Multi-objective Routing Optimization Using Evolutionary Algorithms

Halil Yetgin, Kent Tsz Kan Cheung and Lajos Hanzo
School of ECS, University of Southampton, SO17 1BJ, United Kingdom.
Tel: +44-23-8059 3125, Fax: +44-23-8059 4508
Email: {hy3g09,tktc106,lh}@ecs.soton.ac.uk, http://www-mobile.ecs.soton.ac.uk

Abstract—Wireless ad hoc networks suffer from several limitations, such as routing failures, potentially excessive bandwidth requirements, computational constraints and limited storage capability. Their routing strategy plays a significant role in determining the overall performance of the multi-hop network. However, in conventional network design only one of the desired routing-related objectives is optimized, while other objectives are typically assumed to be the constraints imposed on the problem. In this paper, we invoke the Non-dominated Sorting based Genetic Algorithm-II (NSGA-II) and the Multi-Objective Differential Evolution (MODE) algorithm for finding optimal routes from a given source to a given destination in the face of conflicting design objectives, such as the dissipated energy and the end-to-end delay in a fully-connected arbitrary multi-hop network. Our simulation results show that both the NSGA-II and MODE algorithms are efficient in solving these routing problems and are capable of finding the Pareto-optimal solutions at lower complexity than the 'brute-force' exhaustive search, when the number of nodes is higher than or equal to 10. Additionally, we demonstrate that at the same complexity, the MODE algorithm is capable of finding solutions closer to the Pareto front and typically, converges faster than the NSGA-II algorithm.

I. INTRODUCTION

Routing is one of the most important design issues of multi-hop wireless networks, which has a significant impact on their achievable performance. Hence, efficient routing techniques should be designed for ensuring that the data packets propagate in an 'optimal' manner in terms of several metrics, such as energy consumption, delay, delay jitter, bandwidth and packet loss ratio. In conventional multi-hop networks, only one of the desired objectives is optimized, whereas other objectives are assumed to be constraints of the problem [1]. Nonetheless, in some practical applications finding multiple solutions, each of which is optimal in terms of a single metric may be better than finding a single meritorious solution, which strikes a trade-off amongst several conflicting factors. The way to strike a meritorious trade-off is to consider the multiple design objectives as the components of a single aggregate objective function, for example, using the weighted linear sum based method. However, choosing the most appropriate weights for the various constituent design objectives requires careful consideration depending on the specific importance of the individual design objectives. Naturally, this method only provides a single solution per simulation [2]. In certain applications Multi-objective Optimization (MO) algorithms that provide several optimal solutions may be preferred, since these methods do not necessarily require user-defined objective weights. Furthermore, the drawback of focusing on a single design objective, whilst ignoring other important objectives may be circumvented by multi-objective optimization techniques [1]. We can consider all the objectives simultaneously and generate a set of optimal solutions, which are known as the Pareto solutions [3] of multi-objective problems. However, finding optimal routes for multiple objectives in networks is a NP-Complete problem [4]. Hence, there is a need for efficient heuristic search algorithms based on reduced-complexity Evolutionary Algorithms (EAs) [5].

There is a paucity of contributions in the literature on the issue of multi-objective optimization conceived for routing related issues in multi-hop networks. Nonetheless, for example, Camelo et al. [2] proposed a new method for solving routing problems by considering the Quality of Service (QoS) in Wireless Mesh Networks (WMNs). In this study, the Non-dominated Sorting based Genetic Algorithm-II (NSGA-II) was invoked for finding different alternatives for guaranteeing the QoS requirements in both Voice over Internet Protocol (VoIP) and file transfer. Kotecha and Popat [4] employed a multi-objective Genetic Algorithm (GA) for improving the QoS in Mobile Ad-hoc NETworks (MANETs). Their solution considered the bandwidth, the delay, the traffic emanating from neighboring nodes and the number of hops. They demonstrated that the proposed GA-aided QoS-based routing protocol performs better than conventional methods, such as Ad-hoc On-demand Distance Vector (AODV) and QoS-AODV. Xue et al. [6] conceived a novel multi-objective differential evolution algorithm for routing problems in multi-hop networks by simultaneously optimizing both the delay and the energy consumption. Cui et al. [7] studied MO relying on GAs for determining specific routes, which satisfied different QoS requirements, with a special emphasis on how close their algorithm was capable of approaching the global minimum of each objective, as the number of nodes in the network increased.

