Strength Pareto Evolutionary Algorithm based Multi-Objective Optimization for Shortest Path Routing Problem in Computer Networks

[1]Subbaraj Potti and [2]Chitra Chinnasamy
[1]Department of Electronics and Communication Engineering,
Theni Kammavar Sangam College of Technology,
Koduvilarpatti, Theni, 625 534, Tamilnadu, India
[2]Department of Electronics and Communication Engineering,
Arulmigu Kalasalingam College of Engineering,
Anand Nagar, Krishnankoil, 626 190, Tamil Nadu, India

Abstract: Problem statement: A new multi-objective approach, Strength Pareto Evolutionary Algorithm (SPEA), is presented in this paper to solve the shortest path routing problem. The routing problem is formulated as a multi-objective mathematical programming problem which attempts to minimize both cost and delay objectives simultaneously. **Approach:** SPEA handles the shortest path routing problem as a true multi-objective optimization problem with competing and noncommensurable objectives. **Results:** SPEA combines several features of previous multi-objective evolutionary algorithms in a unique manner. SPEA stores nondominated solutions externally in another continuously-updated population and uses a hierarchical clustering algorithm to provide the decision maker with a manageable pareto-optimal set. SPEA is applied to a 20 node network as well as to large size networks ranging from 50-200 nodes. **Conclusion:** The results demonstrate the capabilities of the proposed approach to generate true and well distributed pareto-optimal nondominated solutions.

Key words: Shortest path routing problem, evolutionary algorithm, multi-objective optimization, clustering, strength pareto evolutionary algorithm, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Non-dominated Sorting Genetic Algorithm (NSGA), Quality of Service (QoS), dynamic programming, communication networks

INTRODUCTION

A computer network is an interconnected group of computers with the ability to exchange data. Today, computer networks are the core of modern communication. Routing problem is one of the important research issues in communication networks (Jayakumar and Gopinath, 2008). An ideal routing algorithm should strive to find an optimal path for packet transmission within a specified time so as to satisfy the Quality of Service (QoS). The objective functions related to cost, time, reliability and risk are appropriated for selecting the most satisfactory route in many communication network optimization problems. Traditionally, the routing problem has been a single-objective problem of minimization of either cost or delay. However, it is necessary to take into account that many real world problems are multi-objective in nature and so is the shortest path routing problem in computer networks.

Current routing protocols use a simple metric and shortest path algorithm so as to work out the routes. In QoS routing, routes must be determined by requirements based on features of the data flows, such as cost, delay, and bandwidth (Murad and Al-Mahadeen, 2007). There are two main goals that are to be achieved by the QoS routing algorithm. The first goal is to find a path that satisfies the QoS requirements. The second goal is to optimize the global network resource utilization (Rouskas and Baldine, 1997). Many applications, such as audio, video conferencing or collaborative environments and distributed interactive simulations have multiple QoS requirements such as bandwidth, packet delay, packet loss, cost (Sriram *et al.*, 1998).

Extensive research has been done on shortest path routing problems. It includes Dynamic Programming for directed networks, Dijkstra labeling algorithm and

Corresponding Author: C. Chitra, Department of Electronics and Communication Engineering,
Arulmigu Kalasalingam College of Engineering, Anand Nagar, Krishnankoil-626 190,
Tamil Nadu, India Tel: +91-4563-289042 Fax: +91-4563-289322

Bellman-Ford successive approximation algorithm (Lawler, 1976). These algorithms have major shortcomings such as they search only for the shortest route and they exhibit high computational complexity for real-time communications. Artificial Neural Networks (ANN) has been examined to solve the shortest path problem relying on their parallel architecture to provide a fast solution (Araujo *et al.*, 2001). However, the ANN approach has several limitations. These include the complexity of the hardware which increases considerably with increasing number of network nodes; at the same time, the reliability of the solution decreases. Secondly, they are less adaptable to topological changes in the network graph. Evolutionary algorithms such as Genetic Algorithm (GA) (Ahn and Ramakrishna, 2002) and Particle Swarm Optimization (PSO) (Mohemmed *et al.*, 2008) have been used. However, the approaches are meant to find single-objective optimization of either cost or delay, mostly cost only. It is apparent that there is a need for more efficient algorithm which gives multi-objective trade-off solutions involving cost, delay and bandwidth.

The simple multi-objective method is to form a composite objective function as the weighted sum of the objectives, where a weight for an objective is proportional to the preference factor assigned to that particular objective. This method of secularizing an objective vector into a single composite objective function converts the multi-objective optimization problem into a single-objective optimization problem. In an ideal multi-objective optimization procedure, multiple trade-off solutions are found. Higher level information is used to choose one of the trade-off solutions. It is realized that, single-objective optimization is a degenerate case of multi-objective optimization. Srinivas and Deb (1994) developed Non-dominated Sorting Genetic Algorithm (NSGA) in which a ranking selection method emphasizes current non-dominated solutions and a niching method maintains diversity in the population. Chitra and Subbaraj (2010) applied NSGA to shortest path routing problem and compared its validity with single-objective optimization. However, NSGA suffers from three weaknesses: computational complexity, non-elitist approach and the need to specify a sharing parameter.

