP. They concluded that label correcting and label setting outperformed other methods for most of the instances they considered, and the two phase method was competitive with other approaches for the BSPP. They also claimed that label correcting is faster than label setting for the BSPP. Other labelling methods for the multiobjective shortest path problem can be found in Sastry et al. (2005) and Guerriero and Musmanno (2001).

Ranking methods, like those proposed in Clímaco and Martins (1982) and Azevedo and Martins (1991), use a k-shortest path routine to solve the MSPP. Starting with the objective value of a Pareto-optimal solution, other paths are obtained in order of increasing value of this objective until the k-best solution is obtained. For the MSPP, the process continues until all Pareto-optimal solutions have been found. Computational results for the BSPP are presented.

In a two phase method, the whole procedure of solving the MSPP is divided into two phases. It was first introduced by Ulungu and Teghem (1995) for solving bi-objective combinatiorial optimisation problems, and later in Przybylski et al. (2010) for solving multiobjective integer programmes with more than two objectives. In phase 1, extreme supported efficient solutions are found. In phase 2, information from supported non-dominated points generated in Phase 1 is used to determine a search area in objective space that is guaranteed to contain all non-supported non-dominated points (Przybylski et al., 2010). For example, in the bi-objective combinatiorial optimisation problems, the search area consists of triangles defined by two consecutive supported efficient solutions. Mote et al. (1991) found the efficient supported paths by relaxing the integrality conditions and solving a simple bi-objective network problem in phase 1, and used a label correcting algorithm to find the nonsupported paths in phase 2. Raith and Ehrgott (2009), tested different implementation strategies for each phase, including label correcting dichotomic, label setting dichotomic and network simplex parametric for phase 1, and bi-objective label correcting, bi-objective label setting and a ranking method for phase 2.

Algorithms for solving the MSPP exist, which do not necessarily generate the complete efficient set. Interactive methods are one such category, where the search for approximate set is performed by considering a limited portion of the efficient set that is defined on the basis of a decision maker's preferences (Current et al., 1990; Granat and Guerriero, 2003). Interested readers on this topic are referred to Coutinho-Rodrigues and Clímaco (1999) and Murthy and Olson (1994).

Solution methods which use a utility function to define preferences among objectives were presented in Loui (1983), Carraway et al. (1990) and Modesti and Sciomachen (1998). In particular, weighting methods were used to transform the MSPP to a single-objective optimisation problem. However, they determined the minimal complete set of Pareto-optimal solutions, and therefore did not find all of those solutions.

In the last decade, considerable effort has also been put into heuristics to solve the MSPP, primarily based on genetic algorithms and ant colony optimisation, although they do not guarantee the optimality of the solutions found. Pangilinan and Janssens (2007) presented a genetic algorithm for the MSPP. Computational experiments were conducted on a set of 270 instances with three objectives and up to 200 nodes and 7960 arcs. Although the approximate set of solutions exhibited good diversity on two of the objectives, the solutions could not be fully evaluated because the efficient set was unavailable. They also compared their algorithms with the results in Gandibleux et al. (2006), which showed that their algorithm was slower than the labelling algorithm presented in Martins (1984). Lin and Gen (2007) proposed a genetic algorithm to solve the BSPP, which used a priority-based encoding to represent a path in the network. They compared their algorithm against the strength of a Pareto evolutionary algorithm (Zitzler and Thiele, 1999), a non-dominated sorting Genetic Algorithm II (NSGA-II) (Deb et al., 2002) and a random-weight Genetic Algorithm (Ishibuchi and Murata, 1998). The authors evaluated their solutions with respect to three measures of performance, including the number of obtained solutions, the proportion of efficient solutions and the average distance of the resulting approximate set from the Pareto front. Computational results on a set of instances with up to 500 nodes and 4978 arcs showed that their algorithm outperformed the others. Ghoseiri and Nadjari (2010) presented an ant colony optimisation algorithm to solve the BSPP and compared it to a label correcting algorithm, although the authors did not cite a reference for the latter. Computational results on randomly generated instances with up to 4000 nodes and

61783 arcs showed that their algorithm produces good quality solutions and runs faster than the label correcting algorithm. For all tested instances, the time saved using their algorithm is between 61.34% and 94.71%. Häckel et al. (2008) presented an ant colony optimisation algorithm for the MSPP, which was then improved by Bezerra et al. (2011) and Bezerra et al. (2013). However, the latter two did not compare their solutions with efficient solutions obtained by exact methods. They tested two types of instances, namely full networks with up to 200 nodes and 39800 arcs and grid networks with up to 1225 nodes and 4760 arcs. Their run-time for the instance with 200 nodes, 100% density and three objectives is 40 seconds, and the run-time for grid network instances is up to 100 seconds. When comparing with NSGA-II (Deb et al., 2002), their algorithm generated better approximations for all instances. When compared with the algorithm of Häckel et al. (2008), their algorithms performed better on the grid instances, while there was no dominance for the full network instances.

2.4.2 Constrainted Shortest Path Problem

In general, the constrained shortest path problem (CSPP) consists of finding the shortest path from a source node to a target node under constraints on the paths in the network (Handler and Zang, 1980; Santos et al., 2007). For example, one or more path-weight side constraints are added (Carlyle et al., 2008); Or the path may be constrained to prohibit the use of specific nodes (Villeneuve and Desaulniers, 2005), to include a specific number of nodes (Deo and Pang, 1984), or to include a given set of nodes such as pick-up node and delivery node as in the case of Desaulniers et al. (2002). Various algorithms have been developed for solving the CSPP. A k-shortest path algorithm and a Lagrangian relaxation algorithm were described in Handler and Zang (1980). A label setting algorithm for the CSPP was presented in Dumitrescu and Boland (2003). An exact algorithm was presented in Santos et al. (2007), which utilised a k-shortest path algorithm by defining a more efficient search direction, where the total traveling cost is minimised subject to an additional time constraint. The computational results showed that this algorithm can solve very large-scale problem instances, up to 40,000 nodes and 800,000 arcs, optimally and performs better than algorithms that are based on k-shortest paths and Lagrangian relaxation. However, they did not compare their algorithm with other exact algorithms, such as label setting and label correcting algorithms. Carlyle et al. (2008) described a Lagrangian relaxation algorithm which was tested on both single-constrained and multi-constrained networks. Royset et al. (2009) presented an "LRE" algorithm, which combines Lagrangian relaxation with an enumeration of near-shortest paths, to solve a constrained shortest path problem for routing various types of military aircraft. A thorough classification and generic formulations for the shortest path problem with resource constraints and the most commonly used solution methods are presented in Irnich and Desaulniers (2005).

2.4.3 Bottleneck and Non-additive Objectives

Let us now briefly review two other types of the MSPP, which received considerable attention in the literature, namely the MSPP with bottleneck objectives and the more general MSPP with non-additive costs in the objectives. A bottleneck objective is a minmax or a maxmin type. Gandibleux et al. (2006) generalised the algorithm described in Martins (1984) to the case when one of the objectives is a bottleneck. Their algorithms were tested on randomly generated networks with different sizes and densities. Computational results showed that their algorithms were not sensitive to the value ranges, and the number of efficient paths grew, not exponentially, with the density and network size. A tri-criterion shortest path problem with at least two bottleneck objectives was studied in de Lima Pinto et al. (2009) and de Lima Pinto and Pascoal (2010), where subgraphs were generated by restricting the set of arcs according to the bottleneck values, and the minimal complete set was generated by computing the Pareto-optimal solution in each subgraph, or the maximal complete set was generated by computing all Pareto-optimal solutions in the subgraphs. The latter paper enhances the method presented in the former by taking into account the objective values of the determined shortest paths to reduce the number of considered subgraphs, and thus reduce the number of solved shortest path problems. A label setting algorithm for the multiobjective shortest path problem with any number of sum and bottleneck objectives was proposed in Iori et al. (2010). Their results showed that the aggregate ordering algorithm performs consistently better than lexicographic ordering. They also confirmed that there is no clear dominance between the performances of the label setting and the label correcting algorithms for this particular problem.

The MSPP with non-additive cost is another special case of the MSPP, see Carraway et al. (1990) and Tsaggouris and Zaroliagis (2004). In Reinhardt and Pisinger (2011), weight functions were considered for the MSPP with a number of non-additive criteria to transform the multiple criteria into a single function, which was then solved by dynamic programming. Results on real-life MSPP with non-additive criteria were also reported. An algorithm for the BSPP with one non-additive cost was proposed in Chen and Nie (2013), which was based on approximating the non-additive function with a piecewise linear function and solving each segment sequentially.

2.5 Research Gaps

In this chapter, an overview will be offered on various problems and issues relevant to three levels of transportation planning, namely strategic, tactical and operational, followed by a literature review and methodological developments. Even though the existing literature has presented realistic models and powerful algorithms for many problems of freight and passenger transportation planning, a number of issues have not yet been explored, hindering the formation of more integrated and more efficient transportation systems. Our review has identified three important factors or algorithms that need to be considered related to freight and passenger intermodal transportation planning problems, which are explained in the rest of this section.

First, most models and algorithms have focused on the traditional economic aspects of intermodal freight transportation. Modelling sustainability and giving consideration to environmental impacts is needed in the light of the growing concerns about the externalities of transport. However, quantifying environmental impacts is a major challenge. The trend toward a more integrated and efficient transportation system with environmental concern is likely to remain.

Second, the MSPP, and in particular the BSPP, is not new to the operations research community. However, limited work exists on unconstrained MSPP with three or more objectives. Furthermore, and to the best of our knowledge, the CMSPP, either with single or multiple resource constraints, has not yet been studied.

Third, exact algorithms, such as labelling, ranking and two-phase, are guaranteed to find the complete set of Pareto-optimal solutions, but may take an exponential number of iterations. In practice, they are typically applicable to small instances only, due to long running times caused by their high complexity. There has been a need for fast and efficient heuristics and metaheuristics for the MSPP over the last ten years. There is very limited existing work presenting heuristics for solving the BSPP and MSPP. Genetic algorithms and ant colony optimisation based algorithms were presented for the MSPP, however, (1) their computational time in solving both the BSPP and MSPP is slow as indicated in Section 2.4.1; and (2) no literature exists evaluating the heuristic solutions compared with solutions using exact methods for the MSPP with three and more objectives. Consequently, more research is needed in developing heuristics and metaheuristics for the MSPP, along with an experimental study comparing exact methods and heuristics.

Chapter 3

Multimode Multicommodity Intermodal Service Network Design

3.1 Introduction

Intermodal transportation concerns the movements of freight or passengers from one point to another using at least two different transport modes. Traditional logistics models have concentrated on minimising operational transportation cost. However, the consideration of the wider objectives and issues, especially related to GHG emissions, leads to new models and technologies. In this chapter, a time-invariant and a time-dependent intermodal transportation model are described that include consideration of GHG emissions, in which CO_2 emissions are explicitly modelled. In our models, the objective is to minimise the total costs in an intermodal system, including the capital cost, operational cost, intermodal transfer cost, the GHG emission cost and the inventory cost, such that a number of commodities are shipped from their origins to their destinations. The inventory cost is only included in the time-dependent model. The decisions to be made comprise: (i) the selection of routes and transport modes and (ii) the flow distribution through the selected route and mode. The resulting green service network design models are non-linear mixed integer programs. A linearisation is proposed to transform each model into an integer linear programming formulation, which is then solved by off-the-shelf optimisation software.

The rest of the chapter is organised as follows: The subsequent section discusses a way of estimating emissions and presents a non-linear service network design model with intermodal transfer cost, which is then linearised. Section 3.3 describes a time-space service network design model, which is an extension of the model developed in Section 3.2. In Section 3.4, a hypothetical case study from the UK is provided and results of computational experiments and analyses for both models is indicated. In Section 3.5, a bi-criteria analysis for the multimode multicommodity network design problem is presented. The chapter concludes in Section 3.6.

3.2 Time-invariant Design

Time-invariant service network design targets tactical issues, including which type of service and transport mode to offer; which traffic routes to operate; and which commodity, and appropriate amount, should be shipped. The time-invariant service network design model is suitable for situations where the actual timing of the demand is unknown or there are high frequency services at each intermodal terminal, or when the timing of the demand is not considered. It is assumed that the designs obtained through a time-invariant model will remain static for the duration of the planning horizon for which the problem is solved.

3.2.1 Estimating Emissions

There are several approaches for estimating the GHG emissions including an energy-based approach and an activity-based approach. For our modelling approach which is at a more tactical level of planning, we have opted to use an activity-based function by McKinnon and Piecyk (2010) to estimate CO_2 emissions in intermodal transportation. This approach is also commonly used: e.g., Treitl et al. (2012) use it to estimate the total transport emissions in a petrochemical distribution network in Europe, and Park et al. (2012) use it to calculate CO_2 emissions in a road network for trucks and railway in intermodal freight transportation in Korea.

According to McKinnon and Piecyk (2010), the total cost of CO_2 emissions of a vehicle carrying a load of l (in tonnes) over a distance of d (in km) is calculated by Equation (3.1) below,

$$l \times d \times e, \tag{3.1}$$

where e is the average CO_2 emission factor (g/tonne-km). To convert CO_2 emissions into monetary units, we adopt the figures provided by the World Bank (The World Bank, 2012); specifically, we use \$100 per ton (=£71.60 per tonne).

The rationale for adopting this CO_2 emissions function is in its ease of use. First, it has the advantage of including variables to measure the total freight weight as well as the corresponding distance transported, while avoiding the elements that are hard to measure or calculate, such as the type of fuels or energy sources and the vehicles. Second, it is applicable to different transportation modes; for a given mode of transportation, the total CO_2 emissions are dependent on the shipping distance and the weight of the commodities.

3.2.2 Problem Description

The problem considered in this chapter is to designate the selection and scheduling of services by assigning a number of vehicles to links, then ship commodities from their origins to their destinations by respecting constraints on the link capacities, flows and requirements for demands. The number of vehicles is defined with respect to the unit carrying capacity regardless of the specific type of truck, train or ship.

The overall objective is to minimise the total cost, including variable, fixed, emission and transfer costs.

The problem is defined on a directed network. Our model assumes that each commodity has a single origin and a single destination. The model permits multiple commodities that might represent distinct physical goods, or the same physical goods but with different origins and destinations. The traffic which flows between the nodes can be expressed in terms of an origin-destination flow matrix. Each vehicle of a given mode has a unit load capacity. The reason for considering weight as opposed to volume is the type of emission function (3.1) used, which is a function of load and transportation modes. However, the model proposed in this section can also be applied when volume is considered as a measure of load. The total transportation cost results from a vehicle moving over a link, including variable and fixed costs. The variable cost per weight unit of commodities is the transportation cost, which covers the carrier's fuel costs, crew costs, overhead costs and administration costs. It is assumed that variable cost is constant over time and that it depends on the link and transport mode. The fixed cost per vehicle consists primarily of handling costs of commodities incurred for moving those commodities on and off the vehicles at the nodes. We assume that each vehicle of the same mode of transport incurs the same fixed cost.

Intermodal transfer cost arises from transferring freight from one transportation mode to another in an intermodal yard (e.g., port and rail yard). In our model, it is assumed that transfer cost does not depend on the combinations of nodes involved in the transfer. While this might be seen as a strong assumption, there are two reasons for doing so. First, the literature on intermodal transportation modelling states that the internal handling costs only depend on the load, e.g., Janic (2007) and Winebrake et al. (2008b) The only case that different combinations of modes for a transfer result in different values is external costs (of emissions) although they do not vary significantly (Winebrake et al., 2008b). The second reason is that an explicit consideration of a combination of modes in a transfer will require a different model, possibly with more variables, to represent the possible combinations. However, the number of such combinations might be large. For example, in instance with $|M| = 4$, there are six possible combinations, whereas if the number of modes increases to 10, there could be up to 45 different combinations. The weight of transferred commodities at a node, which it not the commodities' origins or their destinations, equals the total weight of commodities, which are transported to this node, minus the total weight of commodities, which are transported out of the same node. The intermodal transfer cost at a node equals the absolute value of the transferred commodities' weight times the unit transfer cost. For each of the commodities, there is no transfer cost incurred at its origin and destination. Given the figures obtained from the literature, and to simplify the representation, the unit handling cost at terminals is assumed to be fixed. An alternative model which distinguishes between transfer costs would be worth exploring in future research.

We present three examples in Figures 3.1–3.3 showing how transfer occurs at an intermodal node. In Figure 3.1, 20 units of a commodity are transported by truck at first. When they reach an intermodal node, the transportation mode is changed to ship, so one transfer occurs. In Figure 3.2, 50 units of the commodity are transported to the intermodal node by truck and they are split into two parts, 20 units are transported by ship and 30 units by rail. Two transfers take place. In Figure 3.3, the commodity is brought into the terminal by ship and truck. One transfer occurs at the node. The transportation mode for the 20 units is then changed to truck.

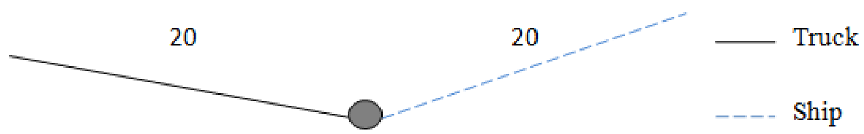


Fig. 3.1 A single transfer from one mode to another

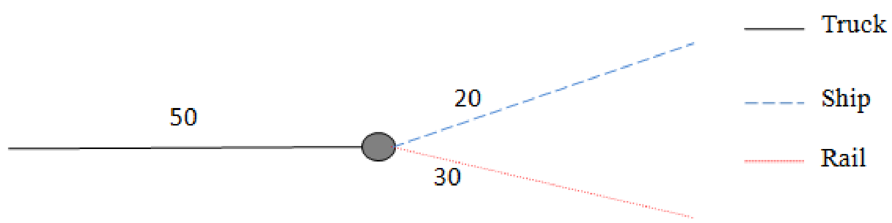


Fig. 3.2 Two transfers from one mode to two others

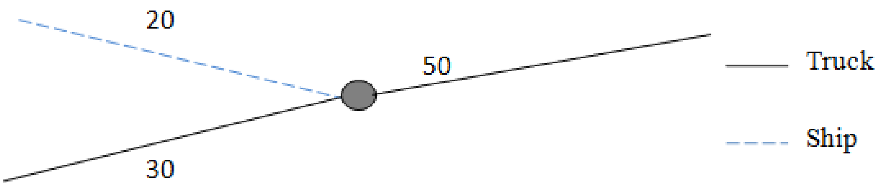


Fig. 3.3 A single transfer from two modes to one

3.2.3 Mathematical Modelling

To formulate the green service network design problem described in the previous section, the notation is defined as shown in Table 3.1.

Using the notation in Table 3.1, a mathematical model for the problem can be written as follows:

$$\text{Minimise} \sum_{k\in K}\sum_{(i,j)\in A}\sum_{m\in M} c_{ij}^{m} x_{ij}^{km} \tag{3.2}$$

$$+\sum_{(i,j)\in A}\sum_{m\in M} f_{ij}^{m} y_{ij}^{m} \tag{3.3}$$

$$+\sum_{k\in K}\sum_{(i,j)\in A}\sum_{m\in M} d_{ij}^{m} p^{m} x_{ij}^{km} \tag{3.4}$$

$$+\frac{1}{2}w\sum_{i\in N}\sum_{k\in K}(\sum_{m\in M}\left|\sum_{j\in N_j^+} x_{ij}^{km} - \sum_{j\in N_j^-} x_{ji}^{km}\right| - h_i^k) \tag{3.5}$$

subject to

$$\sum_{j\in N_j^+}\sum_{m\in M} x_{ij}^{km} - \sum_{j\in N_j^-}\sum_{m\in M} x_{ji}^{km} = b_i^k \qquad \forall i\in N, \forall k\in K \tag{3.6}$$

$$\sum_{k\in K} x_{ij}^{km} \le u_{ij}^{m} y_{ij}^{m} \qquad \forall (i,j)\in A, \forall m\in M \tag{3.7}$$

$$x_{ij}^{km} \ge 0 \qquad \forall (i,j)\in A, \forall k\in K, \forall m\in M \tag{3.8}$$

$$y_{ij}^{m} \in \{0,1,2...\} \qquad \forall (i,j)\in A, \quad \forall m\in M. \tag{3.9}$$

Table 3.1 A summary of mathematical notation

Index	Description
i	for nodes, where $i \in N$
j	for nodes, where $j \in N$, $j \neq i$
m	for modes, where $m \in M$
k	for commodities, where $k \in K$
Parameters	**Explanation**
A	set of arcs
N	set of nodes
$G = (N, A)$	transportation network
K	set of commodities
M	set of modes
$N_i^+ = \{j \in N : (i, j) \in A\}$	The set of outward neighbours for each node
$N_i^- = \{j \in N : (j, i) \in A\}$	The set of inward neighbours for each node
b_i^k	The difference amount for commodity $k \in K$ between coming out of and into node $i \in N$
c_{ij}^m	Unit variable cost on arc $(i, j) \in A$ by mode $m \in M$
$d(k)$	The destination of commodity $k \in K$
d_{ij}^m	Distance of arc $(i, j) \in A$ by mode $m \in M$
f_{ij}^m	Unit fixed cost on arc $(i, j) \in A$ by mode $m \in M$
h_i^k	The absolute value of b_i^k
l_{ij}^{km}	Function of minimum flow of commodity $k \in K$ on arc $(i, j) \in A$ by mode $m \in M$
$o(k)$	The origin of commodity $k \in K$
p^m	Unit emission cost for mode $m \in M$
r^k	The quantity of commodity $k \in K$ that is to be sent from $o(k)$ to $d(k)$
u_{ij}^m	The maximum capacity for arc $(i, j) \in A$ by mode $m \in M$
w	The unit transfer cost for commodities
Decision Variables	**Explanation**
x_{ij}^{km}	Flow variable for commodity $k \in K$ on arc $(i, j) \in A$ by mode $m \in M$
y_{ij}^m	Number of vehicles transported on arc $(i, j) \in A$ by mode $m \in M$

In this model,

$$b_i^k = \begin{cases} r^k, \ i = o(k) \\ -r^k, \ i = d(k) \\ 0, \ \text{otherwise}, \end{cases} \tag{3.10}$$

and

$$h_i^k = \begin{cases} r^k, \ i = o(k) \text{ or } i = d(k) \\ 0, \ \text{otherwise}. \end{cases} \tag{3.11}$$

The model (3.2)–(3.11) presented above is a non-linear, multicommodity multimodal service network design formulation. The objective function measures the total transportation costs. Components (3.2)–(3.5) are variable cost, fixed cost, the emission cost and the intermodal transfer cost, respectively.

A bundle of flow conservation constraints is shown by Equation (3.6), which also expresses the demand requirements. In this case, each commodity has only one origin and one destination. The constraint set (3.7) introduces the capacity constraints. The constraint set (3.8) guarantees that the total flow on arc (i, j) using mode $m \in M$ must not exceed the product of the capacity of each vehicle and the number of vehicles used by mode $m \in M$. If the arc (i, j) is not chosen in the shipping network or the mode $m \in M$ is not used on arc (i, j), the flow on arc (i, j) has to be 0 (i.e., $y_{ij}^m = 0$). Constraint sets (3.8) and (3.9) are to make sure the decision variables x_{ij}^{km}, for the flow of each commodity $k \in K$ and y_{ij}^m for the number of vehicles using mode $m \in M$ on each link, are non-negative and integer, respectively.

The model is an extension of the well-known capacitated multicommodity network design problem (Crainic, 2000), which is challenging to solve. The non-linear nature of the model due to the objective function makes it even more difficult. It is beyond the scope of this chapter to present a bespoke optimisation algorithm for this model. However, we will make use of standard linearisation methods in the literature to convert it into a linear model. This is shown in the next section.

The second part of the objective function shown by component (3.5) is non-linear, due to the absolute value used to model transfers. To linearise, a variable z_i^{km} is used and defined for each $i \in N$, $k \in K$ and $m \in M$. More specifically, z_i^{km} shows the transferred amount of commodity $k \in K$ by mode $m \in M$ at node $i \in N$ if there is a transfer at this node.

Proposition

The component $\left|\sum_{j\in N_j^+} x_{ij}^{km} - \sum_{j\in N_j^-} x_{ji}^{km}\right|$ for $\forall i \in N, \forall k \in K, \forall m \in M$ can be linearised using the following constraints:

$$\sum_{j\in N_j^+} x_{ij}^{km} - \sum_{j\in N_j^-} x_{ji}^{km} \leq z_i^{km}, \forall i \in N, \forall k \in K, \forall m \in M, \tag{3.12}$$

$$\sum_{j\in N_j^-} x_{ji}^{km} - \sum_{j\in N_j^+} x_{ij}^{km} \leq z_i^{km}, \forall i \in N, \forall k \in K, \forall m \in M, \tag{3.13}$$

where $z_i^{km} = \left|\sum_{j\in N_j^+} x_{ij}^{km} - \sum_{j\in N_j^-} x_{ji}^{km}\right|$.

Proof

First, notice that $\sum_{j\in N_j^+} x_{ij}^{km} - \sum_{j\in N_j^-} x_{ji}^{km} = -(\sum_{j\in N_j^-} x_{ji}^{km} - \sum_{j\in N_j^+} x_{ij}^{km})$. Namely, if $\sum_{j\in N_j^+} x_{ij}^{km} - \sum_{j\in N_j^-} x_{ji}^{km} \geq 0$, then $\sum_{j\in N_j^-} x_{ji}^{km} - \sum_{j\in N_j^+} x_{ij}^{km} \leq 0$. If $\sum_{j\in N_j^+} x_{ij}^{km} - \sum_{j\in N_j^-} x_{ji}^{km} < 0$, then $\sum_{j\in N_j^-} x_{ji}^{km} - \sum_{j\in N_j^+} x_{ij}^{km} > 0$.

Now let us consider two cases. If $\sum_{j\in N_j^+} x_{ij}^{km} - \sum_{j\in N_j^-} x_{ji}^{km} \geq 0$, then constraint (3.12) and the minimising objective function (3.2)–(3.5) will guarantee that $z_i^{km} = \sum_{j\in N_j^+} x_{ij}^{km} - \sum_{j\in N_j^-} x_{ji}^{km}$. In the other case where $\sum_{j\in N_j^+} x_{ij}^{km} - \sum_{j\in N_j^-} x_{ji}^{km} < 0$, then constraint (3.13) along with the minimising objective function (3.2)–(3.5) will guarantee $z_i^{km} = \sum_{j\in N_j^-} x_{ji}^{km} - \sum_{j\in N_j^+} x_{ij}^{km}$.

With the new variable z_i^{km}, the formulation can be rewritten as:

$$\text{Minimise} \sum_{k\in K} \sum_{(i,j)\in A} \sum_{m\in M} c_{ij}^m x_{ij}^{km} \tag{3.14}$$

$$+ \sum_{(i,j)\in A} \sum_{m\in M} f_{ij}^m y_{ij}^m \tag{3.15}$$

$$+ \sum_{k\in K} \sum_{(i,j)\in A} \sum_{m\in M} d_{ij}^m p_m x_{ij}^{km} \tag{3.16}$$

$$+ \frac{1}{2} w \sum_{i\in N} \sum_{k\in K} (\sum_{m\in M} z_i^{km} - h_i^k) \tag{3.17}$$

subject to (3.6)–(3.9), (3.12) and (3.13).

The resulting model is now a linear mixed integer program, which can be solved using available optimisation software. In Section 3.4, we present results of computational experiments using this linearised formulation in a case example on data collected from the UK.

3.3 Time-dependent Design

Time-dependent service network design, also called service network design with schedule, is where the time dimension is explicitly taken into consideration and concerns the scheduling of services, including decisions such as when services of different transport modes depart and arrive from origins and reach their destinations, respectively; and how long the commodities are held at intermodal terminals as well as those that are considered by the time-invariant model. To address the schedule of the services, a time-space network will be constructed in the following section.

3.3.1 Problem Description

The model is defined over a given planning horizon, typically one day or one week long, and divided into time periods T; these might be half an hour, one or several hours, or even days. The time-space network consists of copies of nodes representing intermodal terminals and arcs representing connections between these nodes, and is derived from the network $G = (N, A)$ defined in Section 3.2.2. The network is defined by a directed graph $\hat{G} = (\hat{N}, \hat{A})$, where $\hat{N} = \cup_{i \in S} N_i$ is the node set and $N_i = \{i_1, i_2, ..., i_T\}$ represent copies of a super-node N_i for each time period. Each super-node N_i corresponds to a node in the original node set N. The set $\hat{A}$ is the union of all time-indexed arc sets. Arcs between different time representations of the same physical node represent the holding of flow or vehicles, which is expressed as A_w. Arcs in the time-space network between different physical nodes represent movements of vehicles and flow, which is the set A_m. The unit inventory cost at an intermodal terminal for commodities is represented as q.

An illustrative example for the time-space network is shown in Figure 3.4 for T=6. In this figure, the solid lines represent moving services. Black and grey arcs represent services by truck and rail, respectively. In this representation, there is one service from node 1 to node 3 by truck and one service from node 3 to node 2 by rail. The dotted arcs indicate holding services, which may take place at the same in which commodities are waiting to be shipped.

There are M transportation modes and K commodities in the intermodal system. Each commodity has its origin and destination, and specific departure and arrival time within the planning horizon. Figure 3.5 is an example of how commodities are transported in the time-space network.

