A Fuzzy Logic Controller Based Dynamic Routing Algorithm with SPDE based Differential Evolution Approach

Debraj De
Department of Electronics & Telecommunication Engineering, Jadavpur University, Kolkata-32,India
091-033-24305809
debraj00@rediffmail.com

Sonai Ray
Department of Electronics & Telecommunication Engineering, Jadavpur University, Kolkata-32,India
091-033-24617057
ssonai2003@yahoo.com

Amit Konar
Department of Electronics & Telecommunication Engineering, Jadavpur University, Kolkata-32,India
091-033-24162697
amit_konar@vsnl.net

Amita Chatterjee
Centre of Cognitive Science, UGC (UPE), Jadavpur University, Kolkata-700032, India
amita_ju@yahoo.com

ABSTRACT

This paper proposes a novel scheme of Fuzzy Logic based dynamic routing in computer networks. The proposed dynamic routing algorithm is suited for application in Interior Gateway Protocols (IGP) inside an autonomous system (AS), such as a Local Area Networks (LAN). It is used in the protocols to find out a set of best possible routes, where each of the nodes broadcasts link status rather than broadcasting the whole routing table. The Self Adaptive Pareto Differential Evolution (SPDE) algorithm is little modified to apply it in solving efficient and optimal dynamic routing problem. One of the main features of the proposed routing scheme is that it outputs hierarchical quality solutions so that, if one path is blocked, there will be provisions of alternative paths for successful packet transmission in computer networks. The architecture of the proposed dynamic routing scheme is mainly composed of a controller, which makes use of a fuzzy-based decision making system. In a real world dynamic environment the controller finds out the optimal policy that determines weights on the parameters of routing. The total dynamic routing scheme is made to evolve intelligently in changing characteristics of daily network loads and usages. The proposed algorithm grains out the optimal routes for the packets to be transmitted. The paper also reviews the overall performance of the proposed routing scheme by applying it to a number of randomly generated real time computer networks. The fast response of our proposed scheme makes it suitable for real world applications like dynamic routing.

Keywords

Dynamic Routing, Fuzzy Logic Controller, Self Adaptive Pareto Differential Evolution (SPDE).

1. INTRODUCTION

Evolutionary algorithms [2] are a kind of global optimization techniques that use selection and recombination as their primary operators to tackle optimization problems. *Differential evolution* (DE) is a branch of evolutionary algorithms developed by Rainer Storn and Kenneth Price for optimization problems over continuous domains. One of the recent approaches to evolutionary optimization techniques is the *Pareto Differential Evolution* (PDE) algorithm [1].The algorithm was designed for optimization problems with continuous variables and achieved very competitive results compared to other algorithms in the EMO literature. However, there was no obvious way to select the best crossover and mutation rates apart from running the algorithm with different rates, then selecting the best among them. Then another approach came to solve this problem, the Self-Adaptive Pareto Differential Evolution Algorithm (SPDE) [3], which self adapts the crossover and mutation rates.

An optimization approach to dynamic routing in computer networks is described. Characteristics of Computer networks change in very important ways over time. Changes are spatial, temporal and topological. These dynamic, almost instantaneous, events cannot be easily dealt with in static time slices. The routing algorithms need to be improved in order to perform dynamically to take advantage of real time traffic information. Standard algorithms, such as Dijktra's algorithm respond to dynamic changes in network topology, but they are not guaranteed to produce physically optimal route specification [5]. Also they generate only one best solution, while the dynamic routing needs a pool of best solutions, so that if somehow a path becomes inactive there will be provisions of other optimal paths. In this paper, we have proposed a slightly modified SPDE technique used for efficient and optimal fuzzy based dynamic routing.

The paper is organized as follows: Section 2 presents routing model, Fuzzy Logic controller of dynamic routing scheme is proposed in section 3, essence of dynamic routing and proposal of application of SPDE algorithm for dynamic routing is described in section 4 and experimental results are shown in section 5.