In this study, SPEA is applied to route the traffic in communication network. Simulation results considering two objectives are presented for a few sample test networks. A hierarchical clustering technique is implemented to provide the routing problem with a representative and manageable Pareto-optimal set. To analyze and compare the quality of solution produced by SPEA, the same problem is solved as a single-objective optimization by weighted sum method. The effectiveness and potential of the proposed approach to solve the multi-objective shortest path routing problem are demonstrated.

Problem formulation: The routing problem is formulated as a multi-objective mathematical programming problem which attempts to minimize both delay and cost simultaneously, while satisfying the flow conservation constraints (Lin and Gen, 2007).

The topology of a multi-hop network is specified by an undirected graph, where the set of nodes is V and the set of its link is E. There is a cost C_{ij} associated with each link. The costs are specified by the cost matrix C = $[C_{ij}]$, where C_{ij} denotes a cost of transmitting a packet on link (i,j). There are three basic concepts of delay, viz. switching delay, queuing delay and propagation delay. It is specified by the delay matrix D = $[d_{ij}]$, where d_{ij} denotes the propagation delay of transmitting a packet on link (i,j). Since switching delay is a consistent value, it is added to the propagation delay. Also, queuing delay makes little difference in the computation, the total delay is taken as only propagation delay. Source and destination nodes are denoted by S and D, respectively. Each link has the link connection indicator denoted by X_{ij}, providing information on whether the link from node i to node j is included in a routing path or not. If the link is used then the binary variable is 1 else it is 0. A path from node V_i to node V_j is a sequence of nodes from V in which no node appears more than once. A path can also be equivalently represented as a sequence of nodes $(V_i, V_l, . . . ,V_k, V_j)$. For example, referring to Fig. 1, (1, 2), (2, 3), (3, 8) and (8, 14), (14, 20) is a path from node 1 to node 20 (Ahn and Ramakrishna, 2002). The path representation is (1, 2, 3, 8, 14, 20). Thus, the problem is to find a path between the source and destination nodes having minimum total cost and minimum end to end delay.
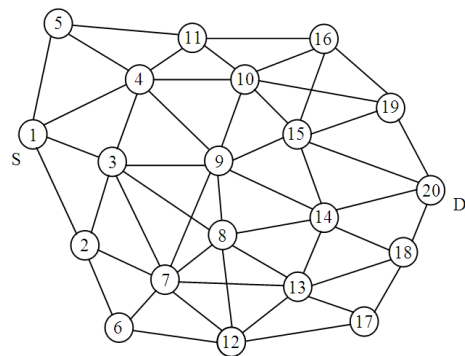

Fig. 1: A simple undirected graph with 20 nodes and 48 edges

Objective functions:
Cost: The total cost function is the sum of cost of link along the path from the source to the destination. The cost can be expressed as:

Minimize:

$$f_1 = \sum_{(i,j)\in E} C_{ij}X_{ij} \qquad (1)$$

Delay: The total delay function is the sum of delay of the link along the path from the source to the destination.

The delay can be expressed as:

Minimize:

$$f_2 = \sum_{(i,j)\in E} d_{ij}X_{ij} \qquad (2)$$

Subject to the constraints:

$$\sum_{(i,j)\in E} X_{ij} - \sum_{(i,j)\in E} X_{ji} = 1; \, i = S \qquad (3)$$

$$\sum_{(i,j)\in E} X_{ij} - \sum_{(i,j)\in E} X_{ji} = -1; \, i = D \qquad (4)$$

$$\sum_{(i,j)\in E} X_{ij} - \sum_{(i,j)\in E} X_{ji} = 0; \, i \neq S, i \neq D \qquad (5)$$

$$X_{ij} = 0 \text{ or } 1 \qquad (6)$$

Constraints (3), (4) and (5) are flow conservation constraints. Constraint (3) ensures that the total flow emerging from ingress node to egress node should be 1. Constraint (4) ensures that the total flow coming towards an egress node should 1. Constraint (5) ensures that for any intermediate node, the sum of their output-flows minus the input-flows should be zero. The variable X_{ij} in (6) takes values 0 or 1, to show whether or not the link (i, j) is used to carry information to the egress node D.