In Figure 3.5, commodity k departs from node 1 at $t = 2$ and at first is shipped to node 3 by truck. At $t = 3$, commodity k arrives at node 3, where it is held for 1

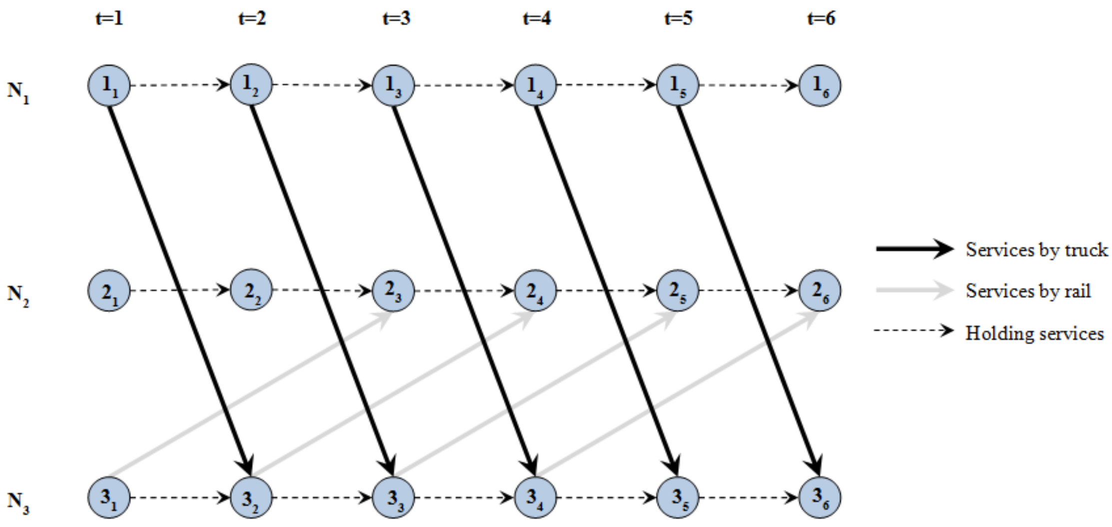


Fig. 3.4 A time-space network with three modes, six time periods and five services

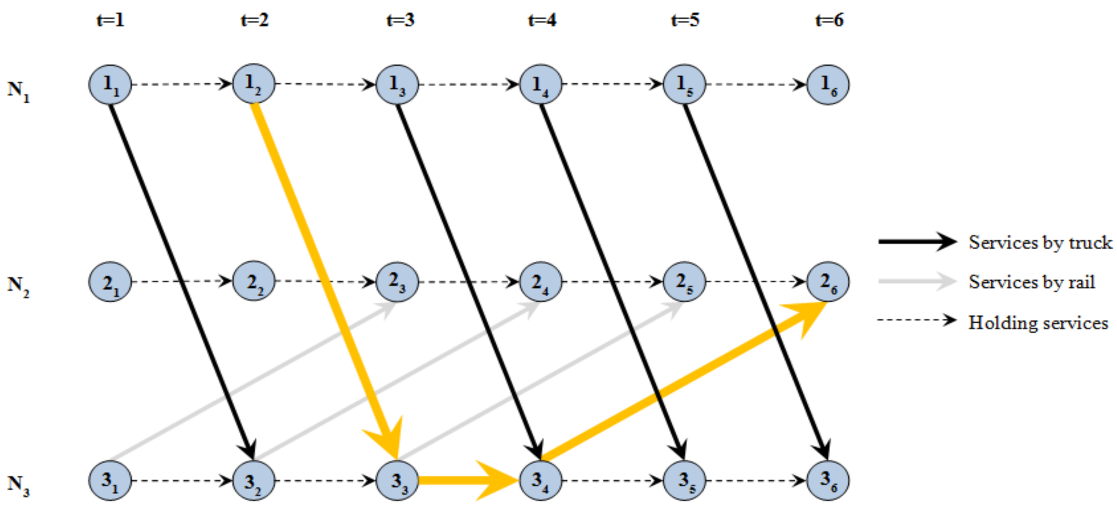


Fig. 3.5 An example of the feasible time-space service plan

time period and then shipped by rail. Commodity k arrives at its destination, node 2, at the due time $t = 6$.

The variable cost per unit weight for a commodity includes the carrier's fuel, crew, overhead and administration costs. It is assumed that variable transportation cost is constant over time and only depends on the link. The fixed cost primarily consists of operators' wages and the handling cost incurred for moving commodities on and off the vehicles. It is assumed that each vehicle of the same mode of transport incurs the same fixed cost. Intermodal transfer cost arises from transferring freight from one transportation mode to another in an intermodal terminal (e.g., port and rail yard). The emissions cost is the cost of the GHG emissions per tonne-km. The inventory cost is that for holding commodities at a node of the network, which is a function of the amount of the commodities and the length of time.

3.3.2 Mathematical Modelling

This section presents a mathematical model to design an intermodal freight transportation system with transfer and emission costs, over a time-space network. This model is based on the multimode multicommodity service network design problem presented in Section 3.2. In this model, the problem is to select: (i) the selection of routes, transport time and transport modes and (ii) the flow distribution of the routes and different modes. The objective is to minimise the total cost, which includes those relevant to variable, fixed, GHG emissions, transfer and inventory costs.

The model uses the same parameters and variables as shown in Section 3.2.3. However, time does not need to be explicitly included in this model given that it is implicitly embedded in the time-space graph construction. A mathematical model for the problem can be written as follows:

$$\text{Minimise} \sum_{k \in K} \sum_{(i,j) \in A_m} \sum_{m \in M} c_{ij}^m x_{ij}^{km} \tag{3.18}$$

$$+ \sum_{(i,j) \in A_m} \sum_{m \in M} f_{ij}^m y_{ij}^m \tag{3.19}$$

$$+ \sum_{k \in K} \sum_{(i,j) \in A_m} \sum_{m \in M} d_{ij}^m p_m x_{ij}^{km} \tag{3.20}$$

$$+ \frac{1}{2} w \sum_{i \in \hat{N}} \sum_{k \in K} \Big(\sum_{m \in M} z_i^{km} - h_i^k \Big) \tag{3.21}$$

$$+ q \sum_{k \in K} \sum_{(i,j) \in A_w} \sum_{m \in M} x_{ij}^{km} \tag{3.22}$$

subject to

$$\sum_{j \in \hat{N}_j^+} \sum_{m \in M} x_{ij}^{km} - \sum_{j \in \hat{N}_j^-} \sum_{m \in M} x_{ji}^{km} = b_i^k \qquad \forall i \in \hat{N}, \forall k \in K \tag{3.23}$$

$$\sum_{j\in\hat{N}_j^+} x_{ij}^{km} - \sum_{j\in\hat{N}_j^-} x_{ji}^{km} \leq z_i^{km},\ \forall i \in \hat{N}, \forall k \in K, \forall m \in M, \tag{3.24}$$

$$\sum_{j\in\hat{N}_j^-} x_{ji}^{km} - \sum_{j\in\hat{N}_j^+} x_{ij}^{km} \leq z_i^{km},\ \forall i \in \hat{N}, \forall k \in K, \forall m \in M, \tag{3.25}$$

$$\sum_{k\in K} x_{ij}^{km} \leq u_{ij}^m y_{ij}^m,\ \forall (i,j) \in A_m, \forall m \in M \tag{3.26}$$

$$x_{ij}^{km} \geq 0, \forall (i,j) \in \hat{A},\ \forall k \in K, \forall m \in M \tag{3.27}$$

$$y_{ij}^m \in \{0, 1, 2...\}\,,\ \forall (i,j) \in A_m, \forall m \in M. \tag{3.28}$$

In this model,

$$b_i^k = \begin{cases} r^k, & i = o(k) \\ -r^k, & i = d(k) \\ 0, & \text{otherwise,} \end{cases} \tag{3.29}$$

and

$$h_i^k = \begin{cases} r^k, & i = o(k) \text{ or } i = d(k) \\ 0, & \text{otherwise.} \end{cases} \tag{3.30}$$

The model (3.18)–(3.30) presented above is a linear, multimode multicommodity time-space network design formulation. The objective function measures the total transportation cost. Components (3.18)–(3.22) capture the variable costs, fixed costs, GHG emission cost, intermodal transfer cost and inventory cost, respectively. A bundle of flow conservation constraints is shown by Equation (3.23), which also expresses the demand requirements. In this case, each commodity has its own origin and destination, representing the departure and due times, respectively. Constraint sets (3.24) and (3.25) are the linearised constraints of our model. Details about the linearisation can be found in Section 3.2.3. The constraint set (3.26) introduces the capacity restrictions, which indicate that the total flow on arc (i, j) using mode $m \in M$ must not exceed the product of the capacity of each vehicle and the number of vehicles using mode $m \in M$. If the arc (i, j) is not chosen in the shipping network or the mode $m \in M$ is not used on arc (i, j) (i.e., $y_{ij}^m = 0$), the flow on arc (i, j) has to be 0. Constraint sets (3.27) and (3.28) model the situation that the decision variables x_{ij}^{km} are non-negative and y_{ij}^m are integers.

3.4 Computational Experiments

3.4.1 Description of the Case Study

Time-invariant Case Study

To demonstrate the application and guide the further development of our intermodal model, this section presents a hypothetical but realistic UK intermodal transportation case example. The network consists of 11 nodes, of which nine are important ports in the UK, namely Edinburgh, Newcastle, Liverpool, Milford Haven, Bristol, Felixstowe, London, Folkestone and Southampton. Two of the nodes are important inland cities, Birmingham and Manchester. A geographical representation of the nodes is shown in Figure 3.6. There are three transportation modes assumed to be running in the network: truck, rail and ship. For Birmingham and Manchester, only rail and truck are available, while for the other nodes all the transportation modes are available. The service network design case example therefore comprises 292 possible directed arcs between 11 nodes. The actual route distance will be considered as opposed to the straight-line distance to make the computational results more realistic. For road and rail journeys, the distance is provided by Travel Footprint Limited (travelfootprint.org); road distance comes from Google Maps and will apply to all road vehicles while for rail, the distance is calculated according to main rail line distances using the most common rail interchanges, see Lane (2006). For sea journeys, distances between pairs of ports are provided by sea-distance.com. There are 30 different commodities to be transported in the network. We randomly generate the origin, destination and demand requirements for each commodity, which is shown in Table 3.2. The data for capacity, variable cost, transfer cost and CO_2 emissions factor of each mode is taken from the literature (see Department for Transport (2005), Andersson et al. (2011), Faulkner (2004), Winebrake et al. (2008a) and McKinnon and Piecyk (2010)) and is summarised in Table 3.3. As can be seen from this table, the capacities, variable cost and emissions tend to vary from one mode to another, although the transfer cost is in this instance the same across all modes (Winebrake et al., 2008b). As for the unit cost of emissions, we use the aforementioned value of £71.60 per tonne (The World Bank, 2012).

Since there are three modes used in our case example, there are three corresponding fixed costs. Estimates of fixed cost in the literature vary significantly. Therefore, we decide to look at a range of scenarios and investigate how these scenarios influence the solutions. Fixed cost is one possible means by which government policy

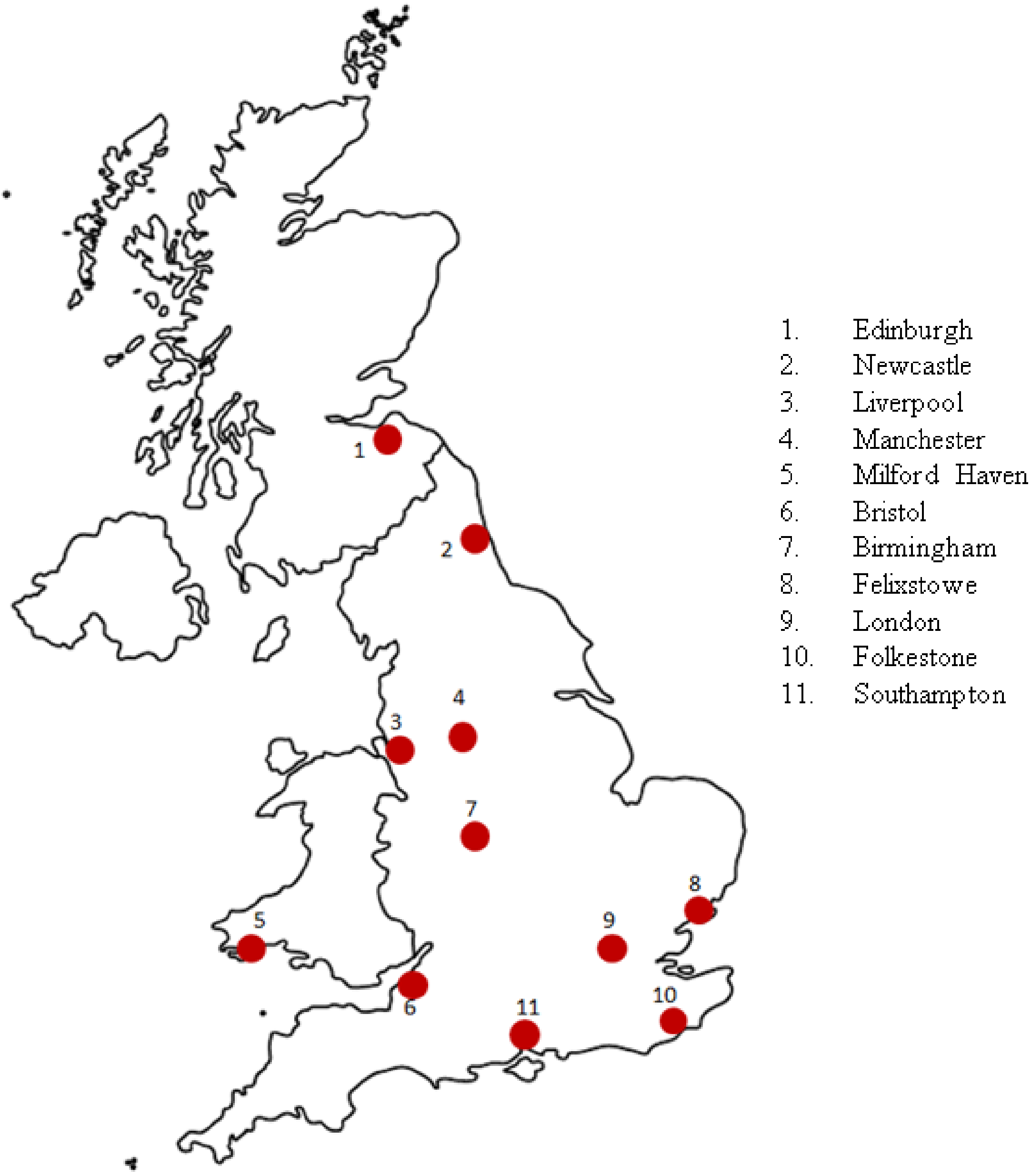


Fig. 3.6 11 UK cities and ports in the network

Table 3.2 Origins, destinations and demands for 30 commodities

Instance	Origin node	Destination node	Required demand (tonne)
1	5	3	212
2	11	9	1182
3	2	11	794
4	5	4	128
5	2	9	182
6	4	6	99
7	7	3	150
8	7	1	343
9	4	9	168
10	3	8	567
11	9	10	960
12	1	3	240
13	2	4	790
14	6	7	83
15	6	8	570
16	6	10	150
17	4	2	1220
18	10	8	110
19	8	6	350
20	9	6	410
21	3	2	300
22	5	2	130
23	11	6	225
24	5	7	850
25	6	7	500
26	2	1	275
27	8	4	150
28	10	11	800
29	5	3	175
30	10	9	87

Table 3.3 Parameters used in the case study

Transportation mode	Capacity (tonne)	Variable cost (£ per ton-mile)	Transfer cost (£ per tonne)	CO_2 (g/tonne-km)
Truck	29	0.036	1.391	62
Rail	397	0.0425	1.391	22
Ship	2970	0.025	1.391	16

can influence the service network design. We denote fixed cost by f_t, f_r and f_s for truck, rail and ship. The relationships between them can be written as:

$$f_r = \alpha f_t, \tag{3.31}$$

$$f_s = \beta f_t, \tag{3.32}$$

where α and β are positive integers.

To see the impact of fixed cost at the design of a service network and send the correct economic signals, a number of scenarios are tested by changing the values of α, β and f_t in our case example. When the value of f_t is fixed, the value of α and β are combinations of 1, 3 and 5. So, for each f_t, there are nine different combinations. It is assumed the fixed costs for truck are £50, £100 and £150, respectively. The number of instances and corresponding fixed costs given different f_r and f_s are shown in Table 3.4. In total, there are 27 combinations.

Table 3.4 Fixed costs(£) for instances with different α and β

Instance	α	β	$f_t = 50$		$f_t = 100$		$f_t = 150$	
			f_r	f_s	f_r	f_s	f_r	f_s
1	1	1	50	50	100	100	150	150
2	1	3	50	150	100	300	150	450
3	1	5	50	250	100	500	150	750
4	3	1	150	50	300	100	450	150
5	3	3	150	150	300	300	450	450
6	3	5	150	250	300	500	450	750
7	5	1	250	50	500	100	750	150
8	5	3	250	150	500	300	750	450
9	5	5	250	250	500	500	750	750

Time-dependent Case Study

The multi-mode, multicommodity time-space network design for intermodal freight transportation in this work is an extension of the model developed in Section 3.3.2. To analyse the application and guide the further development of the time-space model, the same case study is used as described in Section 3.4.1, where the network consists of 11 nodes, shown in Figure 3.6. The time-space model is defined over a 14-hour planning horizon, which is divided into 14 periods. The reason for setting the planning horizon to 14 hours is that all commodities can be shipped from their origin to destination within this period. If the planning horizon is shorter, it is possible that no feasible solutions are found. The resulting time-space network representation for the 11 nodes is shown in Figure 3.7.

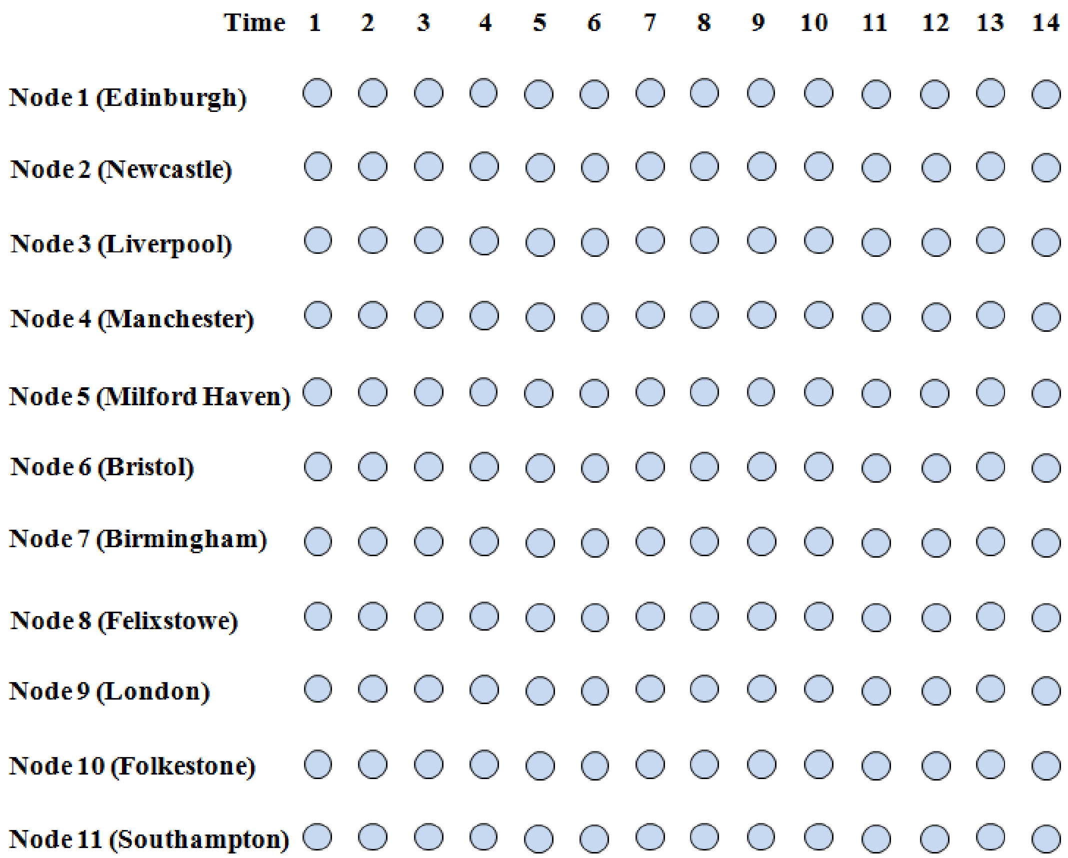


Fig. 3.7 11 nodes in the time-space network in the UK case study

Due to a lack of data on ships, we assume that there are two transportation modes running in a network in Figure 3.7, namely truck and rail. We only consider two transportation modes in building this example, but the model is generally enough to incorporate other modes of transportation if the relevant data is available. For all nodes, both modes are available. All distance data is the same as described for the time-invariant model. Additionally, the average speeds used for truck and rail are those suggested by the Department for Transport (2012) and Forkenbrock (2001), which are 54 mph and 45 mph, respectively. The average traveling times between any two nodes by truck and rail are shown in Tables 3.5 and 3.6, respectively. This is the discretisation of the time line, and it is assumed that the discretisation is fine enough to accommodate the minimum and maximum travel times between any

Table 3.5 The travel times (in hours) between any two super-nodes by truck

Node	1	2	3	4	5	6	7	8	9	10	11
1	0	2	4	4	7	7	5	8	8	9	8
2	2	0	3	3	6	5	4	6	5	6	6
3	4	3	0	1	3	3	2	5	4	5	4
4	4	3	1	0	4	3	2	5	4	5	4
5	7	6	3	4	0	3	4	6	6	6	4
6	7	5	3	3	3	0	2	4	3	4	1
7	5	4	2	2	4	2	0	3	3	4	2
8	8	6	5	5	6	4	3	0	1	2	3
9	8	5	4	4	6	3	3	1	0	1	2
10	9	6	5	5	6	4	4	2	1	0	3
11	8	6	4	4	4	1	2	3	2	3	0

Table 3.6 The travel times (in hours) between any two super-nodes by rail

Node	1	2	3	4	5	6	7	8	9	10	11
1	0	3	5	6	10	9	7	9	9	11	10
2	3	0	4	3	9	7	5	6	6	8	8
3	5	4	0	1	5	4	2	6	5	6	5
4	6	3	1	0	5	4	2	5	4	6	5
5	10	9	5	5	0	3	5	8	6	8	5
6	9	7	4	4	3	0	2	5	3	4	2
7	7	5	2	2	5	2	0	4	3	4	3
8	9	6	6	5	8	5	4	0	2	3	4
9	9	6	5	4	6	3	3	2	0	2	2
10	11	8	6	6	8	4	4	3	2	0	3
11	10	8	5	5	5	2	3	4	2	3	0

pairs of nodes. This idea has also been used in Andersen et al. (2009) and Bauer et al. (2009).

It is assumed that there are hourly services by truck between every pair of super-nodes. As for the rail service data, we use those found in Network Rail (2007). In total, there are 1979 available arcs, including moving arcs and holding arcs. In the resulting network, a small part of which is given in Figure 3.8, shown with two super-nodes and a 5-hour time period. According to this network, it takes three hours by truck to travel from Newcastle to Liverpool, while it is four hours by rail. Since the truck services are hourly, there are two available arcs from node 2 to node 3, which are from time $t = 1$ to $t = 4$ and from $t = 2$ to $t = 5$. Only one rail service is available from node 2 to node 3, and similarly from node 3 to node 2. The dotted arcs represent the idle waiting of commodities at the node, from which these arcs originate.

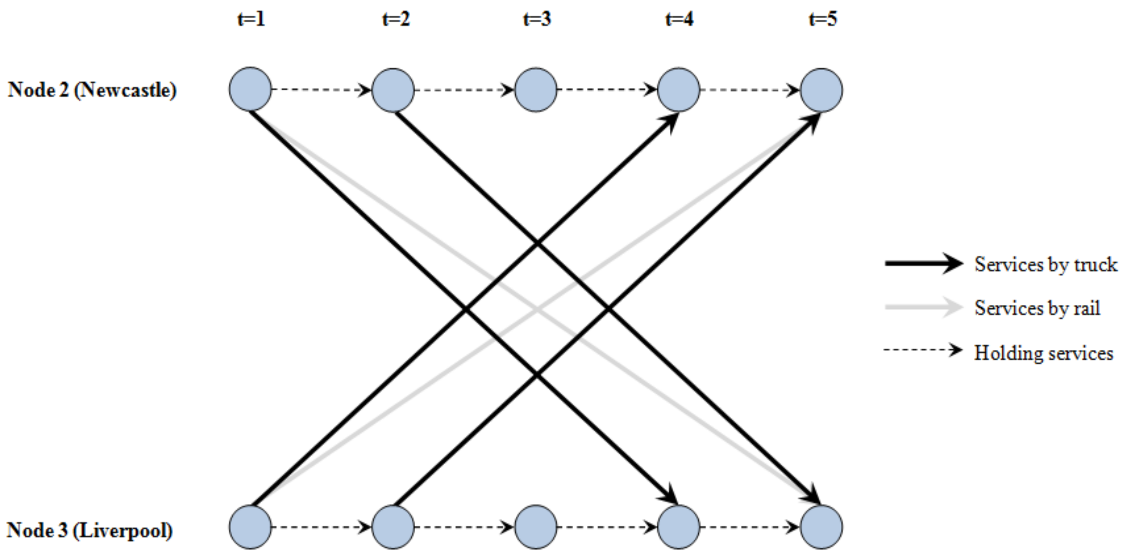


Fig. 3.8 A part of the time-space network in the case study

The data used for transport capacity, unit variable cost, unit transfer cost and CO_2 emissions for each mode, and which were provided in Table 3.3, are from Department for Transport (2005), Andersson et al. (2011), Faulkner (2004), Winebrake et al. (2008a) and McKinnon and Piecyk (2010). As for the fixed costs, it is assumed that they are £50 and £100 for truck and rail, respectively. As for the unit inventory cost, we use £0.107 per tonne per hour (Choong et al., 2002). We randomly generate five sets of data. In each computational experiment, there are 30 commodities with different origins, destinations, departure times and due times, which are shown in detail in Tables B.1–B.5 in Appendix B.

3.4.2 Results and Analysis for the Time-invariant Design

Based on the data set introduced in Section 3.4.1, the computational testing for our model is performed using CPLEX Interactive Optimizer 12.4.0.0 on a Lenovo ThinkPad T410 laptop computer with Intel Core i5 CPU and 4G RAM. In each instance, the resulting integer linear programming formulation has 8760 continuous and 292 integers. The computational time required to solve the integer model to optimality is 1.89 seconds for each instance.

The Effect of Emission and Transfer Costs

In this section, the results of nine experiments are provided when f_t=£50 to show the effect of incorporating transfer and emission costs into the model. For this purpose, the model is first solved without the emission cost Equation (3.16) and transfer cost Equation (3.17), and this is denoted by F. The total cost generated by F is the total operational transportation cost, including variable and fixed costs. Using the solutions obtained, we calculate the resulting emission cost G and transfer cost T, as a consequence of solving F. We then solve the model with objective Equation (3.16) denoted by F(G), with objective Equation (3.17) denoted by F(T) and with both, denoted by F(G+T). Finally, the percentage savings obtained by F(G), F(T) and F(G+T) over the solutions provided by F+G, F+T and F+G+T are calculated, respectively. The results of these experiments are represented in Tables 3.7–3.9, along with the averages calculated across the nine instances.

Table 3.7 Computational results of models with and without emission cost

	F+G		F(G)		Savings	
Instance	Operational cost (£)	Emission cost (£)	Operational cost (£)	Emission cost (£)	Emission cost (%)	Total cost (%)
1	77849	6545	78458	5015	23.38	1.09
2	78949	6545	79558	4991	23.74	1.11
3	79949	6545	80558	4991	23.74	1.09
4	80098	6518	81053	5148	21.02	0.48
5	81235	6744	82190	5158	23.52	0.72
6	82235	6845	83190	5168	24.50	0.81
7	82020	7167	82581	6016	16.06	0.66
8	83134	7076	83696	6016	14.98	0.55
9	84134	7076	84696	6016	14.98	0.55
Average	81067	6785	81776	5391	20.66	0.78

As shown in Table 3.7, F generated at least £6545 in emission costs in each instance. Considering GHG emissions as a part of the objective function, the emission cost decreases significantly, with an average saving of 20.66%. The total cost saved using our model is between 0.48% and 1.11%, which is around £4300–£8000.

It can be seen in Table 3.8 that F generates around £3400 in transfer costs in each instance. In stark contrast, F(T) yields an average saving of 51.97%. The significance of the results shown in both tables is that solutions of a similar cost to

F+G and F+T can be obtained with our models, but those are with significantly fewer emission and transfer costs.

Table 3.8 Computational results of models with and without transfer cost

	F+T		F(T)		Savings	
Instance	Operational cost (£)	Transfer cost (£)	Operational cost (£)	Transfer cost (£)	Transfer cost (%)	Total cost (%)
1	77849	3392	78266	2708	20.17	0.33
2	78949	3396	79466	2498	26.44	0.46
3	79949	3392	81516	1308	61.44	0.62
4	80098	3263	81316	1774	45.63	0.33
5	81235	3431	82755	1315	61.67	0.70
6	82235	3264	83655	1315	59.71	0.62
7	82020	3278	83478	1315	59.88	0.59
8	83134	3435	84415	1315	61.72	0.97
9	84134	3452	85315	1315	20.17	0.33
Average	81067	3367	82242	1651	51.97	0.64

Table 3.9 shows the comparison of results between F+G+T and F(G+T). In this case, emission cost in F(G+T) is reduced by between 6.93% and 11.59% compared to that generated by F. Similarly, intermodal transfer cost is also reduced by up to 54.11%. The average savings in the total cost obtained by using F(G+T) is 0.72%. However, it is worth noting that these solutions again exhibit an average saving of 9.98% on emission cost and 24.75% on transfer cost.

Intermodal vs. Unimodal Transportation

In this set of experiments, we seek to compare unimodal with intermodal transportation. Since Birmingham and Manchester are inland, transportation by ship is not available for these nodes. As a result, truck-only and rail-only models are tested for uni-modal transportation. Scenarios are examined for unimodal and intermodal transportation models with f_t=f_r=f_s=£50, f_t=f_r=f_s=£100 and f_t=f_r=f_s=£150, to remove the effects of different fixed costs when comparing the results. Computational results are summarised in Table 3.10. The third column displays the total cost for shipping commodities. The fourth, fifth, sixth and seventh columns display the variable, fixed, emission and intermodal transfer costs, respectively.

When the fixed cost is £50, the total costs for truck-only, rail-only and intermodal transportation are £112513, £98778 and £87253. Intermodal transportation is 22.4% and 11.7% less costly than truck-only and rail-only transportation. When

Table 3.9 Computational results of models with and without emission and transfer costs

	F+G+T			F(G+T)			Savings		
Instance	Operational cost (£)	Emission cost (£)	Transfer cost (£)	Operational cost (£)	Emission cost (£)	Transfer cost (£)	Emission cost (%)	Transfer cost (%)	Total cost (%)
1	77849	6545	3392	78527	5920	2806	9.55	17.28	0.61
2	78949	6545	3396	79727	5900	2596	9.85	23.56	0.75
3	79949	6545	3392	80627	5900	2596	9.85	23.47	0.85
4	80098	6518	3263	80930	6066	2708	6.93	17.01	0.19
5	81235	6744	3431	82217	6075	2498	9.92	27.19	0.68
6	82235	6845	3264	83217	6075	1498	11.25	54.11	1.68
7	82020	7167	3278	82925	6336	2708	11.59	17.39	0.54
8	83134	7076	3435	84039	6336	2708	10.46	21.16	0.60
9	84134	7076	3452	85039	6336	2708	10.46	21.55	0.61
Average	81067	6785	3367	81916	6105	2536	9.98	24.75	0.72

Table 3.10 Comparison of the results of uni-modal and intermodal transportation

	Modal type	Total cost(£)	Variable cost(£)	Fixed cost(£)	Emission cost(£)	Transfer cost(£)
f_t=f_r=f_s=£50	Truck only	112513	75730	21550	15233	0
	Rail only	98778	91326	2100	5352	0
	Intermodal	87253	75327	3200	5920	2806
f_t=f_r=f_s=£100	Truck only	134019	75775	43000	15244	0
	Rail only	100878	91326	4200	5352	0
	Intermodal	89465	77705	3900	5020	2840
f_t=f_r=f_s=£150	Truck only	155519	75775	64500	15244	0
	Rail only	102978	91326	6300	5352	0
	Intermodal	91415	77705	5850	5020	2840

the fixed cost increases to £100 and £150, the total savings that intermodal transportation affords is up to 41.2%.

