Evolutionary algorithms for the multi-objective shortest path problem
Peer-reviewed author version

Evolutionary Algorithms for the Multiobjective Shortest Path Problem

José Maria A. Pangilinan and Gerrit K. Janssens

Abstract—This paper presents an overview of the multiobjective shortest path problem (MSPP) and a review of essential and recent issues regarding the methods to its solution. The paper further explores a multiobjective evolutionary algorithm as applied to the MSPP and describes its behavior in terms of diversity of solutions, computational complexity, and optimality of solutions. Results show that the evolutionary algorithm can find diverse solutions to the MSPP in polynomial time (based on several network instances) and can be an alternative when other methods are trapped by the tractability problem.

Keywords— Multiobjective evolutionary optimization, genetic algorithms, shortest paths.

I. INTRODUCTION

As in the case of the single-objective shortest path problem, the multiobjective shortest path problem has been studied extensively by various researchers in the fields of optimization, route planning for traffic and transport design [1] and information and communications network design [2], [3]. The MSPP is an extension of the traditional shortest path problem and is concerned with finding a set of efficient paths with respect to two or more objectives that are usually in conflict. For example, the problem of finding optimal routes in communications networks involves minimizing delay while maximizing throughput or finding efficient routes in transportation planning that simultaneously minimize travel cost, path length, and travel time. The concept of optimization in the MSPP or in a multiobjective problem in general is different from the single-objective optimization problem wherein the task is to find a solution that optimizes a single objective function. The task in a multiobjective problem is not to find an optimal solution for each objective function but to find an optimal solution that simultaneously optimizes all objectives. And in most cases, no single optimal solution exists, only a set of efficient or nondominated solutions.

A variety of algorithms and methods such as dynamic programming, label selecting, label correcting, interactive methods, and approximation algorithms have been implemented and investigated with respect to the MSPP [4]. The problem is known to be NP-complete [5]. It has been

J. A. Pangilinan is with the Computer Science Department, Saint Louis University, Baguio, 2600 Philippines. He is a PhD student at the University of Hasselt (e-mail: joeyp@slu.edu.ph).

G. K. Janssens is with the Faculty of Applied Economics, University of Hasselt, Diepenbeek, 3590 Belgium, (e-mail: gerrit.janssens@uhasselt.be).

shown that a set of problems exist wherein the number of Pareto-optimal solutions is exponential which implies that any deterministic algorithm that attempts to solve it is also exponential in terms of runtime complexity at least in the worst case. But some labeling algorithm studies [6], [7] dispute this exponential behavior. They show that the number of efficient paths is not exponential in practice. Other authors avoid the complexity problem by developing new methods that run in polynomial time. For instance, Hansen [8] and Warburton [9] separately develop fully polynomial time approximation schemes (FPTAS) for finding paths that are approximately Pareto-optimal. Interactive procedures [10], [11] similarly avoid the problem of generating the complete set of efficient paths by providing a user-interface that assists the decision-maker to focus only on promising paths and identify better solutions according to preference.

Given the wealth of literature in multiobjective algorithms for the MSPP, there still seems to be a lack of reported review in evolutionary algorithm (EA) applications in relation to the MSPP. Several of the most recent alternative methods focus mostly on execution speed comparisons of different MSPP algorithms but analysis of the salient issues in multiobjective performance analysis such as runtime complexity, diversity, and optimality of nondominated solutions are almost omitted. In order to demonstrate a clearer picture of the advantages and disadvantages of EAs in optimization, this paper attempts to investigate a multiobjective evolutionary algorithm as applied to the MSPP. The paper is organized as follows. Section II gives a brief background of the MSPP. Section III reviews related literature. Section IV presents a multiobjective evolutionary algorithm for the MSPP. Experimental conditions and results are discussed in Section V and Section VI presents the conclusion.