Concepts of multi-objective optimization: Many real-world problems involve simultaneous optimization of several objective functions. Generally, these functions are noncommensurable and often competing and conflicting objectives. Multi-objective optimization with such conflicting objective functions gives rise to a set of optimal solutions, instead of one optimal solution. The reason for the optimality of many solutions is that no one can be considered to be better than any other with respect to all objective functions. These optimal solutions are known as Pareto-optimal solutions (Abido, 2003; 2006; Bueno and Oliveira, 2010; Mendoza *et al.*, 2006; Taher, and Tabei, 2008). A general multi-objective optimization problem consists of a number of objectives to be optimized simultaneously and is associated with a number of equality and inequality constraints. It can be formulated as follows.

Minimize:

$$f_m(x) \; m = 1,2,3, \dots M_{objectives} \qquad (7)$$

Subject to:

$$\lambda_g(x) = 0, \, g = 1,2,3,\dots G \qquad (8)$$

$$\mu_h(x) \leq 0, \, h = 1,2,3,\dots H \qquad (9)$$

Where:
f_m = The mth objective function
x = A decision vector that represents a solution
$M_{objectives}$ = The number of objectives

For a multi-objective optimization problem, any two solutions can have one of two possibilities: one dominates or covers the other or none dominates the other. In a minimization problem, without loss of generality, a solution covers or dominates if the following two conditions are satisfied:

$$\forall k \in \{1,2,\dots,M_{objectives}\}: f_k(x_1) \leq f_k(x_2) \qquad (10)$$

$$\exists l \in \{1,2,\dots,M_{objectives}\}: f_l(x_1) < f_l(x_2) \qquad (11)$$

If any of the above conditions is violated, the solution x_1 does not dominate the solution x_2. If x_1 dominates x_2, the solution, x_1 is called the nondominated solution. The solutions that are nondominated within the entire search space are denoted as Pareto-optimal and constitute the Pareto-optimal set. This set is also known as Pareto-optimal front.

Principle of strength Pareto evolutionary algorithm:
Overview: The studies on evolutionary algorithms have shown that these algorithms can be efficiently used to eliminate most of the difficulties of classical methods that can be summarized as:

- An algorithm has to be applied many times to find multiple Pareto-optimal solutions
- Most algorithms demand some knowledge about the problem being solved
- Some algorithms are sensitive to the shape of the Pareto-optimal front
- The spread of Pareto-optimal solutions depends on efficiency of the single-objective optimizer
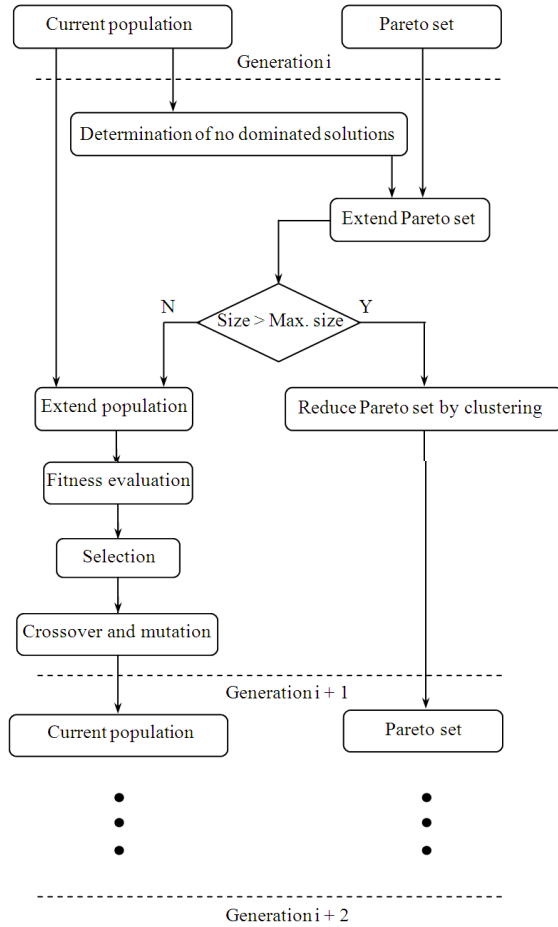
Fig. 2: Strength Pareto evolutionary algorithm

In general, the goal of a multi-objective optimization algorithm is not only guide the search towards the Pareto-optimal front but also maintain population diversity in the set of nondominated solutions (Zitzler and Thiele, 1999; Deb, 2001; 2008).

Procedure: The basic elements of the SPEA technique are briefly stated and defined as follows.

External set: It is a set of Pareto-optimal solutions. These solutions are stored externally and updated continuously. Ultimately, the solutions stored in this set represent the Pareto-optimal set.

Strength of a Pareto optimal solution: It is an assigned real value $s \in [0,1)$ for each individual in the external set. The strength of an individual is proportional to the number of individuals covered by it.

Fitness of population individuals: The fitness of each individual in the population is the sum of the strengths of all external Pareto optimal solutions by which it is covered. The fitness of a population member is determined only from the individuals stored in the external set. This reduces significantly the computational burden of the fitness assignment process. The strength of a Pareto optimal solution is at the same time its fitness.