Additionally, there are comprehensive studies of MO problems that investigate different aspects of Wireless Sensor Networks (WSNs) [5], [8], [9]. For example, Masazade et al. [1] studied the distributed detection problem and the choice of appropriate sensor thresholds, which was also solved with the aid of MO. Minimizing both the error probability and energy consumption of the network were the two objectives the authors focused on. Martins et al. [10] proposed a hybrid multi-objective optimization algorithm for improving the performance of WSNs. This algorithm provided a solution for the Dynamic Coverage and Connectivity Problem (DCCP) in WSNs subjected to node failures. They concluded that the MO approach provides an attractive solution for extending the network’s battery life-time at a slight degradation of the network’s coverage. Perez et al. [11] described a MO model conceived for jointly minimizing both the number of sensors employed and the total energy dissipation of the sensor network, which allowed them to minimize the total deployment cost. Jin et al. [12] investigated the so-called redundant overlapping coverage problem in terms of the network’s high-quality coverage and the power consumption of the network with the aid of a Multi-Objective Differential Evolution (MODE) algorithm.

Although there are numerous examples of employing Multi-Objective Evolutionary Algorithms (MOEAs), to the best of the authors’ knowledge, no comparative study exists between the different algorithms conceived for the same network routing problem. Under this premise, we investigate two MO algorithms. The first is based on the above-mentioned NSGA-II, while the second one is the Multi-Objective Differential Evolution (MODE) algorithm [6], [13]. We employ both algorithms for jointly optimizing the delay and energy consumption of a fully connected network relying on randomly distributed nodes. Furthermore, we compare both algorithms in terms
of the proximity of the solutions to the actual solutions termed as
the Pareto front [3]. Furthermore, we evaluate their complexities
and their rates of convergence, as the number of nodes in the
network increases. We demonstrate that at the same complexity, the
MODE algorithm provides solutions approaching the true Pareto front
more closely than the NSGA-II, and in general exhibits a higher
convergence rate. Additionally, we demonstrate that both algorithms
require substantially less cost-function evaluations to approximate the
true Pareto front for networks of 10 or more nodes, when compared
to an exhaustive search method.

The rest of this paper is organized as follows. In Section II we
describe our network model and quantify the fitness of each routing
solution. In Section III, we illustrate the ideas behind multi-objective
optimization and describe our NSGA-II and MODE algorithms con-
deceived for network routing. In Section IV, we describe our simulation
optimization problem and describe our NSGA-II and MODE algorithms con-
ceived for network routing. In Section IV, we describe our simulation
setup and present our results. Finally, we conclude in Section V.

II. SYSTEM MODEL

We consider a fully-connected network having a single source
and a single destination in a 100x100m² area, where all the nodes
are stationary and randomly distributed according to the uniform
distribution. They can communicate bidirectionally on the same
shared wireless channel. Each node has a unique identifier for its
transmitter and receiver. The Source Node (SN) and Destination
Node (DN) are placed at opposite corners. Transmission between
two nodes is assumed to incur an energy cost of \( E_{i,j} \), which is
proportional to \((d_{i,j})^\alpha\), where \( d_{i,j} \) is the Euclidean distance between
the nodes \( i \) and \( j \) (\( i \neq j \)), while \( \alpha \) is the path-loss exponent, which
is dependent on the propagation environment. We assume that all
other network operations incur a negligible energy consumption cost.
Furthermore, the \( ith \) node has a queue length of \( Q_i \) packets, and we
assume a propagation delay of one time unit. Therefore, given a link
between nodes \( i \) and \( j \), the total transmission delay \( D_{i,j} \) is given by
\( D_{i,j} = Q_i + 1 \), as shown in Fig. 1.