The variable cost of the three truck-only scenarios changes from £75730 to £75775, which is a 0.06% increase. For rail-only scenarios, the variable cost stays the same. Unlike uni-modal results, the variable cost of intermodal scenarios changes from £75327 to £77705, which is 3.1% increase when the fixed cost increases from £50 to £100. A larger range of change indicates that when the fixed cost is less than £100, it has a greater effect on the variable cost in intermodal transportation than that in uni-modal transportation. When the fixed cost increases from £100 to £150, the variable cost of uni-modal and intermodal models does not change. It is possible that variable cost increases as the fixed cost increases, and it will not be affected by the value of the fixed cost when it reaches £77705.

Since the emission factor of truck transportation is the highest, followed by that of ship and of rail, the emission cost for truck-only transportation is nearly three times as costly as that of rail-only and intermodal transportation. It is noticeable that there is only a slight change in the emission cost in truck-only transportation by adjusting the fixed cost, while for rail-only transportation, the emission cost stays the same. In intermodal transportation, the emission cost decreases from £5920 to £5020, a 15.2% decrease. When the fixed cost keeps increasing, the emission cost does not change. A reduction in the emission cost can be obtained by adjusting the fixed cost when the fixed cost is less than £100.

The transfer cost only occurs in intermodal transportation. It is the smallest part of the total cost. When the fixed cost increases from £50 to £150, the transfer cost increases slightly from £2806 to £2840. More details on this are provided in the next section where intermodal transportation results are discussed.

In summary, even if a transfer cost is incurred in intermodal transportation, the total cost can be reduced by 11.3%–41.2% by using intermodal, as opposed to uni-modal, transportation. By adjusting the fixed cost, the variable and emission costs can be reduced in intermodal transportation, while they rarely change in uni-modal transportation. These results would seem to indicate that intermodal transportation is more flexible than uni-modal transportation.

Intermodal Transportation Results

The results presented in Figures 3.9–3.11 show the total cost, including the variable, fixed, emission and intermodal transfer costs, for 9 intermodal instances when f_t=£50, £100 and £150, respectively. From these figures, it is significant that the variable cost, which is around £75000, takes up the largest proportion of the total cost and it does not change greatly across the 27 instances. The intermodal transfer cost takes up the least proportion. For each f_t, the total fixed cost increases when f_r or f_s increases. Hence, the total cost also increases. More details about the total cost and its components are presented in Table 3.11, which shows costs that are normalised to 1 against base case scenarios of f_t=f_r=f_s, for all 27 instances.

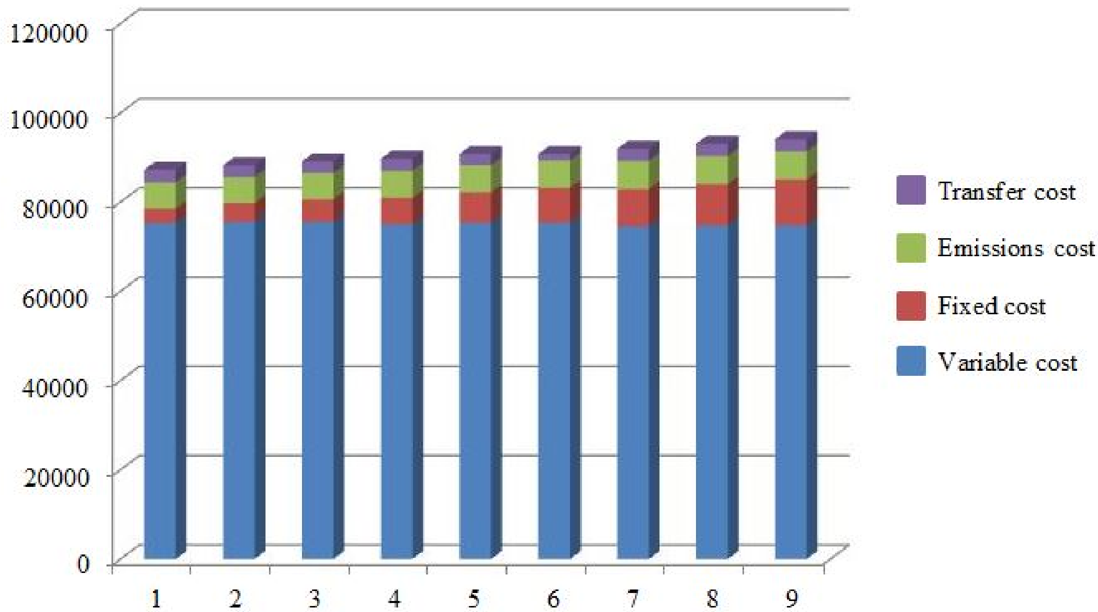


Fig. 3.9 Computational results for nine intermodal instances when f_t=50

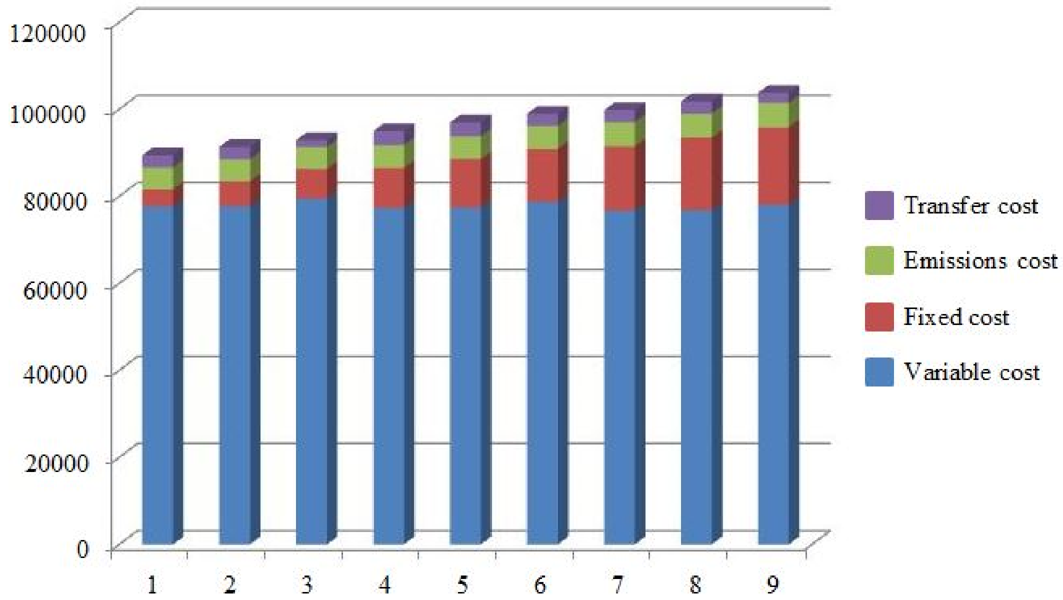


Fig. 3.10 Computational results for nine intermodal instances when f_t=100

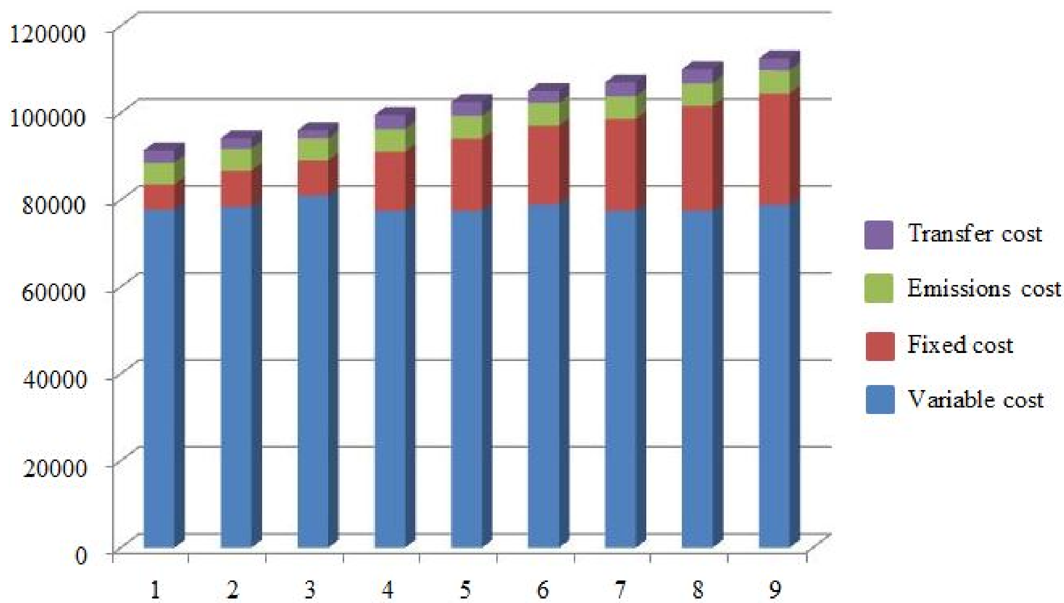


Fig. 3.11 Computational results for nine intermodal instances when f_t=150

Table 3.11 Normalised costs for intermodal instances

f_t	f_r	f_s	Total cost	Variable cost	Fixed cost	Emission cost	Transfer cost
£50	£50	£50	1.000	1.000	1.000	1.000	1.000
		£150	1.011	1.004	1.281	0.997	0.925
		£250	1.021	1.004	1.563	0.997	0.925
	£150	£50	1.028	0.996	1.844	1.025	0.965
		£150	1.041	1.001	2.125	1.026	0.890
		£250	1.041	1.001	2.438	1.026	0.534
	£250	£50	1.054	0.990	2.609	1.070	0.965
		£150	1.067	0.993	2.891	1.070	0.965
		£250	1.078	0.993	3.203	1.070	0.965
£100	£100	£100	1.000	1.000	1.000	1.000	1.000
		£300	1.021	1.000	1.487	0.996	1.000
		£500	1.039	1.023	1.744	1.003	0.559
	£300	£100	1.062	0.995	2.359	1.044	1.129
		£300	1.084	0.997	2.846	1.042	1.129
		£500	1.106	1.013	3.128	1.057	0.963
	£500	£100	1.115	0.987	3.795	1.103	0.982
		£300	1.138	0.988	4.282	1.101	0.982
		£500	1.160	1.005	4.564	1.116	0.816
£150	£150	£150	1.000	1.000	1.000	1.000	1.000
		£450	1.030	1.009	1.436	0.998	0.840
		£750	1.050	1.042	1.385	1.017	0.649
	£450	£150	1.089	0.997	2.333	1.042	1.129
		£450	1.122	0.997	2.846	1.042	1.129
		£750	1.149	1.016	3.103	1.042	0.963
	£750	£150	1.171	0.997	3.615	1.042	1.129
		£450	1.204	0.997	4.128	1.042	1.129
		£750	1.231	1.015	4.385	1.057	0.969

Table 3.11 shows that when f_t=£50, the total cost of the nine instances is normalised to a range of 1.000–1.078. For each f_t, the total fixed cost increases when f_r or f_s is rising. It is interesting that the variable cost stays almost the same when the fixed cost changes. The difference is between 0.1% and 2.3%. In most of the instances, the emission cost slightly increases when f_r or f_s rises. The emission cost increases up to 11.6%. We have mentioned in Section 3.2.1 that the emission cost is measured by the product of distance, the amount of load and emission factors. We also know that the emission factor of truck transportation is the highest, followed by that of ship and of rail. Therefore, when f_r or f_s increases, total truck tonne-miles increase, which results in the total emission cost increasing.

Since transportation by rail and ship have lower emission factors than that by truck, reducing f_r and f_s helps to reduce emission costs in intermodal transportation. Moreover, the transfer cost does not change significantly. From Table 3.10, we already know that the transfer cost is around £2800 when f_t=f_r=f_s. Except for two instances out of the 27, the change of the transfer cost is between 18.4% and 3.1%, which suggests that intermodal transfer is a very important component in an intermodal transportation chain. Consolidation of loads or incentivising certain low emission forms of transportation can be efficiently achieved in intermodal transportation.

Given that there is a fixed cost associated with the use of each vehicle, it is desirable to use the available capacity as much as possible. In other words, it is not very efficient to run empty vehicles or vehicles carrying limited amounts of goods, which will also lead to congestion in available transportation infrastructures. Tables 3.12–3.14 list the total tonnage shipped and the capacity utilisation of three transport modes when f_t=£50, £100 and £150 with the commodities shown in Table 3.2. From Instances #1 to #9 in each table, it is noticeable that more trucks are used and greater total tonnage of commodities are transported by truck when f_r or f_s increases. Except for six instances in total, in which no trucks are used for shipping, the capacity utilisation of truck transportation stays above 91%. In eight out of these 22 instances, the capacity utilisation reaches 100% in truck employment. By adjusting f_t, we could avoid the situation in which there are no trucks used in the transportation network and at the same time, get a relatively high capacity utilisation. In rail transportation, when f_r increases, both total tonnage and the total number of vehicles decreases, while the capacity utilisation increases. One reason for this might be that it is more cost effective to travel further in order to use fewer vehicles when f_r is high. When f_r is low, the employment of more vehicles on direct routes could be more attractive. The capacity utilisation of rail across the 27 instances is greater than 77% for each. In 17 out of the 27 instances, it is greater than 84%. When f_t increases, the total tonnage transported by rail increases. By changing f_t or f_r, we might roughly control the total tonnage of transportation by rail. Since ships have large capacities, the average capacity utilisation of ships is much less than that of truck and rail, with values around 20%. In addition, due to their large capacities, the number of ships used and total tonnage of commodities conveyed by sea do not change as much as those of transportation by truck and rail.

As part of the computational experiments, we have conducted further tests where the variable cost and the capacity of rail are changed and demands are increased. In the case that the variable cost and the capacity of rail are changed to £0.017

Table 3.12 Comparison of the capacity utilisation for three modes when f_t=£50

	Truck			Rail			Ship		
Instance	Total No.	Total tonnage	Capacity utilisation(%)	Total No.	Total tonnage	Capacity utilisation(%)	Total No.	Total tonnage	Capacity utilisation(%)
1	26	741	98.28	27	8265	77.11	11	6306	19.30
2	26	741	98.28	29	9375	81.43	9	5196	19.44
3	26	741	98.28	29	9375	81.43	9	5196	19.44
4	34	939	95.23	24	7666	80.46	12	6323	17.74
5	34	939	95.23	24	8063	84.62	10	6013	20.25
6	34	939	95.23	24	8063	84.62	10	6013	20.25
7	45	1253	96.02	22	7352	84.18	12	6496	18.23
8	45	1253	96.02	22	7352	84.18	10	6410	21.58
9	45	1253	96.02	22	7352	84.18	10	6410	21.58

Table 3.13 Comparison of the capacity utilisation for three modes when f_t=£100

	Truck			Rail			Ship		
Instance	Total No.	Total tonnage	Capacity utilisation(%)	Total No.	Total tonnage	Capacity utilisation(%)	Total No.	Total tonnage	Capacity utilisation(%)
1	0	0	N/A	29	9035	78.48	10	6257	21.07
2	0	0	N/A	31	9748	79.21	9	5544	20.74
3	0	0	N/A	33	10095	77.06	7	4463	21.47
4	6	174	100.00	25	8613	86.78	11	6939	21.24
5	6	174	100.00	25	8613	86.78	10	6766	22.78
6	7	186	91.63	25	8589	86.54	8	6647	27.98
7	17	476	96.55	24	8141	85.44	11	6641	20.33
8	17	476	96.55	24	8141	85.44	10	6468	21.78
9	18	488	93.49	24	8117	85.19	8	6349	26.72

Table 3.14 Comparison of the capacity utilisation for three modes when f_t=£150

	Truck			Rail			Ship		
Instance	Total No.	Total tonnage	Capacity utilisation(%)	Total No.	Total tonnage	Capacity utilisation(%)	Total No.	Total tonnage	Capacity utilisation(%)
1	0	0	N/A	29	9035	78.48	10	6257	21.07
2	0	0	N/A	32	9848	77.52	8	5033	21.18
3	0	0	N/A	34	10772	79.80	4	4078	34.33
4	6	174	100.00	25	8613	86.78	10	6766	22.78
5	6	174	100.00	25	8613	86.78	10	6766	22.78
6	3	87	100.00	26	8688	84.17	8	6647	27.98
7	6	174	100.00	25	8613	86.78	10	6766	22.78
8	6	174	100.00	25	8613	86.78	10	6766	22.78
9	6	174	100.00	25	8613	86.78	8	6671	28.08

per ton-mile and 3360 tonnes (Forkenbrock, 2001), the results with the intermodal model suggest a uni-modal transportation plan with rail being the only mode of transport used, with ship being used very rarely. The capacity utilisation of rail transportation is 18% to 32%. In the event that demands are increased 10-fold, the results do not significantly change, although the capacity utilisation of rail and ship conveyance increases. Rail transportation has a much higher capacity utilisation of more than 96%, with shipping being more used with a capacity utilisation of between 76% and 88%. Computational results for this case can be found in Tables A.1–A.3 in Appendix A.

3.4.3 Results and Analysis for the Time-dependent Design

Based on the data set introduced in Section 3.4.1, the computational testing for the time-dependent model is performed using CPLEX Interactive Optimizer 12.5.0.0 on a Lenovo ThinkPad T410 laptop computer with Intel Core i5 CPU and 4G RAM.

Testing CPLEX Solution Quality

The resulting time-space network using the MIP formulation from Section 3.4.1 is tested with CPLEX 12.5 using the five instances described in Appendix B. Although the instances are of moderate size, CPLEX had difficulties in solving the formulation to provable optimality within two hours for these test sets. To test the effect of the solution time on the resulting optimality gap, we set an absolute tolerance on the gap between the best integer objective and the objective of the best node remaining. When this difference falls below the value of parametre, the MIP optimisation is stopped. Different tolerance limits on the gaps and report the required solution times in Table 3.15 for each instance. The results indicate that when gap tolerance is set equal to 0.7%, CPLEX requires an average solution time of 0.41 seconds. When the tolerance is decreased to 0.5%, all of the instances can also be solved within one second. The minimum gap CPLEX is able to obtain is 0.2% with an average solution time of 489.10 seconds. Since all five instances can be solved to 0.2% optimality within two hours, we use the same tolerance for the remaining set of experiments but allow CPLEX a time limit of 1800 seconds.

Comparisons between the MMND and the MMTND

In the multimode multicommodity network design model (MMND) presented in Section 3.2, the time dimension is not considered. In other words, no departure

Table 3.15 Solution times (in seconds) with different gaps

Gap (%)	Instance #1	Instance #2	Instance #3	Instance #4	Instance #5	Average
0.1	44.34	N/A	N/A	N/A	N/A	N/A
0.2	15.38	937.26	26.6	1450.54	15.74	489.10
0.3	2.61	15.18	3.11	25.83	5.63	10.47
0.4	0.76	2.68	0.75	2.26	0.69	1.43
0.5	0.81	0.94	0.86	0.76	0.72	0.82
0.6	0.41	0.44	0.83	0.83	0.39	0.58
0.7	0.41	0.45	0.41	0.40	0.40	0.41

times and due times are taken into account. It is then interesting to investigate whether or not the total cost and the solutions change when shipping the same set of commodities in the time-space network model (MMTND) and in the MMND. The results provided by the MMND and the MMTND are compared, which are shown in Table 3.16. In particular, the table reports the percentage change in a number of indicators, including the total cost, operational cost, emission cost, number of vehicles (truck and rail), and capacity utilisation (%) in the solutions of the MMTND over the solutions provided by the MMND. The results are shown for five instances, where the last column shows the average. The detailed results can be found in Table C.1 in Appendix C.

Table 3.16 Comparison between the MMND and the MMTND results (%)

	Instance #1	Instance #2	Instance #3	Instance #4	Instance #5	Average
Total cost	4.67	8.45	7.49	12.83	10.01	8.69
Operational cost	4.33	5.84	7.72	11.61	8.69	7.64
Emission cost	8.78	46.41	4.34	30.91	26.28	23.34
No. of vehicles (truck)	50.00	384.44	19.05	334.29	165.38	190.63
No. of vehicles (rail)	-28.13	-40.00	-12.96	-34.62	-42.22	-31.59
Capacity utilisation (truck)	-0.36	0.11	0.59	-1.16	-0.73	-0.31
Capacity utilisation (rail)	-6.55	-5.41	-4.11	2.01	-8.15	-4.44

Table 3.16 shows that all costs increase in the MMTND. In particular, the total cost of the MMTND increases by 8.69% on average in comparison with the MMND, while the average increase in the emission cost is 23.34%. From a company's point

of view, if they want to consider the cost of time in the network system, the relative change rate of the total cost would be an indicator. It is also interesting to note that the emission cost of the MMTND increases significantly in relation to that of the MMND. One possible reason for this is that a greater number of trucks are used when considering the requirements of departure and due time for commodities. In Instance #2, the total number of trucks in the MMTND increases by up to 384.44%. For all five instances, the average increase due to the use of the MMTND is 190.63%, which leads to a higher emission cost. The capacity utilisation of trucks in the MMND and the MMTND does not differ greatly, while rail transportation decreases by 4.44% in the MMTND.

Results of Instances with 10 and 30 Commodities

To see whether our model performs well with a smaller number of commodities, the five instances will be tested by reducing the number of commodities to be transported to 10, which are the first 10 from each instance listed in Tables B.1–B.5. We will then compare the results with those of 30 commodities and report a summary of the results in Table 3.17.

Table 3.17 indicates that the capacity utilisation of vehicles in models with 10 and 30 commodities does not differ significantly. With 10 commodities, the capacity utilisation of truck transportation is around 98%, while for rail it is between 74.55% and 86.15%, indicating that better capacity utilisation is obtained when conveying goods by truck. For 10 commodities, there are no transfers in four out of the five instances, 30 commodities, there are transfers in three out of the five instances. It can be concluded that our model performs well with shipping up to 30 commodities with different origins and destinations.

Lower Bound for Capacity Utilisation

From Table 3.16 presented above, we see that the overall capacity utilisation for truck and rail modes of transport stays above 97% and 73%, respectively. In practice, and in order to increase the capacity utilisation of each vehicle, one may want to impose lower bounds on the amount of commodities carried on each vehicle, or, alternatively, set a minimum capacity utilisation for each vehicle used. This section investigates the effects of such lower bounds on the resulting solutions.

Table 3.17 Computational results for models with 10 and 30 commodities

Instance	1		2		3		4		5	
No. of commodities	10	30	10	30	10	30	10	30	10	30
Total cost (£)	104871	264836	59792	157587	72963	192785	53418	155426	55636	149696
Fixed cost (£)	5000	14450	6000	13600	2550	9700	6950	11000	3200	12950
Variable cost (£)	90932	228966	48545	130090	66416	170287	41304	133045	47945	123839
Emission cost (£)	8939	21139	5247	13676	3997	12798	5165	11380	4410	12826
Transfer cost (£)	0	281	0	221	0	0	0	0	81	81
Total tonnage (truck)	1763	5705	2922	6247	435	2850	3564	4318	1250	5883
Total tonnage (rail)	5623	14700	2796	8366	6156	15236	2048	10479	3248	7797
No. of vehicles (truck)	62	201	102	218	15	100	125	152	44	207
No. of vehicles (rail)	19	46	9	27	18	47	7	34	10	26
Capacity utilisation (truck) (%)	98.05	97.87	98.78	98.81	100.00	98.28	98.32	97.96	97.96	98.00
Capacity utilisation (rail) (%)	74.55	80.50	78.25	78.05	86.15	81.65	73.70	77.63	81.81	75.54

The condition of a minimum capacity utilisation for a transport mode $m \in M$ on an arc $(i, j) \in A_m$ can be expressed as follows,

$$\frac{\sum_{k \in K} x_{ij}^{km}}{y_{ij}^{m}} \geq g, \forall (i, j) \in A_m, \forall m \in M, \tag{3.33}$$

where g is the lower bound of capacity utilisation.

The computational results are compared with different lower bounds of capacity utilisation. Tables 3.18–3.20 show the results obtained on three instances using $g =$ 0, 50% and 70%, respectively.

Table 3.18 Computational result of Instance #1 with different lower bounds of capacity utilisation

	g=0	g=50%	g=70%
Total cost (£)	195292	195377	195437
Operational cost (£)	182311	182173	181578
Emission cost (£)	12981	13204	13859
No. of vehicles (truck)	66	73	94
No. of vehicles (rail)	54	51	46
Capacity utilisation (truck) (%)	96.87	96.50	96.37
Capacity utilisation (rail) (%)	75.71	81.41	86.43

Table 3.19 Computational result of Instance #2 with different lower bounds of capacity utilisation

	g=0	g=50%	g=70%
Total cost (£)	160037	160429	160784
Operational cost (£)	145478	146167	145835
Emission cost (£)	14559	14262	14949
No. of vehicles (truck)	180	172	189
No. of vehicles (rail)	32	35	31
Capacity utilisation (truck) (%)	98.47	98.44	98.21
Capacity utilisation (rail) (%)	76.96	77.34	84.24

Tables 3.18–3.20 provide the relative change of the total cost, the emission cost, the operational cost, the number of vehicles and the average capacity utilisation when the lower bound of the capacity utilisation for each vehicle is increased. When the value of g is increased from 0% to 70%, the change in the total cost rises to £1800, representing a 1.16% increase. The average capacity utilisation of rail continues to increase in all three instances as g is increased from 0% to 70%, while the average capacity utilisation of trucks decreases slightly. The results indicate

Table 3.20 Computational result of Instance #3 with different lower bounds of capacity utilisation

	g=0	g=50%	g=70%
Total cost (£)	157582	158104	159403
Operational cost (£)	146359	145712	146891
Emission cost (£)	11223	12392	12512
No. of vehicles (truck)	94	124	133
No. of vehicles (rail)	41	38	38
Capacity utilisation (truck) (%)	98.39	97.08	97.15
Capacity utilisation (rail) (%)	72.35	77.04	82.29

that imposing an additional constraint on minimum capacity utilisation will cost the operators by about 1.16% more compared with the case in which there are no such constraints.

3.5 Bi-criteria Analysis

The multimode multicommodity network design problem in this chapter has so far been treated as a single-objective optimisation problem. This was made possible by aggregating the two objective functions; one relevant to operational activities, and the other to emissions. In this section, we show how the model can be used to produce non-dominated solutions with regard to the two objectives. This analysis is particularly relevant if costs are not a prevailing factor, or are not readily available, as might be the case for emission costs.

In order to find the trade-offs between minimising the total transportation cost and the amount of CO_2 emissions in the multicommodity multimodal service network design problem, we use the notation shown in Table 3.1 and define the two objective functions below:

$$\text{(OBJ1)} \qquad f_1(x,y,z) = \sum_{k\in K}\sum_{(i,j)\in A}\sum_{m\in M} c_{ij}^m x_{ij}^{km} \tag{3.34}$$

$$+ \sum_{(i,j)\in A}\sum_{m\in M} f_{ij}^m y_{ij}^m \tag{3.35}$$

$$+ \frac{1}{2}w \sum_{i\in N}\sum_{k\in K}(\sum_{m\in M} z_i^{km} - h_i^k) \tag{3.36}$$

$$\text{(OBJ2)} \qquad f_2(x) = \sum_{k\in K}\sum_{(i,j)\in A}\sum_{m\in M} d_{ij}^m p^m x_{ij}^{km}. \tag{3.37}$$

In the above, $f_1(x, y, z)$ measures the total transportation cost, including the variable, fixed and transfer costs, which are detailed in Section 3.2.2. Conversely, $f_2(x)$ measures the total CO_2 emissions for all activities. The bi-objective optimisation problem is formulated as:

$$\text{Minimise} \quad f(x, y, z) = (f_1(x, y, z), f_2(x)) \tag{3.38}$$

$$x \in X, \tag{3.39}$$

$$y \in Y, \tag{3.40}$$

$$z \in Z, \tag{3.41}$$

where X, Y and Z define the feasible regions for variables x, y and z, respectively.

Definition 1 (Pareto-optimality). A point x is said to be a Pareto-optimal or an efficient solution for the MSPP and the CMSPP, if and only if there does not exist any $x^{'} \in X$ such that $f_i(x^{'}) \leq f_i(x)$ for all $i \in \{1, 2, ..., k\}$ where at least one inequality is strict.

Definition 2 (Efficient set). The Pareto-optimal or efficient set $\mathcal{E} = \{x \in X : x \text{ is efficient}\}$.

Definition 3 (Pareto front). The Pareto front $\mathcal{F} = \{f(x) : x \text{ is Pareto-optimal}\}$.

The *Efficient set ($\mathcal{E}$)* is defined on the solution space, and the *Pareto front ($\mathcal{F}$)* is defined on the objective space. In multiobjective optimisation, an *efficient*, or a *Pareto-optimal*, solution is one in which no objective can be improved without worsening at least one other objective (Coello et al., 2007). Decision-makers usually select a particular Pareto-optimal solution based on their preferences on the objectives (Ghoseiri et al., 2004).

According to Hwang and Masud (1979), the methods for solving the multiobjective problems can be classified into four categories depending on the role of the decision maker (DM). No preference methods are methods where the DM is not needed. A priori methods are methods where the DM articulates their preference before optimisation. A criticism of a priori methods is that it is very difficult for the DM to have prior knowledge of the efficient set and to be able to decide their preferences. Interactive methods allow the DM to guide the search by alternating optimisation and preference articulation iteratively. The drawback to the interactive methods is that the DM never-sees the whole efficient set or an approximation of it. A posteriori methods aim to generate a representative set of Pareto-optimal solutions and the DM chooses the best among them. Hence, we will focus on the a posteriori methods. The ϵ-constraint method is a widely used posteriori

Table 3.21 Value of ϵ in the ϵ-constraint method

Iteration No.	Value of ϵ
1	f_2^*
2	$f_2^*+\varsigma$
3	$f_2^*+2\varsigma$
...	...
$q-1$	$f_2^*+(q-2)\varsigma$
q	$f_2(x_1^*)$

method. In it one of the objectives is optimised, subject to the other objective being converted into a constraint by imposing appropriate upper bounds on its value. The ϵ-constraint method can be implemented as follows (Mavrotas, 2009). We first solve the following single-objective optimisation problem:

$$\text{Minimise} \quad f_2(x) \tag{3.42}$$

$$\text{subject to} \quad x \in S_2. \tag{3.43}$$

Let the optimal solution to Equations (3.42)–(3.43) be denoted by x_2^*. The minimised amount of CO_2 emissions is then fixed as $f_2(x_2^*) = \epsilon$. Following this step, we solve following single-objective optimisation model:

$$\text{Minimise} \quad f_1(x, y, z) \tag{3.44}$$

$$\text{subject to} \quad f_2(x) \leq \epsilon, \tag{3.45}$$

$$x \in X, \tag{3.46}$$

$$y \in Y, \tag{3.47}$$

$$z \in Z, \tag{3.48}$$

Let (x_1^*, y_1^*, z_1^*) denote the optimal solution to Equations (3.44)–(3.48). Furthermore, let q be the number of Pareto-optimal solutions that a decision-maker wishes to produce. Then, the value of ϵ is to be increased by ς at every iteration where:

$$\varsigma = (f_2(x_1^*) - f_2(x_2^*))/(q-1). \tag{3.49}$$

The method iterates by increasing the value of ϵ in this way (see Table 3.21), where each iteration generates another solution. By repeatedly relaxing the upper bound on f_2, and resolving f_1 each time, the objectives of solutions can be obtained to construct the image of the Pareto front.