The SPEA algorithm is described in the following steps:

Step 1: Generate an initial population and create an empty external Pareto-optimal set.
Step 2: Update the external Pareto optimal set.
Step 3: Calculate the fitness values of individuals in both external Pareto set and the population.
Step 4: Combine the population and the external set individuals. Select two individuals at random compare their fitness. Select the better one to the mating pool.
Step 5: Perform the crossover and mutation operations according to their probabilities to generate the new population.
Step 6: Terminate if the generation counter exceeds set value.

The computational flowchart of the SPEA approach is presented in Fig. 2.

Implementation of Shortest Path Routing Problem (SPEA):

Initialization: routing path is encoded by a string of positive integers that represent the nodes through which the path passes. Each locus of the string represents an order of a node that is indicated by the gene of the locus. The gene of the first locus is for the source node and the one at the last locus is for the destination node. The length of a routing path should not exceed the maximum length n, where n is the number of nodes in the network. Special difficulties arise when a random sequence of edges usually does not correspond to a path. To overcome such difficulties, an indirect approach is adopted by encoding some guiding information to construct a path. The path is generated by sequential node appending procedure beginning from the specified node 1 and terminating at the specified node n, where n = 20. At each step, there are usually several nodes available for consideration.

Nodes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Priority	2	5	7	6	3	1	9	4	10	8	14	16	30	20	44	70	57	46	35	90

Fig. 3: Example of Priority-based encoding

Each node is assigned a priority with a random mechanism and adds the one with the highest priority into path (Chen and Sun, 2005). A gene in a chromosome is characterized by two factors: 'locus', i.e., the position of gene located within the structure of chromosome and 'allele', i.e., the value the gene takes. In the priority-based encoding method, the position of a gene is used to represent the node and its value is used to represent the priority of the node for constructing a path among candidates. A path can be uniquely determined from this encoding scheme (Mukhef *et al*., 2008; Craveirinha *et al*., 2008).

An example of chromosome generated using priority based encoding scheme is shown in Fig. 3. To find a path from source node 1 to destination node 20, a node which is connected to node 1 is identified first. As seen from Fig. 1, the nodes 2, 3, 4 and 5 are such nodes to be considered. The priorities for them are 5, 7, 6 and 3 respectively. The node 3 has the highest priority and is put into the path. The possible nodes from node 3 are nodes 1, 2, 4, 7, 8 and 9. The priorities of these nodes are 2, 5, 6, 9, 4 and 10 respectively. Since node 9 has a larger value than the other nodes, it is taken as the next node while constructing the path. Then, the set of nodes that are available for next consideration are chosen and the one with the highest priority among them is selected (Pangilinan and Janssens, 2007). The same procedure is repeated until a complete path from the source node1 to the destination node 20 is obtained (1, 3, 9, 15, 20).

Update of external Pareto-optimal set: The external Pareto-optimal set is updated as follows:

- Search the population for the nondominated individuals and copy them to the external Pareto set
- Search the external Pareto set for the nondominated individuals and remove them all from the set
- If the number of the individuals externally stored in the Pareto set exceeds a prespecified maximum size, reduce the set by means of clustering

Fitness assignment: The fitness assignment procedure is a two-stage process. First, the individuals in the external nondominated set P* are ranked.

Afterwards, the individuals in the population P are evaluated (Deb, 2001).

Step 1: Each solution $i \in P*$ is assigned a real value, $S_i \in [0, 1)$, called strength; S_i is proportional to the number, n_i of current population members that an external solution i dominates:

$$S_i = \frac{n_i}{N+1} \qquad (12)$$

Where:

n = The number of individuals in population P that are covered by i,

N = The size of randomly created population. The fitness f_i of i is equal to its strength:

$$f_i = S_i \qquad (13)$$

Step 2: The fitness of an individual $j \in P$ is calculated by summing the strengths of all external nondominated solutions $i \in P*$ that cover j. The fitness of the current population is member j is assigned as one more than the sum of the strength values of all external population members which weakly dominate j:

$$f_j = 1 + \sum_{i \in P^* \wedge i \prec j} S_i \qquad (14)$$

The addition of one makes the fitness of any current population member P to be more than the fitness of any external population member P*. This method of fitness assignment suggests that a solution with a smaller fitness is better.

Selection: The population and the external set individuals are combined and any two individuals at random are selected. Based on their fitness function, the better one is moved to the mating pool.

