

## Multi-Objective Differential Evolution and Its Application to Enterprise Planning

Feng Xue  
Arthur C. Sanderson  
Robert J. Graves

Rensselaer's Electronics Agile Manufacturing Research Institute  
Rensselaer Polytechnic Institute, Troy, NY 12180, USA  
{xuef, sandea, graver}@rpi.edu

**Abstract** — Agility is important to modern enterprises. The effective coordination of large numbers of potential suppliers and manufacturers, demands a scientific methodology rather than just practical experience to make decisions on supply manufacturing planning problems. Particularly in cases where multiple decision objectives are important to process planning, empirical decisions are insufficient. This paper introduces formal methods to solve such multi-objective decision problems involved in general supply manufacturing planning, and specifically describes the extension of differential evolution methods to discrete problem domains. An enterprise planning problem with two objectives—cycle time and cost is used as a principal example. Such multi-objective optimization problems usually are very large and nonlinear. In this paper, the concept of differential evolution, which is well-known in single-objective continuous domain for its fast convergence and adaptive parameter setting, is extended to the discrete domain by introducing greedy probability, mutation probability, and crossover probability. Moreover, this concept is extended to discrete multi-objective optimization problem. The proposed discrete multi-objective differential evolution, or D-MODE algorithm is applied to obtain Pareto solutions of this general planning problem. A practical example in the electronics industry is used as an illustrative example to demonstrate the effectiveness of the proposed D-MODE.

### 1. Introduction

Advances in information technologies are driving fundamental changes in the processes and organizations of global enterprises. Innovations in software, networks, and database systems enable widely distributed organizations to integrate activities, share information, collaborate on decisions, and execute transactions. As a result, it is becoming increasingly uncommon for the creation of product and services in isolation, and they are being realized based on the creation of strategic and dynamic partnerships between suppliers, contract manufacturers, and customers. However, as the numbers of these distinct entities increase and they get more distributed, the complexity of forming efficient partnerships grows; it becomes more difficult to make

ideal assignments with respect to multiple criteria including cost, time, and quality. Fundamental to this complexity is that each assignment has the potential to affect overall product cost, and product realization time, and therefore assignments cannot be considered independent of one another. Due to this complexity it is increasingly difficult to make these dynamic partnerships purely on the basis of prior experience, and it becomes necessary to develop efficient decision-making systems that can automate significant portions of the overall decision task.

As many other engineering applications, this supply manufacturing planning decision-making involves multiple criteria. The ideal solution is that one assignment can be identified which optimizes all criteria simultaneously. However, such ideal solutions can never be obtained in practical applications where outcome criteria may be fundamentally inconsistent. Optimal performance according to one objective, if such an optimum exists, often implies unacceptably low performance in one or more of the other objective dimensions, creating the need for compromise to be reached. In this paper, we consider the identification of multiple solutions that may be used to guide the final decision process.

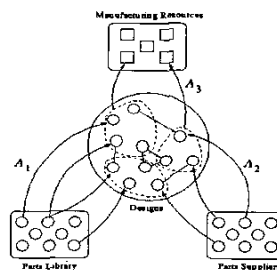


Figure 1: Structure of the design, supplier, manufacturing, planning decision problem. Lines with arrowheads indicate assignments. Dashed lines indicate aggregates. Identical parts in various designs have solid lines between them.

A model of the problem of integrated design, supplier and manufacturing planning for modular products where suppliers and manufacturing resources are network distributed is shown in Figure 1 and described in

[15][16][17]. This planning problem consists of three assignment problems (A1, A2, A3). The assignment problem A1 is the assignment of parts (from a parts library) to a design that satisfies a predetermined functional specification. Multiple designs that satisfy the functional specification are possible. The assignment problem A2 is the assignment of suppliers (from a list of available suppliers) who will supply the parts in a design, and the assignment problem A3 is the assignment of designs to available manufacturing resources. Each of these assignments contributes to overall product cost and product realization time, and has nonlinear (cannot be evaluated as weighted sums) effects on these measures. More detailed aggregation function and the related time and cost components can be found in [15]. A heuristic aggregation to combine product cost  $C(x)$  and product realization time  $T(x)$  for a complete design-supplier-manufacturing assignment  $x$  has been used for evaluation in this prior work

However, selection of an appropriate model of aggregating these two objectives and corresponding parameter to meet practical requirement is not possible in general. Each model would need to appropriately incorporate the nature of the problem itself and the preference structure of decision maker. Therefore, multi-objective optimization techniques must be developed.

