Solving Multi-objective Multicast Routing Problems by Evolutionary Multi-objective Simulated Annealing Algorithms with Variable Neighborhoods

Ying Xu[1,2] · Rong Qu[1]
[1] The Automated Scheduling, Optimisation and Planning (ASAP) Group
School of Computer Science, The University of Nottingham, Nottingham, NG8 1BB, UK
{ yxx| rxq }@cs.nott.ac.uk
[2] School of Computer and Communication, Hunan University
Changsha, 410082, CHINA

Abstract. This paper presents the investigation of an evolutionary multi-objective simulated annealing algorithm with variable neighborhoods to solve the multi-objective multicast routing problems in telecommunications. The hybrid algorithm aims to carry out a more flexible and adaptive exploration in the complex search space by using features of the variable neighborhood search to find more non-dominated solutions in the Pareto front. Different neighborhood strictures have been designed with regard to the set of objectives, aiming to drive the search towards optimising all objectives simultaneously. A large number of simulations have been carried out on benchmark instances and random networks with real world features including cost, delay and link utilisations. Experimental results demonstrate that the proposed evolutionary multi-objective simulated annealing algorithm with variable neighborhoods is able to find high quality non-dominated solutions for the problems tested. In particular, the variable neighborhood structures which are specifically designed for each objective significantly improved the performance of the proposed algorithm compared with variants of the algorithm with a single neighborhood.

Keywords *Multi-objective Optimisation, Evolutionary Multi-objective Simulated Annealing, Variable Neighborhood Search, Multicast Routing*

1. Introduction

In multimedia telecommunications, multicast routing aims to simultaneously transfer information from a source to a group of destinations in computer networks. A solution of a multicast routing problem (MRP) is a multicast tree which spans from the source node to all the destination nodes in the network. Due to the increasing demand of numerous multicast network applications including E-learning, E-commerce and video-conferencing, MRPs have attracted a large amount of research attention over the past decade. In real life applications, a range of Quality of Service (QoS) constraints have been defined, i.e. cost, bandwidth, link utilisation, delay variation, packet lost ratio and hop count, etc. The main goal of the QoS-based multicast routing is to efficiently allocate network resources, balance the network load, reduce congestion hot spots and provide adequate level of QoS for end users while satisfying all required constraints in the underlying computer networks.

When different conflicting objectives are considered simultaneously, MRPs become much more complicated multi-objective problems, as already recognised by many researchers (Roy *et al*, 2002; Cui *et al*, 2003; Roy and Das, 2004; Crichigno and Baran, 2004a; Crichigno and Baran, 2004b; Koyama *et al*, 2004; Diego and Baran, 2005; Li *et al*, 2007). A recent survey on multi-objective optimisation algorithms for a variety of MRPs has been given in (Fabregat *et al*, 2005).

The underlying model of MRPs is the Steiner tree problem in graphs (Hwang and Richards, 1992), one of the well known NP-hard combinatorial optimisation problems (Garey and Johnson, 1979). It is also known that the complexity of finding a feasible route with two independent path constraints is NP-hard (Chen and Nahrestedt, 1998). Due to their complexity and required computational expenses, QoS based MRPs have attracted an increasing research attention in both computer communications and operational research (Salama *et al*, 1997; Diot *et al*, 1997; Chen *et al*, 1998; Yeo *et al*, 2004; Oliveira *et al*, 2005; Masip-Bruin *et al*, 2006). In the rich literature of exact methods, heuristics and meta-heuristics, the majority of research on MRPs has been within the context of single-objective subject to certain constraints (Zhu *et al*, 1995; Guo and Matta, 1999; Haghighat *et al*, 2004; Kun *et al*, 2005; Skorin-Kapov and Kos, 2006; Qu *et al*, 2009).

In contrast to the traditional single-objective algorithms, recently some multi-objective algorithms have been developed for solving MRPs with more realistic features. Table 1 summarises the heuristic algorithms developed for multi-objective MRPs in the literature, categorised by the objectives and constraints considered in the work, and ordered by the year of the publication. Here we concern MRPs where only a single multicast tree is constructed. It can be seen that most multi-objective multicast routing algorithms have been developed based on evolutionary algorithms (Roy *et al*, 2002; Cui *et al*, 2003; Roy and Das, 2004; Crichigno and Baran, 2004a; Crichigno and Baran, 2004b; Koyama *et al*, 2004). Other population based algorithms, such as the ant colony optimisation algorithms in (Diego and Baran, 2005) and the hybrid genetic algorithm with particle swarm optimisation algorithm in (Li *et al*, 2007), have also been investigated on multi-objective MRPs in the literature.

Insert Table 1 somewhere here.

Various multi-objective simulated annealing algorithms have been investigated by researchers for solving a range of combinatorial optimisation problems (Sefarini, 1994; Czyzak and Jaszkiewicz, 1998; Ulungu et al, 1999; Haidine and Lehnert, 2008; Li and Landa-Silva, 2008). In a recent work (Li and Landa-Silva, 2008), a variant of such algorithm named Evolutionary Multi-Objective Simulated Annealing (EMOSA) has been designed and tested on a set of benchmark traveling salesman problem instances. A two-phase strategy in the EMOSA algorithm which tunes search directions and manages similar solutions has shown to be effective in comparison with other multi-objective simulated annealing algorithms.

In this work, we develop the first EMOSA algorithm based on the work in (Li and Landa-Silva, 2008) to solve real-life multi-objective MRPs. Due to the specific features of the multicast tree, which are quite different from those of traveling salesman problems, different neighborhood structures have been designed with respect to different objectives to improve the proposed algorithm. Four objectives are optimised simultaneously without a priori restriction, aiming to minimise (1) the cost, (2) the maximum end-to-end delay, (3) the maximum link utilisation, and (4) the average delay of the multicast tree. Compared with the traditional meta-heuristics with single neighborhood operator, our proposed VEMOSA algorithm (EMOSA with variable neighborhoods) is able to optimise different objectives simultaneously, contributing to advanced multi-objective search algorithms for a wider range of optimisation problems with different features and complex problem structures.

The remainder of this paper is organised as follows. In Section 2, a general definition of the multi-objective optimisation problem is introduced, based on which the multi-objective MRP is formally defined. Sections 3 and 4 present the proposed VEMOSA algorithm and experimental results. Finally, Section 5 concludes the paper.

2. The Multi-objective Multicast Routing Problem

2.1The Multi-objective Optimisation Problem

Multi-Objective Optimisation has become an important and challenging research topic due to the requirement of simultaneous optimisation of several conflicting objectives in many real world optimisation problems. It consists of finding a set of alternative solutions where multiple objectives are optimised at the same time while satisfying all the constraints (Burke and Kendall, 2005). A general multi-objective optimisation problem with *m* objectives and *r* restrictions can be defined as follows (Ulungu and Teghem, 1994):

$$\text{Optimise } F(x) = (f_1(x), f_2(x), \ldots, f_m(x)) \quad (1)$$
$$\text{s.t. } e(x) = (e_1(x), e_2(x), \ldots, e_r(x)) \geq 0$$

where $x = (x_1, x_2, \ldots, x_n) \in X$ is a vector of decision variables (a solution), X denotes the decision space of the set of feasible regions of the solution space; $F(x)$ is the image of x in the m-objective space given by the vector of m-objective functions $f_i(x)$, $i \in \{1, \ldots, m\}$. The set of restrictions $e_j(x) \geq 0$, $j \in \{0, \ldots, r\}$ determines the set of feasible solutions. In general, there is no unique optimal solution but a set of alternative solutions, none of which can be seen as superior to the others when all the objectives are taken into account. If X consists of a discrete set of solutions, then the problem defined in (1) is called the multi-objective combinatorial optimisation problem.