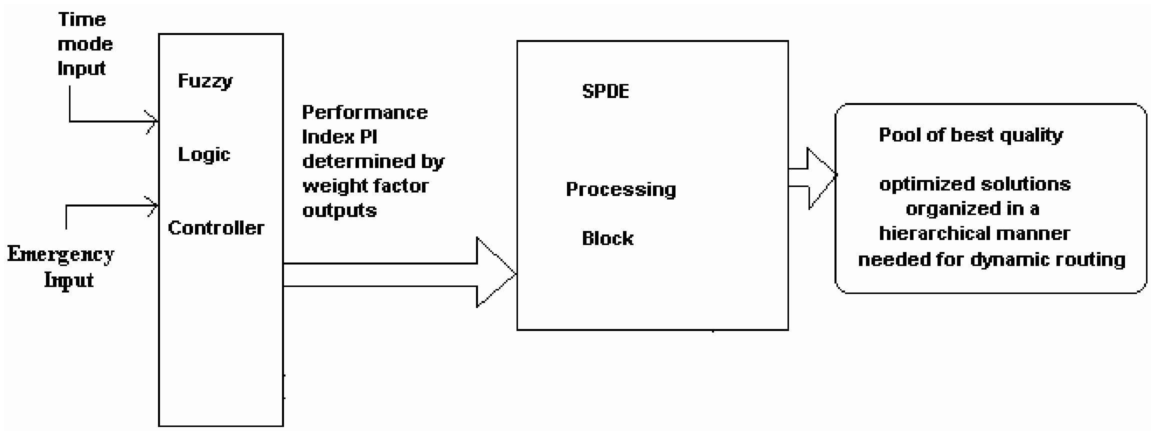

Figure 1. Block diagrammatic representation of total Dynamic Routing solving scheme.

2. Modeling efficient routing policy

The mostly dominant parameters in real-world computer networks for achieving efficient routing are:

a. Cost of path,
b. Speed of path (depending on the type of channel- e.g. optical fiber or wireless etc.),
c. Band width (denotes the channel capacity) of the path between two nodes, and
d. Number of hops needed for a packet to reach the destination node from the source node.

In our paper, we have defined a performance scale, termed as *Performance Index (PI)*, which expresses the Quality of Service (QOS) of the proposed optimizing dynamic routing algorithm. To make the routing efficient, *PI* is to be maximized. Now *PI* is expressed as:

$$PI = \left\{ \sum_{\text{all paths between nodes traversed}} \left(-k_1 C + k_2 S + k_3 B\right) \right\} / H \qquad (1)$$

Where C is Cost, S is Speed, B is Band width, H is number of hops and k_1, k_2, k_3 are three weight factors. The weight factors are termed as follows: Cost index k_1, Speed index k_2 and Bandwidth index k_3.

Variation of three parameters will affect *PI*. Now for path selection by a packet between two nodes will look towards:

a. choosing path with minimum cost,

b. choosing the path with maximum speed of data transmission,

c. choosing path with maximum band width and

d. selecting total path with minimum number of total hops. That will lessen the unwanted delay spent in a node.

From last four requirements and the expression (1) of *PI*, it is concluded that *PI* is to be maximized every time, when a packet has to select a path from all possible free paths.

3. Proposed Fuzzy Logic controller

3.1 Need of Fuzzy controller

A simple example of characteristic load on real world computer networks round the day is presented here in the given table.

Table 1. A sample of daily network load characteristics in real world networks.

Duration	Time	Load type
D1	5-8 am	Low
D2	8-10am	Medium
D3	10am to 8 pm	High
D4	8pm to 5 am	Medium

In a whole day, the various durations with typical characteristic of network load, described earlier in Table 1, as durations D1, D2, D3 and D4 are not discrete in time with crisp boundaries. Rather they are Fuzzy in nature. Since the real time computer network is always unpredictable and busty in nature, the high load in network for data transmission is not bounded to occur for example in duration D3. There can be high data transactions at any time in a day, but with a varying probability. Therefore the choice of Fuzzy system in this paper is justified to offer a solution for real world application for dynamic routing, which is set to work efficiently at any demand of computer network at any time of the day.

3.2 Choice of weight factors

The values of the weight factors k_1, k_2 and k_3 cannot be constant due to varying nature of network load and varying demand of channel capacity as well as speed. Therefore during different duration in a day, different weight factors have to be dominant. For an example, let at midnight, there is mainly use of internet for surfing, email etc. that are done from homes, whose emergency with respect to bandwidth and speed is not so high. At that time, the weight on speed and bandwidth may be lessened. This can be achieved by assigning smaller values of k_2 and k_3. But at that time, there is still a possibility of high network usage that will demand higher values of k_2 and k_3. These all intelligent decision making can be achieved by a well designed fuzzy controller, which will decide the values to be assigned to weights k_1, k_2 and k_3. One very important aspect of the controller is that, a change in these network usage policies will not cause a big deal of problem. Only the Fuzzy temporal rules are needed to be changed in the fuzzy controller adopted, rather than changing the whole structure of the controller. This imposes a great deal of flexibility in the system which makes it suitable for dynamic real world application.