Given these assumptions, we can now describe the ‘cost’ of a particular
N-hop route formulated as, \( R_e = n_0, n_1, \ldots, n_{N-2}, n_{N-1} \),
which is a vector containing both the aggregate energy consumption,
\( E_t \) and the delay, \( D_t \), of each link along the route given by

\[
E_t = \sum_{i=0}^{N-2} E_{i,i+1}, D_t = \sum_{i=0}^{N-2} (Q_i + 1).
\]

We also stipulate the idealized simplifying assumption that our
routing protocols have global knowledge of the network, so that the
SN can evaluate the cost of each potential route leading to the DN.

III. MULTI-OBJECTIVE OPTIMIZATION

Real-world optimization problems usually have to meet multiple
objectives to obtain an attractive solution. Therefore, here multi-
objective optimization is used, since it produces a set of solutions for
conflicting objective functions. By contrast, in conventional single-
objective optimization we find the global optimum by satisfying a
single-component objective function. A multi-objective problem can
be formulated as follows [14]:

\[
\min f(x) = [f_1(x), f_2(x), f_3(x), \ldots, f_n(x)]:
\]

s.t. \( g_j(x) \leq 0, j = 1, \ldots, n_g \),

where \( x = [x_1, x_2, x_3, \ldots, x_N]^T \in \mathbb{R}^N \) is the vector of variables
that has to be optimized, \( f_i, i = 1, \ldots, n_f \), are the Objective
Functions (OFs) and \( g_j, j = 1, \ldots, n_g \), are the constraint functions.

Multi-objective optimization problems can be solved using the
Pareto optimality technique, which was proposed by Edgeworth
and Pareto [15], as defined below for the case of a minimization
problem [16].

Definition 1. (Pareto dominance²) A particular solution vec-
tor \( f(x_1) = [f_1(x_1), f_2(x_1), f_3(x_1), \ldots, f_n(x_1)] \) is said
to dominate another particular solution vector \( f(x_2) = [f_1(x_2), f_2(x_2), f_3(x_2), \ldots, f_n(x_2)] \) if and only if \( f(x_1) \preceq f(x_2) \), i.e. we have \( \forall i \in 1, \ldots, n_f, f_i(x_1) \leq f_i(x_2) \) ∧ \( \exists i \in
1, \ldots, n_f : f_i(x_1) < f_i(x_2) \) where \( n_f \) is the number of OFs in the
optimization problem.

To elaborate further, let us consider three solution vectors \( f(x_1), \)
\( f(x_2) \) and \( f(x_3) \), given that \( f(x_1) = [2.4, 2.3, 2.3, 2.1, 0.3] \)
and \( f(x_2) = [0.8, 1.6] \). From the definition of Pareto dominance, we
can say that \( f(x_2) \) dominates \( f(x_1) \), since each element of \( f(x_2) \)
is unambiguously lower than the corresponding element of \( f(x_1) \).
However, \( f(x_2) \) neither dominates nor it is dominated by \( f(x_3) \),
since we have \( f_1(x_3) < f_2(x_2) \), but \( f_2(x_3) > f_2(x_2) \).

Definition 2. (Pareto optimality) A particular solution \( f(x_1) \) is said
to be Pareto optimal, if and only if there is no \( f(x) \) for which \( f(x_1) \)
is dominated by \( f(x) \).

For the three solution examples given above, solutions \( f(x_2) \) and
\( f(x_3) \) are both Pareto optimal, since they are not dominated by any
other solution. In this case, they are said to lie on the Pareto front
of the objective space. Additionally, solution \( f(x_1) \) is dominated,
²The terminology of ‘dominance’ is a natural one in the context of a
maximization problem. However, when aiming for finding the minimum in an
optimization problem, a ‘dominant’ solution is one, which is associated with
a lower OF value.
and the relationships between $f(x_1)$, $f(x_2)$ and $f(x_3)$ are shown in Fig. 2. The aim of multi-objective optimization is to generate a diverse set of Pareto-optimal solutions so that the user can evaluate the trade-offs between the different objectives.

