An Evolutionary Algorithm for the Multi-objective Shortest Path Problem

Fangguo He Huan Qi Qiong Fan

Institute of Systems Engineering, Huazhong University of Science & Technology, Wuhan 430074, P. R. China

Abstract

In this paper, a multi-objective shortest path evolutionary algorithm (MOSPEA) is presented for an undirected network. The algorithm differs from the others as it uses an external set to maintain non-dominated solutions and different selection strategy. Property of MOSPEA about stochastic convergence is analyzed. Simulation experiments demonstrate the availability and efficiency of MOSPEA. The study shows that MOSPEA may hopefully be a feasible approach for multi-objective shortest path.

Keywords: Shortest path, Evolutionary algorithm, Multi- objective, Combinatorial optimization

1. Introduction

The shortest-path problem is a classical combinatorial optimization problem in network theory. This problem is to find the shortest route between a source and a destination in a given network. Many complicated problems can be described by it, for instance network routing, vehicle routing, path planning and scheduling etc. [1]-[3]. Several good methods have been proposed to solve the problem including the well-known Dijkstra algorithm, the Bellman's dynamic programming algorithm, the Bellman-Ford algorithm, and the heuristic algorithm [4].

For the traditional shortest path problem, the objective function is often single, such as minimum cost or minimum distance. In fact, many shortest-path choosing problem in our real life is a shortest path decision problem based on multi-objective, for instance, we might want to find a path for a lorry that minimizes the distance and cost; or to find the route for data transmission that minimizes its cost, delay and number of hop in a computer network. This optimization problem, called multi-objective shortest path, is required able to deal with multiple objectives, i.e. minimizes each objective function. In principle, multi-objective optimization (MOP) is quite different from the single-objective optimization. Multiple objective are usually conflicting and there exists no best solution to the problem, but a set of Pareto optimal solutions representing the best compromise among the objectives. Pareto optimal solutions are characterized that no other solution exists being superior in all objectives. For this research, a Pareto optimal solution means a Pareto optimal path, and our algorithm will look for the set of Pareto optimal paths.

There are some standard methods for dealing with multi-objective optimization problems, such as the linear weighting method, the distance function method and the constraint method [5]-[6], that have to transform the multi-objective problem into a single objective problem which can be solved using nonlinear optimization. Recently, evolutionary algorithms (EA) were found useful for solving MOP, and many kinds of evolutionary methods are proposed to solve the problem [7]. In this paper we proposed a EA to solve the multi-objective shortest path problem, and tested the algorithm by random network. The experimental results show the EA may hopefully be an efficient approach for multi-objective shortest path based network optimization problems.

2. Problem formulation

Let $G = (V, A)$ be an undirected graph in which $V = \{1,2\ldots n\}$ is the set of vertices (nodes) and $A \subseteq \{(i, j) | i, j = 1, 2 \cdots n; i \neq j\}$ is the set of arcs (edges). Without loss of generality, we only considered the graph in which there exists at most one edge between a pair of ordered nodes. For a given graph, each edge connecting two nodes i and j is specified by a weight vector $W_{ij} = (w_{ij}^1, w_{ij}^2, \ldots w_{ij}^K)$, where W_{ij} is a K-degree vector having K additive arc metrics $w_{ij}^k > 0$. A path from the source node s to the end node t is represented as a sequence of nodes ($s\ l\ m\ \ldots k\ t$). On the additivity of the arc metrics, the path P=(s, $l, \ldots\ k, t$) is specified by a k-degree vector $W(P) = (W_1(P), W_2(P), \ldots W_K(P))$ where for each component $W_k(P)$ the following holds:

$$W_k(P) = \sum_{(i,j)\in P} w_{ij}^k \quad (k = 1, \cdots K)$$

In other words, the vector $W(P)$ of the path P consists of the vector sum of the vectors corresponding to arcs. In general a multi-objective shortest path is a kind of vector optimization problem, which can be described as follows:

$$\textit{Minimize}\ \ W(P) = (W_1(P), W_2(P), \ldots W_K(P))$$
$$\text{s.t} \quad P \in P_{st}$$

where $P_{st,}$ called feasible set, is the set of paths between node s and node t, P decision variable, and $W(P)$ objective vector. In order to measure the performance of different path, we give the following definition.