3.3 Construction of Fuzzy controller

Three Fuzzy controllers are designed whose outputs are k_1, k_2 and k_3, each within a range [0, 1]. Inputs of each controller are:

a) Time mode T (with respect to business hour). This means that the *HIGH* value of membership lies in the business hour in a day. Range of T is [0, 24], i.e. the 24 hours span from 00-00 midnight to 00-00 midnight next day.

b) Emergency degree E, with a range [0, 1]. This denotes the strength of importance of immediate data transmission.

In the Figure 2, the proposed Fuzzy Logic Controller is shown with its output and inputs.

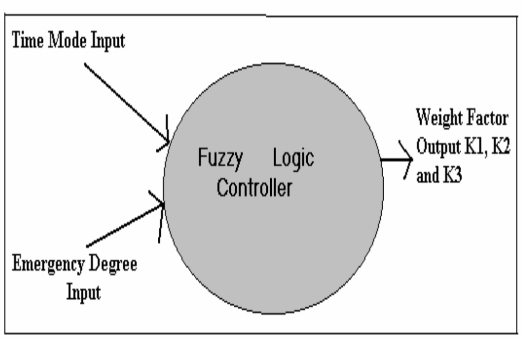

Figure 2. Inputs and outputs of proposed Fuzzy Logic Controller.

The variations of Cost Index k_1, Speed Index k_2 and Band width Index k_3 with Time mode input T and Emergency degree input E are shown in the given Figures 3, 4 and 5, for the defined Fuzzy temporal rules. If the temporal rules change, the characteristics of the plots will also change.

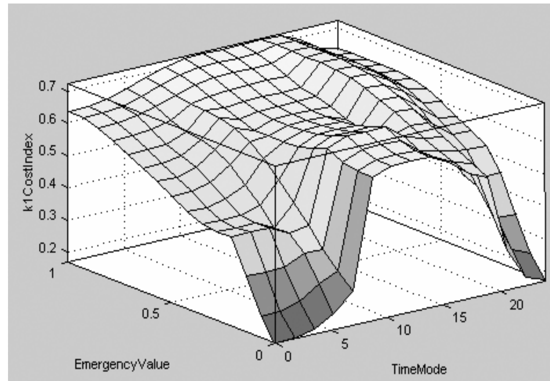

Figure 3. Variation of Cost Index k_1 with Time mode input and Emergency Degree input.

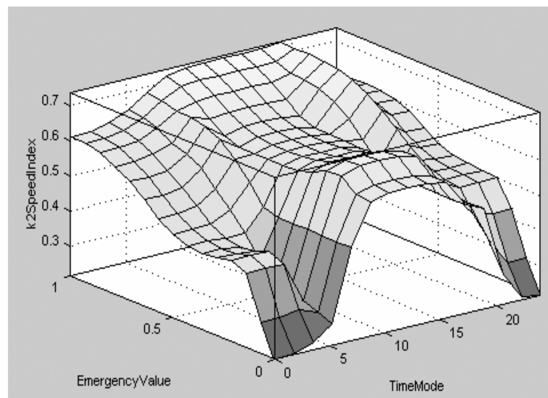

Figure 4. Variation of Speed Index k_2 with Time mode input and Emergency Degree input.

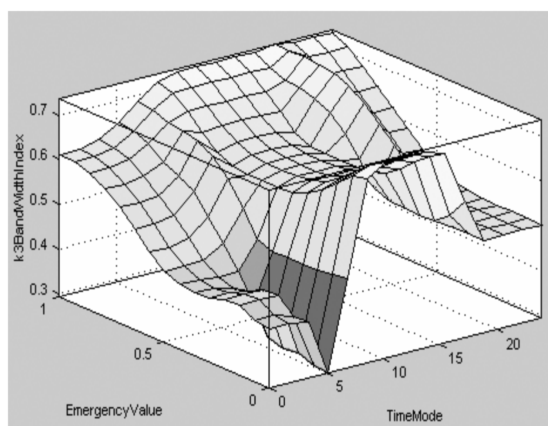

Figure 5. Variation of Band width Index k_3 with Time mode input and Emergency Degree input.