Crossover and mutation:
Crossover: The first genetic operation done to the chromosomes in the mating pool is crossover. The idea behind crossover is to create an information exchange between two chromosomes. By doing so, the algorithm will explore new paths and hopefully be able to find better paths in the process. The crossover scheme that is applied here is Partially Mapped Crossover (PMX).
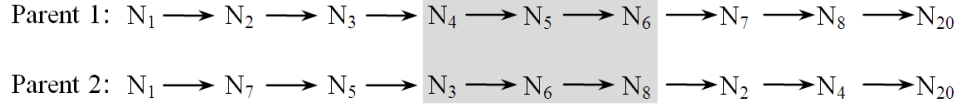
Parent 1: $N_1 \longrightarrow N_2 \longrightarrow N_3 \longrightarrow \boxed{N_4 \longrightarrow N_5 \longrightarrow N_6} \longrightarrow N_7 \longrightarrow N_8 \longrightarrow N_{20}$

Parent 2: $N_1 \longrightarrow N_7 \longrightarrow N_5 \longrightarrow \boxed{N_3 \longrightarrow N_6 \longrightarrow N_8} \longrightarrow N_2 \longrightarrow N_4 \longrightarrow N_{20}$

Fig. 4: Parents for Partially mapped crossover

Offspring 1: $N_1 \longrightarrow N_2 \longrightarrow N_3 \longrightarrow \boxed{N_3 \longrightarrow N_6 \longrightarrow N_8} \longrightarrow N_7 \longrightarrow N_8 \longrightarrow N_{20}$

Offspring 2: $N_1 \longrightarrow N_7 \longrightarrow N_5 \longrightarrow \boxed{N_4 \longrightarrow N_5 \longrightarrow N_6} \longrightarrow N_2 \longrightarrow N_4 \longrightarrow N_{20}$
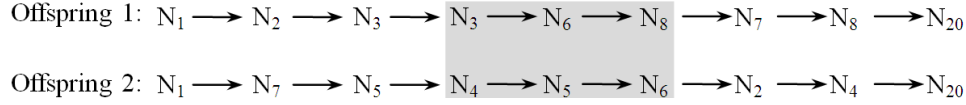
Fig. 5: Offspring's after mapping sections crossed

Offspring 1: $N_1 \longrightarrow N_2 \longrightarrow N_4 \longrightarrow N_3 \longrightarrow N_6 \longrightarrow N_8 \longrightarrow N_7 \longrightarrow N_5 \longrightarrow N_{20}$

Offspring 2: $N_1 \longrightarrow N_7 \longrightarrow N_8 \longrightarrow N_4 \longrightarrow N_5 \longrightarrow N_6 \longrightarrow N_2 \longrightarrow N_3 \longrightarrow N_{20}$
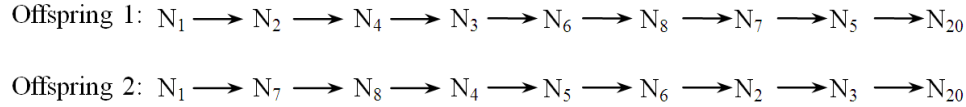
Fig. 6: Production of offsprings after crossover

PMX is a crossover of permutations which guarantees all positions that will be found exactly once in each offspring, i.e., both offspring receive a full complement of genes, followed by the corresponding filling-in of alleles from their parents. PMX proceeds as follows:

- The two chromosomes are aligned
- Two crossing sites are selected uniformly at random along the strings, defining a matching section
- The matching section is used to cross through position-by-position exchange operation
- For illustration, the following two parents are considered

Alleles are moved to their new positions in the offspring. For illustration, the two parents are considered as shown in Fig. 4.

First, two cut points are selected uniformly at random along the parent strings. The sub strings between the cut points are called the mapping sections. Now the mapping section of the first parent is copied into the second offspring and the mapping section of the second parent is copied into the first offsprings as shown in Fig. 5.

Then offspring1 is filled up by copying the first two elements N_1, N_2 of the first parent. In case a node is already present in the offspring it is replaced according to the mapping. Here the mapping is defined as $N_4 \leftrightarrow N_3$, $N_5 \leftrightarrow N_6 \leftrightarrow N_8$. For example the first two elements of parent1 N_1, N_2 are copied as the first two elements of the offspring1. The third element would be N_3, like the first element of the mapped sections in offspring1. So there is already a N_3 present in offspring1. Hence, because of the mapping $N_4 \leftrightarrow N_3$ the third element of the offspring1 is chosen to be N_4. The seventh and ninth

elements of offspring1 can be taken from the first parent. However, the eighth element of the offspring1 would be N_8, which is already present. Because of the mapping $N_5 \leftrightarrow N_6 \leftrightarrow N_8$, it is chosen as N_5. Thus, offspring1 and offspring2 are formed and is shown in Fig. 6.

Here, the source node and the destination nodes are fixed. Each partial route is exchanged and assembled and thus, two new routes are produced. In other crossover techniques available, there is a possibility of loop formation after crossover. In order to avoid this, a repair function must be used as a countermeasure. But in PMX, loop formation is avoided, because, there is no repetition of nodes. The repetition of nodes is avoided by a mapping function. Therefore, PMX finds many new paths without increasing the computational complexity as no repair function is needed.