Definition 1: Let P_1, $P_2 \in P_{st}$, path P_1 is said to dominate path P_2 (also written as $P_1 \succ P_2$) if and only if $W(P_1)$ is partially less than $W(P_2)$, i.e., $\forall\, i \in \{1, \ldots, n\}$, $W_i(P_1) \le W_i(P_2)$ and $\exists\, i \in \{1, \ldots, n\}$: $W_i(P_1) < W_i(P_2)$

Definition 2: A path $P^* \in P_{st}$ is said to be Pareto optimal path or a non-dominated path if and only if there is no path $P \in P_{st}$ such that $W_i(P) \ge W_i(P^*)$ for all $i \in \{1,2,\ldots,n\}$, with at least one strict inequality. The set of all Pareto optimal paths is called the Pareto optimal set, and the set of all non-dominated objective vectors is called the Pareto front (PF).

3. Proposed EA for the problem

In this study, we will use a multi-objective evolutionary algorithm to solve the problem without combining the sub-objective functions into a single objective function. Our algorithm will look for the set of non-dominated paths for the user, and give the liberty to choose the best path from the set depending on the specific requirement. The algorithm called multi-objective shortest-path evolutionary algorithm (MOSPEA) is as follows:

1: Generate random initial population *Pop* with size N and create the empty external set of non-dominated individuals *Pop'*.

2: sort the population based on non-domination and calculate the niche count of each individual

3: Remove solutions within *Pop'* and copy non-dominated members of *Pop* to *Pop'*.

4: Select individuals from *Pop* by using rank and niche count, until the mating pool is filled

5: Perform crossover and mutation to members of the mating pool

6 Combine the offspring population with the external set *Pop'* and perform selection to the recombined members in order to create a new population *Pop*.

7: If maximum number of generations is reached, then copy non-dominated members of *Pop* to replace the one of *Pop'* and stop, else go to Step 3.

3.1. Encoding and initial population

In this paper, we adopt a direct encoding method, which a chromosome represents a path from the source node to the end node. A gene in each chromosome indicates the ID of a node and a gene string uniquely determines a path in the graph. The length of the chromosome is variable, but it should not exceed the maximum length n, where n is the total number of nodes. For example, a path consisting of nodes 1, 5,11 and 14 can be coded as a string (1, 5, 11, 14).

The depth first search algorithm is utilized to produce the initial population. By using of the algorithm repeatedly, a set of possible paths from source node to destination node with specific number N is found, and it is regarded as initial population.

3.2. Fitness value and niche count

In step 2, all individuals in *Pop* are assigned a scalar fitness value. This is accomplished in the following process. First, evaluate the sub-objective functions $W_i(P)$ (i = 1,2…K) for each individual in *Pop* in parallel. Afterwards, the individuals are sorted based on non-domination into each front. The first front F_1 being completely non-dominant set in the current population and the second front F_2 being dominated by the individuals in the first front only and the front goes so on. Individuals in first front are given a rank (fitness) value of 1 and individuals in second are assigned rank value as 2 and so on.

Once the non-dominated sort is complete the niche count is assigned. Since the individuals are selected based on rank and niche count. The niche count is a measure of how similar an individual is to its neighbors. Large average niche count will result in better diversity in the population.

Aiming at the specific problem of this research, the niche count is calculated as below:

$$m(p) = \sum_{q \in Pop} sh(d_{pq}) \qquad (1)$$

Where $m(p)$ denotes the niche count of the individual p, d_{pq} is the distance between individuals p and q and $sh(d_{pq})$ is the sharing function. The expression of d_{pq} and $sh(d_{pq})$ is written as

$$d_{pq} = \sqrt{\sum_i (\Delta W_i(pq))^2}\ , i = 1, 2 \ldots K \quad (2)$$

$$sh(d_{pq}) = \begin{cases} 1 - \dfrac{d_{pq}}{\sigma}, d_{pq} \le \sigma \\ 0, \quad d_{pq} > \sigma \end{cases} \qquad (3)$$

Here $\Delta W_i(pq) = W_i(p) - W_i(q)$, σ is the niche radius, and it is a good estimate of minimal separation expected between the goal solutions. We introduce a new concept of adaptive sharing, i.e., the value of σ is not kept fixed but dynamically updated in every iteration of the algorithm. Let $\Delta W_i = max\{|W_i(p) - W_i(q)|: p,q \in Pop\}$, the niche radius is defined as following

$$\sigma = \frac{1}{2}\sqrt{\sum_{i=1}^{K} \Delta W_i^{\,2}} \qquad (4)$$

3.3. The process of evolution

① Selection

The individuals are selected from the population by using binary tournament selection based on the rank and niche count. An individual is selected in the rank is less than the other or if niche count is greater than the other. For example, given two individuals P_i and P_j, if rank(P_i) < rank(P_j) or if P_i, P_j belong to the same front and $m(P_i) > m(P_j)$ i.e. the niche count is more, then the individual P_i is selected.