4. Proposal of algorithm for Dynamic Routing

4.1 Essence of Dynamic Routing

Dynamic routing performs the same function as static routing except it is more robust. Static routing allows routing tables in specific routers to be set up in a static manner, so network routes for packets are set. But if one or more router on the route goes down or some link become inactive, the destination may become unreachable. In these situations dynamic routing provides multiple best paths in a hierarchical manner in routing tables of routers, so that the best of physically realizable paths are followed.

4.2 SPDE algorithm: little modified to application to solve dynamic routing problem

The solution for the optimal path that packets should follow, can't be a single one for dynamic routing, rather is a set of optimum paths. Very interestingly the Pareto Differential Evolution algorithm (PDE) just not results one solution, but a set of solutions. This is the essence of using Pareto differential algorithm for the application of Dynamic Routing here in this paper. The Self Adaptive Pareto Differential Evolution Algorithm (SPDE) is modified slightly to suit it for construction of dynamic routing algorithm. Here we propose an efficiently optimizing dynamic routing algorithm. A generic version of the adopted algorithm is as follows:

1. Create a random initial population of potential solutions. If any solution has got an unattainable value, it is discarded and a new possible solution is generated randomly again. Here unattainable value means that in the solution, if there are two consecutive nodes with no possible direct path between them.

2. (a) Evaluate the individuals in the population by calculating their corresponding Performance Index (*PI*).

   (b) i. From the population of marked solutions, select at random 3 individuals. Among the three, the solution that gives the maximum value of Performance Index *PI* is selected as the parent a_1. This is done because the property of the solution that is mostly optimum (maximum *PI*), should be inherited in the children when crossover is performed. Other two solutions are marked as supporting parent a_2 and a_3.

   ii. Select at random a variable j in the range [1, 3].

   iii. Crossover rate (x_c): Solutions that are more optimal should have been given more importance in crossover for the sake of inheritance of better properties. To make this effect the normalized crossover rate is chosen as the corresponding values of the *PI* of each solution. Let the crossover rate be:

$$x_c^{child} \leftarrow x_c^{a1} + r1 \times (x_c^{a2} - x_c^{a3}) \quad (2)$$

   Here *r1* is a random variable in the range [0, 1]. If the crossover rate is not [0, 1], repair the crossover rate according to the repair rule.

   iv. Mutation rate (x_m): The individual with lesser crossover rate has a greater need of mutation, hence more mutation rate. Therefore here we have chosen the mutation rate as {1 – crossover rate}. Now let the mutation rate be:

$$x_m^{child} \leftarrow x_m^{a1} + r2 \times (x_m^{a2} - x_m^{a3}) \quad (3)$$

   Here *r2* is a random variable in the range [0, 1]. If the mutation rate is not [0, 1], repair the mutation rate according to the repair rule.

   v. Crossover:

   For each variable i (value 1 or 2 or 3), with some random probability (0, 1) > x_c^{child} or if $i = j$:

   do

   Crossover between that a_i and the individual (between a_1, a_2 and a_3) with maximum value of *PI*.

In this process of crossover, 2 children will be generated: *child1* and *child2*. Now in the solution pool, there will always be the main parent a_1 of the last crossover. And from a_2, a_3, *child1* and *child2* we select two individuals with greater values of *PI* in the solution pool.

   vi. Mutation:

   For each variable i (value 1 or 2 or 3), with some random probability (0,1) > x_m^{child} or if $i = j$,

   do

   Mutation of that individual ai.

In this process of mutation, some mutated individuals will be generated. . Now in the solution pool, the mutated individuals, which have greater values of *PI* than a_2 and a_3. If there is no such then no mutated individual is added to the pool.

3. Go to step 1, until the size of the solution pool i.e. the set of best solutions reaches a minimum value defined at the start. This value of the number of best solutions is dependent on the requirement of best possible routes, which is based on reliability and performance of the routers involved in the network.