Mutation: The objective of mutation is to create diversity in the population. The population undergoes mutation by an actual change or flipping of one of the genes of the candidate chromosomes, thereby keeping away from local optima.

Stopping criteria: Check for stopping criteria. The search will be terminated if the generation counter exceeds set value.

Reducing Pareto set by clustering: In some problems, the Pareto optimal set can be extremely large or even contain an infinite number of solutions. In this case, reducing the set of nondominated solutions without destroying the characteristics of the trade-off front is desirable from the decision maker's point of view. An average linkage based hierarchical clustering algorithm is employed to reduce the Pareto set to manageable

size. It works iteratively by joining the adjacent clusters until the required number of groups is obtained. The algorithm is illustrated in the following steps:

Step 1: Initialize cluster set C; each individual $i \in P^*$ constitutes a distinct cluster.
Step 2: If number of clusters $|C| \leq N^*$, then go to Step 5, else go to Step 3.
Step 3: Calculate the distance of all possible pairs of clusters. The distance d_{xy} between two clusters C_1 and C_2 is defined as the average Euclidean distance of all pairs of solutions ($i \in C_1$ and $j \in C_2$). It is calculated using the following equation:

$$d_{xy} = \frac{1}{n_1 \times n_2} \sum_{x \in C_1, y \in C_2} d(x, y) \tag{15}$$

Where:
n_1 = The numbers of individuals in clusters C_1
n_2 = The numbers of individuals in clusters C_2

Step 4: Determine two clusters with minimal distance d_{xy}. Merge these clusters together. This reduces the number of clusters by one. Go to Step 2.
Step 5: For each cluster, find the centroid and select the nearest individual to the centroid as a representative and remove all other individuals from the cluster.
Step 6: Compute the reduced nondominated set by uniting the representatives of the clusters.

MATERIALS AND METHODS

The SPEA for shortest path search is tested on networks with randomly varying topologies through computer simulations. Simulation runs were carried out 100 times with MATLAB 7.4 software package on IBM PC with Pentium dual core processor.

RESULTS AND DISCUSSION

Simulation results for a fixed network with 20 nodes: In order to test the capability of SPEA for shortest path routing problem, the simulation studies involve the undirected, weighted network topology with 20 nodes depicted in Fig. 1. Each of the links in the network is associated with two QoS parameters, cost and delay. The range of cost varies from 10-250 and the range of delay from 5-200. In general, any evolutionary search algorithm shows improved performance with relatively larger population. As there are no general guidelines available at present for the selection of

parameters, they are chosen by trial and error. For all the runs, the sender is always the first node and the receiver is the twentieth node since that would give the largest number of possible paths in the network. The higher the population size and/or the number of generations, the larger the number of solutions found. Several runs have been carried out to set the parameters and after experimentation, the population size was set as 200. The size of the Pareto-optimal set was chosen as 40. If the number of nondominated Pareto-optimal solutions exceeds this bound, the hierarchical clustering technique is called. Since the population in SPEA is augmented to include the externally stored set for selection process, the population size in SPEA was reduced to 160 individuals only. Crossover and mutation probabilities were chosen as 0.8 and 0.01 respectively, in all simulation runs.

The Pareto-optimal front for the best optimization runs obtained by SPEA is shown in Fig. 7. It is worth mentioning that the Pareto optimal set has 23 nondominated solutions generated by a single simulation run. The average CPU execution time was found to be 59 sec.

For completeness and comparison purposes, the problem was also treated as a single-objective optimization problem by linear combination of cost and delay as follows (Deb, 2008):

Minimize:

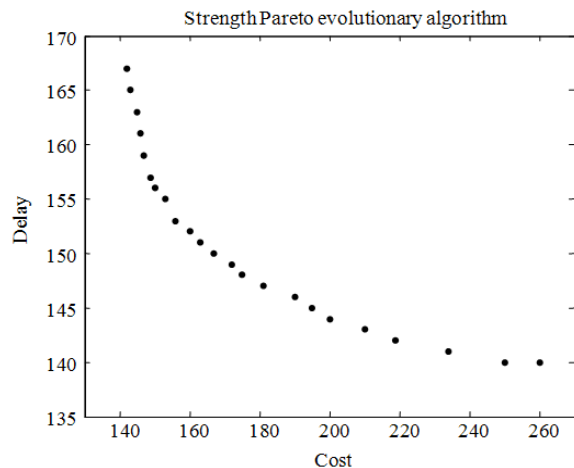$$w f_1 + (1 - w) f_2$$

where, w is the weighting factor.
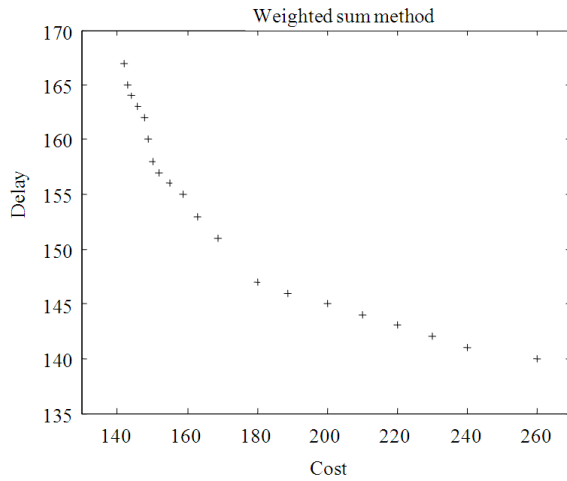

Fig. 7: Pareto-optimal front of SPEA in last generation

Fig. 8: Pareto-optimal front of linear combination in 20 separate runs

Table 1: Route Optimal Rate for networks of varying topologies

Number of nodes	Population size	Route optimal rate	CPU time (in sec)
50	500	96	70
75	500	96	76
100	600	89	85
125	600	87	91
150	750	86	99
175	900	82	102
200	1000	79	112

To generate 20 nondominated solutions, the algorithm was run 20 times with varying w as a random number, w = rand [0,1]. The solutions obtained by weighted sum method for the routing problem considered are plotted in Fig. 8. The execution time is found to be 205 sec just to generate 20 solutions. With the solutions detected by SPEA, the decision-maker has the opportunity to visualize trade-offs and may be inclined to accept a very small violation of the delay requirement for a large cost saving. Comparing the results shown in Fig. 7 and 8, it can be concluded that:

- The 23 solutions shown in Fig. 7 represent the results of the SPEA technique obtained in a single run while the 20 solutions by weighted sum method shown in Fig. 8 has been obtained in 20 separate runs
- The solutions of SPEA approach shown in Fig. 7 have better diversity characteristics and well-distributed solutions over the trade-off surface than weighted sum method
- The number of solutions found depends on the cost and delay data available which is generated randomly

- The CPU execution time of SPEA is much less than that of the weighted sum method as it produces all the Pareto optimal solutions in one single run
- The minimum cost of 142 Units and delay of 140 Units are reached in a single run by SPEA. All the solutions provided are best solutions and it is up to the decision maker to choose one among them depending upon the requirement
- No need to give any weightage for any particular objective in SPEA, whereas in weighted sum method, the solutions are found by varying the weighting factor, w

Simulation results for random network topologies: A serial of optimization runs were conducted with random and varying topologies of 50-200 nodes through computer simulations. The main objective of these simulation experiments is to investigate the quality of solution and convergence speed for different network topologies and number of nodes. For the randomly generated networks with 50-200 nodes with randomly assigned link costs and link delays are investigated for shortest path solution. The cost and delay of the networks are randomly chosen in the interval [10, 500]. In all the cases, the priority based encoding/decoding and partially mapped crossover is used. The quality of solution, in terms of Pareto-optimal front and the performance in terms of route optimal rate are tabulated in Table 1. The route optimal rate is defined as the rate at which the algorithm reached the optimal solution in 100 runs. The population size of different nodes with the obtained route optimal rate is tabulated in Table 2. For all the simulation runs of networks having nodes varying from 50-200, the crossover probability and mutation probability is uniformly set as 0.8 and 0.01 respectively for fair comparison. However, the increase in the number of nodes warrants the population size to be increased.

When the number of nodes and number of edges are increased in this randomly varying topology network, the algorithm works better and finds all nondominated solutions. From the simulation results it is clear that as the population size is increased the route optimal rate also increases. But when the population size is increased the computational time increases.

CONCLUSION

In this study, investigative results on using SPEA to solve the shortest path routing problem have been reported. SPEA is capable of exploring more efficient and non-inferior solutions. This demonstrates that the

search of SPEA span over the entire trade off surface. The SPEA based search used a modified indirect path encoding scheme, called the priority based encoding scheme and partially mapped crossover scheme, so as to widen the scope of search space. It also reduces the probability of invalid path/loop creation during the path construction procedure using a heuristic operator. The performance of the SPEA has been compared with the weighted sum method. The results obtained by SPEA are superior to weighted sum method and also it finds all the possible solutions in a single run, thus saving computer execution time. SPEA is applied to networks of various sizes and it is clear that the algorithm performs well in large network sizes too.