### A. Non-dominated Sorting based Genetic Algorithm-II

In genetic algorithms, there is an initial population of candidate solutions, which are termed as individuals, and the performance of a solution in terms of a given objective function is quantified in terms of its fitness. These individuals are initialized to random values and after each iteration (generation) of the algorithm the values tend to converge upon those corresponding to better solutions given an OF by applying the genetic operators: crossover, mutation and selection, which help to diversify the values of the population and guide them towards higher-fitness solutions.

The NSGA-II is a multi-objective evolutionary optimization algorithm based on non-dominated sorting of the population, which was proposed by Deb et al. [3]. Each candidate solution (individual) represents a route emerging from a given SN and leading to a given DN. In Fig. 3, Parent$_1$ and Parent$_2$ represent a sequence of nodes for the fully-connected multi-hop network, assuming that $S$ is the SN and $D$ is the DN. The candidate solutions in the population are arbitrarily initialized to a random length, i.e. to a random number of hops, with the SN in the first position and the DN in the last position. Naturally, routing loops are prevented, since any individual associated with a loop would perform poorly both in terms of delay and energy consumption. Then, the population is sorted into fronts, $F_1, F_2, \ldots, F_N$, by ensuring that individuals in each preceding front dominate all individuals of all subsequent fronts, as shown in Fig. 2. Given this definition, individuals in the first front, $F_1$, belong to the Pareto front of the current solution set, where the dominated individuals representing longer and higher-energy routes are denoted by small symbols. Furthermore, the concept of 'crowding distance' [3] is introduced in NSGA-II, which is a measure of the Euclidean distance of an individual from its neighbors in the same front of the solution space corresponding to diverse or dissimilar routes. After each generation, the algorithm preserves the specific solutions having the highest crowding distance in each front, which ensures that a high grade of individual-diversity is maintained in the solution space, corresponding to routes having dissimilar attributes. The difference between NSGA-II and its predecessor, NSGA [17], is that the diversity of candidate solutions is maintained without requiring a user-selected parameter. For more details on NSGA-II, the readers are referred to [3], since we now focus our attention on the modified parts of NSGA-II in order to adapt the algorithm to our specific optimization problem. The classic GA operators, such as the crossover and mutation, assists us in evolving the individuals of a specific population by producing new individuals - i.e. new routes - from the existing ones in order to find improved solutions. The classic genetic operators are applied to the so-called parent solution pairs, which are obtained here by the so-called binary tournament selection [3]. We use a common-node, single-point crossover mask with a crossover probability of $P_c$ and highlight the reasoning behind this choice as follows. Routes are formed from the links between the SN and DN passing through intermediate nodes, and the cost of each route is dependent on the cost of the individual links themselves. Therefore, it is more logical to find nodes that are common for both parent routes to perform the crossover operation at, in order to ensure that all but one of the already established links in both parent routes are preserved. Given this method, the crossover operation intelligently searches through the candidate-solution space, rather than doing this haphazardly. For example, in Fig. 3, the common node between Parent$_1$ and Parent$_2$ is $B$, therefore it is chosen as the crossover point. Then Child$_1$ is created by the concatenation of the nodes leading up to and including $B$ from Parent$_1$ with the nodes following $B$ in Parent$_2$, and vice-versa for Child$_2$. Note that the SN and DN cannot be chosen as crossover points, since applying crossover to them does not provide any new solutions. In the NSGA-II algorithm, both tournament selection and crossover are applied twice to the current population of individuals in order to provide $N$ new solutions, if the original population size was $N$.