II. BACKGROUND

Given a directed graph $G = (V, E)$, where V is set of vertices (nodes) and E the set of edges (arcs) with cardinality $|V| = n$ and $|E| = m$ and a d-dimensional function vector $c:E \rightarrow [\Re^+]^d$. Each edge e belonging to E is associated with a cost vector $c(e)$. A source vertex s and a sink vertex t are identified. A path p is a sequence of vertices and arcs from s to t. The cost vector $C(p)$ for linear functions of path p is the sum of the cost vectors of its edges, that is $C(p) = \Sigma_{e \in p} c(e)$ whereas $C(p) = \min_{e \in p} c(e)$ for min-max functions. Given the two vertices s and t, let $P(s, t)$ denote the set of all s-t paths in G. If all

objectives are to be minimized, a path $p \in P(s, t)$ dominates a path $q \in P(s, t)$ iff $C_i(p) \le C_i(q)$, $i = 1,\dots,d$ and we write $p \preceq q$. A path p is Pareto-optimal if it is not dominated by any other path and the set of nondominated solutions (paths) is called the Pareto-optimal set. The objective of the MSPP is to compute the set of nondominated solutions that is the Pareto-optimal set $\mathcal{P}$ of $P(s,t)$ with respect to c.

The problem of the single-source multiobjective shortest path is to find the set of all paths from s to all other vertices in G, i.e. to find the Pareto-optimal set of $P(s,t)$, $\forall t \in V \setminus \{s\}$. Figure 1 illustrates an example of a directed graph with 3 objective functions that have to be minimized simultaneously. The corresponding efficient paths from the source node s to all other nodes are : for $t=1$ that path is $s \rightarrow 1$; for $t=2$, $s \rightarrow 1 \rightarrow 2$; for $t = 3$, $s \rightarrow 1 \rightarrow 2 \rightarrow 3$; for $t = 4$, , $s \rightarrow 1 \rightarrow 2 \rightarrow 4$; for $t = 5$, $s \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 5$ and $s \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 5$.
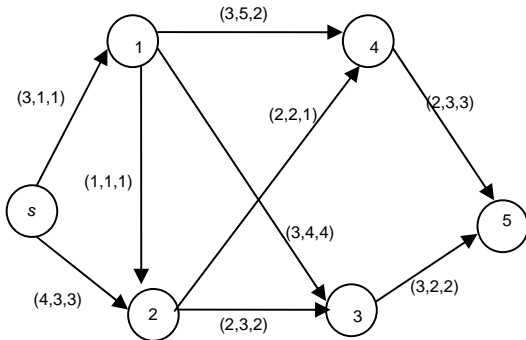

Fig.1 Example of a graph with 3 objectives to be minimized. There are 2 efficient paths from $s \rightarrow 5$: $s \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 5$ and $s \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 5$.

III. LITERATURE REVIEW

Martins and Santos [12] outline a labeling algorithm for the multiobjective shortest paths problem and presents an analysis in terms of finiteness and optimality concepts and reports that any instance of the MSPP is bounded if and only if there are no absorbent cycles in the network. They show a set of networks wherein the labeling algorithm only determines nondominated labels. On the contrary, Mooney and Winstanley [13] state that Martins' labeling algorithm works well in theory but is prohibitive in practice in terms of its implementation due to memory costs.

A recent study by Gandibleux *et al.* [6] reports a concise description of the MSPP and clearly narrates the most salient issues to its solution. Their study recalls Martins' labeling algorithm and attempts to improve it. Their new algorithm extends Martins' algorithm by introducing a procedure that can solve MSPP that have multiple linear functions and a max min function. Since it is an extension of Martins' algorithm, the generation of all nondominated paths remains intractable in polynomial time. However, experimental results of their study say otherwise. Their algorithm is tested on a variety of test instances and results show that in terms of size and complexity, optimizing simultaneous linear and max min functions does not behave exponentially. They also show that

their algorithm is not sensitive to different cost ranges and that density and network size increase the number of efficient solutions. An independent study by [7] also shows that the cardinality of efficient paths in a bicriteria shortest path problem is not exponential as long as the instances are bounded by potential characteristics as defined in their experiment. They conclude with emphasis that it is still preferable to work with complete information rather than falling back on approximations.

Guerriero and Musmanno [14] examine several label-selection and node-selection methods that can find Pareto-optimal solutions to the MSPP with respect to execution time. The performance of the different algorithms was measured using random and grid networks and results show that label-selection methods are generally faster than node-selection methods and that parallel computing is necessary in the design of efficient methods.

While some researchers focus on exhaustive solutions or on improvements thereof, other researchers are more concerned with better runtime solutions. Tsaggouris and Zaroliagis [15] present an improved fully polynomial time approximation scheme (FPTAS) for the multicriteria shortest path problem and a new generic method for obtaining FPTAS to any multiobjective optimization problem with non-linear objectives. They show how their results can be used to obtain efficient approximate solutions to the multiple constrained path problem and to the non-additive shortest path problem. Their algorithm builds upon an iterative process that extends and merges sets of node labels representing paths which departs from earlier methods using rounding and scaling techniques on the input edge costs. The algorithm resembles the Bellman-Ford method but implements the label sets as arrays of polynomial size by relaxing the requirements for strict Pareto optimality.