A numerical example will now be presented to illustrate the application of the ϵ-constraint method on the multimode multicommodity network design problem. The method is implemented in C and each sub-problem is solved to optimality by ILOG CPLEX Interactive Optimizer 12.4. Therefore, the solutions generated are efficient. The instances are based on the data presented in Section 3.4.1 with the fixed-cost scenario where f_t=£50, f_r=£150 and f_s=£250.

Assuming q=45, the step-size is calculated as ς=0.7 tonne. The resulting Pareto front is shown in Figure 3.12.

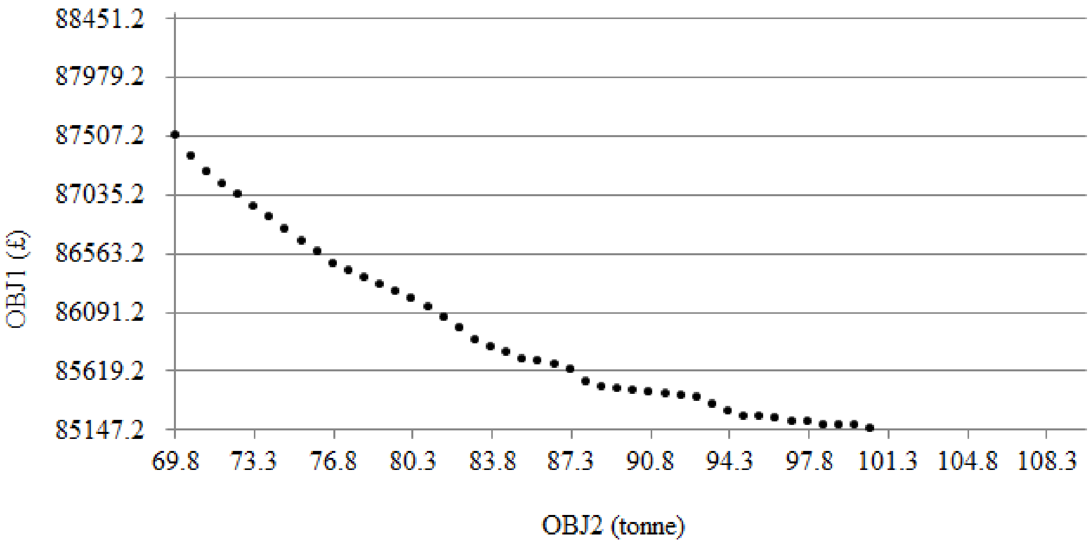


Fig. 3.12 The Pareto front of the the bi-criteria model

To select the most preferred solution from this set of solutions, we have also implemented a normalised distance method, for which the two objectives are normalised as follows:

$$\bar{f}_1(x,y,z) = \frac{f_1(x,y,z) - f_1(x_1^*,y_1^*,z_1^*)}{f_1(x_2^*,y_2^*,z_2^*) - f_1(x_1^*,y_1^*,z_1^*)} \in [0,1], \tag{3.50}$$

$$\bar{f}_2(x) = \frac{f_2(x) - f_2(x_2^*)}{f_2(x_1^*) - f_2(x_2^*)} \in [0,1], \tag{3.51}$$

where $\bar{f}_1$ and $\bar{f}_2$ are the normalised value of OBJ1 and OBJ2, respectively. Because this is a minimization problem, the ideal point is the minimum point $(0,0)$, as shown in Figure 3.13, the solution that is closest can be considered as the most preferred solution.

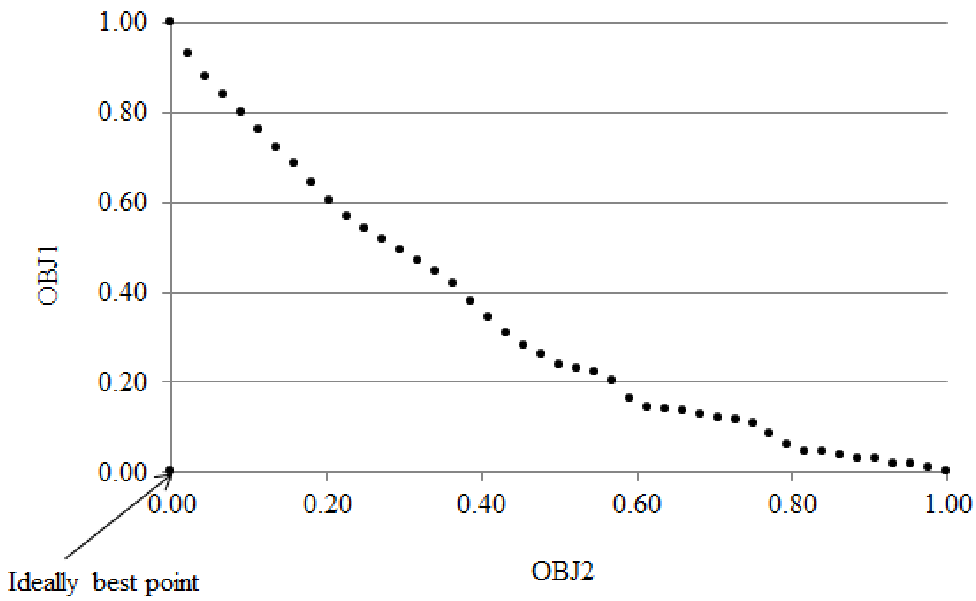


Fig. 3.13 The normalised Pareto front of the bi-criteria model

As can be seen in Figure 3.13, the normalised vectors of the solution that yields the minimum distance from the ideal point is $\bar{f}_1$=0.31 and $\bar{f}_2$=0.43. Mapping this on the original functions, the values of the corresponding vectors are OBJ1=£85869.20 and OBJ2=83.1 tonnes.

3.6 Conclusions

In this chapter, two models are proposed for building service network design models to minimise the total cost in intermodal transportation by taking the GHG emission cost and the intermodal transfer cost into account, which are more realistic than than current models. The method is based on McKinnon and Piecyk (2010)'s work that enables us to transfer the GHG emission to the same unit as the transportation cost. The total cost includes the internal cost, including operational cost and fixed cost, and external cost, including GHG emission cost, transfer cost and inventory cost. The transfer cost is in a non-linear expression, which is then linearised. Using these models, multi-commodities with different origins and destinations are transported in the intermodal systems in a cost-efficient and emission-efficient way. Using the time-dependent model, commodities can be shipped at promised departure date and arrive at the due date. Computational testing is presented for both models. All computational testing is performed using C# for data processing and CPLEX for the optimisation. The realistic case studies have suggested that the proposed models provide cost-efficient and emission-efficient ways for transporting commodities.

One important detail to remember is that the adjusted fixed cost is used for the purpose of sending the correct economic signals. We believe that it provides an opportunity for considerable cost reduction while adjusting fixed costs. By changing the value of the fixed costs of the three modes of transport, the trade-offs between emissions and transfer costs can be analysed.

The contributions of this chapter against existing knowledge can be summarised as follows:

- We extend the existing service network design models with the inclusion of a GHG emission and intermodal transfer cost, the latter needing a linearisation. A time-invariant and a time-dependent model are described which, to the best of our knowledge, are the first to explicitly include a GHG emission and intermodal transfer cost in their objectives, resulting in a new formulation for green intermodal transportation.

- We also describe a bi-objective optimisation formulation for the same problem as an alternative way to account for GHG emissions in intermodal transportation. We not only extend the existing service network design models to the single objective formulation, but have also shown how it is possible to formulate the problem as a bi-objective optimisation one, which forms a different way considering GHG emissions in intermodal transportation.

- We describe a case study building on secondary data collected from the UK using which the two models are tested. The data includes distances using different transportation modes, unit variable costs, unit transfer costs, capacities of different vehicles, unit emission costs, unit inventory costs and schedules of trucks and rail between each pair of nodes. Our data is available for use by other researchers to test different methodologies and models for intermodal transportation.

Chapter 4

Labelling Algorithms for Multiobjective Shortest Path Problems

4.1 Introduction

In the multiobjective shortest path problem (MSPP), multiple objectives are present which need to be optimised simultaneously. For example, in network optimisation problems, it is increasingly necessary to take several objectives such as travel cost, travel time, distance, greenhouse gas emissions (GHG) and transfer times into account. If there is no conflict between the objectives, then a solution can be found where every objective attains its optimum. In real world applications, the objectives are at least partly conflicting or are measured in different units (Miettinen, 1999). It is therefore not possible to find a solution that would be optimal for all the objectives at the same time. In this case, the aim is to find efficient solutions, also known as Pareto-optimal solutions, which are defined in Section 3.5. The MSPP is known to be an NP-hard problem (Garey, 1979). It is known that there are a set of problem instances for which the number of Pareto-optimal solutions increase exponentially with the size of the problem, which implies that the computational effort also increases exponentially (Garey, 1979).

The MSPP with additional restrictions, namely the constrained multiobjective shortest path problem (CMSPP), also belongs to the class of NP hard problems. The application of constraints to the MSPP frequently arises in practice. For example, a passenger who wishes to travel from one location to another and wishes to minimise financial cost, carbon footprint and make few transfers and yet also needs to arrive at the destination within a limited time is faced with the CMSPP. A review of the existing literature on the MSPP and the CMSPP is presented in Section 2.4.

The MSPP, and in particular the bi-objective SPP, with or without constraints, have been previously studied and largely solved using labelling algorithms. Very limited work exists on the unconstrained MSPP with three or more objectives (Ehrgott and Gandibleux, 2002). Furthermore, to the best of our knowledge, the CMSPP, either with single or multiple resource constraints, has not yet been studied. This chapter aims to fill this gap by presenting a computational study of a number of algorithms to solve such problems, all based on labelling, given their popularity and success in solving the bi-objective SPP.

The rest of the chapter is structured as follows. The next section describes the MSPP and the CMSPP in more formal terms. Sections 4.2 and 4.3 present labelling algorithms for the MSPP and the CMSPP, respectively. Computational results are presented in Section 4.4. Conclusions are given in Section 4.5.

4.1.1 Problem Description

A formal description of the CMSPP is presented as follows: The CMSPP is defined on a directed network $G(N, A)$, where N is the set of nodes and A is the set of arcs (i, j) joining nodes $i \in N$, $j \in N$, $i \neq j$. A source node s and a target node t are identified. A path p_{st} is a sequence of nodes and arcs from s to t. We use the set $K = \{1, 2, ..., k\}$ to denote the index set of the objective functions. For $k = 1$, we have the classical case of single objective shortest path problem. A cost vector $c_{ij} = (c_{ij}^1, c_{ij}^2, ..., c_{ij}^k)$ is associated with each arc $(i, j) \in A$. In a road network, for example, the cost vector $c_{ij} = (c_{ij}^1, c_{ij}^2, c_{ij}^3, c_{ij}^4)$ could represent time ($k = 1$), distance ($k = 2$), cost ($k = 3$) and greenhouse gas emissions ($k = 4$) for traversing arc (i, j). The objectives of a path p_{ij} is $c(p_{ij}) = (c^1(p_{ij}), c^2(p_{ij}), ..., c^k(p_{ij}))$.The set $M = \{1, 2, ..., m\}$ represents the index set of the constraints. For each arc $(i, j) \in A$, u_{ij}^h represents the amount of resource $h \in M$ consumed, the total of which is constrained by an upper bound U^h. For example, U^h might represent the maximum allowable travel budget for a passenger. The CMSPP is to find the shortest paths from the single-origin s to the single-destination t.

The CMSPP can be formulated as a network flow problem as follows:

$$\text{Minimise} \qquad z = \left(\sum_{(i,j)\in A} c_{ij}^1 x_{ij}, ..., \sum_{(i,j)\in A} c_{ij}^k x_{ij} \right) \tag{4.1}$$

subject to

$$\sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} = \begin{cases} 1 & \text{if} \quad i = s \\ 0 & \text{if} \quad i \neq s, t \\ -1 & \text{if} \quad i = t \end{cases} \tag{4.2}$$

$$\sum_{(i,j)\in A} u_{ij}^h x_{ij} \leq U^h, h \in M \tag{4.3}$$

$$x_{ij} \in \{0, 1\}, \text{for all } (i, j) \in A, \tag{4.4}$$

where x_{ij} is a binary variable which equals 1 if arc (i, j) is used in a shortest path from s to t, and 0 otherwise. Constraints (4.2) ensure a balance of flow at each node and constraints (4.3) model the resource restrictions. If $M = \emptyset$, or, alternatively if $U^h = \infty$ for all $h \in M$, then the formulation becomes that of the MSPP.

4.1.2 Dominance

Let us now briefly describe the concept of dominance in minimisation problems, which will be used in the algorithms described later in the chapter. If $X = \{x_1, ..., x_k\}$ and $Y = \{y_1, ..., y_k\}$ are two objective vectors, then X is said to dominate Y if and only if $x_l \leq y_l$ holds for all $l \in \{1, ..., k\}$ and a strict inequality holds for at least one $l \in \{1, ..., k\}$. The solution is *efficient* if its objectives are not dominated by that of any other feasible solutions. A path p_{st} in G from the source node $s \in N$ to the target node $t \in N \setminus \{s\}$ is a sequence $(s, i_1), (i_1, i_2), ..., (i_l, t)$ of arcs in A. For example, suppose five paths from the the source node s to the target node t exist, as shown in Figure 4.1, namely, s-1-t, s-2-t, s-2-3-t, s-3-t and s-3-2-t, with cost vectors (7,5,16), (7,8,5), (10,17,9), (4,16,8) and (7,9,12). The cost vector (7,9,12) of path s-3-2-t is dominated by the cost vector (7,8,5) of path s-2-t under the first two criteria. The cost vector (10,17,9) of path s-2-3-t is dominated by two cost vectors, (7,8,5) of path s-2-t and (4,16,8) of path s-3-t, under the three criteria. However, none of the cost vectors (7,5,16), (7,8,5), and (4,16,8) are dominated by another.

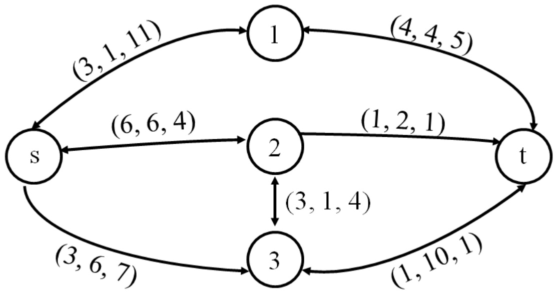


Fig. 4.1 An MSPP example with three minimising objectives

More formally, a path p_{st} from node s to node t dominates another path q_{st} if the objectives $c(p_{st}) = \sum_{(i,j)\in p_{st}} c_{ij}$ and $c(q_{st})$ are such that $c^l(p_{st}) \leq c^l(q_{st})$ holds for all $l \in \{1, ..., k\}$ and at least one l makes the inequality strict. Therefore, for the MSPP, we are looking for a set of efficient paths. Similarly, the CMSPP is concerned with finding a set of efficient paths which additionally satisfy constraints (4.3).

4.2 Labelling Algorithms for the MSPP

As in the classical Dijkstra algorithm, labelling algorithms extend paths from the source node to the rest of the network by labelling the nodes. In this section, we present three labelling algorithms for the MSPP. Two are label setting, one of which is the one described in Martins (1984) and the other is new. The last is a label correcting algorithm which extends the one described in Brumbaugh-Smith and Shier (1989).

4.2.1 Label Setting MSPP Algorithms

Martins' Algorithm

The algorithm described in Martins (1984), which are denoted here by ULS (unconstrained label setting), is a label selection method, in which all labels are treated separately. Let $i \in N$ be some node of $G(N, A)$. The rth label associated with node i is represented as $l^r(i)$=$[(c^1(p_{si}), c^2(p_{si}), ..., c^k(p_{si})), (j, r')]_r$, where $j \in N \setminus \{i\}$ and r' indicates some label of node j, for which $c_r(p_{si}) = c_{r'}(p_{sj}) + c_{ji}$. $L(i)$ is the list of $l^r(i)$ at a particular node $i \in N$. Each time a label of node j is generated, it is put into a temporary set $L(j)$. Labels in $L(j)$ are then checked and those representing a dominated path from s to j are deleted. All labels are put in the temporary set L, where $L = \bigcup_{i \in N} L(i)$. Of these labels, the lexicographically smallest label is found and removed from L. From this lexicographically smallest label of some node $i \in N$, a temporary label is assigned to every node $j \in N \setminus \{i\}$ such that $(i, j) \in A$. The lexicographically smallest ordering method guarantees that this selected label belongs to non-dominated paths. We present the steps of this procedure in more detail in Algorithm 1.

The Aggregate Algorithm

The aggregate algorithm (ULS$^+$) is obtained by replacing the lexicographic ordering of the labels in ULS with an aggregate ordering. This idea is inspired by Iori et al. (2010). Using the same notation as in Section 4.2.1, we attach additional aggregate information $c^{k+1}(p_{si})$ to each label $l^r(i)$=$[(c^1(p_{si}), c^2(p_{si}), ..., c^k(p_{si})), (j, r')]_r$, where $(j, i) \in A$ and $c^{k+1}(p_{si})$ are defined by the sum of the objective function coefficients for path p_{si}, namely $c^{k+1}(p_{si}) = c^1(p_{si}) + c^2(p_{si}) + ... + c^k(p_{si})$. At each iteration, we select a label from the temporary set for which $c^{k+1}(p_{si})$ is the minimum. As the arc costs are non-negative, this selected label will belong to a non-dominated path and will never be dominated by another temporary label (Iori et al., 2010). A full description of the aggregate algorithm is not needed as it follows the same steps as in Algorithm 1, and only differs with respect to selecting the label with the smallest value of $c^{k+1}(p_{si})$ from the set L.

Algorithm 1 ULS

procedure ULS (network (N, A), cost vector $c = (c^1, c^2, ..., c^k)$, the source node s)
  Initialisation: $L(s)$=$\{[(0, 0, ..., 0), (-, -)]_1\}$, $L(i) = \emptyset$, for all $i \in N \setminus \{s\}$, L=$\{[(0, 0, ..., 0), (-, -)]_1\}$
  while $L \neq \emptyset$ **do**
    remove the lexicographically smallest label from L, let this be the r'th label associated with node j
    for all $i \in N$ such that $(j, i) \in A$ **do**
      $c_r(p_{si}) = c_{r'}(p_{sj}) + c_{ji}$ and let $[(c^1(p_{si}), c^2(p_{si}), ..., c^k(p_{si})), (j, r')]_r$ be a new temporary label of node i
      if any label in $L(i)$ represents a dominated path from s to i **then**
        remove the labels that represent dominated paths
      end if
    end for
  end while
end procedure
Output: a set of efficient paths from the node s to the node t

4.2.2 Label Correcting MSPP Algorithm

Approaches to unconstrained label correcting algorithms (ULC) employ either label-selection or node-selection. We opt for node-selection policy here, also used by Skriver and Andersen (2000) and Raith and Ehrgott (2009). Node-selection means that a node is selected in any iteration of the algorithm and all labels of this node are extended along all outgoing arcs. A bi-objective label correcting algorithm is described in Brumbaugh-Smith and Shier (1989) where the First-in First-out (FIFO) principle is used to select nodes. We generalise this approach to solve the MSPP with any number of objectives, which is explained below and shown in Algorithm 2 in greater detail.

Let the rth label associated with node $i \in N$ be represented as $g^r(i)$=$[(c^1(p_{si}), c^2(p_{si}), ..., c^k(p_{si}))]_r$. $\mathcal{L}$ is the list of nodes with modified labels that have not yet been reconsidered, treated in FIFO order. $L(i)$ is the list of $l^r(i)$ at a particular node i. Each time a node i in $\mathcal{L}$ is selected, all labels in the set $L(i)$ are extended along all outgoing arcs (i, j). The dominated labels in the set $L(j)$ are then eliminated for all $j \in N \setminus \{i\}$. Whenever the set $L(j)$ changes, we add node j to $\mathcal{L}$ if it is not already there. The algorithm terminates when the set $\mathcal{L}$ is empty.

Algorithm 2 ULC

procedure ULC (network (N, A), cost vector $c = (c^1, c^2, ..., c^k)$, the source node s)
  Initialisation: $L(s) = \{[(0, 0, ..., 0)]_1\}$, $L(i) = \emptyset$, $i \in N \setminus \{s\}$, $\mathcal{L} = \{s\}$
  while $\mathcal{L} \neq \emptyset$ **do**
    remove the first node i from $\mathcal{L}$
    for all $j \in N$, such that $(i, j) \in A$ **do**
      $c_{r'}(p_{sj}) = c_r(p_{si}) + c_{ij}$ and let $[(c^1(p_{sj}), c^2(p_{sj}), ..., c^k(p_{sj}))]_{r'}$ be a new label of node j
      if any label in $L(j)$ represents a dominated path from s to j **then**
        remove the labels that represent dominated paths
        if $L(j)$ has changed and $j \notin \mathcal{L}$ **then**
          $\mathcal{L} \leftarrow \mathcal{L} \cup j$
        end if
      end if
    end for
  end while
end procedure
Output: a set of efficient paths from the node s **to the node** t

4.3 Labelling Algorithms for the CMSPP

In this section, two labelling algorithms are presented for the CMSPP, where both use feasibility and dominance checks within the algorithms. An illustrative example is provided for such a situation using the instance in Figure 4.1, to which one constraint is added. Subsequently, each arc is associated a vector $\left[(c^1_{ij}, c^2_{ij}, c^3_{ij}), u^1_{ij}\right]$ indicating the objective function coefficients and the resource consumption. It is assumed that the limit on the resource usage is 10. The resulting CMSPP instance is shown in Figure 4.2.

In this example, there are two paths from the source node s to node 2 with the following cost and constraint vectors: $[(6, 6, 4), 7]$ of path s-2 and $[(6, 7, 11), 2]$ of path s-3-2. Feasibility checking is done first. Since the limitation on the resource usage is 10, we keep both vectors. If we check the dominance of the cost vectors only, $(6, 7, 11)$ is dominated by $(6, 6, 4)$. The vector $[(7, 8, 5), 12]$ corresponding to path s-2-t will therefore be eliminated when applying feasibility checking because 12 is greater than the resource usage 10. As a result, the vector $[(8, 9, 12), 7]$ corresponding to path s-3-2-t, which happens to be an efficient solution for this instance, is eliminated during the dominance checking. Applying dominance checking to the cost vectors

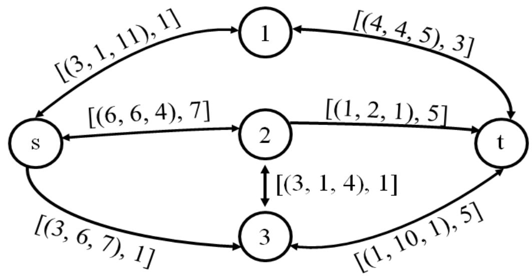


Fig. 4.2 A CMSPP example with three objectives and one constraint

implies that the remaining set of efficient solutions will be incomplete. This example shows that dominance checking should be performed on both cost and constraint vectors. For example, the vectors $[(6,6,4),7]$ and $[(6,7,11),2]$ corresponding to paths s-2 and s-3-2, respectively, are not dominated by one other. The resulting efficient solutions from the source node s to the target node t are $[(7,5,16),4]$ for path s-1-t, $[(8,9,12),7]$ for path s-3-2-t and $[(4,16,8),6]$ for path s-3-t.

The next section will present two types of algorithms for the CMSPP, namely label setting and label correcting.

4.3.1 Label Setting CMSPP Algorithms

In this section, we introduce a label setting CMSPP algorithm which is based on the aggregate algorithm, shown in Algorithm 3. The reason for selecting the aggregate algorithm as the basis, as opposed to the Martins' algorithm, is that the former computationally outperforms the latter. This will be shown in Section 4.4.1. The same notation is used as in Sections 4.1.1 and 4.2.1, namely the rth label associated with node i is represented as $l^r(i)$=$[(c^1(p_{si}), c^2(p_{si}), ..., c^k(p_{si}))$, $(u^1(p_{si}), u^2(p_{si}), ..., u^m(p_{si}))(j, r')]_r$, where $j \in N \setminus \{i\}$ and r' indicates some label of node j, for which $c_r(p_{si}) = c_{r'}(p_{sj}) + c_{ji}$, $u_r(p_{si}) = u_{r'}(p_{sj}) + u_{ji}$. $L(i)$ is the list of $l^r(i)$ at a particular node i, $L = \bigcup_{i \in N} L(i)$. This algorithm first checks the constraint vectors when each temporary label is set, which rules out "expensive "paths by considering the present set of labels, which is followed by dominance checking. The output is a set of efficient paths from the source node s to the target node t, derived from the set of paths that satisfy the resource constraints.

Algorithm 3 CLS

procedure CLS (network (N, A), cost vector $c = (c^1, c^2, ..., c^k)$, constraint vector $u = (u^1, u^2, ..., u^m)$, upper bound for constraints $U = (U^1, U^2, ..., U^m)$, the source node s)

 Initialisation: $L(s) = \{[(0, 0, ..., 0), (0, 0, ..., 0), (-, -)]_1\}$, $L(i) = \emptyset$, $i \in N \setminus \{s\}$, $L = \{[(0, 0, ..., 0), (0, 0, ..., 0), (-, -)]_1\}$

 while $L \neq \emptyset$ **do**

  remove the label with smallest value of $c^{k+1}(p_{si})$ from L, let it be the rth label associated with node i

  for all $j \in N$, such that $(i, j) \in A$ **do**

   $c(p_{sj}) = c_r(p_{si}) + c_{ij}$, $u(p_{sj}) = u_r(p_{si}) + u_{ij}$ and let

$[(c^1(p_{sj}), c^2(p_{sj}), ..., c^k(p_{sj})), (u^1(p_{sj}), u^2(p_{sj}), ..., u^m(p_{sj})), (i, r)]_{r'}$ be a new temporary label of node j

   if $\exists h \in \{1, 2, ..., m\}$, $u^h(p_{sj}) > U^h$ **then**

    remove $l^{r'}(j)$

    if any label in $L(j)$ represents a dominated path from s to j **then**

     remove the labels that represent dominated paths

    end if

   end if

  end for

 end while

end procedure

Output: a set of efficient paths from the node s to the node t

4.3.2 Label Correcting CMSPP Algorithms

The label correcting algorithm for the CMSPP described here extends the one for the MSPP. Using the notation defined earlier, the rth label associated with node i is represented as $g^r(i)$=$[(c^1(p_{si}), c^2(p_{si}), ..., c^k(p_{si})), (u^1(p_{si}), u^2(p_{si}), ..., u^m(p_{si}))]_r$. $L(i)$ is the list of $l^r(i)$ at a particular node i; $\mathcal{L} = \{s\}$, and $\mathcal{L}$ is the list of nodes with modified labels that have not yet been reconsidered, treated in FIFO order. Algorithm 4 checks the feasibility when each label is generated, and then tests the dominance of the labels in $L(i)$. The procedure yields all efficient paths from the source node s to the target node t.

Algorithm 4 CLC

---

procedure CLC (cost vector $c = (c^1, c^2, ..., c^k)$, constraint vector $u = (u^1, u^2, ..., u^m)$, upper bound for constraints $U = (U^1, U^2, ..., U^m)$, the source node s)
    Initialisation: $L(s) = \{[(0, 0, ..., 0), (0, 0, ..., 0)]_1\}$, $L(i) = \emptyset$, $i \in N \setminus \{s\}$,$\mathcal{L} = \{s\}$
    while $\mathcal{L} \neq \emptyset$ **do**
        remove the first node i from $\mathcal{L}$
        for all $j \in N$, such that $(i, j) \in A$ **do**
            $c_{r'}(p_{sj}) = c_r(p_{si}) + c_{ij}$, $u(_{r'}p_{sj}) = u_r(p_{si}) + u_{ij}$ and let
$[(c^1(p_{sj}), c^2(p_{sj}), ..., c^k(p_{sj})), (u^1(p_{sj}), u^2(p_{sj}), ..., u^m(p_{sj}))]_{r'}$ be a new label of node j
            if $\exists h \in \{1, 2, ..., m\}$, $u^h(p_{sj}) > U^h$ **then**
                remove $l^{r'}(j)$
                if any label in $L(j)$ represents a dominated path from s to j **then**
                    remove the labels that represent dominated paths
                    if $L(j)$ has changed and $j \notin \mathcal{L}$ **then**
                        $\mathcal{L} \leftarrow \mathcal{L} \cup j$
                    end if
                end if
            end if
        end for
    end while
end procedure
Output: a set of efficient paths from node s to node t

4.4 Computational Results

In this section, the computational results are indicated using the five algorithms described for the MSPP and the CMSPP. The algorithms are coded using C++ and compiled using the optimisation option -$O3$. All numerical tests are performed on the Iridis 4 computer cluster with dual 2.6 GHz Intel Sandybridge processors and 64 GB of memory (http://cmg.soton.ac.uk/iridis). The computational results are carried out on the basis of a different number of criteria k. In the computational experiments, we assume the source node s is node 1 and the target node t is node $|N|$. All algorithms are tested on two classes of networks, namely random and grid.

Random and grid networks are both well recognised and have been used extensively by researchers to test algorithms for the bi-objective SPP (Mote et al., 1991; Raith and Ehrgott, 2009) and for the MSPP (Guerriero and Musmanno, 2001). Let $|A| = a$ and $|N| = n$. The percentage density d of the network $G(N, A)$ is defined as $100a/(n(n-1))$. A graph is said to be a full network with 100% density if $\forall i, j \in N, i \neq j$, $\exists (i, j) \in A$. Random networks can have different densities. For all random instances, values of the cost vector have been drawn from a uniform distribution in the range $[1, 100]$, although the effect of changing this interval will be tested in Section 4.4.1. In grid networks, each node in the network is connected with two, three or four neighbours along one or more dimensions, and connections are between adjacent nodes. All bi-objective shortest path grid network instances used here are those generated in Raith and Ehrgott (2009). In these instances, the costs are randomly chosen from a uniform distribution in the range $[1, 10]$.

The remainder of this section first presents the results for the MSPP, then for the CMSPP. Although several works already exist in the literature presenting the results for the former (see, for example, Paixão and Santos (2013); Raith and Ehrgott (2009); Skriver (2000)), we still include the MSPP here as it is a special case of the CMSPP and therefore used to show consistency of results across the two main problems.