The mutation operator is applied to each new individual. For each node - excluding the SN and DN - mutation is applied with a probability of $P_m$. In our implementation of the mutation operator, there are three possible modifications, which occur with equal probability: node exchange, as well as node removal or insertion. In case of node exchange, the current node is exchanged for a randomly-selected node, as shown in Fig. 4, where node $A$ in the original individual is exchanged to node $K$. In node removal or node insertion, the current node is deleted from the route, or a new node is inserted before this node. For example, in Fig. 4, node $A$ is removed in the related example. In the 'inserting' example, node $E$ is inserted before node $A$. After mutation has been applied to each new individual, any potential routing loops are removed from each solution. In our fully-connected network, these GA operators are simple to implement, since any new route is valid. Our specific NSGA-II is summarized in Algorithm 1.
Algorithm 1 The NSGA-II algorithm adapted to our routing problem.

```
initialize(P₀);
for g := 1 to G_max do
    F = non-dominated-sort(Pₙ);
    F = calculate-crowding-distance(F);
    Rₙ = get-strongest-N-individuals(F);
    Sₙ = selection(Rₙ) ⋃ selection(Rₙ);
    Cₙ = crossover(Sₙ);
    Mₙ = mutation(Cₙ);
    Rₙ₊₁ = Pₙ ⋃ Rₙ
end for
```

B. MultiObjective Differential Evolution Algorithm

The philosophy of Differential Evolution (DE) is that of relying on the individuals in the current population for guiding the optimization process. In classic DE [18], the new individuals are created using the appropriately weighted difference between the current individuals of the specific generation. We follow a similar methodology to [6], [13], which used both the current population as well as the non-dominated solutions for generating new solutions. By contrast, we only use the non-dominated solutions, because applying common-node, single-point crossover using a solution also from the current population may produce a solution has no relation to either the original individual or the non-dominated solution. In order to generate new individuals, with reference to Fig. 2, we use the common-node, single-point crossover for each parent individual \( f(x₁) \) of the current population and a randomly selected node from the set \( Sᵢ \), which consists of individuals in the Pareto front that dominate \( f(x₁) \). If the set \( Sᵢ \) is empty, in other words \( f(x₁) \) lies on the Pareto front, then no crossover is performed. However, apart from the crossover process the rest of the algorithm operates like our implementation of NSGA-II in Algorithm 1.

IV. EXPERIMENTAL RESULTS

We generate a network supporting \( N \) random uniformly distributed nodes. Each node has a queue length of \( Q \) packets and each link has an energy cost, as previously detailed in Section II. For our simulations, a path-loss exponent of \( \alpha = 3 \) is selected. To simplify the analysis, we set the queue length of each node to zero, therefore the delay is equal to the number of hops, \( N_{hops} \) in units of time. For our MOEAs, the probability of crossover was set to \( P_c = 0.9 \), and the probability of mutation was \( P_m = 0.5 \).

A. True Pareto front

Let us now continue by invoking our MOEA for the optimization of the routing in networks of \( N \) nodes, \( N \in \{6, 8, 10, 12\} \). We used the true Pareto front, as the best-case performance lower-bound relying on an exhaustive search. For \( N \) nodes the number of distinct routes \( R_N \), excluding loops, is given by

\[
R_N = \sum_{n=0}^{N-2} \frac{(N-2)!}{(N-2-n)!},
\]

which is simply the summation of the total number of route permutations for each route having \( N_{hops} = n + 1 \) hops, given the total of \( N \) nodes. As it will be shown in Section IV-B, even for a modest-size network of \( N = 12 \) nodes the number of route evaluations becomes prohibitively high. For our randomly generated uniformly distributed nodes, the Pareto solutions obtained using exhaustive search are presented in Table I. These solutions may be used for evaluating the performance of our MOEAs, when employed within the same network deployments. As shown in Table I, in a small-size network, for example \( N = 6 \), there are less Pareto solutions, since the network is not dense enough to provide many attractive routing solutions. As the network size increases above \( N = 8 \), the number of Pareto solutions becomes generally higher, and furthermore, the achievable energy reduction is improved, since there is a higher diversity of routes available. For example, when considering \( N \in \{10, 12\} \) and increasing the number of hops \( N_{hops} \) from 1 to 2 provides an energy reduction of 75%. However, as the value of \( N_{hops} \) increases beyond 2, the potential energy reduction generally results in diminishing returns.