Granat and Guerriero [1] introduce an interactive procedure for the MSPP based on a reference point labeling algorithm. The algorithm converts the multiobjective problem into a parametric single-objective problem whereby the efficient paths are found. The algorithm was tested on grid and random networks and its performance was measured based on execution time. They conclude that an interactive method, from their experimental results, is encouraging and does not require the generation of the whole Pareto-optimal set (which avoids the intractability problem). Likewise, [10] suggests an interactive method that incorporates an efficient *k*-shortest path algorithm in identifying Pareto-optimal paths in a bi-objective shortest path problem. The algorithm was tested against other MSPP algorithms on 39 network instances. They conclude that their *k*-shortest path algorithm performs better in terms of execution time.

From a different perspective, evolutionary algorithms (EAs) have been extensively analyzed in single objective optimization problems but only a few researchers have applied EAs to the multiobjective shortest path problem either as the main problem or as a sub-problem in relation to route planning, traffic and transport design, information systems

and communications network design. Gen and Lin [2] use a multiobjective hybrid genetic algorithm (GA) to improve solutions to the bicriteria network design problem (finding shortest paths) with two conflicting objectives of minimizing cost and maximizing flow. The paper shows how the performance of a multiobjective genetic algorithm can be improved by hybridization with fuzzy logic control and local search. The results show a positive effect of hybridization, that is, an improvement in the convergence of the Pareto front.

Kumar and Banerjee [3] present an algorithm for multicriteria network design (shortest paths and spanning trees) with two objectives of optimizing network delay and cost subject to satisfaction of reliability and flow constraints. They tested an evolutionary algorithm approach, Pareto Converging Genetic Algorithm (PCGA), to design different sized networks and found that EAs scale better in larger networks than two traditional approaches namely branch exchange heuristics and exhaustive search. They conclude that the primary advantage of EAs to solve multiobjective optimization problems is their diversity of solutions generated in polynomial time. Crichigno and Baran [17] demonstrate similar representations (spanning trees) to Kumar's for a multicast algorithm. The basic difference between both algorithms is the latter adopts the Strength Pareto Evolutionary Algorithm (SPEA) in generating efficient solutions to the multicast routing problem.

A contemporary report by [13] shows the behavior of an elitist genetic algorithm as applied to the MSPP in the field of geographic information systems (GIS). The experiment compares the runtime performance (execution time) of the EA against a modified version of Dijkstra's algorithm on several artificial and real road networks. The results show that the EA competes well with the modified Dijkstra approach in terms of execution time and that the EA converges quickly to the Pareto-optimal paths.

IV. Multiobjective evolutionary algorithm

Evolutionary algorithms are adaptive heuristic search algorithms based on the evolutionary ideas of natural selection and genetics. As such they represent an intelligent exploitation of a random search used to solve optimization problems. Although randomized, EAs are by no means random instead they exploit historical information to direct the search into the region of better performance within the search space. At each generation, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and breeding them together using operators borrowed from natural genetics. This process leads to the evolution of populations of individuals that are better suited to their environment than their ancestors, just as in natural adaptation.

A complete discussion of multiobjective evolutionary algorithms (MOEA) can be found in [17]. Also, [18] gives a summary of current approaches in MOEA and emphasizes the importance of new approaches in exploiting the capabilities of evolutionary algorithms in multiobjective optimization.

A number of MOEAs exist and Zitzler and Thiele's [19] SPEA2 is regarded as one of the better elitist multiobjective evolutionary algorithms. It has three favorable characteristics. First, SPEA2 features an excellent fitness assignment strategy that accounts for an individual's strength in terms of the individuals that it dominates and the strength of its dominators. Second, SPEA2 incorporates a density estimation technique which discriminates individuals efficiently, and third, it has an archive truncation method that prevents boundary solutions from elimination. The algorithm is summarized below.

Input: N : population size
           $\overline{N}$: archive size
           T : maximum number of generations
Output: A : non-dominated set of solutions

Step 1: Initialization. Generate an initial population $\mathbf{P}_0$ and create the empty archive (external set) $\overline{P}_0 = \varnothing$. Set t=0.