4.4.1 Computational Results for the MSPP

In this section, the results are presented for randomly generated and grid network instances. The effect of changing the objective function coefficients on the performance of the algorithms is also tested.

Random Network Instances with Two Objectives

We first conduct the experiments with two objective functions to allow the results to be put in context with respect to the existing results reported in the literature. The comparisons are performed on a total of 30 randomly generated instances with $n \in \{100, 200\}$ and $d \in \{5\%, 10\%, 20\%, 50\%, 100\%\}$. A comparison of computational times of Martins' algorithm (ULS), the aggregate algorithm (ULS$^+$) and the label correcting algorithm (ULC) is shown in Table 4.1. Each row is an average of three instances for a particular configuration (n, d) of the parameters. The average total numbers of Pareto-optimal solutions are also reported under the column entitled "#A".

Table 4.1 Average solution times (in seconds) on randomly generated MSPP instances with two objectives

n	d(%)	ULS	ULS$^+$	ULC	#A
100	5	<0.2	<0.2	<0.2	3
	10	<0.2	<0.2	<0.2	10
	20	<0.2	<0.2	<0.2	8
	50	0.34	<0.2	<0.2	5
	100	11.71	3.90	1.90	8
200	5	0.34	<0.2	<0.2	5
	10	2.30	<0.2	<0.2	10
	20	10.32	1.86	0.57	13
	50	14.25	6.70	1.95	14
	100	813.44	325.71	71.32	15

As Table 4.1 shows, all instances with 100 nodes and less than 20% density networks are solved in less than 0.2 seconds. Instances with 100 nodes and full networks can be solved within 11.71 seconds using Martins' algorithm, followed by 3.90 seconds using the aggregate algorithm and 1.90 seconds using the label correcting algorithm. Using the label correcting algorithm, all instances with 200 nodes and less than 50% density networks can be solved in less than two seconds. For the instance with 200 nodes and a full network, the run-time is 71.32 seconds using the label correcting algorithm, which is only 8.77% of the time required by Martins' algorithm.

Clímaco and Martins (1982) presented a path/tree handling procedure for the bi-objective shortest path problem and reported their computational performance, shown in Table 4.2. Skriver and Andersen (2000) presented two label correcting algorithms (Skriver1 and Skriver2), which are improvements of the Brumbaugh-

Smith and Shier (1989)'s algorithm (Brum). Computational results, which are reported in Table 4.3, are tested using a randomly generated network.

Table 4.2 A part of the bi-objective results reported in Clímaco and Martins (1982)

n	$d(\%)$	Run-time (in seconds)	#A
100	4.81	10.888	4
	4.71	10.630	4
	4.82	18.280	4
	5.10	139.084	6
	5.19	12.088	4
	5.24	22.036	6
	5.18	38.568	7
200	5.11	22.958	4
	5.13	2.432	1
	5.04	6.402	2
	5.07	45.636	5
	4.82	13.412	4
	5.30	10.924	3
	5.09	53.450	7

Table 4.3 A part of the average solution times (in seconds) reported in Skriver and Andersen (2000)

n	$d(\%)$	Brum	Skriver1	Skriver2
100	2-4	2.52	7.33	1.84
100	11	12.59	N/R*	11.05
200	2-4	18.27	57.86	13.59
200	2	9.01	N/R*	4.12
200	3	18.34	N/R*	12.16
200	11	79.55	N/R*	73.50

*N/R: Not reported.

In Table 4.2, the instances with 100 nodes and up to 5.25% density are solved between 10.630 and 139.084 seconds, while we report less than 0.2 seconds run-time. The instances with 200 nodes and around 5% density are solved between 2.432 and 45.636 seconds, while we report 0.34 seconds taken by Martins' algorithm and less than 0.2 seconds by the aggregate and label correcting algorithms. In Table 4.3, reported run-times range between 1.84 and 79.5 seconds, while we report less than 0.2 seconds run-time for instances with the same number of nodes and even greater density. Although the data we use is not exactly the same as the one in the literature, we select the reported results using randomly generated instances with the same number of nodes and density. The results show that our run-times for the shortest path problem with two objectives are very competitive.

Random Network Instances with Three and More Objectives

A comparison of computational times of Martins' algorithm (ULS), the aggregate algorithm (ULS$^+$)and the label correcting algorithm (ULC) is shown in Table 4.4. The comparisons are performed on a total of 90 instances with $k \in \{2, 3, 4, 5\}$, $n \in \{100, 200\}$ and $d \in \{5\%, 10\%, 20\%, 50\%, 100\%\}$. Each row is an average of three instances for a particular configuration (n, d, k) of the parameters.

Table 4.4 Average solution times (in seconds) on randomly generated MSPP instances with three and more objectives

n	$d(\%)$	k	ULS	ULS$^+$	ULC
100	10	3	0.54	<0.2	<0.2
		4	2.18	0.62	<0.2
		5	4.21	1.28	<0.2
	20	3	4.13	1.03	0.23
		4	21.97	6.63	1.61
		5	71.70	15.63	3.36
	50	3	3.88	1.57	0.66
		4	28.26	10.73	4.58
		5	97.41	44.00	15.27
	100	3	476.45	141.20	45.44
		4	4444.05	1989.78	825.96
		5	>2 hours	>2 hours	3314.77
200	5	3	6.02	0.85	<0.2
		4	10.45	2.99	<0.2
		5	69.29	10.72	0.73
	10	3	61.59	5.63	0.77
		4	289.56	33.46	4.36
		5	1780.81	249.21	13.34
	20	3	654.50	56.88	8.92
		4	2481.39	733.78	119.60
		5	>2 hours	2513.50	498.25
	50	3	544.75	88.91	20.68
		4	2302.36	1110.59	234.10
		5	>2 hours	3420.02	1091.54
	100	3	>2 hours	3711.21	1701.47
		4	>2 hours	>2 hours	>2 hours
		5	>2 hours	>2 hours	>2 hours

For all instances with $n = 100$, $d = 5\%$ and $k = 2, 3, 4, 5$, the solution times are less than 0.2 seconds. The results would seem to show that the average solution time grows exponentially with the density, network size and the number of objectives. For example, when $n = 100$, $d = 100\%$, the average solution time for ULC for an instance with three objectives is 45.44 seconds, whereas for instances with four and five objectives the average solution times increase to 825.96 and 3314.77 seconds, respectively. When $n = 200$, $d = 100\%$, $k = 4$ or 5, the MSPP cannot be solved within a time limit of two hours by any of the three algorithms. The results suggest that the label correcting algorithm consistently outperforms the other two in terms of the solution time, followed by the aggregate algorithm and then Martins' algorithm. For example, when $n = 200$, $d = 5\%$, $k = 5$, the label correcting algorithm requires 0.73 seconds to solve the instance, which is 1.05% and 6.81% of the time required by Martins' algorithm and the aggregate algorithm, respectively. The aggregate algorithm and label correcting algorithm are able to solve instances with 200 nodes, 50% density and five objectives for which Martins' algorithm fails. The label correcting algorithm has further solved instances with 100 nodes, 100% density and five objectives where the others could not. The general conclusion from the results presented in Table 4.4 is that the larger the size of the network, the better the reduction in computational times the label correcting algorithm yields.

Grid Network Instances

To test the algorithms on grid networks, the 33 instances, which are generated and solved in Raith and Ehrgott (2009) using a near shortest path algorithm, are used. For these instances, Raith and Ehrgott (2009) report the run-time and number of efficient solutions, where the run-time is only a part of the total solution time representing the generation of all efficient paths' labels together with the actual paths. The same number of solutions is obtained as reported in Raith and Ehrgott (2009). In our experiments, we only report instances that can be solved within two hours, for which the corresponding run-times are all less than 0.2 seconds. The results are reported in Table 4.5. In the table, the first column shows the instance name, and the second shows the size defined by the height h and width w of the grid. Columns titled n and a show the number of nodes and arcs in the instance, respectively. The next two columns show the individual run-times for each instance, solved once by ULS$^+$ and once by ULC. Column #U shows the cardinality of the set of efficient paths, and finally the last column shows the total number of efficient paths including equivalent paths.

Table 4.5 Computational results including the solution times (in seconds) on grid MSPP instances

	$h \times w$	n	a	ULS$^+$ run-time	ULC run-time	#U	#A
G01	30×40	1202	4720	18.04	12.26	37	40
G02	20×80	1602	6240	1024.54	783.38	80	90
G04	90×50	4502	17900	1208.89	1140.84	46	75
G12	50×50	10002	39600	216.46	190.48	52	56
G15	2450×2	4902	19596	0.25	<0.2	6	7
G16	1225×4	4902	19592	0.45	0.29	6	6
G17	612×8	4898	19586	2.14	1.39	10	10
G18	288×17	4898	19550	18.74	15.03	15	16
G19	196×25	4902	19550	74.37	62.68	18	18
G20	140×35	4902	19530	294.30	238.48	32	36
G21	111×44	4886	19448	717.80	598.77	54	61
G22	92×53	4878	19398	1240.45	1237.97	53	62
G23	79×62	4900	19468	4893.86	4826.45	77	91

As can be seen from Table 4.5, label correcting algorithm appears to perform better than the aggregate label setting algorithm for grid networks. However, the difference in the solution times is not as much as is seen for random networks, which can be up to 85%. For grid networks, the most significant difference in run-time appears in Instance G02, for which ULC requires 33.54% less time than ULS$^+$. In another instance, namely G22, the difference is negligible.

Effect of Changing Objective Function Coefficients

To test the effects of the objective function coefficients on the solution time, the random network instances is tested with coefficient ranges $[1, 100]$, $[1, 1000]$ and $[1, 10000]$. In the light of the results presented in Table 4.4, we only compare the aggregate algorithm with the label correcting algorithm, as Martins' algorithm is dominated with respect to computational times. The results are presented in Table 4.6. Each row is an average of three instances for each particular configuration of the parameters.

The results in Table 4.6 show that the label correcting algorithm still performs better than the label setting algorithm for the instances tested. For smaller and sparser networks, the solution time only slightly increases when the range of the

Table 4.6 Average solution times (in seconds) with different coefficient ranges on randomly generated instances with $k = 3$

(n, d)	ULS$^+$			ULC		
	$[1, 100]$	$[1, 1000]$	$[1, 10000]$	$[1, 100]$	$[1, 1000]$	$[1, 10000]$
(100,5)	<0.2	<0.2	<0.2	<0.2	<0.2	<0.2
(100,10)	<0.2	<0.2	0.25	<0.2	<0.2	<0.2
(100,20)	1.03	1.30	1.93	0.23	0.52	0.52
(100,50)	1.57	11.61	31.70	0.66	8.29	8.46
(100,100)	141.20	147.46	377.99	45.44	64.09	203.12
(200,5)	0.85	0.87	1.79	<0.2	<0.2	<0.2
(200,10)	5.63	6.51	13.33	0.77	1.94	1.99
(200,20)	56.88	90.66	164.63	8.92	18.83	19.59
(200,50)	88.91	941.45	1666.54	20.68	472.07	554.43
(200,100)	3711.21	5291.66	6200.55	1701.47	3726.95	3742.04

objective function coefficients increases from $[1, 100]$ to $[1, 1000]$, or from $[1, 1000]$ to $[1, 10000]$. For networks with 100 and 200 nodes, and with more than 50% density, the change is more significant. For example, when $(n, d) = (200, 50)$, the instance using the range $[1, 10000]$ requires an average of 1666.54 seconds by ULS$^+$, while the range $[1, 100]$ takes 88.91 seconds, which is nearly a 19-fold increase. Similarly, for the same parameter configuration, the range $[1, 10000]$ takes more than 23 times longer than when the coefficients are drawn from the range $[1, 100]$ by ULC.

4.4.2 Computational Results for the CMSPP

Random Network Results

In this section, the computational results are reported which were obtained with Algorithms 3 and 4. The computational results are carried out on random networks with different numbers of objectives and with m constraints, with the vector $U = (U^1, U^2, ..., U^m)$ representing the upper bounds of the constraints. The comparisons are performed on a total of 90 instances with $k \in \{2, 3, 4\}$, $n \in \{50, 100, 200\}$ and $d \in \{5\%, 10\%, 20\%, 50\%, 100\%\}$. Each row is an average of three instances for each particular configuration of the parameters.

It is assumed that both the arc vector cost and the arc vector constraint are drawn from a uniform distribution in the range $[0, 100]$. A comparison of solution times of CLS and CLC with a single constraint and $U^1 = 200$ is shown in Table 4.7 and Figures 4.3–4.7.

Table 4.7 Average solution times (in seconds) for the CMSPP with $m = 1$

n	d (%)	k	CLS	CLC
100	5	2	<0.2	<0.2
		3	<0.2	<0.2
		4	<0.2	<0.2
	10	2	<0.2	<0.2
		3	0.22	<0.2
		4	0.39	<0.2
	20	2	<0.2	0.20
		3	2.95	0.82
		4	7.37	2.57
	50	2	1.38	0.72
		3	8.81	5.30
		4	35.03	17.54
	100	2	109.22	55.82
		3	1355.22	724.79
		4	4131.98	1181.25
200	5	2	0.302	<0.2
		3	0.86	<0.2
		4	2.73	0.28
	10	2	2.87	0.51
		3	16.07	2.72
		4	68.11	9.67
	20	2	47.97	10.32
		3	412.72	73.21
		4	1475.52	272.74
	50	2	96.62	31.10
		3	838.69	220.60
		4	2540.51	863.70
	100	2	2956.97	2082.07
		3	>2 hours	>2 hours
		4	>2 hours	>2 hours

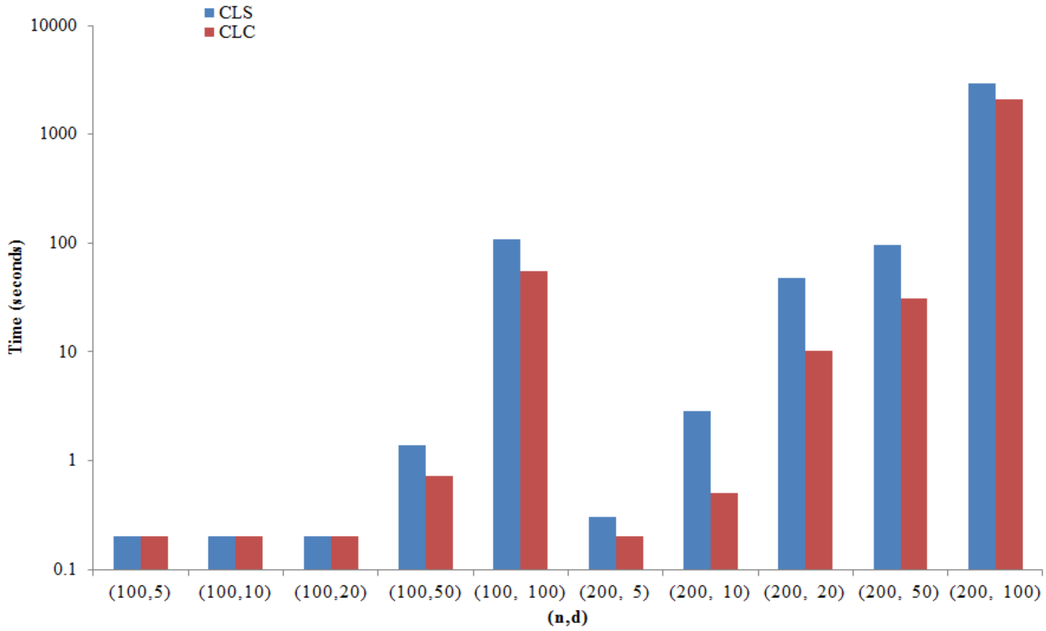


Fig. 4.3 Comparison of solution times for instances with $m = 1$ and $k = 2$ tested using CLS and CLC

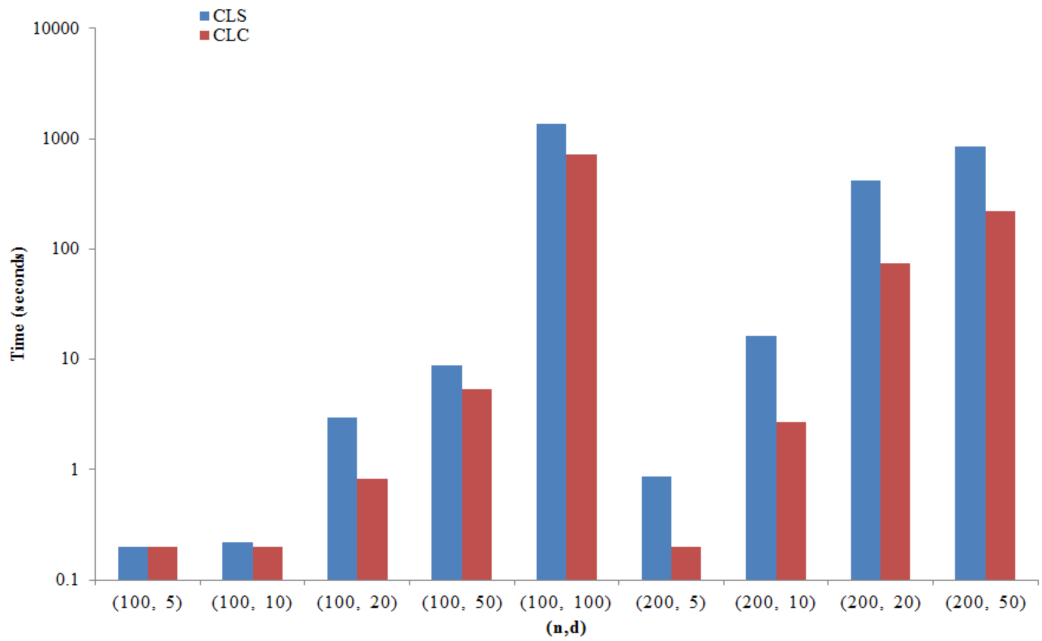


Fig. 4.4 Comparison of solution times for instances with $m = 1$ and $k = 3$ tested using CLS and CLC

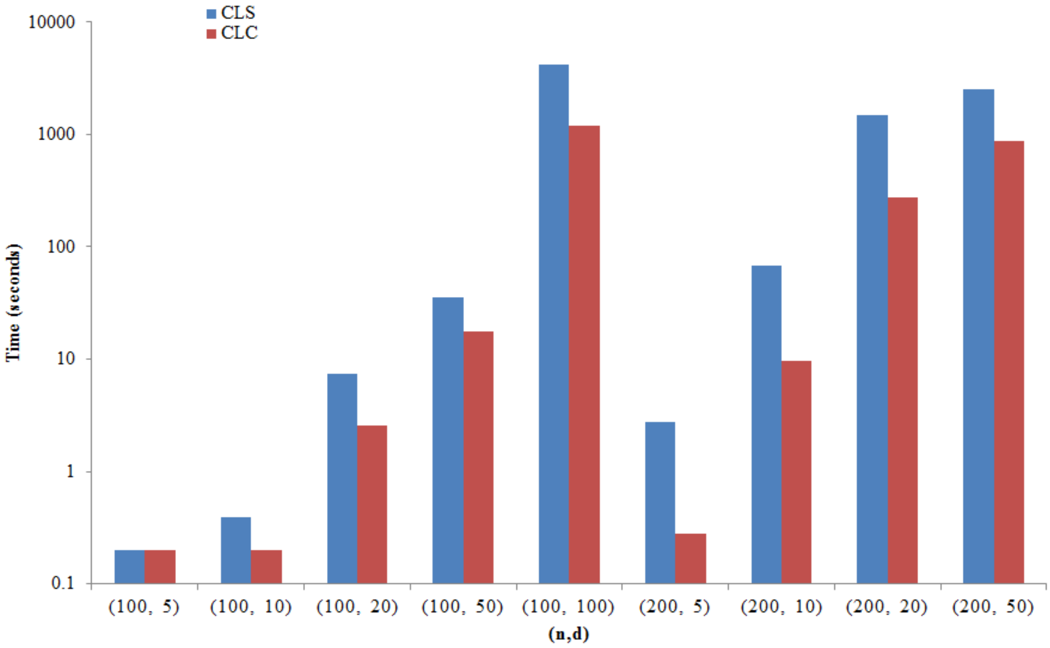


Fig. 4.5 Comparison of solution times for instances with $m = 1$ and $k = 4$ tested using CLS and CLC

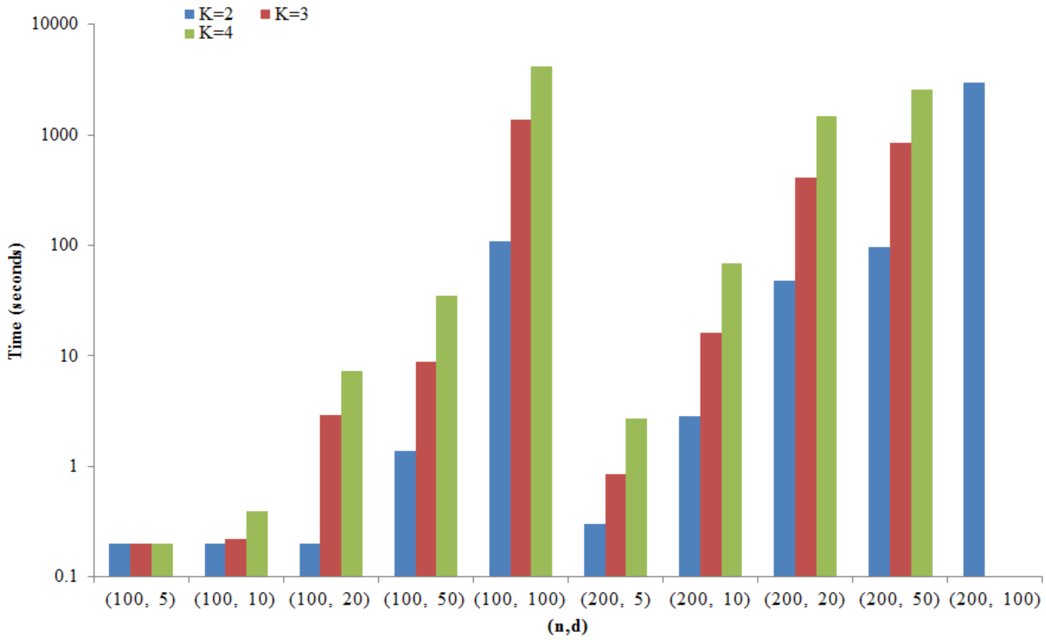


Fig. 4.6 Average solution times for instances with $m = 1$ tested using CLS

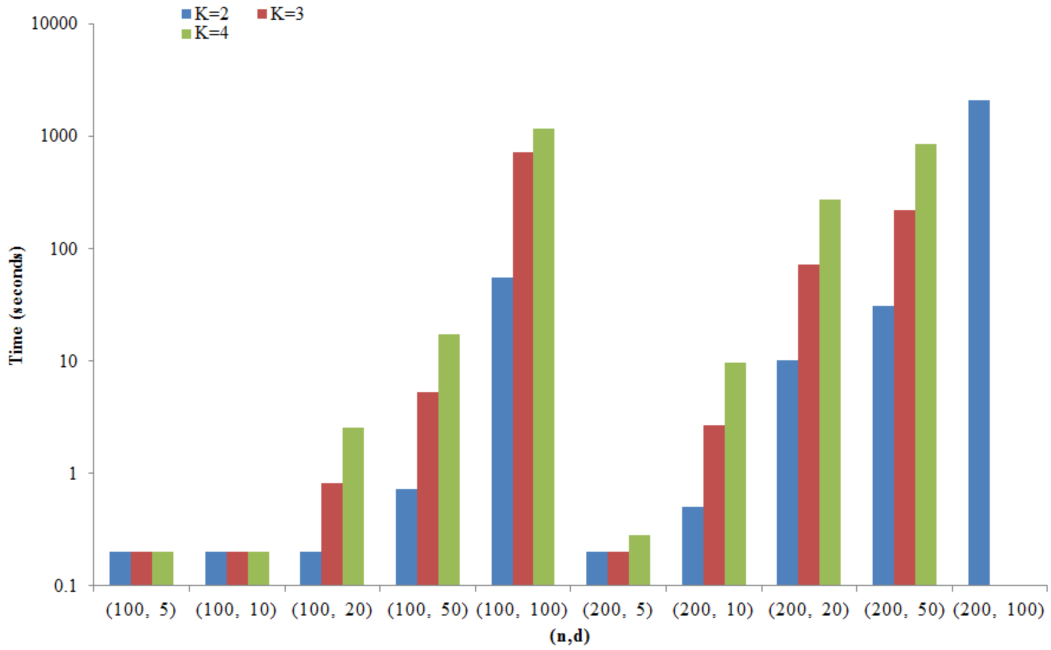


Fig. 4.7 Average solution times for instances with $m = 1$ tested using CLC

Our experiments show that the label correcting algorithm is clearly better for the CMSPP than the label setting algorithm. The savings solution time by using the former is around 50% when $n = 100$. When $n = 200$ and $d \leq 100\%$, the label correcting algorithm appears to perform much better than the label setting algorithm. The savings percentage of the solution time is between 66% and 89.74%. The greater the number of the objectives, the more difficult the problem becomes. When $n = 100$ and $d = 5$ and 10%, most of the instances can be solved within 0.2 seconds. When $n = 200$, $d = 100\%$ and $k = 3$ or 4, however, none of the instances can be solved within a time limit of two hours.

To test the effects of the increase in the number of constraints, the following parameter settings are used. Both the arc costs and constraint coefficients are chosen from a uniform distribution in the range of $[0, 100]$. The upper bounds are set as $u^1 = u^2 = u^3 = 200$, while the comparisons are performed on 90 instances with $n \in \{100, 200\}$, $d \in \{5\%, 10\%, 20\%, 50\%, 100\%\}$ and $m \in \{1, 2, 3\}$.

Table 4.8 and Figures 4.8–4.12 show the comparison in the solution times required by CLS and CLC for when the number of constraints ranges from 1 to 3 and for when $k = 2$. Each result shown is an average of three instances for a particular configuration of the parameters.

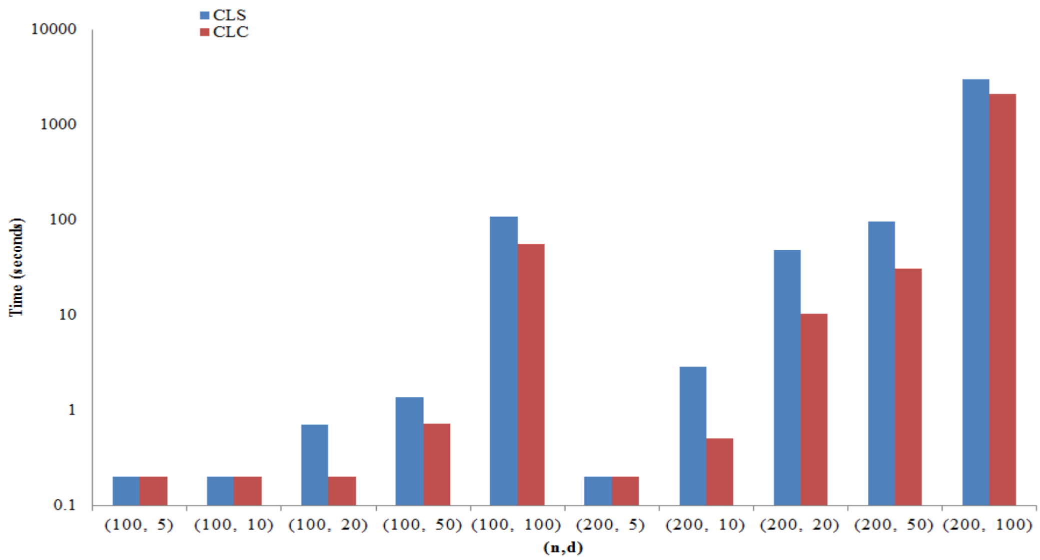


Fig. 4.8 Comparison of solution times for instances with $k = 2$ and $m = 1$ tested using CLS and CLC

Table 4.8 Average solution times (in seconds) for the CMSPP with $k = 2$

n	$d(\%)$	m	CLS	CLC
		1	<0.2	<0.2
	5	2	<0.2	<0.2
		3	<0.2	<0.2
		1	<0.2	<0.2
	10	2	<0.2	<0.2
		3	<0.2	<0.2
		1	0.71	0.20
100	20	2	1.70	0.46
		3	2.24	0.65
		1	1.38	0.72
	50	2	10.46	4.41
		3	24.13	11.31
		1	109.22	55.82
	100	2	1020.37	475.29
		3	2777.99	1332.18
		1	<0.2	<0.2
	5	2	0.26	<0.2
		3	0.302	<0.2
		1	2.87	0.51
	10	2	7.67	1.36
		3	8.48	1.45
		1	47.97	10.32
200	20	2	252.83	39.49
		3	435.93	71.56
		1	96.62	31.10
	50	2	624.90	203.49
		3	2052.19	519.63
		1	2956.97	2082.07
	100	2	>2 hours	>2 hours
		3	>2 hours	>2 hours

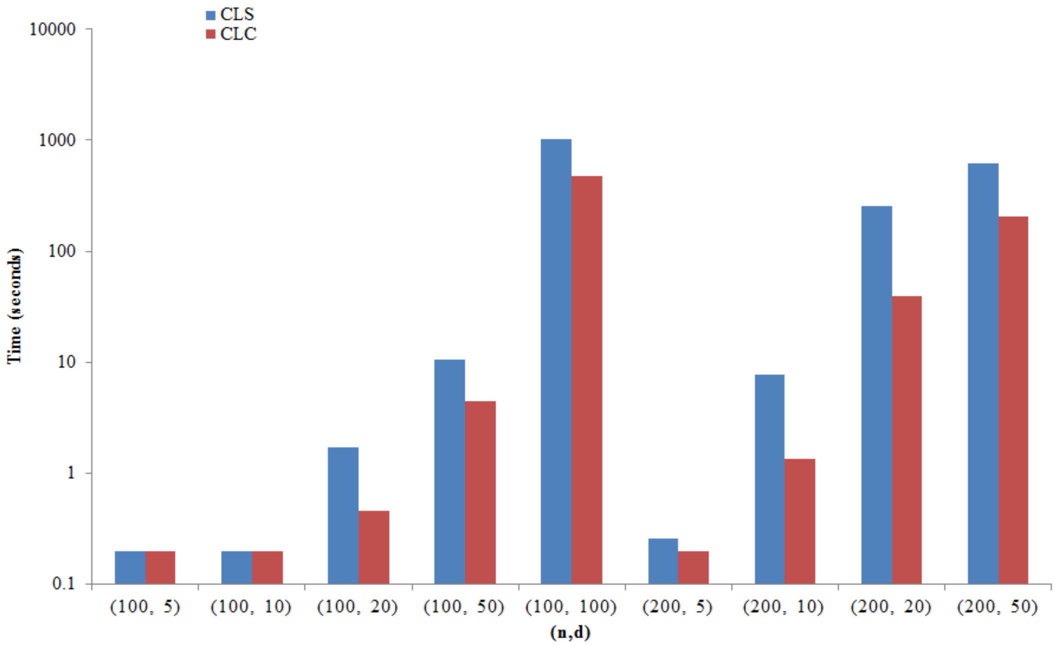


Fig. 4.9 Comparison of solution times for instances with $k = 2$ and $m = 2$ tested using CLS and CLC