Without loss of generality, we assume hereinafter in a minimisation context, for any two decision vectors u, $v \in X$, u is said to dominate v, denoted by $u \prec v$, iff $f_i(u) \leq f_i(v)$ for $i \in \{1, \ldots, m\}$, and there exists at least one objective $j \in \{1, \ldots, m\}$ satisfying $f_j(u) < f_j(v)$. A solution x^* is said to be Pareto-optimal if no solution in X dominates x^*. The set of all Pareto-optimal solutions in X is called the Pareto-optimal set. The objective vectors of all solutions in the Pareto-optimal set in X is called the Pareto-optimal front (PF). Many meta-heuristic algorithms, including genetic algorithms, simulated annealing, tabu search, particle swarm optimisation and ant colony optimisation, have been investigated to tackle different multi-objective optimisation problems (Landa-Silva *et al*, 2004; Gandibleux *et al*, 2004; Konak *et al*, 2006; Liu *et al*, 2006). The majority of research in this area has focused on generating a set of non-dominated Pareto optimal solutions (Ulungu and Teghem, 1994; Ehrgott and Gandibleux, 2000).

In this work, we focus on investigating the first simulated annealing based multi-objective approaches for multi-objective MRPs. An excellent survey of multi-objective simulated annealing algorithms for solving both single and multi-objective optimisation problems and their performance comparisons can be found in (Suman and Kumar, 2006).

2.2. The Multi-objective Multicast Routing Problem Formulation

By using a computer network topology, we model the MRP by a directed graph $G = (V, E)$ with $|V| = n$ nodes and $|E| = l$ links. The rest of the paper uses the following notations:

$(i, j) \in E$: the link from node i to node j, $i, j \in V$.

$c_{ij} \in R^+$: the cost of link (i, j) .

$d_{ij} \in R^+$: the delay of link (i, j) .

$z_{ij} \in R^+$: the capacity of link (i, j), measured in Mbps.

$t_{ij} \in R^+$: the current traffic of link (i, j), measured in Mbps.

$s \in V$: the source node of a multicast group.

$R \subseteq V - \{s\}$: the set of destinations of a multicast group.

$r_d \in R$: the destinations in a multicast group.

$|R|$: The cardinality of R, i.e. the number of destinations, also called group size.

$\phi \in R^+$: the traffic demand (bandwidth requirement) of a multicast request, measured in Mbps.

$T(s, R)$: the multicast tree with the source node s spanning all destinations $r_d \in R$.

$p_T(s, r_d) \subseteq T(s, R)$: the path connecting the source node s and a destination $r_d \in R$.

$d(p_T(s, r_d))$: the delay of path $p_T(s, r_d)$, given by: $d(p_T(s, r_d)) = \sum_{(i,j) \in p_T(s, r_d)} d_{ij} \quad r_d \in R$.

Using the above definitions, in this paper a MRP is defined as a multi-objective optimisation problem that consists of finding a multicast tree while minimising the following four objectives (Crichigno and Baran, 2004a):

The cost of the multicast tree:

$$C(T) = \phi \cdot \sum_{(i,j)\in T} c_{ij} \tag{2}$$

The maximal end-to-end delay of the multicast tree:

$$DM(T) = Max\{d(p_T(s, r_d))\},\ r_d \in R \tag{3}$$

The maximal link utilisation:

$$\alpha(T) = Max\left\{\frac{\phi + t_{ij}}{z_{ij}}\right\},\ (i,j)\in T \tag{4}$$

The average delay of the multicast tree:

$$DA(T) = \frac{1}{|R|}\sum_{r_d\in R} d(p_T(s, r_d)) \tag{5}$$

The problem is subject to a link capacity constraint as follows:

$$\phi + t_{ij} \le z_{ij},\ \forall (i,j)\in T(s,R) \tag{6}$$

3. The Variable Neighborhood EMOSA (VEMOSA) Approach

3.1 The Evolutionary Multi-objective Simulated Annealing (EMOSA) Algorithm

Simulated annealing is one of the mostly studied stochastic local search algorithms in global optimisation (Kirkpatrick *et al*, 1983). The basic idea of the algorithm comes from the physical annealing process of heating and controlled cooling of a material to form a larger size of better state. During the search of simulated annealing, moves to non-improving neighborhood solutions are accepted in the algorithm with a given probability depending on the temperature at that stage and the difference between the fitness of the neighborhood solutions. As a result, the search has a chance to escape from local optima, making simulated annealing a very effective meta-heuristic when exploring the search space of complex combinatorial optimisation problems.

A recent Evolutionary Multi-objective Simulated Annealing (EMOSA) algorithm developed in (Li and Landa-Silva, 2008) has shown to be effective for solving traveling salesman problems. In the EMOSA algorithm, a population of solutions evolves by using strategies of the cooling schedule in simulated annealing algorithms. We describe the main features of EMOSA as follows:

1) The weighted scalarising function in the EMOSA algorithm

One general technique in multi-objective optimisation is to combine different objectives into a single composite function. In EMOSA, a weighted scalarising function is used to evaluate the fitness of solutions. A Pareto-optimal solution x^* is obtained if it is the unique global minimum of the following scalar optimisation problem:

$$\text{Minimise } g(x, \lambda) = \sum_{i=1}^{m} \lambda_i f_i(x),\ x \in X \tag{7}$$

where λ is a weight vector, $\lambda_i \in [0, 1]$, $i = 1, \ldots, m$ and $\sum_{i=1}^{m} \lambda_i = 1$, i.e. the weights are uniformly assigned to m objectives, and each λ_i is associated with one objective function $f_i(x)$. The function $g(x, \lambda)$ in (7) is called the weighted scalarising function, and is one of many scalarising functions used in the literature (Miettinen, 1999).

The competition between members of population in EMOSA is managed by adaptation strategies using the scalarising function defined in (7). During the evolution of EMOSA, a neighboring solution x' replaces similar solutions in the current population if it is better with respect to the scalarising function. The similarity between solutions is measured by the Euclidean distance between their weight vectors.

2) The acceptance criterion in the EMOSA algorithm

In EMOSA, the acceptance decision of a new solution has to take into account all the objectives in the problem. The acceptance probability in the standard simulated annealing has been modified by using the weighted scalarising function $g(x, \lambda)$ in (7). Given an incumbent solution x and its neighborhood $N(x)$, the probability of x moving to its neighbor $x' \in N(x)$ is thus defined as follows:

$$P(x, x', \lambda, T) = \begin{cases} 1 & \text{if } \Delta g(x, x', \lambda) < 0 \\ e^{\frac{-\Delta g(x, x', \lambda)}{T}} & \text{otherwise} \end{cases} \tag{8}$$

where $T > 0$ is the current temperature, $\Delta g(x, x', \lambda) = g(x', \lambda) - g(x, \lambda)$ is the difference between x and x', defined by using the weighted scalarising function in (7).

3) The two-phase strategy in the EMOSA algorithm

In addition to defining a starting temperature, a final temperature, and a temperature decrement in the cooling schedule, in EMOSA, a threshold temperature TP_c is also defined within a two-phase strategy for tuning the weight vectors. During the evolution of EMOSA, in the first phase when $T \geq TP_c$, all weight vectors remain unchanged. In the second phase when $T < TP_c$, the weight vectors are adaptively changed according to the closeness of solutions to their non-dominated neighbor in the population.