Step 2: Fitness assignment: Compute the fitness values of individuals in P_t and $\overline{P}_t$. Strength of an individual $\mathbf{i}$

$$S(\mathbf{i}) = |\{\mathbf{j} \mid \mathbf{j} \in P_t + \overline{P}_t \land \mathbf{i} \prec \mathbf{j}\}| \qquad (1)$$

where $|.|$ denotes the cardinality of a set, + stands for multi-set union and the symbol $\prec$ corresponds to the Pareto dominance relation extended to individuals.

$$\textit{Raw fitness} \qquad R(\mathbf{i}) = \sum_{\mathbf{j} \in P_t + \overline{P}_t, \mathbf{j} \prec \mathbf{i}} S(\mathbf{j}) \qquad (2)$$

For each individual $\mathbf{i}$ the distances (in objective space) to all individuals $\mathbf{j}$ in the archive and in the population are calculated and stored in a list. After sorting the list in increasing order, the k-th element gives the distance sought, denoted as σ_i^k. Two is added to the denominator to ensure that its value is greater than zero and $D(\mathbf{i}) < 1$.

$$\textit{Density} \qquad D(\mathbf{i}) = \frac{1}{\sigma_i^k + 2}; k = \sqrt{N + \overline{N}} \qquad (3)$$

$$\textit{Fitness} \qquad F(\mathbf{i}) = R(\mathbf{i}) + D(\mathbf{i}) \qquad (4)$$

Step 3: Environmental Selection. Copy all non-dominated individuals in P_t and $\overline{P}_t$ to $\overline{P}_{t+1}$. If size of $\overline{P}_{t+1}$ exceeds $\overline{N}$ then reduce $\overline{P}_{t+1}$ by means of the truncation operator, otherwise if size of $\overline{P}_{t+1}$ is less than $\overline{N}$ then fill $\overline{P}_{t+1}$ with dominated individuals in P_t and $\overline{P}_t$.

Step 4: Termination. If $t \geq T$ or another stopping criterion is satisfied then set A is the set of decision vectors represented by the nondominated individuals in $\overline{P}_{t+1}$. Stop.

Step 5: Mating selection. Perform binary tournament selection with replacement on $\overline{P}_{t+1}$ in order to fill the mating pool.

Step 6: Variation. Apply recombination and mutation operators to the mating pool and set P_{t+1} to the resulting population. Increment generation counter (t = t +1) and go to Step 2.

The public-domain package, Platform and Programming Language Independent Interface for Search Algorithms (PISA) is a ready-to-use package that implements SPEA2 for multi-objective optimization problems [20]. The package makes it possible for researchers and practitioners to specify and implement representation-independent selection modules while allowing users to define problem-specific variation operators. This paper employs the PISA platform to implement SPEA2 as the selection and fitness function operator for the MSPP. The description of genetic representation and operators follows in the next section.

A. Genetic Representation

A chromosome or an individual consists of integer-ID nodes that form a path from the source node to a sink node. The length of the chromosome is variable and may not be greater than the number of nodes, n.

B. Initial Population

A path or a chromosome is generated randomly in an ordered sequence from the source node to the sink node. The ID of the source node s is assigned to the first locus (array index) of the chromosome. The ID of a randomly generated node v_i is assigned to the second locus such that v_i belongs to the set of nodes connected to the source node s. This procedure continues iteratively for the succeeding nodes until a simple path to the sink node t is created.

Given the process to create a feasible path, the problem of population size N remains an issue. It is known that the population size of solutions increases exponentially with the number of objectives in the MSPP. There are two common options to respond to this problem, use a large population or integrate a dynamic population sizing procedure in the GA. The latter has been implemented for single-objective EAs and has shown promising results but dynamic sizing remains a challenge to MOEAs with regard to the MSPP. Hence there is no better alternative in our case but to estimate the size of the initial population. Deb's [17] approximation chart for finding the minimum population size in relation to the number of objectives as a guide such that the exploration of the search space is used in this research.

C. Fitness Function

SPEA2 first assigns a strength value $S(p)$, to each path p from the archive ($\overline{N}$) and population (N) representing the number of solutions p dominates. Refer to (1). Then the raw fitness $R(p)$ of each path p is calculated which measures the strength of p's dominators. The raw fitness acts as a niching mechanism but poorly performs when most paths in $M = N + \overline{N}$ are non-dominated, i.e. the population forms new solutions in only a few clusters, in effect compromising exploration of the search space. Refer to (2). This phenomenon is called genetic drift.