② Crossover and mutation

Two chromosomes can be crossed if they have at least one pair of same gene (node) except for source and destination nodes. If there are more pairs of common genes, one pair is randomly chosen and the locus of each node becomes a crossing site of each chromosome. It is possible that loops are formed during crossover. The strategy of repair operator is adopted to eliminate the infeasible chromosomes. Firstly, examine loops in the offspring chromosome and mark the location of their start and end. Secondly, find out the largest loop and remove the genes within the largest loop from the chromosome. The proposed repair operator is described in following example:

parent1: **| 1 2 7 3 5 6 9 |** ***parent2:*** **| 1 3 4 6 7 8 9 |**
offspring1: **| 1 2 7 3 4 6 7 8 9 |** (repair)→ **| 1 2 7 8 9 |**
offspring2: **| 1 3 5 6 9 |**

In order to perform a mutation, a node, called mutation point, is randomly selected from the chosen path. Then, starting from the node, we use the depth first search to find the following nodes until the end node is found. It is noticed that loops may be produced in the course of mutation, thus the repair operator is also performed to the offspring chromosomes.

3.4. Recombination and Selection

The offspring population is combined with the external set of non-dominated individuals *Pop'*, and selection is performed to form the individuals of the next generation. Similarly, Population is now sorted based on non-domination. The new generation is filled by each front subsequently until the population size exceeds the current population size. If by adding all the individuals in front F_i the population exceeds N then individuals in front F_i are selected based on their niche count in the descending order until the population size is N. Obviously, elitism is ensured for all the previous and current best individuals are added in the population.

3.5. The analysis of Convergence

Definition 3: A evolutionary algorithm is said to converges to the global optimal solution S^* with probability one if the following holds:

$$p(\lim_{t\to\infty} \left\| S^{(t)} - S^* \right\| \le \varepsilon) = 1$$

where $S^{(t)}$ is the solution of t-th iteration.

Lemma 1: A evolutionary algorithm converges to the global optimal solution with probability one if the algorithm could meet the following two conditions [8]:

(1) Let I be feasible set, ∀ x, $x' \in$ I, x' is reachable from x by means of crossover and mutation;

(2) The solution sequence Q^1, Q^2, ...Q^t is monotone, i.e., ∀ t: every number of Q^{t+1} is not more inferior than that of Q^t.

Theorem1 (MOSPEA convergence): MOSPEA converges to the global optimal with probability one.

Proof: It was proved that MOSPEA would converge with probability one if it could satisfy the two conditions of lemma 1. By the proposed algorithm, mutation and crossover operators allow every point in a search space to be visited. Thus ∀ P, $P' \in P_{st}$, P' is reachable from P.

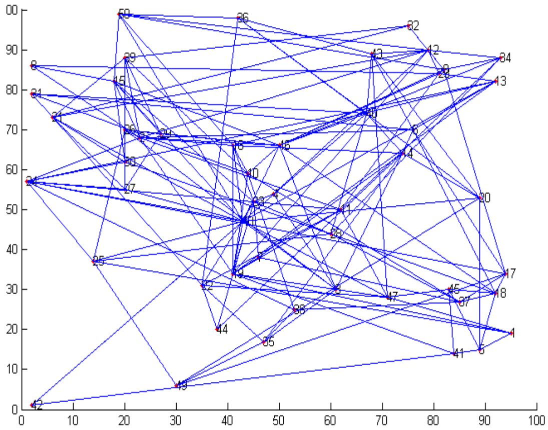
We keep in mind that the proposed algorithm use a external set *Pop'* as all elitist mechanism in order to maintain the best solutions, i.e., in every iteration the current best individual strings survive. Intuitively, as the algorithm iterating, the fitness of the strings in *Pop'* does not decrease. Thus, the sequence $Pop'^{(1)}$, $Pop'^{(2)}$, ...$Pop'^{(t)}$ is monotone.