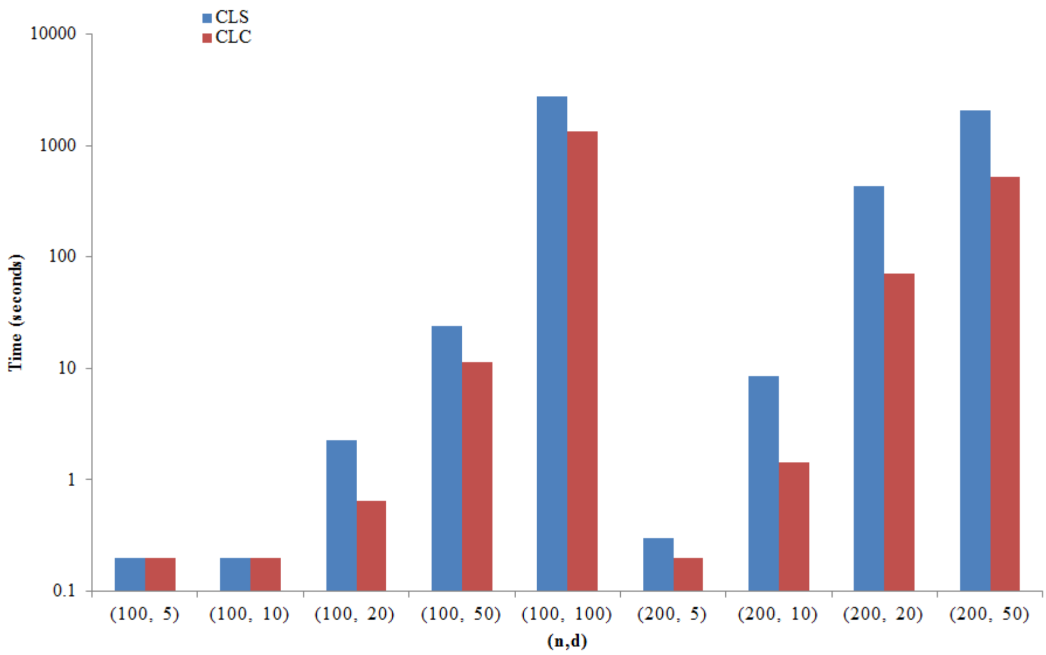


Fig. 4.10 Comparison of solution times for instances with $k = 2$ and $m = 3$ tested using CLS and CLC

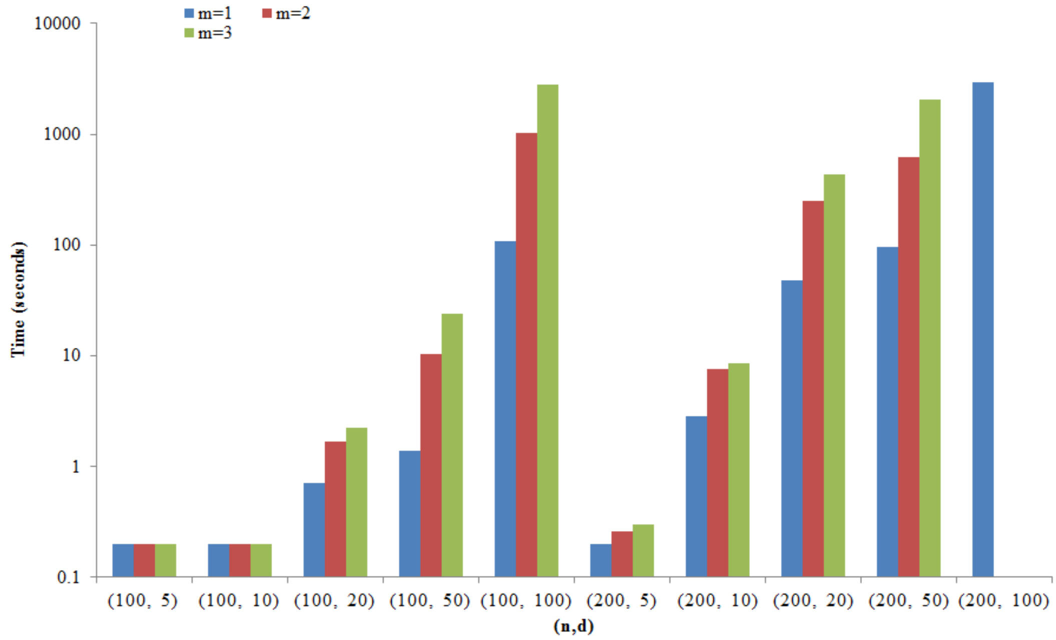


Fig. 4.11 Average solution times for instances with $k = 2$ tested using CLS

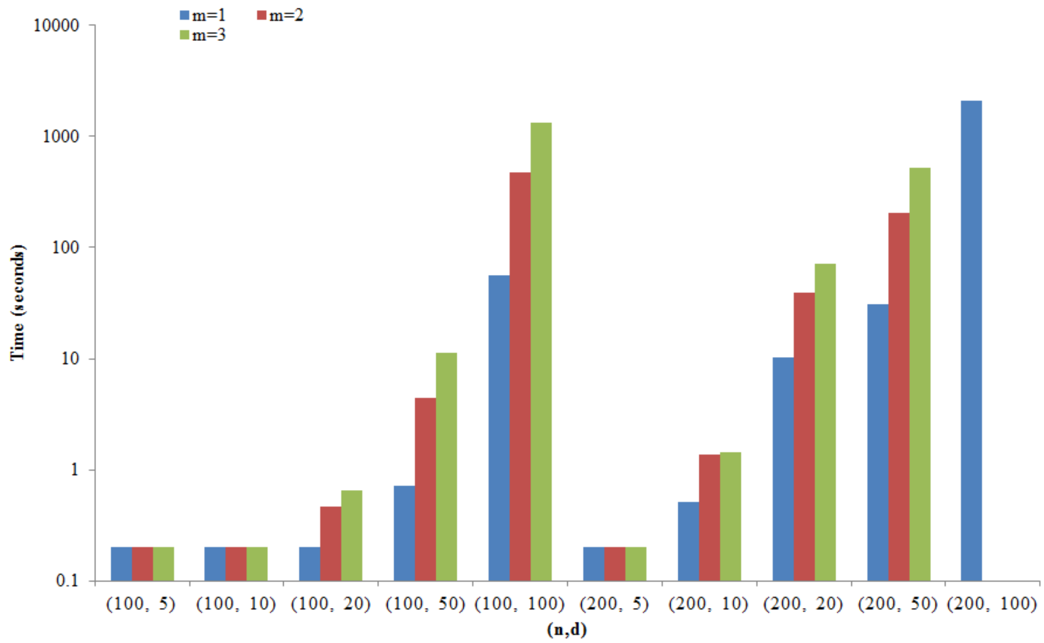


Fig. 4.12 Average solution times for instances with $k = 2$ tested using CLC

Figures 4.8–4.10 suggest that CLC generally performs better than CLS for the CMSPP. For instances with $n = 100$ and $d = 5, 10$ and 20%, there are no significant benefits to using different labelling algorithms as the solution times are negligible. Only for larger and denser networks does CLC show any advantages. Figures 4.11 and 4.12 show that the number of constraints significantly affects the solution times for networks with more than a 50% density. The higher the number of constraints, the slower the solutions are obtained. The likely reason for this behaviour is that more constraints will increase the number of objectives when using CMSPP algorithms, which leads to a slower process for finding the efficient solutions.

Correlated Networks

Let us now consider the case where the constraint vector is correlated to the objective vector. This case might arise in practice, for example, when a passenger would like to travel with a minimised cost and number of transfers, but has a travel time constraint. The travel time in this case depends on both the travel cost and the number of transfers. To further investigate the effect of correlations between constraint and objective vectors, we generate random network instances with two objectives, c_{ij}^1 and c_{ij}^2, and one constraint u_{ij}. The constraint vector is obtained by the following expression:

$$u_{ij} = \alpha c_{ij}^1 + (1 - \alpha)c_{ij}^2, \tag{4.5}$$

where $0 \leq \alpha \leq 1$. The computational results for six such networks are shown in Table 4.9 for five different values of α.

Table 4.9 Average solution times (in seconds) for the CMSPP with two objectives and one correlated constraint solved by CLC

n	$d(\%)$	no correlation	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 0.9$
100	50	0.72	<0.2	<0.2	<0.2	<0.2	<0.2
100	100	55.82	1.98	2.03	2.03	1.98	1.91
200	10	0.51	<0.2	<0.2	<0.2	<0.2	<0.2
200	20	10.32	0.62	0.65	0.66	0.64	0.61
200	50	31.10	2.32	2.33	2.35	2.32	2.30
200	100	2082.07	57.26	63.99	63.18	59.90	52.33

From Table 4.9, it can be observed that all instances using a correlated constraint vector require much less solution time than an uncorrelated one. The reasoning is that there are fewer non-dominated labels generated during the process depending on this correlation, which results in less computational time. For instances with 200 nodes and 100% density, the solution time without correlation is 2082.07 seconds, while all correlated instances use only about 2.5% of the solution time. Instances with α values of 0.1 and 0.9 are the most efficient, while instances with α values of 0.3 and 0.5 require slightly more computational time than other instances.

CMSPP Instances from the Literature

To the best of our knowledge, there are only two published papers on the constrained shortest path problem with a single objective, described in detail by Zhang et al. (2013), who use a heuristic algorithm to solve them. The first instance is a transportation network with 20 nodes and 28 edges, and the second is an instance of the delay-constrained least-cost problem with 33 nodes and with 113 edges. Using our algorithms, we optimally solved both instances within 0.2 seconds to confirm that the heuristic solutions reported in Zhang et al. (2013) are indeed optimal. In the case of the latter, the objective of the efficient solution we found is 21.804 units, which is in contrast to the value 21.8135 reported in Zhang et al. (2013).

4.5 Conclusions

Multiobjective shortest path problems are computationally harder than single objective ones. Labelling algorithms are considered the most efficient exact methods for the MSPP. In this chapter, five labelling algorithms are described for solving the MSPP and the CMSPP to find all optimal paths from a source node to a target node in a graph. All five algorithms generate the complete efficient set. An aggregate label setting algorithm is introduced for the MSPP, which is a new algorithm inspired by Iori et al. (2010), and compares with Martins' algorithm and the label correcting algorithm. Martins' algorithm and the aggregate algorithm are both label setting algorithms. They differ in the label selecting mechanism. Martins' algorithm selects the lexicographically label, while the aggregate algorithm selects the label with smallest aggregate of objectives. Since there are few works in the literature reporting the computational results of labelling algorithms with three and more objectives, these three algorithms are compared using instances with different numbers of objectives, which range from two to five. A computational study is provided on different labelling algorithms for the MSPP and CMSPP.

An aggregate label setting algorithm and a label correcting algorithm are introduced for the CMSPP. Both are based on algorithms for the MSPP. Computational results show that label correcting algorithms consistently perform better than label setting algorithms for both the MSPP and the CMSPP. This phenomenon has already been observed for the MSPP and we have presented here, for what we believe to be the first time, that a similar behaviour is observed for the more generalised CMSPP problem. The results appear to show that solution times depend on the size and density of the network, the number of objectives and the objective function coefficients. For the CMSPP, the solution times also depend on the number of constraints, the upper bounds and objective vectors. Using the algorithms, we were able to optimally solve two test instances for the CMSPP from the literature, which were previously only solved using heuristics.

We are also the first to test instances with positive correlations between constraint and objective vectors for the CMSPP. In the labelling algorithm, the more objectives which conflict, the more labels are generated. Correlated vectors are the reason for fewer conflicted objectives, which leads to fewer non-dominated labels being generated. Computational results show that all positively correlated instances take only up to 2.5% of the run-time compared with non-correlated instances.

The contributions of this chapter can be summarised as follows:

1. We first extend and enhance existing labelling algorithms for the MSPP to deal with cases with any number of objectives.

2. We describe, for what we believe to be the first time in the literature, labelling algorithms for the CMSPP with any number of objectives and constraints.

3. We present computational experiments to compare and evaluate labelling algorithms for this class of problems.

Chapter 5

Local Search for Multiobjective Shortest Path Problems

5.1 Introduction

The multiobjective shortest path problem (MSPP) is an extension of the classical shortest path problem which has a wide range of practical applications such as traffic routing, telecommunications and resource allocation. A variety of algorithms and methods such as dynamic programming, labelling algorithms, ranking methods, utility function methods and interactive methods have been implemented and investigated for the MSPP. However, solving the MSPP to optimality poses challenges in terms of the computational power and time required as was demonstrated through the computational experiments presented in Chapter 4. Heuristic methods can be used to speed up the process of finding approximate Pareto-optimal solutions. A brief overview on the MSPP using heuristic algorithms was presented in Chapter 2.

Table 5.1 presents a tabulated summary of the existing heuristic algorithms for the MSPP. Our review identified two heuristics for the bi-objective SPP, namely genetic and ant colony algorithms, proposed by Lin and Gen (2007) and Ghoseiri and Nadjari (2010). Both papers provided the computational results tested on randomly generated instances. Ghoseiri and Nadjari (2010) reported the computational run-time of their heuristic, which is between 5.29% and 61.34% of the label correcting algorithm run-time. As for the MSPP, ant colony and genetic algorithms have been proposed by Häckel et al. (2008). All proposed algorithms were tested on instances with three objectives. The reported run-time is slower than the labelling algorithm in Pangilinan and Janssens (2007), which is 40 seconds for random networks and 100 seconds for grid networks in Bezerra et al. (2011) and Bezerra et al. (2013), respectively.

This chapter proposes two simple local search methods for the MSPP, in order that the total run-time to find the approximate Pareto-optimal set can be reduced considerably. The chapter also presents computational experiments to test the efficiency and the effectiveness of the local search algorithms, by comparing with the efficient set obtained by labelling algorithms. The purpose of developing local search algorithms for the MSPP is twofold:

Firstly, many heuristics are problem-specific, so that a method which works for one problem cannot be used to solve a different one (Reeves, 1993). To the best of our knowledge, using local search methods for the MSPP is a non-trivial task, and our aim is to fill this gap.

Secondly, genetic and ant colony optimisation algorithms described earlier in the literature for the MSPP are slow, as indicated above, and there is a need for heuristics that run fast, particularly for use in practical situations such as SAT-NAV

Table 5.1 Heuristics for the MSPP

Problem	Reference	Solution approach	Type of instances	Run-time	Results compared with the efficient set
Bi-objective SPP	Lin and Gen (2007)	genetic algorithm	random	N/R*	Yes
Bi-objective SPP	Ghoseiri and Nadjari (2010)	ant colony	random	between 196 and 2529 seconds	Yes
MSPP	Häckel et al. (2008)	ant colony	N/R*	N/R*	Yes (not referenced)
MSPP	Pangilinan and Janssens (2007)	genetic algorithm	random	N/R*, slower than labelling algorithm	No
MSPP	Bezerra et al. (2011) Bezerra et al. (2013)	ant colony	random and grid	random: 40 seconds, grid: 100 seconds	No

*N/R: Not reported.

and mobile phone applications. The local search algorithms described here are fit for such a purpose as they are first in the literature.

This chapter is organised as follows: Section 5.2 presents a brief overview of local search algorithms and then describes the algorithms developed for the MSPP; Section 5.3 presents the experimental results and comparisons with the labelling algorithm described in Chapter 4; and Section 5.4 presents the conclusions.

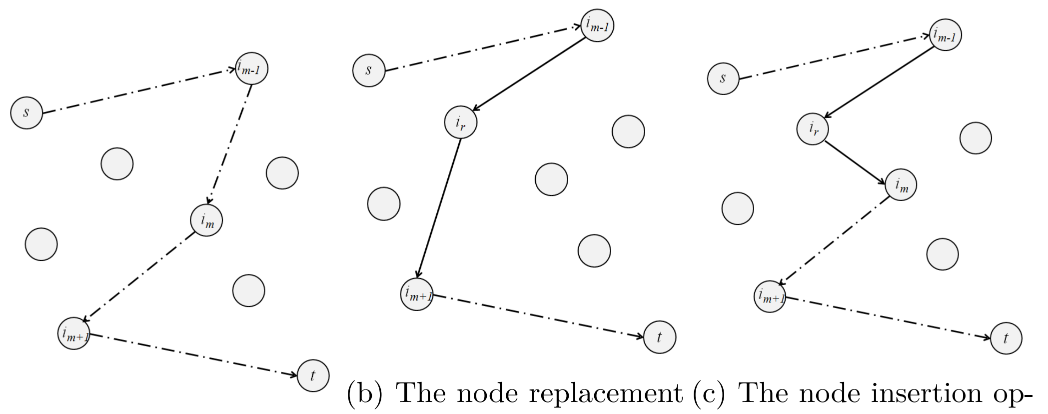
5.2 Local Search Methods

Local search methods are applied to a wide range of optimisation problems, due to the fact that they are generally able to produce good quality solutions within very short computation times. Local search consists of moving from a feasible solution to an optimisation problem and on to another in its neighbourhood according to some well-defined rules (Pirlot, 1996), and proceeding iteratively in such a fashion until a termination criterion is met. One such rule is that two feasible solutions are neighbours if they differ exactly on one variable. A given neighbour can be obtained by performing an elementary move on a feasible solution. A common criterion for selecting a neighbour solution is to pick up the best in the neighbourhood with respect to the objective of the optimisation problem being solved. We now present two moves used in the MSPP local search methods, and illustrate how they can be applied in our algorithms in Section 5.2.1.

5.2.1 Two Neighbourhood Moves for the MSPP

The following definition will be used in defining both moves. A path p^0_{st} in G from *source node* $s \in N$ to *target node* $t \in N \setminus \{s\}$ is defined as $p^0_{st} = (s, i_1, i_2, ..., i_l, t)$ with cost vector $c(p^0_{st}) = (c^1(p^0_{st}), c^2(p^0_{st}), ..., c^k(p^0_{st}))$, where $i_1, i_2, ..., i_l \in N \setminus \{s \cup t\}$ and $(s, i_1), (i_1, i_2), ..., (i_l, t) \in A$. Figure 5.1a shows such a path.

Node replacement This operator repeatedly replaces i_m in the initial path p^0_{st}, where $m \in \{1, 2, ..., l\}$, by another node i_r not in path p^0_{st} such that from the path $p^0_{st} = (s, i_1, i_2, ..., i_{m-1}, i_m, i_{m+1}, ..., i_l, t)$, a new path is obtained, i.e., $p^r_{st} = (s, i_1, i_2, ..., i_{m-1}, i_r, i_{m+1}, ..., i_l, t)$, where $(i_{m-1}, i_r) \in A$ and $(i_r, i_{m+1}) \in A$. The move is illustrated in Figure 5.1b. If $c(p^r_{st})$ is not dominated by $c(p^0_{st})$, then we keep the new path p^r_{st}. If it is, then we select another i_k not in p^0_{st} and replace it with i_m. We then check the dominance. For the first new path generated, we check the dominance with the p^0_{st}. In order to save computational time, for other new paths we only check the dominance with respect to the previous new path, not p^0_{st}, and



(a) A feasible solution (b) The node replacement operator (c) The node insertion operator

delete the dominated path. The operator iterates α times for all $m \in \{1, 2, ..., l\}$. The dominance of all undeleted paths is checked at the end of the algorithms.

Node insertion This operator randomly inserts another node i_r, which is not included in p^0_{st}, between two adjacent nodes i_m and i_{m+1}, where $m, m+1 \in \{1, 2, ..., l\}$, such that from the initial path $p^0_{st} = (s, i_1, i_2, ..., i_{m-1}, i_m, i_{m+1}, ..., i_l, t)$, a new path $p^r_{st} = (s, i_1, i_2, ..., i_m, i_r, i_{m+1}, ..., i_l, t)$ is obtained, where $(i_{m-1}, i_r) \in A$ and $(i_r, i_{m+1}) \in A$. The move is illustrated in Figure 5.1c. If $c(p^r_{st})$ is not dominated by $c(p^0_{st})$, then we keep this path. If it is, then we select another i_k not in p^0_{st} and insert it between i_m and i_{m+1}. We then check the dominance with the p^0_{st}. This step is repeated until a new path is generated.

5.2.2 Local Search Algorithms for the MSPP

This section presents an algorithmic description of the two local search algorithms we propose for the MSPP. The procedure initialises by applying Dijkstra's algorithm (Dijkstra, 1959) $|K|$ times, one for each objective $K = \{1, 2, ..., k\}$. As a result, we obtain k initial solutions, whose corresponding paths are denoted as $p^1_{st}, p^2_{st}, ..., p^k_{st}$. In this step, we choose Dijkstra's algorithm, as opposed to generating arbitrary solutions, as we guarantee the optimality of the solutions on which the improvement operators would be applied in later steps. Furthermore, we generate k initial solutions instead of one solution so that the solution space is better represented with respect to all k objectives. We now present two variations of the local search procedure, shown in Algorithms 5 and 6, respectively.

Algorithm 5 MSPP local search algorithm

procedure MSPP LOCAL SEARCH ALGORITHM (network (N, A), cost vector $c = (c^1, c^2, ..., c^k)$, the source node s)
  initialisation: generate k initial solutions using Dijkstra's algorithm for each objective $1, 2, ..., k$, denoted the initial solution set as P_1
  for $w = 1$ to χ **do**
    apply the node replacement operator for each solution in P_w to obtain the solution set S^w
    if any solution in S^w represents a dominated path from s to t **then**
      remove the solutions that represent the dominated paths
    end if
    selection: randomly select β solutions in S^w as the initial solutions in S^{w+1}
  end for
  Let $S = \cup_{w \in \{1,2,...,\chi\}} S^w$
  if any solution in S represents a dominated path from s to t **then**
    remove the solutions which represent the dominated path
  end if
end procedure Output: an approximate set of the efficient paths from the source node s to the target node t

Algorithm 6 MSPP local search algorithm*

procedure MSPP LOCAL SEARCH ALGORITHM* (network (N, A), cost vector $c = (c^1, c^2, ..., c^k)$, the source node s)
    initialisation: generate k initial solutions using Dijkstra's algorithm for each objective $1, 2, ..., k$, denote the initial solution set as P_1
    for $w = 1$ to χ **do**
        apply the node replacement operator for all solutions in P_w to obtain the solution set S^w
        if the node replacement operator iterates on the same node for δ times and the new paths are all dominated **then**
            apply the insertion operator
        end if
        if any solution in S^w represents a dominated path from s to t **then**
            remove the solutions which represent the dominated paths
        end if
        selection: randomly select β solutions in S^w as the initial solutions in S^{w+1}
    end for
    Let $S = \cup_{w \in \{1,2,...,\chi\}} S^w$
    if any solution in S represents a dominated path from s to t **then**
        remove the solutions which represent the dominated paths
    end if
end procedure Output: an approximate set of the efficient paths from the source node s to the target node t

In Algorithm 5, a node replacement operator is used to replace all nodes except for the source and target nodes in the initial set of paths, and consequently obtain the solution set S^1. We then randomly select β new initial solutions from S^1 and apply the node replacement operator to all solutions therein to obtain S^2. This step is repeated χ times. In Algorithm 6, in addition to the node replacement operator, we use an node insertion operator for exploring different neighbourhoods. The output of these two algorithms is an approximate set of the efficient paths from the source node s to the target node t.

5.3 Computational Results

In this section, the results of computational experiments conducted to test the efficiency and the effectiveness of the local search algorithms are presented, the solutions of which are compared to the efficient set obtained by the label correcting algorithm presented in Chapter 4. The label correcting algorithm yields the complete Pareto-optimal set of solutions whereas local search algorithms generate an approximate set. We perform the comparisons through four different solution quality indicators which are presented in Section 5.3.1. The fine-tuning of the local search parameters is presented in Section 5.3.2. In Section 5.3.3, we analyse the local search results and compare them with the label correcting solutions.

The computational experiments are tested on two types of networks, namely randomly generated and grid networks. The algorithms are coded using C++ and compiled using the optimisation option -$O3$. All numerical tests are performed on the Iridis 4 computer cluster with dual 2.6 GHz Intel Sandybridge processors and 64 GB of memory (http://cmg.soton.ac.uk/iridis). All randomly generated instances used here are the same as those generated in Chapter 4. For randomly generated instances, the computational results are carried out on the basis of a different number of criteria k and with 50% and 100% density, which are more difficult to solve by the exact algorithm and enable direct comparisons with exact algorithms to be made. Table 5.2 shows specifications of the randomly generated instances. Grid network instances are those with two objectives and described in Raith and Ehrgott (2009), the specifications of which were presented in Table 4.5 in Chapter 4.

5.3.1 Solution Quality Indicators

A quantitative comparison of the performance of different heuristic algorithms in multiobjective optimisation is less straightforward than single objective optimisation. Finding the right measures for evaluating the quality of the heuristic solutions is as important as the algorithms themselves. In single-objective optimisation problems, the quality of the solutions can be defined by the objective function itself: the smaller (or larger) the objective value, the better the solution. However, the output of multiobjective optimisation is a set of solutions. Zitzler et al. (2000, 2003) defined a number of measures to evaluate the solution quality assessment in multiobjective optimisation. Some of these are easy to apply in bi-objective optimisation. Since we have three and more objectives in our problems, the following measures of

Table 5.2 Specifications of randomly generated instances

Instance	n	$d(\%)$	k	number of efficient solutions
1	100	50	3	35
2	100	50	4	101
3	100	50	5	86
4	100	100	3	48
5	100	100	4	226
6	100	100	5	241
7	200	50	3	58
8	200	50	4	56
9	200	50	5	401
10	200	100	3	70
11	200	100	4	240
12	200	100	5	508

performance have been used: (1) the number of solutions obtained; (2) the ratio of the number of efficient solutions found by local search algorithms to the total number of efficient solutions found by exact algorithms; (3) the average distance between an approximate set and the efficient set found by exact methods; and (4) the ϵ-indicator method.

Let S be the approximate set that is generated by the local search algorithms and S^* be the efficient set generated by the labelling algorithm. We now define four approaches to evaluate the closeness and representativeness of S from S^*. The first three approaches are used by Lin and Gen (2007), and the fourth, namely the average distance method, is used by Ghoseiri and Nadjari (2010) to evaluate the quality of their heuristic solutions for the MSPP.

Number $|S|$ of solutions obtained: This measures the number of solutions in the approximate set S obtained by local search algorithms.

Ratio $R(S)$ of the number of efficient solutions found by local search to the total number of efficient solutions found by labelling algorithms: Here, one evaluates each approximate set on the basis of the number of efficient solutions found. The greater the value of $R(S)$ is, the better the heuristic solution set is. The $R(S)$ measure can be written as follows:

$$R(S) = \frac{|S \cap S^*|}{|S^*|} \times 100\%. \tag{5.1}$$

Average distance $D(S)$: This measures the average distance between the objectives of each point in S^* to those of its closest point in S as follows:

$$D(S) = \frac{1}{|S^*|} \sum_{b \in S^*} \min \{\|b - a\| \, ; a \in S\}, \tag{5.2}$$

where $\|\cdot\|$ is Euclidean distance metric. The smaller the value of $D(S)$, the better the approximate set is.

ϵ-indicator I_ϵ: This is the minimum factor for which we need to multiply the objectives of all the elements of a reference set S^* in order to have all the objectives of its elements dominated or equal to the objectives of the elements in S (Lizárraga et al., 2008). Smaller values of $I_\epsilon(S, S^*)$ mean that the set S is more similar to the reference set S^*. The smaller the value of I_ϵ the better the approximate set is. If $S = S \cap S^*$ then $I_\epsilon = 1.00$. The ϵ-indicator I_ϵ for two objective vectors $a = (a^1, a^2, ..., a^k) \in S$ and $b = (b^1, b^2, ..., b^k) \in S^*$ can be calculated as follows:

$$\epsilon_{a,b} = \max_{1 \le i \le k} \frac{a^i}{b^i} \quad \forall a \in S, b \in S^* \tag{5.3}$$

$$\epsilon_b = \min_{a \in S} \epsilon_{a,b} \quad \forall b \in S^* \tag{5.4}$$

$$I_\epsilon(S, S^*) = \max_{b \in S^*} \epsilon_b \tag{5.5}$$

or equivalently

$$\epsilon_{a,b} = \max_{b \in S^*} \min_{a \in S} \max_{1 \le i \le k} \frac{a^i}{b^i}. \tag{5.6}$$

5.3.2 Fine-tuning of the Parameters

Four parameters, namely α, β, χ and δ are used in Algorithms 5 and 6. This section describes how the parameters were fine-tuned. For this purpose, several possible combinations of the parameters were tested, which were then evaluated on the basis of their performance using the four indicators mentioned in Section 5.3.1. The procedure stops when no improvements are observed. To fine-tune the algorithms in a reasonable amount of computational time, we use the same combination of parameters for the randomly generated instances with same number of nodes and density. For example, the best combination of parameters selected for Instance #3 shown in Table 5.2 has also been used for Instances #1 and #2. The fine-tuning has been performed using Instances #3, #6, #9 and #12, where the results are reported in Tables 5.3–5.6. In the tables, each row presents average results over five runs.

Table 5.3 Fine-tuning of results for Instance #3 using Algorithm 5

ID	α	β	χ	$\|S\|$	$R(S)(\%)$	$D(S)$	I_ϵ
a	40	50	30	42	48.84	43.12	1.00
b	40	50	40	49	56.98	37.50	1.00
c	50	40	40	64	74.42	21.35	1.00
d	50	50	40	66	76.74	20.79	1.00
e	50	50	50	68	79.07	18.40	1.00
f	60	50	50	64	74.42	20.42	1.00
g	50	60	50	51	59.30	35.73	1.00
h	50	50	60	52	60.47	34.73	1.00

Table 5.4 Fine-tuning of results for Instance #6 using Algorithm 5

ID	α	β	χ	$\|S\|$	$R(S)(\%)$	$D(S)$	I_ϵ
a	40	80	50	130	29.88	68.00	1.96
b	50	70	50	130	30.71	62.59	1.96
c	50	80	40	119	32.78	65.24	1.96
d	50	80	50	129	33.20	62.59	1.96
e	50	80	60	125	32.37	65.83	1.96
f	50	90	50	126	31.54	64.01	1.96
g	50	100	50	130	31.95	62.59	1.96
h	60	80	50	114	31.95	65.21	1.96

As can be seen in Table 5.3, the qualities of solutions are getting better when α, β and χ increase. The best performance is found when (α, β, χ)=(50,50,50), which is selected as the combination of parameters to use for Instance #3. As for Table 5.4, the value of I_ϵ turns out to be the same for all tests. Furthermore, there is not one particular combination which yields the best performance. Tests b, d and g have the three best values. In this case, we assume $R(S)$ is more important than $|S|$ and select the combination (50,80,50) of parameters. Using the same procedure, (80,60,40) is selected for Instance #9 and (100,100,40) is selected for Instance #12 as shown in Tables 5.5 and 5.6, respectively.