SPEA2 introduces a density estimator (3), a fitness sharing mechanism to avoid genetic drift. The density estimator is defined as the inverse of the distance of an individual in objective space to the k-th nearest neighbor. The density value is then added to the raw fitness value to give the final fitness function (4). The run-time of computing the fitness function is governed by the density estimator which is $O(M^2 \log M)$ [20].

D. Selection

SPEA2 offers two selection procedures, environmental and mating selection. The environmental selection is concerned with choosing individuals that will have to move on to the next generation archive $\overline{P}_{t+1}$ from the current archive $\overline{P}_t$ and population P_t. SPEA2 maintains an archive $\overline{P}_t$ in each generation and is composed of the "best" individuals of a fixed size $\overline{N}$ which is equal to the population size N. Two usual situations may occur in selection. First, the number of non-dominated solutions in $\overline{P}_{t+1}$ is less than $\overline{N}$. SPEA2 resolves this by adding the "best" dominated individuals from $\overline{P}_t + P_t$ to $\overline{P}_{t+1}$. Second, the number of non-dominated solutions for the next generation is greater than $\overline{N}$. SPEA2 uses a truncation procedure whereby the individual with the minimum distance to another individual is truncated until $|\overline{P}_{t+1}| = \overline{N}$. SPEA2 implements binary tournament selection with replacement to fill in the mating pool. This type of mating selects two solutions at a time in each tournament. Their fitness values are evaluated and the better solution is placed in the mating pool. The truncation operator dominates the runtime complexity of the selection procedure and takes $O(M^2 \log M)$ on the average and $O(M^3)$ on the worst case [20].

E. Recombination

The crossover scheme is an adaptation of the one-point crossover. For each pair of paths a locus is randomly selected from one of the chromosomes (the shorter path in terms of number of nodes) and the node ID of the locus is matched with the genes in the other chromosome. If there is a match then crossover is performed otherwise two new paths are selected for crossover until the mating pool is empty. It should be easy to see that the loci of both individuals need not be the same.

F. Mutation

In the mutation operator, a locus is randomly selected from the chromosome. The algorithm proceeds by employing the method in the initialization process (as described previously) to create a new path, only the start node is replaced by the locus.

V. EXPERIMENTS AND RESULTS

A. Experimental Design

The experiments intend to show the behavior of a multiobjective evolutionary algorithm as applied to the MSPP in terms of diversity and optimality of solutions, and computational complexity. The datasets are random networks that have been generated by [6]. Nine configurations are

selected for the random networks: (a) number of vertices: 50, 100, 200; (b) density of the network: 5%, 10%, 20%; (c) range of cost values $c_p(i,j)$: [1, 100], p=1, 2, 3. Five instances are generated for each network configuration and two objective configurations are considered for finding efficient paths, (3-S) and (2-S|1-M). S-type objectives are sum problems that are to be minimized whereas M-type objectives are max-min problems that are to be maximized.

The population size of 100 is estimated from [17] with the assumption that 5% of nondominated paths are present in the initial population. The probability for mutation and recombination is 1.0 and 0.2 respectively. Initial tests show that decreasing the mutation probability and increasing the recombination probability reduces the number of efficient solutions.

Efficient paths are found from a single source node (Node 1) to a single sink node (Node 50, 100, 200) for each objective configuration. For each network configuration, the number of efficient paths is computed from three different generation runs, 50, 100, and 200 respectively.

B. Results

After examination of the solutions from 270 combinations, Fig. 2 shows that the number of efficient paths generally increases as the number of generations increases. In the 3-S configuration, there is a growth of efficient paths as network density increases when the network size is 50 and 100 but this trend is not evident when n=200. The graph also shows that there are more efficient paths generated in the 2-S|1-M than in the 3-S configuration but the number of efficient paths decreases with the increase in density and network size.
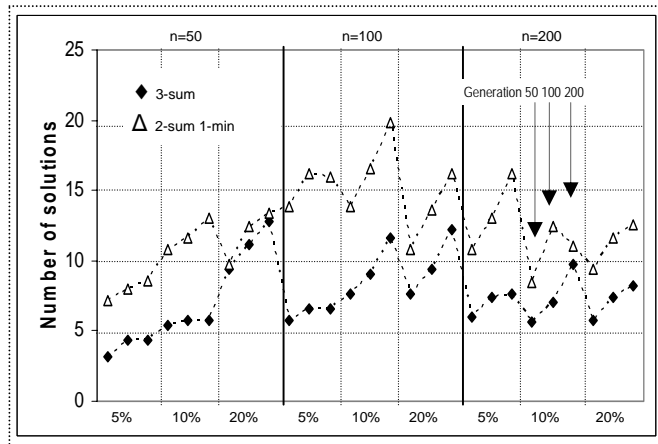