3.2 The Overall Procedure of the VEMOSA Algorithm

Based on the EMOSA algorithm developed in (Li and Landa-Silva, 2008), we propose in this work a variable neighborhood based EMOSA (VEMOSA) for solving MRPs. In VEMOSA, a simple binary vector of n bits, where n is the number of nodes, has been used to represent the multicast tree in the population during the evolution. The value of 1 to a bit indicates that the corresponding node is included in the multicast tree. This solution representation has been widely used in the literature to represent multicast trees.

The pseudo code of the VEMOSA algorithm is presented in Fig. 1. The algorithm starts with a random initial population of solutions where multicast trees T_i are generated by starting from the source node and randomly connecting the next node until all the destination nodes have been added to the tree. Each solution T_i is initially associated with a random weight vector λ^i. An external non-dominated solution set is maintained to store non-dominated solutions during the evolution and is returned as the solutions for the problem after the evolution is finished.

Insert Fig. 1 somewhere here.

During the evolution, for each multicast tree T_i in the population, a neighboring tree T_i' is generated by using a randomly chosen neighborhood structure N_k. The neighboring tree is accepted as the current tree with a probability $P(T_i, T_i', \lambda^i, TP)$ based on the current temperature TP and the weight vector λ^i, defined by the adopted acceptance criteria in simulated annealing, see formula (8). The external non-dominated solution set is updated if T_i' is not dominated by the current tree T_i. Note that in VEMOSA, if $k_{max} = 1$ then it becomes the EMOSA algorithm with only one neighborhood structure.

To maintain a population of diverse solutions, two adaptation strategies have been used in VEMOSA. Each newly generated neighboring tree replaces its closest member in the current population if it is better with respect to the scalarising function in (7). At the later stage of the evolution, when the current temperature is decreased to below a threshold, the weight vectors of each solution are adaptively adjusted to tune the search directions near the Pareto front. The VEMOSA algorithm stops either when the temperature drops to below the final temperature or after a given period of time.

3.3 The Variable Neighborhood Structures in VEMOSA

Variable neighborhood search is a recent meta-heuristic, jointly invented by Hansen and Mladenovic (Hansen and Mladenovic, 2001), for solving combinatorial and global optimisation problems. Compared with other single neighborhood based meta-heuristics, variable neighborhood search is more flexible and effective due to its ability to escape from local optima by traversing among different search spaces defined by different neighborhood structures. However, little research attention has been given on applying variable neighborhood search to solve multi-objective optimisation problems (e.g. Liu *et al*, 2006; Adibi and Zandieh, 2009).

In solving multi-objective problems, an important research issue concerned is the diversification and the intensification of the search for Pareto optimal solutions. In the current literature, almost all multi-objective simulated annealing algorithms use only one neighborhood structure. We investigate in this paper the important issue of how to appropriately define neighborhood structures targeting at different objectives of multi-objective MRPs. Our aim is to devise neighborhood structures which effectively drive the search towards diverse solutions with regard to different individual objectives, rather than to select solutions within those obtained during the search which, if not properly designed, may miss some of the Pareto optimal solutions in the search space of complex multi-objective MRPs. We integrate the EMOSA algorithm with a set of neighborhood structures designed for individual objectives during the search process. Two EMOSA algorithms have been evaluated and compared based on the EMOSA algorithm described above, namely 1) EMOSA with a single neighborhood structure; and 2) VEMOSA with a set of variable neighborhood structures. Both algorithms represent new approaches in solving MRPs in the literature.

In VEMOSA, two types of neighborhood operators have been devised as the transition mechanism to generate neighboring solutions.

- **The path-switching operator**. This type of neighborhood operator operates on paths in the multicast tree. A backup-path-set of *k*-best paths for each objective is constructed from the source node *s* to each destination by using the *k*-shortest path algorithm (Eppstein, 1998). For *m* objectives, there are $m \times k$ paths in the backup-path-set for each destination (in this work m = 4 as formulated in Section 2.2., see Fig. 3 for an example). In our algorithms, we keep the same *k*-best paths for the two delay related objectives in (3) and (5). To generate a neighboring tree, a destination node r_d is randomly chosen and the path from the source to r_d in the current tree is replaced by a randomly selected path in the backup-path-set of the chosen destination. The mechanism of the backup-path-set is developed based on a similar idea of routing table designed in (Crichigno and Baran, 2004a), which shows to be effective to provide alternative paths in multicast trees.
- **The node-switching operator.** This neighborhood operator operates on nodes in the multicast tree. A neighboring tree is generated by removing a random node (excluding the source and destination nodes) in

the current tree. A minimum spanning tree is then generated by using the Prim's algorithm (Betsekas and Gallager, 1992) with the remaining nodes in the tree. The Prim's algorithm is a well known algorithm in graph theory that finds a tree which spans a subset of links and all the nodes in a graph, where the total weight of all the links in the tree is minimised.

In this work, the neighborhood structure in EMOSA is based on the path-switching operator. In VEMOSA, besides the neighborhood structure defined in EMOSA, four additional neighborhood structures based on both the path-switching and the node-switching operators are devised for each objective. By employing different neighborhood structures specifically target at individual objectives, VEMOSA is expected to make intensified search while optimising all objectives simultaneously.

For ease of understanding, we present the benchmark NSF (National Science Foundation) network (Cui *et al*, 2003) with 14 nodes in Fig. 2 and an example backup-path-set of the path-switching operator in Fig. 3. The NSF network is a major part of early 1990s Internet backbone and has been tested in the existing literature by a number of researchers as a benchmark MRP.

Insert Fig.2 and Fig.3 somewhere here.

For the MRP in Fig. 2.(a), a random multicast tree and an optimal solution is given in Fig. 2.(b) and Fig. 2.(c), respectively. To generate a neighboring tree, the set of backup-paths of destination r_d = 0 is shown in Fig. 3, i.e. the k least cost paths for objective (2), the k least delay paths for objectives (3) and (5), and the k least link utilisation paths for objective (4). We list only 5 paths for each objective in Fig. 3.

3.4 Adaptation Strategies in VEMOSA

To diversify the search and avoid getting trapped in local optima, we adopted the two adaptation strategies in EMOSA (Li and Landa-Silva, 2008) to tune the search directions in VEMOSA during the evolution. Both adaptive strategies have shown to help improving the performance of the EMOSA algorithm.

- The first strategy concerns the competition between the closest members in the current population. After a new solution T_i' is generated, the closest solution T_j measured by the Euclidean distance on their weight vectors in the current population is selected. VEMOSA will replace T_j by T_i' if T_j is worse than T_i' by comparing their weighted scalarising function values, i.e. $g(T_i', \lambda^j) < g(T_j, \lambda^j)$. The competition is an intensification strategy which improves the quality of solutions in the current population.
- The second strategy adaptively tunes the weight vectors according to the closest non-dominated neighboring solution in the current population after the current temperature is below a given threshold. This is similar to that is used in EMOSA (Li and Landa-Silva, 2008), aiming to tune the directions of the

search when it is getting closer to the Pareto front at the later stage of the evolution. The second strategy is intended to diversify the search directions along the Pareto front.

4. Performance Evaluation

We first evaluate the impact of neighborhood structures and adaptation strategies within the VEMOSA algorithms upon two random networks with two objectives. Based on the observations obtained, we test two multi-objective multicast routing benchmark problems, namely the NSF network in Fig. 2.(a) and the NTT network in Fig. 4 (see Crichigno and Baran, 2004b) with all four objectives defined in Section 2.2, and compared our results with those of existing approaches in the literature. Finally, we demonstrate the effectiveness and efficiency of our algorithms upon a set of random networks of different sizes with all the four objectives defined.