4. Illustrative example

In the example, a undirected network with 50 nodes is generate random based on Waxman's model [9]. In the network, the edge's probability can be expressed by

$$p(u, v) = \alpha \exp(-d(u,v)/\beta D)$$

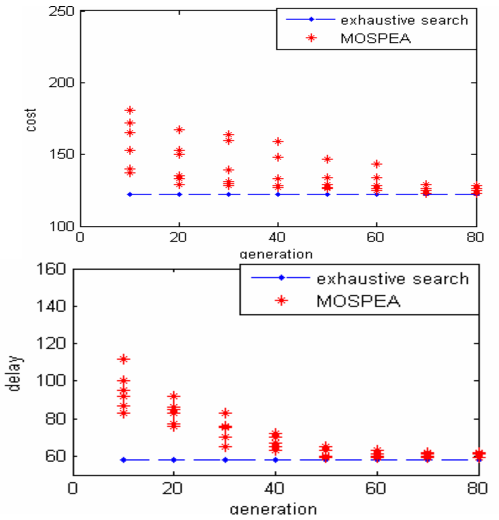
Where $d(u,v)$ is geometric distance from node u to node v, D is maximum distance between two nodes, parameter β determines the scale of short edge and long edge of the network, and parameter α determines the average degree of the network. Fig. 1 shows the topology of the data transmission network we consider. For any link, we can define a three-degree vector $W_{ij} = (w_{ij}^1, w_{ij}^2, w_{ij}^3)$, where three metrics w_{ij}^1, w_{ij}^2, w_{ij}^3 denote cost, delay and number of hop, respectively. Assumed that the cost w_{ij}^1 and the delay w_{ij}^2 are all randomly generated and $w_{ij}^3 = 1$, here w_{ij}^1, w_{ij}^2 is uniformly distributed in the range of [20, 30], [1, 20], respectively. Let 1 be source node and 50 be end node of a path. In this example, the multi-objective shortest path problem is to find a shortest path from source node 1 to destination node 50, such that the cost, delay and number of hop are minimal.



.

Fig. 1: The topology of network

We utilize the proposed algorithm to simulate the model, which were performed with MATLAB. The parameters setting for our experiment are population size 20, the maximum generation 80, crossover rate 0.7 and mutation rate 0.2. Our algorithm is compared to the exhaustive search method, which finds the optimal values of the three parameters by exhaustively searching them one after another. Fig.2 shows that how these Pareto-optimal fronts are developed and converged towards global-optimal solution, and the table 1 explains the ratio of Pareto paths during evolution.



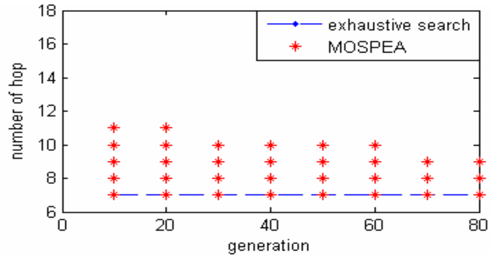


Fig. 2: Evolutionary process

generation	20	40	60	80
ratio	6%	23%	41%	60%

Table 1: The ratio of Pareto paths

5. Conclusions

The shortest path problem is currently an active area of research. In real need this optimization problem is often required able to deal with different and conflicting multiple objectives. In this paper, we proposed a feasible multi-objective shortest-path evolutionary algorithm (MOSPEA) for solving the problem. The algorithm differs from the others as it uses an external set to maintain non-dominated solutions and different selection strategy. Property of MOSPEA about stochastic convergence is analyzed. Simulation experiments demonstrate the availability and efficiency of MOSPEA. Our study shows that MOSPEA may hopefully be an efficient approach for multi-objective shortest path.

References

[1] P.V. Mieghem and A. K. Fernando., Concepts of Exact QoS Routing Algorithms. *IEEE/ACM Transactions on networking*, 12: 851-864, 2004.

[2] J. Chen and F. Lu, Implementation and Evaluation of the Shortest Path Labeling Algorithms in Transportation Networks. *Journal of Image and Graphics*, 10: 1134-1138, 2005.

[3] Q. Gao, J. Z. Luo, A tabu-search-based fast QoS multicast routing optimal algorithm. *Journal of Software*, 15: 1877-1884, 2004.

[4] W. A. Chang and R. S. Ramakrishna, A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations. *IEEE Transactions on evolutionary computation*, 6: 566-579, 2002.

[5] Manolis Papadrakakis et al, Multi-objective optimization of skeletal structures under static and seismic loading conditions. *Eng. Opt.*, 34: 645–669, 2002.

[6] M. Zeleny, Multiple Criteria Decision Making. *McGraw-Hill*, New York, 1982.

[7] T. Lan and S.R Liu., Approaches of Evolutionary Multiobjective Optimization. *Control and Decision*, 21: 601-611, 2006.

[8] D.V. Veldhuizen and G.B. Lamont, Evolutionary computation and convergence to a pareto front. *In: The genetic programming 1998 conference*, pp..221-228, 1988.

[9] B.M. Waxman, Routing of multipoint connections. *IEEE Journal of Selected Area in Communications*, pp. 1617–1622, 1988.