Fig. 2 Average number of paths for all resolutions

In terms of the diversity of solutions, Fig. 3 and Fig. 4 show value path plots of a sample result from all the resolutions examined for the 3-S and 2-S|1-M objective configurations. The value path plot provides information on how good an algorithm is in finding diverse solutions and good trade-off nondominated solutions for problems having more than two objectives. A good spread of solutions over a range implies that an algorithm is good in finding diverse solutions. And a large change in slope between objectives implies good trade-off nondominated solutions. Both figures show that the MOEA solutions are well spread over the range for objectives 1 and 2 but not so well spread for objective 3. Also, the figures illustrate that the MOEA finds good trade-off nondominated paths for both objective configurations.
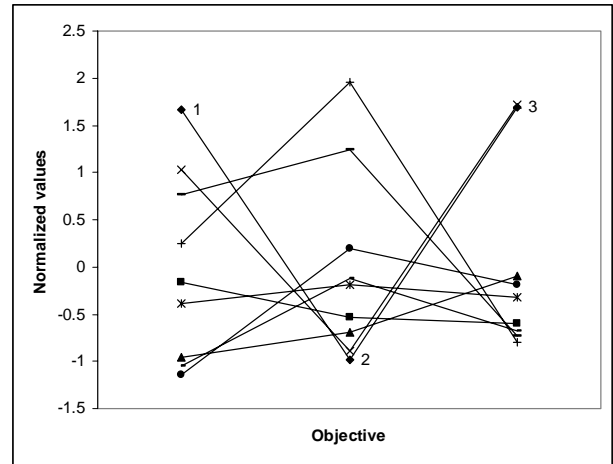

Fig. 3 A value path plot for 3-S configuration, 200-node network with 20% density.
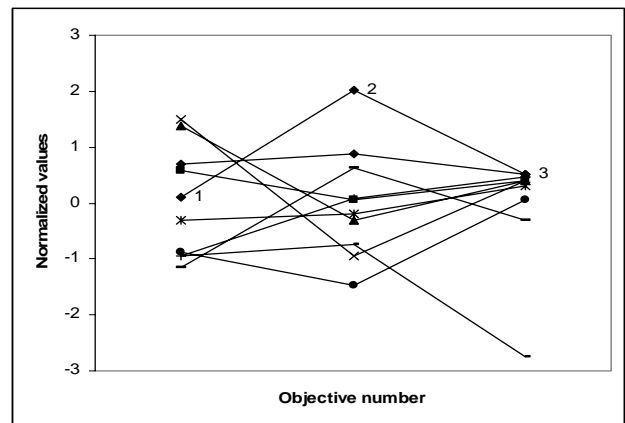

Fig. 4 A value path plot for 2-S|1-M configuration, 200-node network with 20% density.

Another issue for the evaluation of the MOEA is the closeness of its solutions to the Pareto-optimal solutions. In the absence of the Pareto-optimal solutions of the MSPP instances, it is difficult to evaluate the MOEA in terms of proximity to the Pareto-optimal fronts. However, the MOEA as shown in Fig. 2 generates more efficient paths as the number of generation increases regardless of density and network size. This means that the MOEA nondominated paths improve and move to the location of the Pareto-optimal solutions. While there is no assurance that the nondominated solutions will converge to the Pareto-optimal front or the maximal set of efficient paths, the MOEA finds a subset of diverse and good nondominated trade-off solutions at each generation and improves as the number of generation increases.

With regard to computational complexity, the MOEA is a

polynomial time algorithm with respect to the number of nodes and edges of a network. Although the CPU runtime takes longer as compared to the results in [6], unlike exhaustive algorithms, the MOEA does not suffer from intractability and memory problems when the density is high and the network size is large.