5. Experimental results

In the experimental work, for establishing the efficiency of our proposed algorithm, some real world sample networks are chosen. One sample taken is shown in the following diagram, with 14 routers at 14 nodes; Node 1 is the starting node, while node 14 is the final destination node:

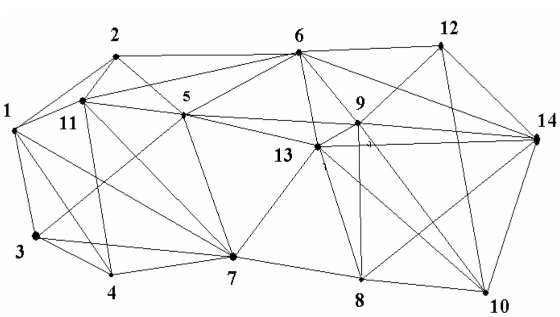

Figure 6. One sample network chosen for experiment.

In the approach of testing the proposed algorithm, the *Emergency value* (*E*) is taken as 0.6 and the *Time Mode* (*T*) is taken 10-00 am. So the values of the weight factors k1, k2 and k3 are obtained from the fuzzy controllers according to *E* and *T*. Every possible direct path between two nodes is randomly assigned cost value, speed value and band width value each of which is [0, 1]. For the network shown in figure 6, the dynamic routing problem is solved using SPDE algorithm.

Finally the pool of solutions (size of the pool was selected to be 8) proposed by our algorithm for the optimized routed path is (path is represented by expressing the nodes to be visited in order):

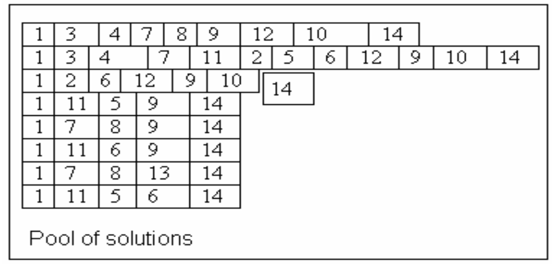

Figure 7. Resultant pool of best solutions.

A significantly good point of outcome is that it took only 6 iterations to generate pool of solutions within expected quality level of maximum *PI* index value; whereas the standard Dijktra's Shortest Path First Algorithm took 13 iterations to produce the best solution [1-11-5-6-9-14].

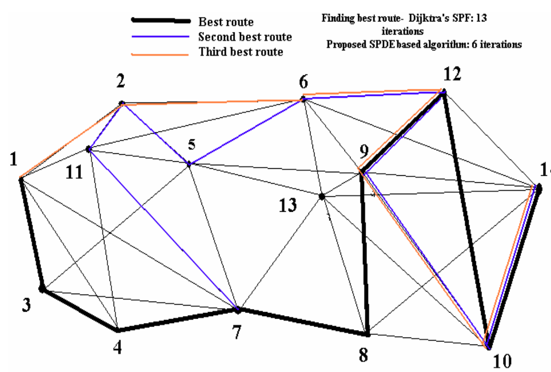

Figure 8. Three best routes (higher *PI*) found for the sample network chosen.

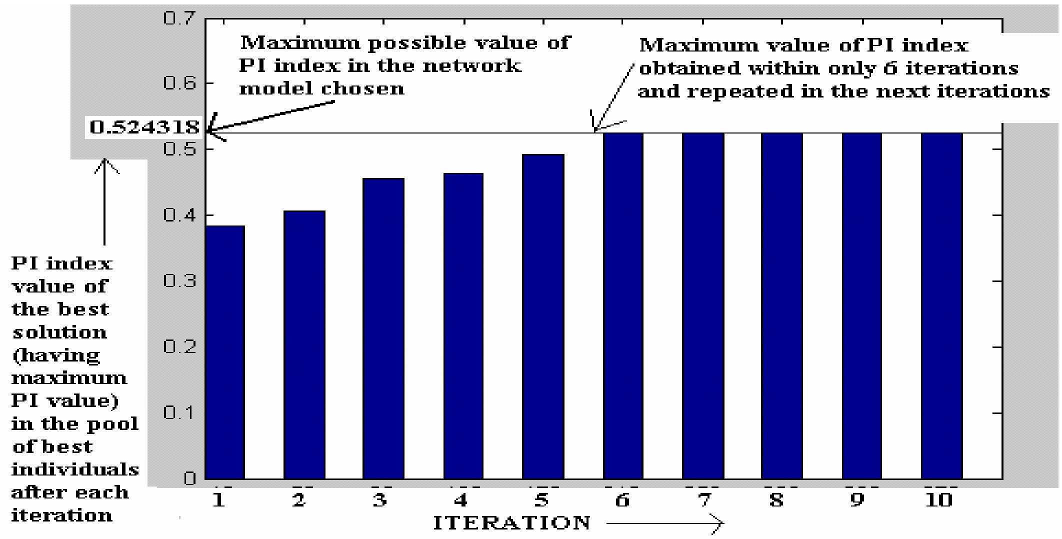

Figure 9. Bar plot of PI value of the best individual in the solution pool vs. iteration value.