Insert Fig. 4 somewhere here.

4.1 Parameter Settings

The random networks in our simulations are generated by the multicast routing simulator (MRSIM) implemented in C++ based on Salama's generator (Salama *et al*, 1997). MRSIM generates random network topologies by using the Waxman's graph generation algorithm (Waxman BM, 1988). In the random networks, the average node degree is set as 4. The link capacity z_{ij} is set as 1.5Mbps, the delay d_{ij} is generated depending on the length of the link, the cost c_{ij} is assigned a random value between (0,100] which represents the bandwidth consumption on the link, and the current traffic t_{ij} is randomly loaded with around 50% of its total link capacity.

The parameter settings for all EMOSA and VEMOSA algorithms are given in Table 2. They have been determined after a set of extensive tests. It is well known that simulated annealing with a higher starting temperature leads to better results however at a much longer computational time. Our purpose here is to find the balance between the quality of solutions and the expenses of computations. For fair comparisons, we keep the same parameter settings for both the VEMOSA and EMOSA algorithms on all the networks tested in this paper. All simulations have been run on a Windows XP computer with PVI 3.4GHZ, 1G RAM. To encourage scientific comparisons, the detailed information of all the problems and experimental results are provided at http://www.cs.nott.ac.uk/~rxq/benchmarks.htm.

Insert Table 2 somewhere here.

The following notations are used in the analysis of experimental results:

- VEMOSA1 and VEMOSA2: VEMOSA with and without adaptation
- EMOSA1 and EMOSA2: EMOSA with and without adaptation

4.2 Variable Neighborhoods and Adaptation Strategies in VEMOSA

We first compare VEMOSA and EMOSA to identify the impact of different neighborhood structures on the performance the proposed multi-objective algorithms. In addition, we evaluate the effect of adaptation strategies in these variants of algorithms.

To visually compare the Pareto optimal front generated by variants of algorithms, in this set of experiments we consider only two objectives, namely the cost and the delay as defined in Section 2.2, on two random networks of $|V| =50$ and $|V| =100$ with different group sizes, i.e. the number of destination nodes $|R| = 20\%*|V|$ and $|R| = 30\%*|V|$. The computational time of each algorithm for each run is set to 160 seconds. If within the given computational time the temperature drops to the final temperature, the algorithms will reheat the current temperature to the initial temperature and start again. The best results obtained are returned as the solutions for the problem.

The union of non-dominated solutions found by each algorithm after 20 runs for the random network of size $|V| =50$ with different group sizes are shown in Fig. 5. It can be clearly seen that both VEMOSA algorithms with five neighborhoods found much better non-dominated solutions than the two EMOSA algorithms with single neighborhood structure. This is due to that the variable neighborhoods in VEMOSA are devised to effectively deal with different individual objectives simultaneously. Within both VEMOSA and EMOSA, the adaptation strategies do not seem to make significant contributions, shown by the similar performance with the interweaving Pareto fronts found by the algorithm with and without adaptation strategies for this small network of 50 nodes.

Similarly, in Fig. 6, the union of non-dominated solutions found by 20 runs of the four algorithms shows that VEMOSA outperforms EMOSA with a much larger scale on the larger network of 100 nodes. It becomes clearer that variable neighborhoods can contribute to a better performance of VEMOSA on larger networks. It is interesting to see that with single neighborhood structure, the adaptation strategies in EMOSA1 significantly improve the algorithm performance compared against EMOSA2. However, both VEMOSA algorithms with variable neighborhood structures have similar performances. This shows that the adaptation strategies have less impact on VEMOSA, where variable neighborhoods can also effectively drive the search directions towards diverse Pareto optimal solutions of different objectives. Our proposed VEMOSA algorithm outperforms EMOSA even without adaptation strategies which carefully concern the directions of the search.

Insert Fig. 5 and Fig. 6 somewhere here.

4.3 The VEMOSA and EMOSA Algorithms on Benchmark Instances

4.3.1 The VEMOSA and EMOSA algorithms for the NSF network with Four Objectives

For the NSF benchmark problem in Fig. 2.(a), an optimal Pareto Front (*PF*) that consists of 16 solutions have been found by an exhaustive search in (Crichigno and Baran, 2004b). We test the same four variants of VEMOSA and EMOSA algorithms on the MRP of the NSF network with the four objectives defined in Section 2.2. Table 3 presents the results of each algorithm within 100 runs, compared against those of MOEA (Crichigno and Baran, 2004a) and the Improved MOEA (Crichigno and Baran, 2004b) in the literature.

Insert Table 3 somewhere here.

For this small NSF problem, our proposed VEMOSA and EMOSA algorithms outperform the two MOEA approaches in the literature. Among all algorithms, both VEMOSA algorithms perform the best, i.e. always find all 16 Pareto optimal solutions in all 100 runs. This, together with our above observations for the random networks with two objectives, supports the conclusion that VEMOSA is more effective than EMOSA and MOEA algorithms with only one neighborhood structure for multi-objective MRPs. It is not surprising to see that the VEMOSA algorithms spend longer computational time to obtain better results compared with the EMOSA algorithms when using the same parameter settings.

4.3.2. The VEMOSA and EMOSA algorithms for the NTT network

We also test our proposed four algorithms on the NTT network in Fig. 4 with four objectives and different multicast groups, shown in Table 4. As the traffic t_{ij} for each link in the NTT network is randomly loaded with around 50% of its total link capacity in our experiments and in (Diego and Baran, 2005), we could not provide comparisons on the exact same experimental data. We therefore report and analyse results of our proposed four algorithms in this paper.

Insert Table 4 somewhere here.

To obtain an approximation of the Pareto optimal solution set, denoted by Y_{PF}, the following six-step procedure in (Diego and Baran, 2005) is also used here to generate Y_{PF}:

1) Each algorithm is run 10 times. In this work four variants of EMOSA and VEMOSA algorithms have been used.

2) For each algorithm, 10 sets of non-dominated solutions Y_1, Y_2, …, Y_{10} are obtained, one from each run.

3) For each algorithm, the aggregate population Y_T is obtained, where $Y_T = \bigcup_{i=1}^{10} Y_i$.

4) Dominated solutions are deleted from Y_T to obtain the non-dominated solution set for each algorithm, denoted by Y_{alg}. Here, *alg* represents an algorithm, i.e. $Y_{VEMOSA1}$, $Y_{VEMOSA2}$, Y_{EMOSA1}, Y_{EMOSA2}.

5) A set of solutions *Y'* combining the non-dominated solutions from all algorithms is obtained, i.e. $Y' = \bigcup Y_{alg}$

6) Non-dominated solutions in *Y'* are selected as an approximation of the true Pareto optimal solution set Y_{PF}.

Table 5 presents the total number of non-dominated solutions in Y_{PF} obtained by our four variants of EMOSA and VEMOSA algorithms for the NTT network by using the above six-step procedure. Results are obtained for the NTT network with two different multicast groups by using 160 seconds and 320 seconds, respectively. Our proposed four algorithms are tested with regard to the obtained Y_{PF} on the NTT network, results shown in Table 6 and Table 7.

Insert Table 5 somewhere here.

Insert Table 6 and Table 7 somewhere here.