Using the best combination of parameters found in Algorithm 5, we ran Algorithm 6 using different values of δ. Tables 5.7–5.10 report the fine-tuning with Instances #3, #6, #9 and #12, respectively.

As shown in Table 5.7, the value of I_ϵ is 1.00 for all values of δ, which indicates that all heuristic solutions obtained are optimal. Three other indicators indicate

Table 5.5 Fine-tuning of results for Instance #9 using Algorithm 5

ID	α	β	χ	$\|S\|$	$R(S)(\%)$	$D(S)$	I_ϵ
a	60	50	40	332	64.59	20.63	2.45
b	80	50	20	335	68.58	19.09	2.74
c	80	50	30	332	67.33	20.06	2.45
d	80	50	40	337	71.32	18.08	2.45
e	80	60	40	333	71.82	17.91	2.45
f	90	60	40	335	69.83	18.63	2.45
g	80	70	40	331	69.08	18.57	2.45
h	80	60	50	349	69.58	19.01	2.45

Table 5.6 Fine-tuning of results for Instance #12 using Algorithm 5

ID	α	β	χ	$\|S\|$	$R(S)(\%)$	$D(S)$	I_ϵ
a	80	100	40	386	41.73	39.18	1.95
b	80	100	50	397	42.91	38.11	1.88
c	80	100	30	389	41.93	39.10	1.88
d	80	120	40	395	42.91	38.42	1.88
e	100	100	40	404	44.69	35.54	1.88
f	130	100	40	395	42.72	38.68	1.88
g	100	120	40	393	42.32	38.87	1.88
h	100	100	50	394	42.72	38.72	1.88

Table 5.7 Fine-tuning of results for Instance #3 using Algorithm 6

ID	δ	$\|S\|$	$R(S)(\%)$	$D(S)$	I_ϵ
a	20	60	63.95	30.51	1.00
b	40	63	73.26	23.06	1.00
c	70	67	77.91	20.60	1.00
d	80	67	77.91	21.31	1.00
e	100	65	75.58	19.33	1.00
f	130	68	79.07	18.40	1.00
g	140	67	77.91	18.75	1.00
h	150	67	77.91	19.52	1.00

Table 5.8 Fine-tuning of results for Instance #6 using Algorithm 6

ID	δ	$\lvert S \rvert$	$R(S)(\%)$	$D(S)$	I_ϵ
a	30	130	33.20	80.00	1.96
b	40	130	33.20	80.00	1.96
c	50	130	33.20	80.0	1.96
d	60	130	33.20	80.0	1.96
e	70	130	33.20	80.0	1.96
f	80	130	33.20	80.00	1.96
g	90	130	33.20	80.00	1.96
h	100	130	33.20	88.00	1.96

Table 5.9 Fine-tuning of results for Instance #9 using Algorithm 6

ID	δ	$\lvert S \rvert$	$R(S)(\%)$	$D(S)$	I_ϵ
a	70	333	60.60	22.70	2.45
b	80	351	61.85	20.69	3.67
c	90	364	62.34	21.17	2.74
d	100	353	62.34	21.77	2.45
e	110	345	64.09	20.98	2.45
f	120	327	66.83	20.44	2.45
g	130	389	62.59	20.95	3.67
h	140	366	61.60	21.75	2.85

Table 5.10 Fine-tuning of results for Instance #12 using Algorithm 6

ID	δ	$\lvert S \rvert$	$R(S)(\%)$	$D(S)$	I_ϵ
a	40	374	38.78	40.59	1.88
b	60	381	40.94	39.49	1.88
c	80	387	42.13	39.02	1.88
d	100	392	42.52	38.82	1.88
e	110	395	42.91	38.08	1.88
f	120	390	42.32	38.87	1.88
g	130	391	42.32	38.87	1.88
h	140	392	42.32	38.87	1.88

better performance when the value of δ is increased. In this case, test f yields the best performance. When we continue to increase the value of δ, the performance deteriorates. For this instance, the value 130 is detected. In Table 5.8, the four indicators yield the same results for all tests. Since the greater the parameter is, and the slower the algorithm becomes, we select the smallest value 30. Following the same procedure, δ=120 and 110 are selected for Instances #9 and #12, respectively. A summary of the best combinations of parameters for different instance sizes in a randomly generated network is shown in Table 5.11, which is used in the next section.

Table 5.11 Parameters used for randomly generated networks in Algorithms 5 and 6

n	$d(\%)$	α	β	χ	δ
100	50	50	50	50	130
100	100	50	80	50	30
200	50	80	60	40	120
200	100	100	100	40	110

As for fine-tuning α, β, χ and δ for grid network instances, we set (α, β, χ)=(30,30,30) in the first run of Algorithm 5. When values of α, β and χ are increased, the performance measures of the instances confirm that there are no significant differences between the solution sets. We take Instance G01 as an example. Table 5.12 shows the values of indicators for Instance G01 using different combinations of α, β and χ. Table 5.13 shows the procedure of fine-tuning δ for G01. For other grid network instances, the same values of indicators are obtained while the values of the parameters are increased incrementally. Comparing fine-tuning of results in Tables 5.12 and 5.13, the performance of Algorithm 6 is similar to that of Algorithm 5. This results in the fact that the local search algorithms are not sensitive to the parameters for the grid network instances. Thus α, β, χ and δ are respectively set equal to 30, 30, 30 and 30, and will be used in Section 5.3.3.

5.3.3 Solution Performance Analysis

In this section, the results of how the local search algorithms run on two types of instances are reported and analysed, namely randomly generated and grid network instances. Table 5.14 summarises the performance measures of Algorithms 5 and 6 for the 12 randomly generated instances introduced in Table 5.2, as well as the run-time (in seconds) by the label correcting (LC) and the two local search algorithms. The LC algorithm is run on the same computer as the local search

Table 5.12 Fine-tuning of results for Instance G01 using Algorithm 5

ID	α	β	χ	$\|S\|$	$R(S)(\%)$	$D(S)$	I_ϵ
a	30	30	30	46	5.41	33.80	1.28
b	40	30	30	46	5.41	33.80	1.28
c	30	40	30	46	5.41	33.80	1.28
d	30	30	40	46	5.41	33.80	1.28
e	40	40	40	46	5.41	33.80	1.28
f	50	50	50	46	5.41	33.80	1.28
g	80	80	80	46	5.41	33.80	1.28
h	100	100	100	46	5.41	33.80	1.28

Table 5.13 Fine-tuning of results for Instance G01 using Algorithm 6

ID	δ	$\|S\|$	$R(S)(\%)$	$D(S)$	I_ϵ
a	30	46	5.41	33.80	1.28
b	40	46	5.41	33.80	1.28
c	50	46	5.41	33.80	1.28
d	60	46	5.41	33.80	1.28
e	70	46	5.41	33.80	1.28
f	80	46	5.41	33.80	1.28
g	90	46	5.41	33.80	1.28
h	100	46	5.41	33.80	1.28

algorithms. Values are averaged over the five runs for each column, in all reported results hereafter.

It is worth noting that for all instances, local search algorithms are much faster than LC. The run-times for all tested instances are less than seven seconds by the use of either algorithm. For Instance #11, it takes more than two hours to solve the MSPP using LC, while it takes less than two seconds using the local search algorithms. Instance #11 also yields more than 50% efficient solutions. There is not a significant difference between Algorithms 5 and 6 in relation to $|S|$. The greatest difference in $|S|$ is six for Instance #9. As for the indicator $R(S)$, more than 50% efficient solutions were identified in six instances out of 12, in which three instances are with 100 nodes and 50% density. Instance #3 has the highest $R(S)$, which is 79.07%, identified by both Algorithms 5 and 6. The lowest $R(S)$ is obtained by Instance #6 using both Algorithms 5 and 6. However, this also guarantees one third of the efficient solutions. The higher the $R(S)$ is, the smaller the $D(S)$ and I_ϵ are. The performance appears to show that local search algorithms are highly effective in obtaining a good approximate set.

Table 5.14 Summary of results on randomly generated instances

Instance	Algorithm 5					Algorithm 6					LC run-time (in seconds)
	$\lvert S\rvert$	$R(S)(\%)$	$D(S)$	I_ϵ	run-time (in seconds)	$\lvert S\rvert$	$R(S)(\%)$	$D(S)$	I_ϵ	run-time (in seconds)	
1	23	65.14	12.80	1.50	< 0.2	23	66.86	12.77	1.50	< 0.2	0.90
2	58	57.23	22.36	2.15	< 0.2	58	57.03	26.91	2.15	0.30	4.41
3	68	79.07	18.40	1.25	0.51	68	79.07	18.40	1.25	1.06	12.68
4	18	37.08	30.66	2.84	0.61	22	45.00	30.00	2.84	0.34	52.11
5	81	48.67	26.28	2.2	1.08	86	51.81	24.02	2.2	3.06	800.73
6	80	33.20	62.59	1.96	3.21	80	33.20	80.00	1.96	1.68	3385.41
7	28	47.59	21.38	1.56	< 0.2	26	44.83	21.52	2.07	0.29	19.99
8	31	55.36	30.17	3.14	0.48	31	55.00	28.58	3.25	0.62	237.72
9	333	71.82	17.91	2.45	1.23	327	66.83	20.44	2.45	1.52	922.29
10	30	43.14	14.66	1.89	0.28	32	46.29	14.26	1.63	0.75	1790.41
11	141	58.83	25.62	1.73	1.67	138	57.50	26.26	1.92	1.70	>2 hours
12	227	44.69	35.54	1.88	6.39	218	42.91	38.08	1.88	5.80	>2 hours

Table 5.15 Summary of results on grid network instances

Instance	$h \times w$	n	a	$\|S\|$	$R(S)(\%)$	$D(S)$	I_e	run-time (in seconds)	LC run-time (in seconds)
G01	30×40	1202	4720	46	5.41	33.80	1.28	0.23	12.26
G02	20×80	1602	6240	57	6.67	47.76	1.22	0.53	783.38
G04	90×50	4502	17900	26	8.70	25.75	1.27	0.87	1140.84
G12	50×50	10002	39600	58	7.69	43.72	1.29	1.61	190.48
G15	2450×2	4902	19596	2	33.33	8.05	3.14	<0.2	<0.2
G16	1225×4	4902	19592	4	33.33	7.52	2.00	<0.2	0.29
G17	612×8	4898	19586	10	20.00	12.75	1.59	<0.2	1.39
G18	288×17	4898	19550	6	13.33	15.17	1.48	0.25	15.03
G19	196×25	4902	19550	8	11.11	25.14	1.39	0.26	62.68
G20	140×35	4902	19530	33	12.50	41.39	1.45	0.54	238.48
G21	111×44	4886	19448	54	3.70	28.38	1.21	1.71	598.77
G22	92×53	4878	19398	64	9.43	25.81	1.19	2.23	1237.97
G23	79×62	4900	19468	62	5.19	45.24	1.26	2.26	4826.45

Table 5.15 shows the performance measures of local search solutions for grid network instances. Regarding time requirements, all instances can be solved within three seconds. Nine out of 12 instances can be solved in less than one second. The LC algorithm is grossly outperformed by the local search algorithms with time savings up to 95.32%. In relation to the indicator $R(S)$, G15 and G16 have the highest value of 33.33%, which is even less than the smallest value of random instances in Table 5.14. The worst case is G21, which reported the smallest $R(S)$ of 3.70%. This means that local search performs better in finding efficient solutions for randomly generated instances than for grid network instances. As for the other indicators, all are similar in value scale with randomly generated instances, which are in acceptable scales.

Figures 5.2–5.5 illustrate the approximate Pareto front constructed by local search algorithms compared with the Pareto front generated by the label correcting, respectively, for four grid network instances, namely G01, G16, G17 and G21. As can be seen, all four approximate Pareto front using local search methods are located near the Pareto front generated by the labelling algorithm. Some duplicate solutions can also be observed. For G21, although a small proportion of $|S^*|$ is obtained, the relatively small values of $D(S)$ and I_ϵ indicate that the local search algorithms are capable of finding a good approximate Pareto-optimal set.

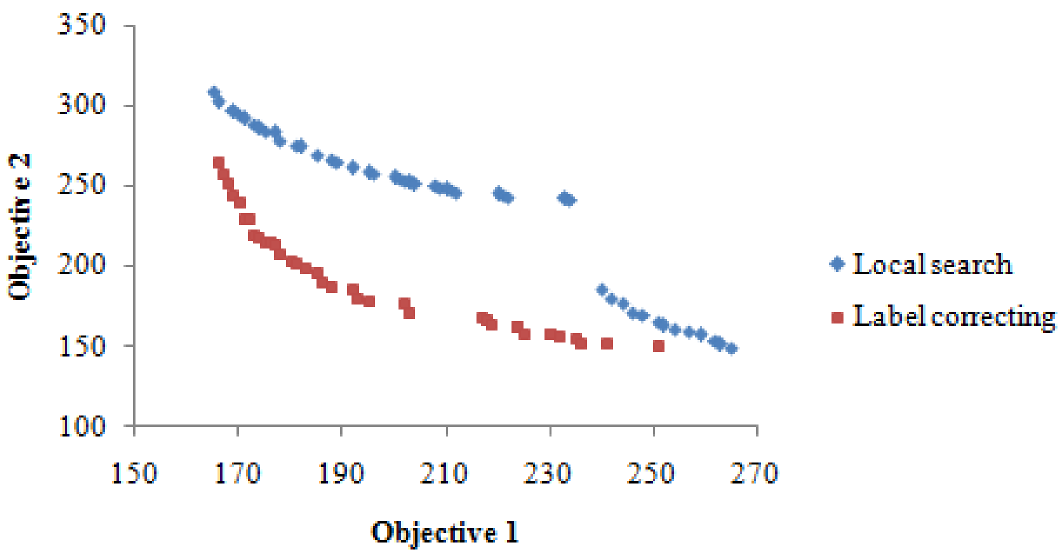


Fig. 5.2 The approximate Pareto front generated by the local search algorithm for Instance G01

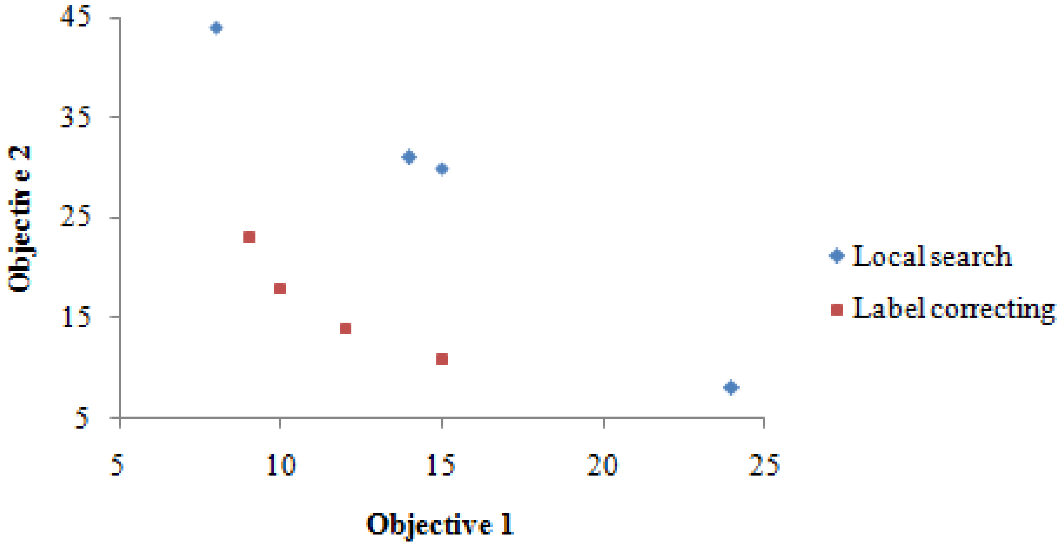


Fig. 5.3 The approximate Pareto front generated by the local search algorithm for Instance G16

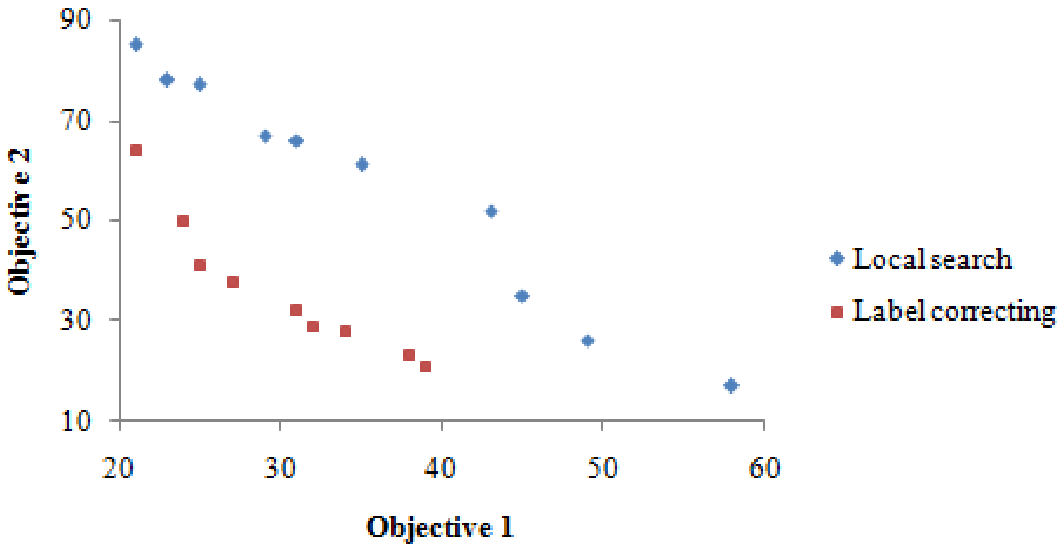


Fig. 5.4 The approximate Pareto front generated by the local search algorithm for Instance G17

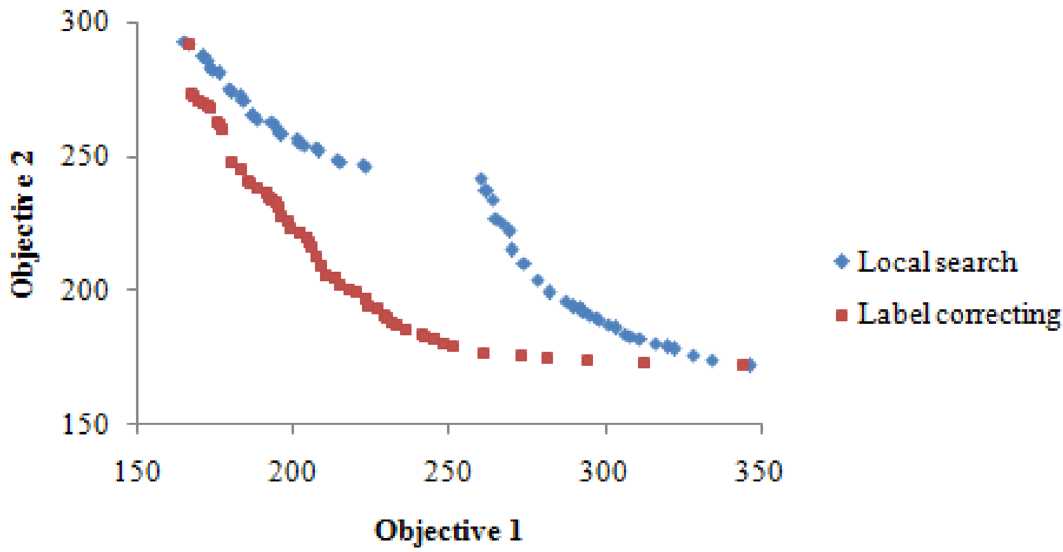


Fig. 5.5 The approximate Pareto front generated by the local search algorithm for Instance G21

5.4 Conclusions

In this chapter, two local search algorithms are proposed for the MSPP. They enable the MSPP to be solved much faster than when using labelling algorithms. We developed two moves for the MSPP, namely node replacement and node insertion which operate between two nodes in a route. Since the MSPP is known to be a data sensitive problem, each algorithm uses different moves or combinations of moves and is tested on data of different types of networks.

The contributions of this chapter against existing knowledge can be summarised as follows:

- We present two local search algorithms for the MSPP, for what we believe to be the first time in the literature, such that they will run fast. Two different types of neighbourhoods are used in the algorithms. Four parameters are used, the values of which are set through a fine-tuning process. The nature of the local search, though, is also such that it can be embedded in a metaheuristic algorithm for this type of problem.

- The contribution of this work is not limited to new solution algorithms for the MSPP. In the existing literature, one of the key aspects that has not yet been fully evaluated is the run-time and performance evaluation of heuristics for the MSPP as compared with the Pareto-optimal solutions obtained by exact methods. One main contribution of this chapter is that we describe and use four solution quality indicators as well as the run-time to evaluate the effectiveness of the local search algorithms.

- We provide computational experiments on randomly generated and grid network instances. Regarding the time requirements, the computational run-time does not increase exponentially as the size of network or the number of objectives is increased. Contrary to other heuristics found in the literature for the MSPP, our algorithms are capable of finding good-quality solutions and are also able to detect efficient solutions within short time scales.

- Another main contribution is that we provide a computational study of local search algorithms using different operators. Instances with different types of network are used to test the performance of different algorithms. A node insertion operator was used to help avoid local optima when finding the approximate set. Computational results show that algorithms using the node replacement and node insertion operators perform better on randomly generated instances. These two algorithms are not dominated by each other according to the four indicators and the run-time. Therefore, we can conclude that the node replacement operator performs well in avoiding local optima for randomly generated data. This effect can also be explained by the procedure of the node replacement operator; each time we remove a node, the operator chooses the replacement node from the whole range of networks as opposed to a local area network. An additional interesting phenomenon was observed: Algorithms using the node replacement operator do not perform well in the grid network instances. This can be explained by the grid characterisation of data; in grid networks, there is at most one replacement node, and the search procedure is therefore limited.

Chapter 6

Conclusions

6.1 Overview

In this thesis, we studied three research questions under two main and interdependent research topics. The first research topic concerns the service network design models for multicommodity multimode freight transportation, where for multiple commodities and a given set of demands, the model determines the routes, transport modes, and the flow distribution. The first research question related to modelling and solving the intermodal service network design problem, studied in Chapter 3. The second research topic is multiobjective shortest path problems, either without or with constraints (MSPP and CMSPP, respectively). The second and third research questions considered designing and testing exact and heuristic algorithms for the MSPP and the CMSPP, studied in Chapters 4 and 5. The next section highlights the important findings and specific contributions of the thesis. Limitations of presented techniques and methodologies are given in Section 6.3, followed by recommendations of areas for future research in Section 6.4.

6.2 Summary of the Main Contributions

To address the first research question, the thesis described time-invariant and time-dependent service network design models for intermodal freight transportation, where internal and external costs, including those of greenhouse gas (GHG) emissions, intermodal transfers, and inventories, where the later was captured in the time-dependent model, were minimised. Chapter 3 provided two ways to consider GHG emissions in intermodal freight transportation. One is to convert the GHG emissions to GHG cost, and incorporate it into the objective function. The other method is to design a bi-objective model, which enables the representation of competition between players. Additionally, the model we presented was, to the best of our knowledge, the first to explicitly include intermodal transfer cost in the objective in modelling a green intermodal transportation system. A hypothetical but realistic case study of the UK including 11 locations was presented for our investigation. Prior to this study, a comprehensive case study of green service network design models using both time-invariant and time-dependent data generated from the UK transportation network does not seem to have been conducted. Our data is available for use by other researchers to compare different methodologies to improve the design of green intermodal services. Another main finding of Chapter 3 on a realistic case study is that the total cost can be greatly reduced, between 11.3% and 41.2%, by using intermodal transportation as opposed to uni-modal transportation. Furthermore, in intermodal transportation, the GHG cost can be reduced by 6.94%–11.59% by designing models that explicitly account for emissions and transfer costs as opposed to models without these two costs.

To address the second research question, Chapter 4 first presented computational testing for the MSPP on different types of networks using a variety of labelling algorithms, including Martins' algorithm, the aggregate label setting algorithm, and label correcting algorithm. Even though these algorithms are not new methods for solving the MSPP, one contribution of this thesis is to report the computational comparison using these methods, for the first time, on the MSPP with three and more objectives. The computational results indicated that label correcting performs better than other labelling algorithms, followed by aggregate label setting algorithms and Martins' algorithm. In addition, Chapter 4 proposed two labelling algorithms, for what we believe to be the first time in the literature for the CMSPP, with any number of objectives and constraints. Computational experiments showed that label correcting algorithms consistently perform better than label setting algorithms. The computational run-time increased exponentially as the size of network or the number of objectives was increased, paralleling the results of the MSPP. Another important finding was that the solution times are largely reduced if the constraints and the objective vectors are correlated.

The fifth chapter of the thesis addressed the third research question and described heuristics for the MSPP. To overcome the run-time limitations of the existing heuristics for the MSPP in literature, namely genetic and ant-colony algorithms, we proposed two local search algorithms for the MSPP. Two operators, based on replacement and insertion, were developed for use within the local search algorithms. The node replacement operator was used in the first algorithm, while both operators were used in the second. Computational results showed that local search algorithms greatly outperformed the label correcting algorithm by yielding run-time savings of up to 2136%, and are faster than all previously described heuristic methods for the MSPP presented in the literature. The run-time is calculated using the instances that can be solved within two hours using label correcting algorithms. For the instances which solved more than two hours, the run-time savings will be even greater. It is well known that heuristics give no guarantee of the optimality of any solutions found. Therefore, one of the most obvious questions for any heuristic is how well it will perform. Chapter 5 is believed to be the first time in the literature that four performance indicators, along with the run-time, to evaluate the local search algorithms for the MSPP, have been presented. Results showed that between 33.20% and 79.07% efficient solutions were found when testing on randomly generated network instances, while testing on grid network instances, between 3.70% and 33.33% efficient solutions were found using local search algorithms. There was no dominance of other indicators between the two algorithms. This effect showed that local search algorithms are able to produce good quality solutions on both randomly generated and grid networks, but more efficient solutions were obtained by the random instances.

6.3 Limitations of the Research

It is acknowledged that there are some limitations of the research, which are detailed below:

1. Two external costs, namely those of GHG emissions and transfers, were considered in modelling the intermodal network design problem. However, there are other types of costs that would arise, such as those related to congestion and delay. The cost due to road and rail congestion might cause an increase in the total cost. Interested readers are referred to De Camargo et al. (2009) and Elhedhli and Hu (2005), who describe ways in which costs of the congestion in hub-and-spoke network design problems can be included in the objective function. This research has offered, as a starting point, mathematical models solved by mixed integer programming in which mainly GHG emissions are considered, but the modelling framework described here can also take congestion into account, although new and efficient solution methods might be needed for the resulting models.

2. It is well known that heuristics should be evaluated not only in general, but on instances with particular characteristics. Even though we presented more comprehensive computational results than other existing work in the literature to evaluate the performance of local search algorithms, we did not perform tests on 'any particular instance', such as a real network. However, it is very difficult to obtain all efficient solutions of a real network due to the fact that real networks are large, which is very difficult to solve. To our knowledge, none of the algorithms, both exact algorithms and heuristics in the literature, have tested the real data for the MSPP with three and more objectives.

3. All models and algorithms studied in this thesis assumed deterministic and static data in the underlying problems. When data exhibits stochasticity, as might be the case in travel times or demands, the remaining problems will be significantly more difficult to solve and will require new approaches.

6.4 Future Research Directions

In this section, we present an outlook into relevant future research:

- In order to design more realistic models, one may consider modelling and solving green service network design problems with congestion cost. This can also be applied to intermodal passenger transportation considering possible traffic delays on road and rail, as well as at intermodal yards.
- The question of whether other operators or heuristics can be devised to solve the MSPP more efficiently is still open. Even though genetic algorithms and ant colony algorithms were presented in the literature to solve the MSPP, and we presented local search algorithms for this class of problems, more work is needed in this area. Ehrgott and Gandibleux (2000, 2002, 2008) provide a framework of heuristics, metaheuristics and the combined approaches for multiobjective combinatorial optimization, which is a good starting point.
- Heuristic algorithms are also needed for the CMSPP, on which we are not aware of any existing work. The local search algorithms for the MSPP described in this thesis are able to obtain good quality solutions, and would be a starting point in designing heuristics for the CMSPP.

Appendix A

Computational Results with Demands Increased 10-fold

Tables A.1–A.3 list the capacity utilisation for three transport modes when f_t=£50, £100 and £150 with 10-fold of commodities in Table 3.2.

Table A.1 Comparison of capacity utilisation for three modes when f_t = £50 and demand is increased 10-fold

	Truck			Rail			Ship		
	No. of vehicles	Total tonnage	Capacity utilisation(%)	No. of vehicles	Total tonnage	Capacity utilisation(%)	No. of vehicles	Total tonnage	Capacity utilisation(%)
1	241	6970	99.73	207	79080	96.23	27	64370	80.27
2	245	7105	100.00	210	80624	96.71	25	62629	84.35
3	245	7105	100.00	211	80624	96.25	25	62614	84.33
4	289	8365	99.81	199	77805	98.48	28	63427	76.27
5	290	8380	99.64	200	77775	97.95	27	63355	79.01
6	292	8437	99.63	199	77718	98.37	27	63298	78.94
7	428	12370	99.66	188	73697	98.74	28	63120	75.90
8	430	12428	99.66	188	73630	98.65	27	63064	78.64
9	430	12443	99.78	188	73634	98.66	27	63092	78.68

Table A.2 Comparison of capacity utilisation for three modes when f_t = £100 and demand is increased 10-fold

	Truck			Rail			Ship		
	No. of vehicles	Total tonnage	Capacity utilisation(%)	No. of vehicles	Total tonnage	Capacity utilisation(%)	No. of vehicles	Total tonnage	Capacity utilisation(%)
1	0	0	N/A	225	86050	96.33	28	65670	78.97
2	0	0	N/A	229	87638	96.40	26	64100	83.01
3	0	0	N/A	229	87638	96.40	24	63042	88.44
4	10	279	96.21	220	85773	98.21	29	65583	76.14
5	11	308	96.55	220	85767	98.20	27	65211	81.32
6	10	279	96.21	221	85796	97.79	26	64625	83.69
7	22	627	98.28	217	85429	99.16	29	65671	76.25
8	39	1119	98.94	216	84935	99.05	27	65122	81.21
9	39	1119	98.94	216	84935	99.05	27	65122	81.21

Table A.3 Comparison of capacity utilisation for three modes when f_t=£150 and demand is increased 10-fold

	Truck			Rail			Ship		
	No. of vehicles	Total tonnage	Capacity utilisation(%)	No. of vehicles	Total tonnage	Capacity utilisation(%)	No. of vehicles	Total tonnage	Capacity utilisation(%)
1	0	0	N/A	225	86050	96.33	28	65670	78.97
2	0	0	N/A	229	87638	96.40	24	63042	88.44
3	0	0	N/A	229	87638	96.40	24	63042	88.44
4	5	134	92.41	220	86028	98.50	29	65865	76.47
5	12	337	96.84	219	85738	98.61	27	65324	81.46
6	5	134	92.41	220	86028	98.50	26	64530	83.57
7	15	424	97.47	217	85325	99.04	29	66285	76.96
8	16	453	97.63	217	85294	99.01	27	65919	82.20
9	16	453	97.63	217	85294	99.01	26	65125	84.34

Appendix B

Five Instances Used for the Time-Dependent Case Study in Chapter 3

Tables B.1–B.5 list the five sets of commodities, which are part of the data in the case study in Chapter 3. Each instance includes 30 commodities. The columns show its origin, departure time, destination, due time and the required amount, respectively.