Fig. 9 clearly shows that the best possible individual in the solution pool (having the maximum possible *PI* value of 0.524318) is obtained very quickly in just 6^{th} iteration, by applying the SPDE based dynamic routing algorithm. The maximum value of 0.524318 remains the PI value of the best solution in the solution pool in the next iterations.

6. Conclusion and future work

In this paper, the Self Adaptive Pareto Differential Evolutionary algorithm (SPDE) is modified to for using it in dynamic routing of real world networks. The proposed fuzzy controller very efficiently supports the dynamic routing in unpredictable nature of computer networks. We have tested our proposed algorithm on various sizes of networks and set of good solutions are obtained in small time, which makes it suitable for a real time online applications. The dynamic routing algorithm, being online, is a flexible one for the case where there is addition or deletion of nodes in the network.

For future work, we intend to form a real time decision maker on dynamic routing which incorporates SPDE and machine intelligence with classifier system. We also intend to add more networking and routing parameters, specific for different type of communications, for e.g. AD HOC networks, satellite communication etc.

7. References

[1] Coello, C. A. C. (1999). A Comprehensive Survey of Evolutionary-based Multiobjective Optimisation Techniques. *Knowledge and Information Systems*, 1(3):269 – 308.

[2] D.B. Fogel. *Evolutionary Computation: towards a new philosophy of machine intelligence.* IEEE Press, New York, NY, 1995.

[3] Hussein A. Abbass. The Self-Adaptive Pareto Differential Evolution Algorithm, in *Congress on Evolutionary Computation (CEC'2002)*, Vol. 1, pp. 831--836, IEEE Service Center, Piscataway, New Jersey, May 2002.

[4] H.A. Abbass, R. Sarkar, and C. Newton. A pareto differential evolution approach to vector optimisation problems. *The IEEE Congress on Evolutionary Computation, Seoul, Korea*, pages 971–978, 2001.

[5] Dan Costelloe, Peter Mooney and Adam Winstanley. Multi-Objective Optimisation and Dynamic Routing Algorithms in Transportation Networks, *First International Conference on Geographic Information Science*, Savannah, Georgia, October 2000.

[6] Carlos A. Coello Coello. A Short Tutorial on Evolutionary Multiobjective Optimization. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 21-40. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.

[7] Carlos A. Coello Coello.Evolutionary Multiobjective Optimization: Current and Future Challenges, in Jose Benitez, Oscar Cordon, Frank Hoffmann and Rajkumar Roy (editors), *Advances in Soft Computing---Engineering, Design and Manufacturing*, pp. 243--256, Springer-Verlag, ISBN 1-85233-755-9, September 2003 .

[8] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

[9] J. Horn, N. Nafpliotis, and D.E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. *Proceedings of the First IEEE Conference on Evolutionary Computation*, 1:82–87, 1994.

[10] J.D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 93–100, 1985.

[11] J. Knowles and D. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, 8(2):149–172, 2000.

[12] J. Knowles and D. Corne. The pareto archived evolution strategy: a new baseline algorithm for multiobjective optimization. *In 1999 Congress on Evolutionary Computation, Washington D.C., IEEE Service Centre*, pages 98–105, 1999.

[13] Johan Andersson. A Survey of Multiobjective Optimization in Engineering Design, Technical Report No. LiTH-IKP-R-1097, Department of Mechanical Engineering, Linköping University, 2000.

[14] Kiam Heong Ang and Yun Li. Multi-Objective Benchmark Studies for Evolutionary Computation, *2001 Genetic and Evolutionary Computation Conference. Workshop Program*, San Francisco, California, pp. 393--396, July 2001.

[15] Richard J. Balling. Pareto sets in decision-based design, *Journal of Engineering Valuation and Cost Analysis*, Vol. 3, pp. 189--198, 2001.