Table 6 shows that the VEMOSA algorithms found more non-dominated solutions in Y_{PF} than that of EMOSA for each run. It is interesting to see that EMOSA2 outperforms EMOSA1, meaning that the adaptation strategies of tuning search directions does not help to improve the algorithm performance for the small multicast group instance. VEMOSA1 and VEMOSA2 have similar performance, again showing that adapting search directions does not make significant contribution to the VEMOSA algorithms.

Results in Table 7 show that for the NTT network with large multicast group, VEMOSA algorithms outperform EMOSA algorithms, demonstrating that variable neighborhoods help the algorithms to find more non-dominated solutions. EMOSA1 with the adaptation strategy found more solutions in Y_{PF} than that of EMOSA2, which does not find a single solution in Y_{PF}. Compared with the results in Table 6, this indicates that the adaptation strategies improve the algorithm performance for the large multicast group instance. However, if given longer computational time, even without adaptation strategies, VEMOSA2 find more non-dominated solutions than that of VEMOSA1, again showing that the adaptation strategies have less impact to the performance of VEMOSA algorithms.

4.4 The VEMOSA Algorithms on Random Networks

Finally, we test the four algorithms on random networks of size $|V| = 50$ and $|V| = 100$ with different group sizes of $|R| = 20\%*|V|$ and $|R| = 30\%*|V|$ and with all four objectives defined in Section 2.2. Table 8 presents the total number of the estimated non-dominated solutions obtained by the four algorithms on the random networks by using the six-step procedure in Section 4.3.2.

Insert Table 8 somewhere here.

Table 9 and Table 10 present the results of the four variants of algorithms on the two random networks with different group sizes in the given 360 seconds. Average results from 10 runs for each problem show that both VEMOSA algorithms significantly outperform the EMOSA algorithms in finding more non-dominated solutions in Y_{PF} in Table 8. The scale of differences again is much larger for larger problems, demonstrating the effectiveness of the VEMOSA algorithms for larger multi-objective MRPs. We observe again that the two VEMOSA algorithms have similar performances with regard to the average number of non-dominated solutions found, i.e. the number of solutions in Y_{PF} for problems of different group sizes. The adaptation strategies have small impact on the performance of VEMOSA algorithms.

To summarise, the extensive experimental results on a range of different multi-objective MRPs with different features demonstrate the efficiency and effectiveness of our proposed evolutionary simulated annealing algorithms based on variable neighborhoods. Variable neighborhood structures which are specifically designed for different objectives significantly improve the performance of the proposed algorithms, especially for problems of larger network size. The VEMOSA algorithms work very well even without the adaptation strategies, which have shown to be important in the EMOSA algorithms for both MRPs in our experiments and the travelling salesman problems in (Li and Landa-Silva, 2008).

5. Conclusions

In this paper, we investigate the hybridisation of an evolutionary simulated annealing algorithm with variable neighborhoods for solving the multi-objective multicast routing problems in telecommunications. Instead of employing a single neighborhood structure, a number of neighborhood structures are designed to deal with each specific objective during the evolution in the complex search space of multi-objective multicast routing problems. By comparing the evolutionary multi-objective simulated annealing (EMOSA) algorithm and its hybridisation with variable neighborhoods (VEMOSA), we demonstrate the efficiency and effectiveness of the VEMOSA algorithms via an extensive set of experiments for solving both the benchmark and random multi-objective multicast routing problems with different number of objectives.

With a single neighborhood, the conventional EMOSA algorithms is not able to identify multiple objectives simultaneously, thus cannot find some of the Pareto optimal solutions. The proposed VEMOSA algorithms overcome this weakness and obtain a highest number of non-dominated solutions for all the multi-objective multicast routing problems tested. It is observed that properly defined variable neighborhood structures can significantly improve the performance of multi-objective simulated annealing algorithms. In addition, the competition between the members of solutions and the adaptive tuning of the search directions shown to be effective in EMOSA. However, they seem to play a less important role in the VEMOSA algorithms. The VEMOSA algorithms have shown to be highly effective even without the adaptation strategies for solving the multi-objective multicast routing problems.

The proposed VEMOSA algorithm has shown to be a promising approach for solving multicast routing problems with multiple objectives. In our future work, we intend to investigate the influence of specific neighborhood structures within the VEMOSA algorithm for solving multicast routing problems with wider range of real world features. It is also interesting to extend our VEMOSA algorithms to solve other multi-objective optimisation problems.

References

Adibi MA, Zandieh M, Amiri M (2010). Multi-objective scheduling of dynamic job shop using variable neighborhood search. Expert Systems with Applications. 37: 282-287.

Betsekas D, Gallager R (1992). Data networks (2nd edition). Englewood Cliffs, HJ: Prentice-Hall.

Burke EK, Kendall G (2005). Multi-objective Optimisation. In: Search Methodologies: Introductory Tutorials in Optimisation and Decision Support Methodologies. Springer.

Chen S, Nahrstedt K (1998). An overview of quality of service routing for next-generation high-speed networks: problems and solutions. IEEE Network, Special Issue on Transmission and Distribution of Digital Video. 12: 64-79.

Crichigno J, Baran B (2004a). Multiobjective Multicast Routing Algorithm. In: LNCS, 3124: 1029-1034.

Crichigno J, Baran B (2004b). Multiobjective Multicast Routing Algorithm for Traffic Engineering. In: Proceedings of the 13th International Conference on Computer Communications and Networks , pp 301-306. Springer, Heidelberg.

Cui X, Lin C, Wei Y (2003). A Multiobjective Model for QoS Multicast Routing Based on Genetic Algorithm. 2003 International Conference on Computer Networks and Mobile Computing (ICCNMC'03), pp 49.

Czyzak P, Jaszkiewicz A (1998). Pareto simulated annealing - a meta-heuristic technique for multiobjective combinatorial optimization. Journal of Multi-Criteria Decision Analysis, 7: 34-37.

Diot C, Dabbous W, Crowcroft J (1997). Multipoint communication: a survey of protocols, functions, and mechanisms. IEEE Journal on Selected Areas in Communications, 15: 277-290.

Diego P, Baran B (2005). Solving Multiobjective Multicast Routing Problem with a new Ant Colony Optimization Approach. Proceedings of the 3rd international IFIP/ACM Latin American conference on Networking Cali, Columbia. pp 11-19.

Eppstein D (1998). Finding the k shortest paths, SIAM J. Computing. 28: 652-673.

Ehrgott M, Gandibleux X (2000). A Survey and Annotated Bibliography of Multiobjective Combinatorial Optimization. OR Spectrum. 22: 425-460.

Fabregat R, Donoso Y, Baran B, Solano F, Marzo JL (2005). Multi-objective optimization scheme for multicast flows: a survey, a model and a MOEA solution. In: Proceedings of the 3rd international IFIP/ACM Latin American Conference on Networking (LANC '05). ACM, New York.

Garey MR, Johnson DS (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, New York.

Guo L, Matta I (1999). QDMR: An efficient QoS dependent multicast routing algorithm. In: Proceedings of the 5th IEEE Real Time Technology and Applications Symposium. pp 213-222.

Gandibleux X, Sevaux M, Sörensen K, T'kindt V(2004). Metaheuristics for Multiobjective Optimization. Lecture Notes in Economics and Mathematical Systems: Springer.

Haghighat AT, Faez K, Dehghan M, Mowlaei A, Ghahremani Y (2004). GA-based heuristic algorithms for bandwidth-delay-constrained least-cost multicast routing, Computer Communications, 27: 111-127.

Haidine A, Lehnert R (2008). Multi-case multi-objective simulated Annealing (MC-MOSA): New approach for adapt simulated annealing to multi-objective optimization. International Journal of Information Technology. 4: 197-205.