Table B.1 Commodities for Instance #1

No.	Origin	Destination	Departure time	Due time	Required demand (tonne)
1	1	8	1	13	938
2	3	10	2	12	575
3	5	4	1	13	931
4	10	11	2	14	917
5	9	6	1	12	533
6	10	1	1	14	796
7	5	9	1	13	918
8	4	10	2	12	438
9	3	10	2	14	373
10	5	1	1	14	967
11	9	2	2	13	235
12	9	11	2	14	705
13	11	10	1	13	744
14	3	11	1	12	962
15	11	5	1	14	895
16	11	5	1	14	370
17	10	5	2	14	218
18	11	7	1	12	554
19	11	3	1	13	119
20	2	6	1	13	426
21	9	7	1	13	972
22	2	8	1	12	894
23	1	8	2	14	298
24	3	6	2	14	990
25	5	2	2	13	283
26	8	5	1	13	718
27	11	1	1	14	711
28	11	2	1	12	848
29	1	6	2	14	530
30	8	4	2	12	560

Table B.2 Commodities for Instance #2

No.	Origin	Destination	Departure time	Due time	Required demand (tonne)
1	2	11	1	14	143
2	1	4	1	14	308
3	11	1	2	13	764
4	2	9	2	13	612
5	2	3	2	13	170
6	11	7	1	12	726
7	3	7	2	14	334
8	4	6	1	12	648
9	2	8	1	13	428
10	7	2	1	12	832
11	4	9	2	13	221
12	1	9	1	13	190
13	8	5	2	13	108
14	6	4	2	12	840
15	11	2	2	14	802
16	7	6	2	13	469
17	11	4	1	13	463
18	10	7	2	13	600
19	4	2	2	13	162
20	8	6	2	13	281
21	5	8	1	14	231
22	8	11	1	14	560
23	11	6	2	14	728
24	2	7	2	14	735
25	4	3	1	13	488
26	8	6	1	13	402
27	9	1	2	12	265
28	1	7	1	12	245
29	8	10	1	14	323
30	10	1	1	13	505

Table B.3 Commodities for Instance #3

No.	Origin	Destination	Departure time	Due time	Required demand (tonne)
1	6	3	1	14	272
2	2	1	2	13	756
3	8	2	1	14	884
4	8	2	1	14	965
5	7	4	2	14	352
6	10	2	2	14	838
7	9	11	2	14	562
8	6	8	2	12	239
9	10	7	1	13	750
10	3	10	2	13	973
11	11	5	1	14	751
12	5	11	2	12	399
13	3	2	1	13	450
14	11	4	1	13	893
15	11	10	1	13	369
16	4	9	2	14	299
17	11	7	1	12	698
18	5	7	1	14	921
19	5	9	1	12	648
20	8	3	1	12	431
21	8	7	2	12	715
22	2	8	2	13	445
23	9	7	1	12	836
24	9	3	2	14	870
25	17	5	6	14	378
26	11	9	1	14	263
27	7	1	1	14	445
28	3	8	2	13	560
29	10	11	1	12	956
30	3	2	1	13	166

Table B.4 Commodities for Instance #4

No.	Origin	Destination	Departure time	Due time	Required demand (tonne)
1	9	11	2	13	170
2	3	11	2	13	939
3	3	9	1	14	319
4	5	4	1	13	548
5	5	11	2	12	600
6	10	11	2	12	150
7	2	8	1	14	362
8	6	3	2	14	338
9	11	10	1	13	964
10	4	7	1	13	578
11	1	8	2	14	932
12	4	11	2	14	456
13	10	1	2	14	237
14	9	5	2	13	538
15	11	4	1	14	290
16	11	10	2	13	389
17	4	8	2	12	338
18	2	4	2	14	213
19	6	8	1	13	334
20	5	1	2	14	151
21	11	4	1	14	779
22	19	9	8	14	688
23	1	7	1	14	983
24	6	1	2	13	249
25	7	1	1	12	126
26	4	2	2	13	617
27	7	9	1	12	474
28	4	7	2	13	708
29	11	9	2	12	774
30	2	1	1	13	214

Table B.5 Commodities for Instance #5

No.	Origin	Destination	Departure time	Due time	Required demand (tonne)
1	9	6	1	14	223
2	7	2	1	13	231
3	9	6	1	14	694
4	2	9	2	13	989
5	1	9	1	13	428
6	10	8	2	12	858
7	10	1	2	14	100
8	11	3	1	13	255
9	1	6	2	12	289
10	10	1	1	12	373
11	3	7	1	14	193
12	11	5	1	13	849
13	11	7	2	14	108
14	11	1	1	14	537
15	11	9	1	13	182
16	5	9	2	12	418
17	5	2	1	13	316
18	11	9	1	13	801
19	7	3	2	14	245
20	1	3	2	13	137
21	11	8	1	13	741
22	8	5	1	12	191
23	7	10	1	12	995
24	4	1	2	13	232
25	9	11	1	12	523
26	8	7	1	12	585
27	1	6	2	14	143
28	5	11	2	14	200
29	11	4	1	13	634
30	5	3	2	13	615

Appendix C

Computational comparisons between the MMND and the MMTND

Table C.1 Computational comparisons between the MMND and the MMTND

	Instance #1		Instance #2		Instance #3		Instance #4		Instance #5	
	MMND	MMTND	MMND	MMTND	MMND	MMTND	MMND	MMTND	MMND	MMTND
Total cost(£)	253023	264836	145308	157587	179355	192785	137757	155426	136081	149696
Fixed cost(£)	13100	14450	6750	13600	9600	9700	6950	11000	8400	12950
Variable cost(£)	220490	228966	129216	130090	157489	170287	122114	133045	117524	123839
Emission cost(£)	19433	21139	9341	13676	12266	12798	8693	11380	10157	12826
Transfer cost(£)	0	281	0	221	0	0	0	0	0	81
Total tonnage (truck)	3817	5705	1288	6247	2380	2850	1006	4318	2233	5883
Total tonnage (rail)	21886	14700	14740	8366	18254	15236	15710	10479	14693	7797
No. of vehicles (truck)	134	201	45	218	84	100	35	152	78	207
No. of vehicles (rail)	64	46	45	27	54	47	52	34	45	26
Capacity utilisation (truck)(%)	98.22	97.87	98.70	98.81	97.70	98.28	99.11	97.96	98.72	98.00
Capacity utilisation (rail)(%)	86.14	80.50	82.51	78.05	85.15	81.65	76.10	77.63	82.24	75.54

References

Alumur, S. and Kara, B. Y. (2008). Network hub location problems: The state of the art. *European Journal of Operational Research*, 190(1):1–21.

Andersen, J., Crainic, T. G., and Christiansen, M. (2009). Service network design with management and coordination of multiple fleets. *European Journal of Operational Research*, 193(2):377 – 389.

Andersson, E., Berg, M., Nelldal, B., and Fröidh, O. (2011). TOSCA: Rail freight transport : Techno-economic analysis of energy and greenhouse gas reductions. Technical report, Rail Vehicles, The KTH Railway Group, Traffic and Logistics.

Azadeh, A., Farahani, M. H., Eivazy, H., Nazari-Shirkouhi, S., and Asadipour, G. (2013). A hybrid meta-heuristic algorithm for optimization of crew scheduling. *Applied Soft Computing*, 13(1):158–164.

Azevedo, J. A. and Martins, E. Q. V. (1991). An algorithm for the multiobjective shortest path problem on acyclic networks. *Investigacao Operacional*, 11(1):52–69.

Balakrishnan, A., Magnanti, T. L., and Wong, R. T. (1989). A dual-ascent procedure for large-scale uncapacitated network design. *Operations Research*, 37(5):716–740.

Bauer, J., Bektaş, T., and Crainic, T. G. (2009). Minimizing greenhouse gas emissions in intermodal freight transport: an application to rail service design. *Journal of the Operational Research Society*, 61(3):530–542.

Bektaş, T. and Crainic, T. G. (2007). A brief overview of intermodal transportation. In Taylor, G. D., editor, *Logistics Engineering Handbook*. CRC Press, Boca Raton.

Bezerra, L., Goldbarg, E., Buriol, L., and Goldbarg, M. (2011). Grace: A generational randomized ACO for the multi-objective shortest path problem. In Takahashi, R., Deb, K., Wanner, E., and Greco, S., editors, *Evolutionary Multi-Criterion Optimization*, volume 6576 of *Lecture Notes in Computer Science*, pages 535–549. Springer Berlin Heidelberg.

Bezerra, L. C., Goldbarg, E. F., Goldbarg, M. C., and Buriol, L. S. (2013). Analyzing the impact of MOACO components: An algorithmic study on the multi-objective shortest path problem. *Expert Systems with Applications*, 40(1):345–355.

Bielli, M., Bielli, A., and Rossi, R. (2011). Trends in models and algorithms for fleet management. *Procedia - Social and Behavioral Sciences*, 20(0):4–18.

Brumbaugh-Smith, J. and Shier, D. (1989). An empirical investigation of some bicriterion shortest path algorithms. *European Journal of Operational Research*, 43(2):216–224.

Bruynooghe, M. (1972). An optimal method of choice of investments in a transport network. *PTRC Proceedings*.

Campbell, J. F., Ernst, A. T., and Krishnamoorthy, M. (2005a). Hub arc location problems: Part I: Introduction and results. *Management Science*, 51(10):1540–1555.

Campbell, J. F., Ernst, A. T., and Krishnamoorthy, M. (2005b). Hub arc location problems: Part II: Formulations and optimal algorithms. *Management Science*, 51(10):1556–1571.

Carlyle, W. M., Royset, J. O., and Wood, R. K. (2008). Lagrangian relaxation and enumeration for solving constrained shortest-path problems. *Networks*, 52(4):256–270.

Carraway, R. L., Morin, T. L., and Moskowitz, H. (1990). Generalized dynamic programming for multicriteria optimization. *European Journal of Operational Research*, 44(1):95–104.

Chen, P. and Nie, Y. (2013). Bicriterion shortest path problem with a general nonadditive cost. *Transportation Research Part B: Methodological*, 57(0):419–435.

Cherkassky, B. V., Goldberg, A. V., and Radzik, T. (1996). Shortest paths algorithms: theory and experimental evaluation. *Mathematical Programming*, 73(2):129–174.

Choong, S. T., Cole, M. H., and Kutanoglu, E. (2002). Empty container management for intermodal transportation networks. *Transportation Research Part E: Logistics and Transportation Review*, 38(6):423 – 438.

Clímaco, J., Craveirinha, J., and Pascoal, M. (2003). A bicriterion approach for routing problems in multimedia networks. *Networks*, 41(4):206–220.

Clímaco, J. C. N. and Martins, E. Q. V. (1982). A bicriterion shortest path algorithm. *European Journal of Operational Research*, 11(4):399–404.

Clímaco, J. C. N. and Pascoal, M. M. B. (2012). Multicriteria path and tree problems: discussion on exact algorithms and applications. *International Transactions in Operational Research*, 19(1-2):63–98.

Coello, C. A. C., Lamont, G. B., and Van Veldhuisen, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems*. Springer-Verlag New York, Inc., NJ, USA.

Cordeau, J., Gendreau, M., Laporte, G., Potvin, J., and Semet, F. (2002). A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, 53(5):512–522.

Cordeau, J., Laporte, G., Savelsbergh, M. W. P., and Vigo, D. (2007). Vehicle routing. In Barnhart, C. and Laporte, G., editors, *Transportation: Vol.14 of Handbooks on Operations Research and Management Science*, pages 347–428. Elsevier, Amsterdam.

Coutinho-Rodrigues, J. and Clímaco, J.C.N.and Current, J. (1999). An interactive bi-objective shortest path approach: searching for unsupported nondominated solutions. *Computers & Operations Research*, 26(8):789–798.

Crainic, T. G. (2000). Service network design in freight transportation. *European Journal of Operational Research*, 122(2):272–288.

Crainic, T. G. and Kim, K. H. (2007). Intermodal transportation. In Barnhart, C. and Laporte, G., editors, *Transportation, Handbooks in Operations Research and Management Science*, pages 467–537. Elsevier, North Holland, Netherland.

Crainic, T. G. and Laporte, G. (1997). Planning models for freight transportation. *European Journal of Operational Research*, 97(3):409–438.

Crainic, T. G. and Laporte, G. (1998). *Fleet management and logistics*. Kluwer Academic, London, Dordrecht, Boston.

Crainic, T. G. and Roy, J. (1992). Design of regular intercity driver routes for the ltl motor carrier industry. *Transportation Science*, 26(4):280–295.

Current, J. R. and Marsh, M. (1993). Multiobjective transportation network design and routing problems: Taxonomy and annotation. *European Journal of Operational Research*, 65(1):4–19.

Current, J. R., Revelle, C. S., and Cohon, J. L. (1990). An interactive approach to identify the best compromise solution for two objective shortest path problems. *Computers & Operations Research*, 17(2):187–198.

Dantzig, G. B. (1966). All shortest routes in a graph. In *Theory of Graphs: International Symposium, Rome*. Gordon and Breach, New York.

De Camargo, R. S., Miranda Jr, G., Ferreira, R. P. M., and Luna, H. P. (2009). Multiple allocation hub-and-spoke network design under hub congestion. *Computers & Operations Research*, 36(12):3097–3106.

de Lima Pinto, L., Bornstein, C. T., and Maculan, N. (2009). The tricriterion shortest path problem with at least two bottleneck objective functions. *European Journal of Operational Research*, 198(2):387–391.

de Lima Pinto, L. and Pascoal, M. M. (2010). On algorithms for the tricriteria shortest path problem with two bottleneck objective functions. *Computers & Operations Research*, 37(10):1774 – 1779.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197.

Dejax, P. J. and Crainic, T. G. (1987). Survey paper-a review of empty flows and fleet management models in freight transportation. *Transportation Science*, 21(4):227–248.

Deo, N. and Pang, C.-Y. (1984). Shortest-path algorithms: Taxonomy and annotation. *Networks*, 14(2):275–323.

Department for Transport (2005). Truck specifications for best operational efficiency. Technical report, Department for Transport, London, UK. available at: http://www.dft.gov.uk/rmd/project.asp?intProjectID=12043(accessed 28.11.2013).

Department for Transport (2012). Free flow speeds statistics. Technical report, Department for Transport, London, UK. available at: https://www.gov.uk/government/statistical-data-sets/spe01-vehicle-speeds(accessed 01.08.2014).

Department for Transport (2014). Transport statistics Great Britain 2014. Technical report, Department for Transport, London, UK. Available at: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/389592/tsgb-2014.pdf (accessed 30.03.2015).

Desaulniers, G., Desrosiers, J., Erdmann, A., Solomon, M. M., and Soumis, F. (2002). VRP with Pickup and Delivery. In Toth, P. and Vigo, D., editors, *The Vehicle Routing Problem*, Chapter 9, pages 225–242. Siam, Philadelphia.

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271.

Drezner, Z. and Hamacher, H. (2002). *Facility location: Applications and Theory*. Springer-Verlag Berlin Heidelberg, Berlin.

Dumitrescu, I. and Boland, N. (2003). Improved preprocessing, labeling and scaling algorithms for the weight-constrained shortest path problem. *Networks*, 42(3):135–153.

Ehrgott, M. and Gandibleux, X. (2000). A survey and annotated bibliography of multiobjective combinatorial optimization. *OR-Spektrum*, 22(4):425–460.

Ehrgott, M. and Gandibleux, X. (2002). Multiobjective combinatorial optimization — theory, methodology, and applications. In Ehrgott, M. and Gandibleux, X., editors, *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys*, volume 52 of *International Series in Operations Research & Management Science*, pages 369–444. Springer US.

Ehrgott, M. and Gandibleux, X. (2008). Hybrid metaheuristics for multi-objective combinatorial optimization. In Blum, C., Aguilera, M., Roli, A., and Sampels, M., editors, *Hybrid Metaheuristics*, volume 114 of *Studies in Computational Intelligence*, pages 221–259. Springer Berlin Heidelberg.

Elhedhli, S. and Hu, F. X. (2005). Hub-and-spoke network design with congestion. *Computers & Operations Research*, 32(6):1615–1632.

Farahani, R. Z., Hekmatfar, M., Arabani, A. B., and Nikbakhsh, E. (2013). Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering*, 64(4):1096–1109.

Faulkner, D. (2004). Shipping safety: a matter of concern. *Proceedings of IMarEST-Part B-Journal of Marine Design and Operations*, 2004(5):37–56.

Floyd, R. W. (1962). Algorithm 97: Shortest path. *Communications of the ACM*, 5(6):345.

Ford, L. R. (1956). Network flow theory. Technical report, Rand Corporation, Santa Monica, CA.

Forkenbrock, D. J. (2001). Comparison of external costs of rail and truck freight transportation. *Transportation Research Part A: Policy and Practice*, 35(4):321–337.

Fujimura, K. (1996). Path planning with multiple objectives. *IEEE Robotics and Automation Magazine*, 3(1):33–38.

Gabrel, V. and Vanderpooten, D. (2002). Enumeration and interactive selection of efficient paths in a multiple criteria graph for scheduling an earth observing satellite. *European Journal of Operational Research*, 139(3):533–542.

Gandibleux, X., Beugnies, F., and Randriamasy, S. (2006). Martins' algorithm revisited for multi-objective shortest path problems with a maxmin cost function. *4OR*, 4(1):47–59.

Gao, Z., Wu, J., and Sun, H. (2005). Solution algorithm for the bi-level discrete network design problem. *Transportation Research Part B: Methodological*, 39(6):479–495.

Garey, M. R. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., Sanfrancisco, USA.

Ghoseiri, K. and Nadjari, B. (2010). An ant colony optimization algorithm for the bi-objective shortest path problem. *Applied Soft Computing*, 10(4):1237–1246.

Ghoseiri, K., Szidarovszky, F., and Asgharpour, M. J. (2004). A multi-objective train scheduling model and solution. *Transportation Research Part B: Methodological*, 38(10):927–952.

Granat, J. and Guerriero, F. (2003). The interactive analysis of the multicriteria shortest path problem by the reference point method. *European Journal of Operational Research*, 151(1):103–118.

Guerriero, F. and Musmanno, R. (2001). Label correcting methods to solve multicriteria shortest path problems. *Journal of Optimization Theory and Applications*, 111(3):589–613.

Häckel, S., Fischer, M., Zechel, D., and Teich, T. (2008). A multi-objective ant colony approach for pareto-optimization using dynamic programming. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, GECCO '08, pages 33–40, New York, USA. ACM.

Handler, G. Y. and Zang, I. (1980). A dual algorithm for the constrained shortest path problem. *Networks*, 10(4):293–309.

Hansen, P. (1980). Bicriterion path problems. In Fandel, G. and Gal, T., editors, *Multiple Criteria Decision Making Theory and Application*, volume 177 of *Lecture Notes in Economics and Mathematical Systems*, pages 109–127. Springer Berlin Heidelberg.

Hekmatfar, M. and Farahani, R. Z. (2009). *Facility Location: Concepts, Models, Algorithms and Case Studies.* Physica-Verlag Heidelberg.

Holguín-Veras, J. and Jara-Dírz, S. (2006). Preliminary insights into optimal pricing and space allocation at intermodal terminals with elastic arrivals and capacity constraint. *Networks and Spatial Economics*, 6(1):25–38.

Holmberg, K. and Yuan, D. (2000). A Lagrangian heuristic based branch-and-bound approach for the capacitated network design problem. *Operations Research*, 48(3):461–481.

Hwang, C. and Masud, A. (1979). Multiple objective decision making. methods and applications: A state of the art survey.

Iori, M., Martello, S., and Pretolani, D. (2010). An aggregate label setting policy for the multi-objective shortest path problem. *European Journal of Operational Research*, 207(3):1489–1496.

Irnich, S. and Desaulniers, G. (2005). Shortest path problems with resource constraints. In Desaulniers, G., Desrosiers, J., and Solomon, M., editors, *Column Generation*, pages 33–65. Springer US.

Ishibuchi, H. and Murata, T. (1998). A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, 28(3):392–403.

Janic, M. (2007). Modelling the full costs of an intermodal and road freight transport network. *Transportation Research Part D: Transport and Environment*, 12(1):33–44.

Janic, M. and Bontekoning, Y. (2002). Intermodal freight transport in Europe: an overview and prospective research agenda. *Proceedings Focus Group*, 1:1–21.

Kerbache, L. and Smith, J. M. (2000). Multi-objective routing within large scale facilities using open finite queueing networks. *European Journal of Operational Research*, 121(1):105–123.

Klose, A. and Drexl, A. (2005). Facility location models for distribution system design. *European Journal of Operational Research*, 162(1):4–29.

Lane, B. (2006). Life cycle assessment of vehicle fuels and technologies. Technical report, Ecolane Transport Consultancy, Bristol UK. http://www.ecolane.co.uk/content/dcs/Camden_LCA_Report_FINAL_10_03_2006.pdf.

Laporte, G. (2007). What you should know about the vehicle routing problem. *Naval Research Logistics*, 54(8):811–819.

Laporte, G., Nickel, S., and Saldanha da Gama, F., editors (2015). *Location Science*. Springer International Publishing, Switzerland.

Lin, L. and Gen, M. (2007). A bicriteria shortest path routing problems by hybrid genetic algorithm in communication networks. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 4577–4582.

Lizárraga, G., Hernández, A., and Botello, S. (2008). A set of test cases for performance measures in multiobjective optimization. In Gelbukh, A. and Morales, E., editors, *MICAI 2008: Advances in Artificial Intelligence*, volume 5317 of *Lecture Notes in Computer Science*, pages 429–439. Springer Berlin Heidelberg.

Loui, R. P. (1983). Optimal paths in graphs with stochastic or multidimensional weights. *Communications of the ACM*, 26(9):670–676.

Macharis, C. and Bontekoning, Y. M. (2004). Opportunities for or in intermodal freight transport research: A review. *European Journal of Operational Research*, 153(2):400–416.

Machuca, E., Mandow, L., Pérez, J., and Ruiz-Sepulveda, A. (2012). A comparison of heuristic best-first algorithms for bicriterion shortest path problems. *European Journal of Operational Research*, 217(1):44–53.

Magnanti, T. L. and Wong, R. T. (1984). Network design and transportation planning: Models and algorithms. *Transportation Science*, 18(1):1–55.

Mandow, L. and Pérez, J. L. (2008). Multiobjective A* search with consistent heuristics. *Journal of the ACM*, 57(5):27:1–27:25.

Martins, E. Q. V. (1984). On a multicriteria shortest path problem. *European Journal of Operational Research*, 16(2):236–245.

Mavrotas, G. (2009). Effective implementation of the ϵ-constraint method in multi-objective mathematical programming problems. *Applied Mathematics and Computation*, 213(2):455–465.

McKinnon, A. and Piecyk, M. (2010). Measuring and managing co_2 emissions in European chemical transport. Technical report, Logistics Research Centre, Heriot-Watt University, Edinburgh,UK.

Miettinen, K. (1999). *Nonlinear multiobjective optimization*. Springer, New York, US.

Modesti, P. and Sciomachen, A. (1998). A utility measure for finding multiobjective shortest paths in urban multimodal transportation networks. *European Journal of Operational Research*, 111(3):495–508.

Mote, J., Murthy, I., and Olson, D. L. (1991). A parametric approach to solving bicriterion shortest path problems. *European Journal of Operational Research*, 53(1):81–92.

Murthy, I. and Olson, D. L. (1994). An interactive procedure using domination cones for bicriterion shortest path problems. *European Journal of Operational Research*, 72(2):417–431.

Network Rail (2007). Freight: Route utilisation strategy. Technical report, Network Rail, London.

Oliver, J., Janssens-Maenhout, G., and Peters, J. (2012). *Trends in global CO_2 emissions: 2012 report*. PBL Netherlands Environmental Assessment Agency, Netherlands.

Paixão, J. M. and Santos, J. L. (2013). Labeling methods for the general case of the multi-objective shortest path problem – a computational study. In Madureira, A., Reis, C., and Marques, V., editors, *Computational Intelligence and Decision Making*, volume 61 of *Intelligent Systems, Control and Automation: Science and Engineering*, pages 489–502. Springer Netherlands.

Pangilinan, J. M. A. and Janssens, G. (2007). Evolutionary algorithms for the multi-objective shortest path problem. *International Journal of Applied Science, Engineering and Technology*, 4(1):205–210.

Park, D., Kim, N. S., Park, H., and Kim, K. (2012). Estimating trade-off among logistics cost, CO_2 and time: A case study of container transportation systems in Korea. *International Journal of Urban Sciences*, 16(1):85–98.

Pirlot, M. (1996). General local search methods. *European Journal of Operational Research*, 92(3):493–511.

Przybylski, A., Gandibleux, X., and Ehrgott, M. (2010). A two phase method for multi-objective integer programming and its application to the assignment problem with three objectives. *Discrete Optimization*, 7(3):149 – 165.

Qu, Y., Bektaş, T., and Bennell, J. (2014). Sustainability SI: Multimode multicommodity network design model for intermodal freight transportation with transfer and emission costs. *Networks and Spatial Economics*. (in press).

Raith, A. and Ehrgott, M. (2009). A comparison of solution strategies for biobjective shortest path problems. *Computers & Operations Research*, 36(4):1299–1331.

Reeves, C. R., editor (1993). *Modern Heuristic Techniques for Combinatorial Problems*. John Wiley & Sons, Inc., New York, NY, USA.

Reinhardt, L. B. and Pisinger, D. (2011). Multi-objective and multi-constrained non-additive shortest path problems. *Computers & Operations Research*, 38(3):605–616.

Royset, J. O., Carlyle, W. M., and Wood, R. K. (2009). Routing military aircraft with a constrained shortest-path algorithm. *Military Operations Research*, 14(3):31 – 52.

Santos, L., Coutinho-Rodrigues, J., and R., C. J. (2007). An improved solution algorithm for the constrained shortest path problem. *Transportation Research Part B: Methodological*, 41(7):756–771.

Sastry, V. N., Janakiraman, T. N., and Mohideen, S. I. (2005). New polynomial time algorithms to compute a set of pareto optimal paths for multi-objective shortest path problems. *International Journal of Computer Mathematics*, 82(3):289–300.

Skriver, A. and Andersen, K. (2000). A label correcting approach for solving bicriterion shortest-path problems. *Computers & Operations Research*, 27(6):507–524.

Skriver, A. J. (2000). A classification of bicriterion shortest path (bsp) algorithms. *Asia Pacific Journal of Operational Research*, 17(2):199–212.

SteadieSeifi, M., Dellaert, N., Nuijten, W., Woensel, T. V., and Raoufi, R. (2014). Multimodal freight transportation planning: A literature review. *European Journal of Operational Research*, 233(1):1–15.

Steenbrink, P. A. (1974). Transport network optimization in the Dutch integral transportation study. *Transportation Research*, 8(1):11–27.

Stewart, B. and White, III, C. (1991). Multiobjective A*. *Journal of the ACM*, 38(4):775–814.

The World Bank (2012). Carbon finance unit. Technical report, The World Bank, Washington DC, USA. available at: http://web.worldbank.org/WBSITE/EXTERNAL/TOPICS/ENVIRONMENT/EXTCARBONFINANCE/0,,contentMDK:21848927~menuPK:4125939~pagePK:64168445~piPK:64168309~theSitePK:4125853,00.html(accessed 28.11.2013).

Toth, P. and Vigo, D. (2014). *Vehicle Routing: Problems, Methods, and Applications*, volume 18. SIAM.

Treitl, S., Nolz, P., and Jammernegg, W. (2012). Incorporating environmental aspects in an inventory routing problem. a case study from the petrochemical industry. *Flexible Services and Manufacturing Journal*, 26(1-2):143–169.

Tsaggouris, G. and Zaroliagis, C. (2004). Non-additive shortest paths. In Albers, S. and Radzik, T., editors, *Algorithms-ESA 2004*, volume 3221 of *Lecture Notes in Computer Science*, pages 822–834. Springer Berlin Heidelberg.

Tung, C. T. and Chew, K. L. (1992). A multicriteria pareto-optimal path algorithm. *European Journal of Operational Research*, 62(2):203–209.

Ulungu, E. and Teghem, J. (1995). The two phases method: An efficient procedure to solve bi-objective combinatorial optimization problems. *Foundations of Computing and Decision Sciences*, 20(2):149 – 165.

Vidović, M., Zečević, S., Kilibarda, M., Vlajić, J., Bjelić, N., and Tadić, S. (2011). The p-hub model with hub-catchment areas, existing hubs, and simulation: A case study of Serbian intermodal terminals. *Networks and Spatial Economics*, 11(2):295–314.

Villeneuve, D. and Desaulniers, G. (2005). The shortest path problem with forbidden paths. *European Journal of Operational Research*, 165(1):97–107.

Winebrake, J. J., Corbett, J. J., Falzarano, A., Hawker, J. S., Korfmacher, K., Ketha, S., and Zilora, S. (2008a). Assessing energy, environmental, and economic tradeoffs in intermodal freight transportation. *Journal of the Air & Waste Management Association*, 58(8):1004–1013.

Winebrake, J. J., Corbett, J. J., Hawker, J. S., and Korfmacher, K. (2008b). *Intermodal Freight Transport in the Great Lakes: Development and Application of a Great Lakes Geographic Intermodal Freight Transport Model.* Great Lakes Maritime Research Institute, Rochester, NY.

Yang, H. and Bell, M. G. H. (1998). Models and algorithms for road network design: a review and some new developments. *Transport Reviews*, 18(3):257–278.

Zhan, F. B. and Noon, C. E. (1998). Shortest path algorithms: an evaluation using real road networks. *Transportation Science*, 32(1):65–73.

Zhang, X.and Zhang, Y., Hu, Y., Deng, Y., and Mahadevan, S. (2013). An adaptive amoeba algorithm for constrained shortest paths. *Expert Systems with Applications*, 40(18):7607–7616.

Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195.

Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *Evolutionary Computation, IEEE Transactions on*, 3(4):257–271.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., and da Fonseca, V. (2003). Performance assessment of multiobjective optimizers: an analysis and review. *Evolutionary Computation, IEEE Transactions on*, 7(2):117–132.

Zografos, K. G. and Regan, A. C. (2004). Current challenges for intermodal freight transport and logistics in Europe and the United States. *Transportation Research Record: Journal of the Transportation Research Board*, 1873:70–78.