ACKNOWLEDGEMENT

REFERENCES

Abido, M.A., 2003. Environmental/Economic power dispatch using multiobjective evolutionary algorithms. IEEE Trans. Power Syst., 18: 1529-1537. ISSN: 0885-8950

Abido, M.A., 2006. Multiobjective evolutionary algorithms for electric power dispatch problem. IEEE Trans. Evolutionary Comput., 10: 315-329. ISSN: 1089-778X

Ahn, C.W. and R.S. Ramakrishna, 2002. A genetic algorithm for shortest path routing problem and the sizing of populations. IEEE Trans. Evolut. Comput., 6: 566-579. ISSN: 1089-778X

Araujo. F., B. Ribeiro and L. Rodrigues, 2001. A neural network for shortest path computation. IEEE Trans. Neural Networks, 12: 1067-1073. ISSN: 1045-9227

Bueno, M.L.P. and G.M.B. Oliveira, 2010. Multiobjective evolutionary algorithms and a combined heuristic for route reconnection applied to multicast flow routing. Proceeding of the 10th IEEE International Conference on Computer and Information Technology, Bradford, June 29-July 1, 2010, pp: 464-417.

Chen, H. and B. Sun, 2005. Multicast routing optimization algorithm with bandwidth and delay constraints based on GA. J. Commun. Comput., 2: 63-67. http://74.125.155.132/scholar?q=cache:h51XCGuk K7cJ:scholar.google.com/+Multicast+routing+opti mization+algorithm+with+bandwidth+and+delay+ constraints+based+on+GA&hl=en&as_sdt=2000

Chitra, C. and P. Subbaraj, 2010. A nondominated sorting genetic algorithm for shortest path routing problem. Int. J. Comput. Sci., 5: 55-63.

Craveirinha, J., R. Girao-Silva and J. Climaco, 2008. A meta-model for multiobjective routing in MPLS networks. Central Eur. J. Operat. Res. (Springer), 16: 79-105. DOI: 10.1007/s10100-007-0044-9

Deb. K., 2001. Multi-Objective Optimization using Evolutionary Algorithms. 1st Edn., John Wiley and Sons, ISBN: 10: 047187339X, pp: 518.

Deb, K. and S. Tiwari, 2008. Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization. Eur. J. Operat. Res., 185: 1062-1087. DOI: 10.1016/J.EJOR.2006.06.042

Jayakumar, G. and G. Gopinath, 2008. Performance Comparison of Two On-demand Routing Protocols for Ad-hoc Networks based on Random Way Point Mobility Model. Am. J. Applied Sc., 5: 659-. DOI: 10.3844/.2008.659.664

Lawler, E.L., 1976. Combinatorial Optimization in Networks and Matroids, Holt, Rinehart and Winston. New York.

Lin, L. and M. Gen, 2007. A bicriteria sortest path routing problems by hybrid genetic algorithms in communication networks. Proceeding of the IEEE Congress Evolutionary Computer, Sep. 25-28, Singapore, pp: 4577-4582.

Mendoza, F., J.L. Bernal-Agustín and J.A. Domínguez-Navarro, 2006. NSGA and SPEA Applied to multiobjective design of power distribution systems. IEEE Trans. Power Syst., 21: 1938-1945. ISSN: 0885-8950

Mohemmed, A.W., N.C. Sahoo and T.K. Geok, 2008. Solving shortest path problem using particle swarm optimization. Applied Soft Comput., 8: 1643-1653. DOI: 10.1016/J.ASOC.2008.01.002

Mukhef, H.A., E.M. Farhan and M.R. Jassim, 2008. Generalized shortest path problem in uncertain environment based on PSO. J. Comput. Sci., 4: 349-352. http://www.scipub.org/fulltext/jcs/jcs44349-352.pdf

Murad, A.M. and B. Al-Mahadeen, 2007. Adding Quality of Service Extensions to the Enhanced Associativity Based Routing Protocol for Mobile Ad Hoc Networks (MANET). Am. J. Applied Sci., 4: 874-879. DOI: 10.3844/.2007.874.879

Pangilinan, J.M.A. and G.K. Janssens, 2007. Evolutionary algorithms for the multi-objective shortest path problem. world academy of science. Eng. Technol., 25: 205-210.

Rouskas, G.N. and I. Baldine, 1997. Multicast routing with end-to-end delay and delay variation constraints. IEEE J. Selected Areas Commun., 15: 346-356.

Srinivas, N. and K. Deb, 1994. Multi-objective optimization using non-dominated sorting in genetic algorithm. Evolut. Comput., 2: 221-248.

Sriram, R., G. Manimaran and S.R. Murthy, C., 1998. Preferred link based delay-constrained least-cost routing in wide area networks. Comput. Commun., 21: 1655-1669.

Taher, S.A. and S.M.H. Tabei, 2008. A Multi-Objective HPSO Algorithm Approach for Optimally Location of UPFC in Deregulated Power Systems. Am. J. Applied Sci., 5: 835-843. DOI: 10.3844/.2008.835.843

Zitzler, E. and L. Thiele, 1999. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. IEEE Trans. Evolut. Comput., 3: 257-271.