Hansen P, Mladenovi'c N (2001). Variable neighborhood search: principles and applications. European Journal of Operations Research. 130: 449-467.

Hwang FK, Richards DS (1992). Steiner tree problems. Networks. 22: 55-89.

Koyama A, Barolli L, Matsumoto K, Apduhan BO (2004). A GA-based multi-purpose optimization algorithm for QoS routing. Proceedings of the 18th International Conference on Advanced Information Networking and Applications (AINA 2004), 1: 23-28.

Kirkpatric S, Gelatt CD, Vecchi MP (1983). Optimization by simulated annealing. Science. 220: 671-680.

Kun Z, Heng W, Feng-Yu L (2005). Distributed multicast routing for delay and delay variation-bounded Steiner tree using simulated annealing. Computer Communications. 28: 1356-1370.

Konak A, David WC, Alice ES (2006). Multi-objective optimization using genetic algorithms: A tutorial. Reliability Engineering and system Safety. 91: 992-1007.

Landa-Silva JD, Burke EK, Petrovic S (2004). An Introduction to Multiobjective Metaheuristics for Scheduling and Timetabling. In: Xavier Gandibleux, Marc Sevaux, Kenneth Sorensen, Vincent T'kindt (eds). Metaheuristic for Multiobjective Optimisation. Lecture Notes in Economics and Mathematical Systems: Springer, pp 91- 129.

Li C, Cao C, Li Y, Yu Y (2007). Hybrid of Genetic Algorithm and Particle Swarm Optimization for Multicast QoS Routing. IEEE International Conference on Control and Automation. Guangzhou, China.

Li H, Landa-Silva D (2008). Evolutionary multi-objective simulated annealing with adaptive and competitive search direction. In: Proceedings of the 2008 IEEE Congress on Evolutionary Computation (CEC 2008). pp 3310-3317. IEEE Press: Hong Kong.

Liu H, Abraham A, Choi O, Moon SH (2006). Variable neighborhood particle swarm optimization for multi-objective flexible job-shop scheduling problems. SEAL 2006. LNCS 4247, pp 197-204.

Miettinen K (1999). Nonlinear Multiobjective Optimization. Kluwer Academic Pulishers: Boston, USA.

Masip-Bruin X, Yannuzzi M, Domingo-Pascual J, Fonte A, Curado M, Monteiro E, Kuipers F, Van Mieghem P, Avallone S, Ventre G, Aranda-Gutierrez P, Hollick M, Steinmetz R, Iannone L, Salamatian K (2006). Research challenges in QoS routing. Computer Communications. 29: 563-581.

Oliveira CAS, Pardalos PM (2005). A survey of combinatorial optimization problems in multicast routing. Computers & Operations Research. 32: 1953-1981.

Qu R, Xu Y, Kendall G (2009). A Variable Neighborhood Descent Search Algorithm for Delay-Constrained Least-Cost Multicast Routing, In: Online Proceedings of Learning and Intelligent OptimizatioN (LION3), Trento, Italy.

Roy A, Banerjee N, Das SK (2002). An efficient Multi-Objective QoS-Routing Algorithm for wireless-Multicasting. IEEE 55th Vehicular Technology Conference. pp 1160-1164.

Roy A, Das SK (2004). QM2RP: a QoS-based Mobile Multicast Routing Protocol using Multi-Objective Genetic Algorithm. Wireless Networks. 10: 271-286.

Salama HF, Reeves DS, Viniotis Y (1997). Evaluation of multicast routing algorithms for real-time communication on high-speed networks, IEEE Journal on Selected Areas in Communications. 15: 332-345.

Sefarini P (1994). Simulated annealing for multiobjective optimization problems. In: Tzeng, Wang and Pan (eds). Proceedings of the tenth international conference on multiple criteria decision making. pp 283-292.

Skorin-Kapov N, Kos M (2006). A GRASP heuristic for the delay-constrained multicast routing problem, Telecommunication Systems. 32: 55-69.

Suman B, Kumar P (2006). A survey of simulated annealing as a tool for single and multiobjective optimization. Journal of Operational Research Society. 57: 1143-1160.

Ulungu EL, Teghem J (1994). Multiobjective combinatorial optimization problems: A survey. Journal of Multi-Criteria Decision analysis. 3: 83-104.

Ulungu EL, Teghem J, Fortemps PH, Tuyttens D (1999). MOSA method: A tool for solving multiobjective combinatorial optimization problems. Journal of Multi-Criteria Decision Analysis, 8: 221-236.

Waxman BM (1988). Routing of multipoint connections. IEEE Journal on Selected Areas in Communications. 6: 1617-1622.

Yeo CK, Lee BS, Er MH (2004). A survey of application level multicast techniques, Computer Communications. 27: 1547-1568.

Zhu Q, Parsa M, Garcia-Luna-Aceves JJ (1995). A source-based algorithm for delay-constrained minimum-Cost multicasting, In: Proceedings of the 14th Annual Joint Conference of the IEEE Computer and Communication (INFOCOM'95), pp 377-385. IEEE Computer Society Press: Boston, Massachusetts.

```
VEMOSA (m, k_max, TP_max, TP_min, TP_c)
 { // m: the number of objectives; k_max: the number of neighborhood structures;
   // TP_max / TP_min: the starting / final temperatures; TP_c: the threshold temperature for tuning weight vectors;
  Initialization:
  Create the initial population P by randomly generating T_1, …, T_pop multicast trees; // pop: population size
  Produce pop distinct random weight vectors λ^1, …, λ^pop with a uniform distribution;
  Form the external non-dominated solution set NDS by the non-dominated solutions in P;
  Set the current temperature TP = TP_max;
  while TP ≥ TP_min do {
      foreach T_i ∈ P do {
          lc = 0;
          while lc < LC do { // LC: the number of local search moves between two consecutive temperatures
              Generate a neighbor T_i' ∈ N_k (T_i) by a random k ∈ {1, ..., k_max}, set lc = lc + 1;
              if T_i does not dominate T_i' then
                 Update the external NDS set;
              Replace the current solution T_i by T_i' with the probability P(T_i, T_i', λ^i, TP); // see formula (8)
              Find the closest solution T_j ∈ P (T_j ≠ T_i) of T_i'
              if (g(T_i', λ^j) – g(T_j, λ^j) then
                 T_j = T_i' // adaptation strategy 1: replace the closest solution in P, see Section 3.4
          end of while loop }
      }
      TP = TP – α; // α: temperature decrement
      if TP < TP_c then // adaptation strategy 2: tune the search direction, see Section 3.4
      foreach T_i ∈ P do {
          Find the closest non-dominated solution T of T_i from P
          foreach obj ∈ {1, …, m} do {
              if f_obj(T) < f_obj(T_i) then λ^i_obj = μ λ^i_obj; // μ: the constant for tuning the search direction
              else λ^i_obj = λ^i_obj / μ; }
      }
  end of while loop }
  return the non-dominated solution set NDS;
}
```

Fig. 1. The pseudo code of the VEMOSA algorithm.