VI. CONCLUSION

This paper has presented the feasibility of a multiobjective evolutionary algorithm as applied to the multiobjective shortest path problem. Results show that the MOEA is a good alternative in finding a subset of efficient solutions to the MSPP. It is particularly beneficial when intractability and memory issues become obstructions to finding efficient solutions to the MSPP-related problems.

REFERENCES

[1] J. Granat and F. Guerriero, "The interactive analysis of the multicriteria shortest path problem by the reference point method", *European Journal of Operational Research*, vol. 151, pp. 103-111, 2003.

[2] M. Gen and L. Lin, "Multiobjective genetic algorithm for solving network design problem", presented at the 20th Fuzzy Systems Symposium, Kitakyushu, Japan, June 2004.

[3] R. Kumar and N. Banerjee, "Multicriteria network design using evolutionary algorithm", in *Lecture Notes in Computer Science* vol. 2724, Berlin: Springer Verlag, 2003, pp. 2179-2190.

[4] M. Ehrgott and X. Gandibleux, "A survey and annotated bibliography of multi-objective combinatorial optimization", *OR Spektrum* vol. 22, pp. 425-460, 2000.

[5] J. Garey, *Computers and Intractability: A Guide to the Theory of NP-completeness*, San Francisco, CA : Freeman.,1979.

[6] X. Gandibleux, F. Beugnies, and S. Randriamasy, "Martins' algorithm revisited for multi-objective shortest path problems with a MaxMin cost function", *4OR A Quarterly Journal of Operations Research*, vol. 4, pp. 47-59, 2006.

[7] M. Müller-Hannemann, and K. Weihe, "Pareto Shortest Paths is Often Feasible in Practice", in *Lecture Notes in Computer Science*, vol 2141, G. Brodal, , D. Frigioni, A. Marchetti-Spaccamela Eds. Berlin: Springer Verlag, 2001, pp. 185-198.

[8] P. Hansen, "Bicriteria path problems", in *Lecture Notes in Economics and Mathematical Systems,* vol 177, G. Fandel & T. Gal Eds. Berlin: Springer Verlag, 1979, pp.109-127.

[9] A. Warburton, "Approximation of Pareto-optima in multiple-objective shortest path problems", *Operations Research,* vol. 35, pp. 70-79. 1987.

[10] J. Coutinho-Rodrigues, J. Climaco, and J. Current, "An interactive bi-objective shortest path approach: searching for unsupported nondominated solutions", *Computers & Operations Research* vol. 26, pp. 789-798, 1999.

[11] J. Granat and F. Guerriero, "The interactive analysis of the multicriteria shortest path problem by the reference point method", *European Journal of Operational Research,*, vol. 151, pp. 103-111, 2003.

[12] E. Martins and J. Santos, "The labeling algorithm for the multiobjective shortest path problem", Departamento de Matematica, Universidade de Coimbra, Portugal, Tech. Rep. TR-99/005, 1999.

[13] P. Mooney and A. Winstanley, "An evolutionary algorithm for multicriteria path optimization problems", *International Journal of Geographical Information Science*, vol. 20, pp. 401-423, 2006.

[14] F. Guerriero and R. Musmanno, "Label correcting methods to solve multicriteria shortest path problems", *Journal of Optimization Theory and Applications,* vol. 111, pp. 589-613, 2001.

[15] G. Tsaggouris and C. Zaroliagis, "Multi-objective optimization: improved FPTAS for shortest paths and non-linear objectives with applications", Computer Technology Institute, University of Patras, Greece , Tech. Rep. TR 2006/03/01, 2006.

[16] J. Crichigno and B. Baran, "A multicast routing algorithm using multi-objective optimization", in *Lecture Notes in Computer Science*, vol 3124, J. Neuman de Souza, D. Dini and P. Lorenz Eds. Berlin: Springer Verlag, 2004, pp. 1107-1113.

[17] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, Chichester, UK: J. Wiley & Sons, 2004.

[18] C. Coello, "An updated survey of GA-based multiobjective optimization techniques", *ACM Computing Surveys,* vol. 32, pp. 109–143, 2000.

[19] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative study and the strength Pareto approach", *IEEE Transactions on Evolutionary Computation*, vol. 3, pp. 257-271, 1999.

[20] Zitzler, Eckart, Bleuler, S., Laumanns, M. and L. Thiele, "PISA- a platform and programming language independent interface for search algorithms", Swiss Federal Institute of Technology, Tech. Rep. TIK-Report No. 154, 2003.