B. Performance evaluation of MOEAs

We present the results of applying our MOEAs in Fig. 5 for the optimization problem of \( N = 12 \). For these results, the number of individuals for both MOEAs is 48, whilst the number of generations is fixed to \( N_{gen} = 50 \), so that the difference in performance can be illustrated. It is clear from Fig. 5 that increasing \( N_{hops} \) reduces the energy cost, which is expected due to the reduced-distance-based relaying gain. However, the reduction in energy cost exhibits diminishing returns upon increasing the number of hops beyond \( N_{hops} > 4 \). Additionally, the end-to-end delay increases as the number of hops \( N_{hops} \) increases, and this represents a fundamental trade-off in this multi-hop network. At the same complexity, the MODE algorithm provides solutions that are closer to the true Pareto front than the NSGA-II for \( N_{hops} \geq 5 \), namely when the search space becomes larger. MODE is capable of searching through this increased solution-space more efficiently by combining each generation with the individuals in the current Pareto front, whereas NSGA-II uses only tournament selection, which may not find sufficiently meritorious individuals that would provide improved routes with the aid of crossover. The algorithms may also be compared on the basis of their average rates of convergence and complexities. We may define
the proximity of the solutions obtained to the true Pareto front in terms of the percentage difference between them. Let us define the set of Pareto solutions obtained from Section IV-A as $P_N$, which contains elements of $P_{N,h}$, where $N$ is the number of nodes in the network, while the subscript $h$ indicates the number of hops associated with that particular solution. Furthermore, let us define the current solutions in the MOEAs as $C_{N,h}$, which have the minimum energy cost. Then we may define the difference, $\Delta_{N,g}$, between the current solutions of the MOEAs and the true Pareto front for a $N$ node network as

$$\Delta_{N,g} = \frac{N-2}{\sum_{h=1}^{N-2} \frac{C_{N,h} - P_{N,h}}{P_{N,h}}}.$$  

As shown in Fig. 6, although both the NSGA-II and the MODE algorithms are capable of approaching the Pareto front, the rate of convergence for MODE is higher at $N = 12$ than that of the NSGA-II. However, as shown in Table II, the average number of cost function evaluations required for approaching the Pareto front within 1% are similar for both algorithms. This illustrates that although the convergence rate of MODE is initially higher, the number of generations needed for approaching the Pareto front is ultimately similar for both algorithms. We can also see that the number of Cost Function (CF) evaluations is significantly lower for the MOEAs than that of the Exhaustive Search (ES) method.

V. CONCLUSIONS

Multi-hop networks suffer from a higher end-to-end delay than their single-hop counterparts. Furthermore, since they rely on batteries and can be affected by long aggregate queuing times at each node along the route, minimizing their energy requirements is of prime importance. In this paper, we optimized the routing using the combined CF of Eq. 1 with the aid of novel multi-objective optimization algorithms in a fully-connected arbitrary multi-hop network in order to measure the performance of the NSGA-II and of the MODE algorithms, which were appropriately adapted to suit our routing problem. We demonstrated that both MODE and NSGA-II are applicable for these kind of routing problems, and that they are capable of finding the Pareto-optimal solutions at a significantly lower complexity than an exhaustive search method, when the number of nodes is higher than or equal to 10. Additionally, we demonstrated that, at the same complexity, the MODE algorithm is capable of finding solutions closer to the Pareto front and in general, converges faster than the NSGA-II. The Pareto optimal solutions we obtained from the results demonstrated that the conflicting multiple objectives may indeed be jointly optimized.

REFERENCES