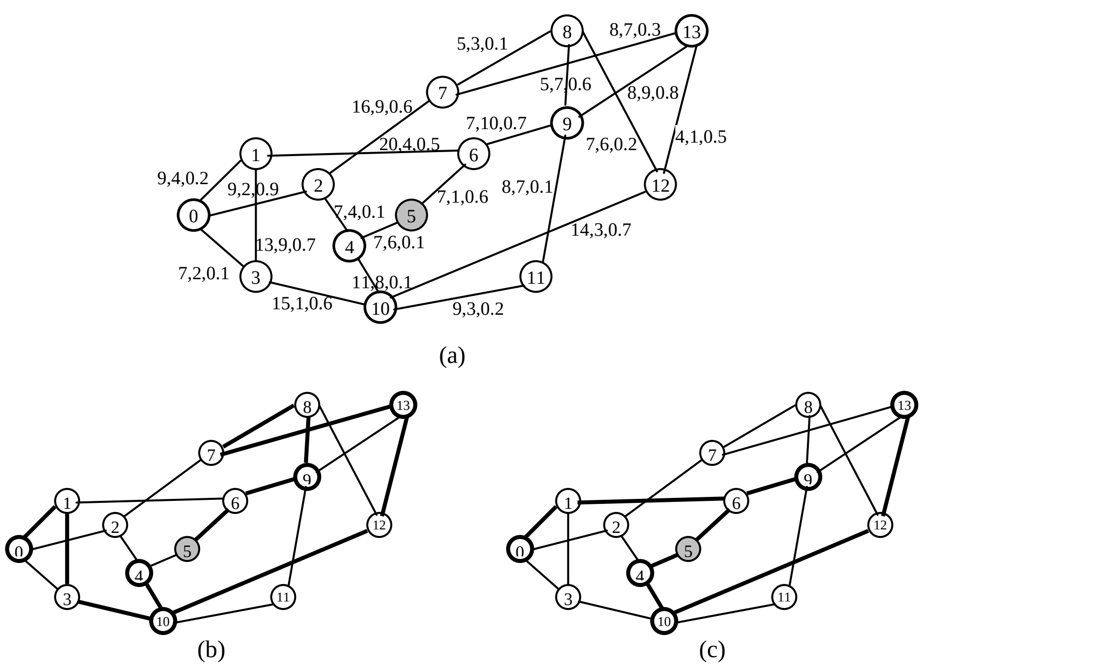


Fig. 2. The NSF network and two example solutions (multicast trees). (a) The NSF (National Science Foundation) network. The delay d_{ij}, cost c_{ij} and current traffic t_{ij} are shown on each link. The shaded node 5 is the source node; the destinations $R = \{0, 4, 9, 10, 13\}$. For each link the traffic demand $\phi = 0.2$Mbps, capacity $z_{ij} = 1.5$Mbps. (b) An example of a random initial solution, $C(T) = 10.8$, $DM(T) = 87$, $\alpha(T) = 0.6$, $DA(T) = 48.8$; (c) An example of an optimal solution, $C(T) = 7.4$, $DM(T) = 36$, $\alpha(T) = 0.6$, $DA(T) = 22.2$.

ID	Paths
1	5-6-1-0
2	5-4-2-0
3	5-6-9-11-10-3-0
4	5-6-1-3-0
5	5-4-10-3-0
(a) least cost paths	

ID	Paths
1	5-4-2-0
2	5-6-1-0
3	5-4-10-3-0
4	5-4-2-7-8-9-6-1-0
5	5-6-9-8-7-2-0
(b) least delay paths	

ID	Paths
1	5-4-10-3-0
2	5-6-1-0
3	5-4-2-0
4	5-4-10-11-9-6-1-0
5	5-4-10-3-1-0
(c) least link utilisation paths	

Fig. 3. An example of the backup-path-set by using the path-switching operator.

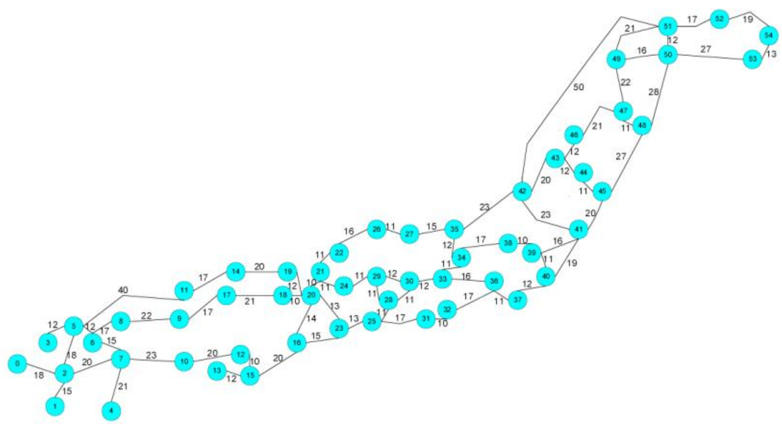
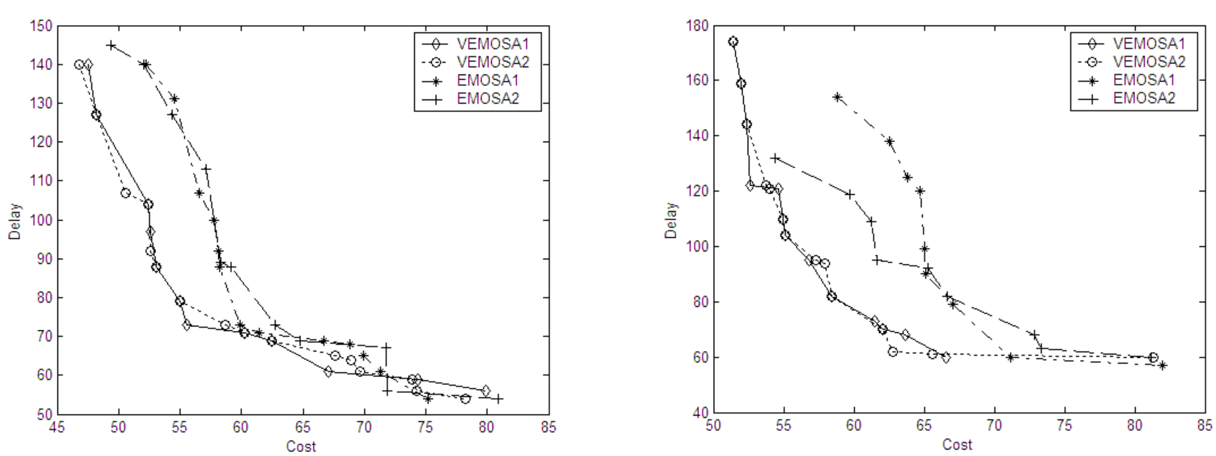


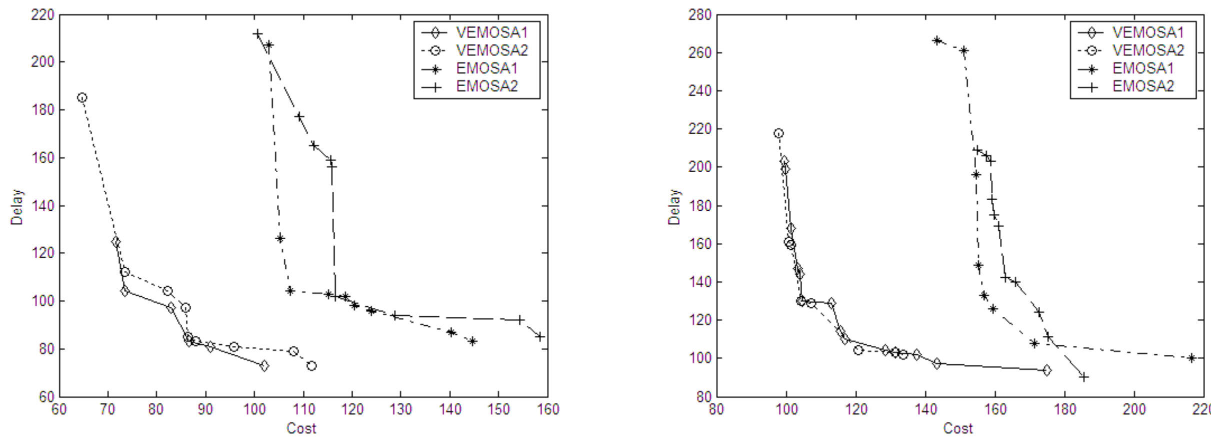
Fig. 4. The NTT (Nippon Telephone Telegraph of Japan) network topology with 55 nodes and 142 links.



(a) Cost vs. Delay (|*V*| = 50, |*R*| = 10)

(b) Cost vs. Delay (|*V*| = 50, |*R*| = 15)

Fig. 5. The non-dominated solutions found by variants of algorithms in 160 seconds for the random network of |*V*| =50 with different group sizes and two objectives.



(a) Cost vs. Delay (|*V*| = 100, |*R*| = 20) (b) Cost vs. Delay (|*V*| = 100, |*R*| = 30)

Fig. 6. The non-dominated solutions found by variants of algorithms in 160 seconds for the random network of |*V*| = 100 with different group sizes and two objectives.

Table 1. Categorised multi-objective multicast routing algorithms by the objectives and constraints considered in the problem. MOEA: multi-objective evolutionary algorithm.

Heuristic Algorithms	Objectives					Constraints	
	Cost	**Delay**	**Link Utilisation**	**Bandwidth**	**Packet Loss**	**Link Capacity**	**Delay**
A MOEA based on non-dominated sorting (Roy *et al*, 2002)		X	X	X			
A multi-objective genetic algorithm (Cui *et al*, 2003)		X		X	X		X
A QoS-based mobile multicast routing protocol based on a MOEA (Roy and Das, 2004)		X	X	X			
A MOEA (Crichigno and Baran, 2004a) and an improved MOEA (Crichigno and Baran, 2004b) based on a Strength Pareto Evolutionary Algorithm	X	X	X			X	
A multi-objective genetic algorithm (Koyama *et al*, 2004)	X	X					
A multi-objective ant colony optimisation system and a multi-objective Max-Min ant system (Diego and Baran, 2005)	X	X	X			X	
A hybrid genetic algorithm and a particle swarm optimisation algorithm (Li *et al*, 2007)	X	X	X			X	

Table 2. Parameter settings of EMOSA and VEMOSA algorithms.

Algorithm parameters	**Settings**
population size	50
starting, final temperatures and temperature decrement	100, 5 and 5
number of iterations between two consecutive temperatures	25
temperature threshold for tuning weight vectors	50
constant μ in the adaptation strategy for tuning the search direction	1.05
number of paths for each objective in the backup-path-set for path-switching operator	25

Table 3. The maximum, minimum and average number of non-dominated solutions found by different algorithms for the NSF network in Fig. 2.(a) and the computational time of each algorithm. |*NDS*|: the number of non-dominated solutions; *PF*: the optimal Pareto front, here |*PF*| = 16.

Algorithms	**Max |*NDS*|**	**Min |*NDS*|**	**Average |*NDS*|**	**Computational time (sec)**
VEMOSA1	16	16	16	12.719
VEMOSA2	16	16	16	12.528
EMOSA1	16	15	15.98	5.346
EMOSA2	16	14	15	5.426
MOEA (Crichigno and Baran, 2004a)	16	10	12.72	/
MOEA (Crichigno and Baran, 2004b)	16	12	13.54	/

Table 4. Two different multicast groups within the NTT network in Fig. 4.

Group	**Source**	**Destination Set *R***	**|*R*|**
group1 (small)	5	{0,1,8,10,22,32,38,43,53}	9
group2 (large)	4	{0,1,3,5,6,9,10,11,12,17,19,21,22,23,25,33,34,37,41,44,46,47,52,54}	24

Table 5. The total number of non-dominated solutions for each multicast group of the NTT network in Fig 4 with two different multicast groups shown in Table 4.

Computational Time	160 seconds		320 seconds	
Multicast Group	group1	group2	group1	group2
$\lvert Y_{PF} \rvert$	6	9	6	11

Table 6. Four variants of algorithms for the NTT network in Fig. 4 with small multicast group 1. The best results are in bold.

Algorithms	**Time = 160 seconds**			**Time = 320 seconds**		
	Average Solutions in Y_{PF}	**Average Dominated Solutions**	**Average Total Solutions**	**Average Solutions in Y_{PF}**	**Average Dominated Solutions**	**Average Total Solutions**
VEMOSA1	**3.8**	8.2	12	**4**	6	10
VEMOSA2	**3.8**	8.6	12.4	**4**	5	9
EMOSA1	2.2	12.8	15	2.8	13.2	16
EMOSA2	3.4	10	14.4	3	9	12

Table 7. Four variants of algorithms for the NTT network in Fig. 4 with large multicast group 2. The best results are in bold.

Algorithms	**Time = 160 seconds**			**Time = 320 seconds**		
	Average Solutions in Y_{PF}	**Average Dominated Solutions**	**Average Total Solutions**	**Average Solutions in Y_{PF}**	**Average Dominated Solutions**	**Average Total Solutions**
VEMOSA1	**2**	8	10	1	10.2	11.2
VEMOSA2	1.8	8.2	10	**1.8**	7.4	9.2
EMOSA1	0.6	15.4	16	1	21.8	22.8
EMOSA2	0	27.8	27.8	0	17	17

Table 8. The total number non-dominated solutions of Y_{PF} on the random networks.

Network Size	$\lvert V\rvert = 50$		$\lvert V\rvert = 100$	
Group Size	$\lvert R\rvert = 10$	$\lvert R\rvert = 15$	$\lvert R\rvert = 20$	$\lvert R\rvert = 30$
$\lvert Y_{PF} \rvert$	96	164	46	46

Table 9. Comparison of different algorithms on random network of $|V| = 50$ with different group sizes. Computational time = 360 seconds. The best results are in bold.

Algorithms	**$\lvert R\rvert = 10$**			**$\lvert R\rvert = 15$**		
	Average Solutions in Y_{PF}	**Average Dominated Solutions**	**Average Total Solutions**	**Average Solutions in Y_{PF}**	**Average Dominated Solutions**	**Average Total Solutions**
VEMOSA1	17.4	19.8	37.2	**28**	94.2	122.2
VEMOSA2	**17.8**	21.8	39.6	23	92.2	115.2
EMOSA1	8.4	97.8	106.2	0.8	96.6	97.4
EMOSA2	4	79.6	83.6	8.6	37.8	46.4

Table 10. Comparison of different algorithms on random network of $|V| = 100$ with different group sizes. Computational time = 360 seconds. The best results are in bold.

Algorithms	**$\lvert R\rvert = 20$**			**$\lvert R\rvert = 30$**		
	Average Solutions in Y_{PF}	**Average Dominated Solutions**	**Average Total Solutions**	**Average Solutions in Y_{PF}**	**Average Dominated Solutions**	**Average Total Solutions**
VEMOSA1	**4.8**	26.4	31.2	3.4	15.6	17
VEMOSA2	4.6	26.4	31	**6.8**	11	34
EMOSA1	0.4	40.2	40.6	0	19	19
EMOSA2	0	51.4	51.4	0	27	27

List of Figures

List of Tables