In mathematical notation, a multi-objective optimization problem (MOOP) can be loosely posed as (without loss of any generality, minimization of all objectives is assumed):

$$\min_{x \in \Omega} Z(x) = \begin{bmatrix} z_1(x) \\ z_2(x) \\ \vdots \\ z_k(x) \end{bmatrix},$$

where  $\Omega = \{x | h(x) = 0, g(x) \leq 0\}$ , and  $x$  is decision variable of dimension  $n$ .  $Z: \mathbb{R}^n \rightarrow \mathbb{R}^k$ ,  $h: \mathbb{R}^n \rightarrow \mathbb{R}^{m_1}$ ,  $g: \mathbb{R}^n \rightarrow \mathbb{R}^{m_2}$ ,  $k$  is the number of objectives,  $m_1$  and  $m_2$  are the number of equality and inequality constraints, respectively.

In practical application, there is no solution that can minimize all the  $k$  objectives simultaneously. As a result, multi-objective optimization problems tend to be characterized by a family of alternatives that must be considered equivalent in the absence of information concerning the relevance of each objective relative to the others. These alternatives are referred to as Pareto optimal solutions, which have the same meaning with efficient, non-inferior, or non-dominated solutions. A Pareto optimal solution is defined as follows:

**Definition:** The vector  $Z(\bar{x})$  is said to dominate another vector  $Z(\bar{x})$ , denoted by  $Z(\bar{x}) \prec Z(\bar{x})$ , if and only if  $z_i(\bar{x}) \leq z_i(\bar{x})$  for all  $i \in \{1, 2, \dots, k\}$  and  $z_j(\bar{x}) < z_j(\bar{x})$  for some  $j \in \{1, 2, \dots, k\}$ . A solution  $x^* \in \Omega$  is said to be Pareto optimal solution for MOOP if and only if there does not exist  $x \in \Omega$  that satisfies  $Z(x) \prec Z(x^*)$ .

Evolutionary algorithms have gained a lot of interest on optimization (single objective) and learning area and have been applied to various practical problems. In recent 15 years, a new area of evolutionary multi-objective optimization has grown considerably. Evolutionary algorithms deal simultaneously with a set of possible solutions. This characteristic allows to find an entire set of Pareto optimal solutions in a single run of the algorithm, instead of have to perform a series of separate runs as in the case of the traditional mathematical programming techniques. Additionally, evolutionary algorithms are less susceptible to the shape or continuity of the Pareto front, whereas these two issues are concerns for mathematical programming techniques.

The Vector Evaluated Evolutionary Algorithm (VEEA) was the first practical approach for multi-criteria optimization using EAs, in which Schaffer extended Grefenstette's GENESIS program to include multiple criteria [11][12]. In this scheme,  $N/k$  sub-populations of equal size are assigned one to each criterion (where  $k$  is the number of criteria and  $N$  is the total population size). A modified selection operator performs proportional selection for each sub-population according to each objective function, while recombination and mutation operations are allowed to cross sub-population boundaries.

There also are other versions of evolutionary algorithms to attempt to promote the generation of multiple non-dominated solutions such as Fourman [5], Kursawe [9], Hajela and Lin [7]. However, none makes direct use of the actual definition of Pareto optimality. The concept of Pareto-based fitness assignment was first proposed by Goldberg [6], as a means of assigning equal probability of reproduction to all non-dominated individuals in the population. This method is consisted of assigning rank 1 to the non-dominated individuals and removing them from contention, then finding a new set of non-dominated individuals, ranked 2, and so forth.

Fonseca and Fleming [4] have proposed a slightly different scheme, whereby an individual's rank corresponds to the number of individuals in the current population by which it is dominated. Non-dominated individuals are, therefore, all assigned the same rank, while dominated ones are penalized according to the population density in the corresponding region of the trade-off surface. Srinivas and Deb [13] have

implemented a similar sorting and fitness assignment procedure, called NSGA, but based on Goldberg's version of Pareto ranking. Horn et al. [8] proposed the tournament selection method based on Pareto dominance. The more recent algorithms include NSGA-II [3], and the SPEA algorithm [18].

## 2. Discrete Differential Evolution

Differential Evolution (DE) is a branch of evolutionary algorithm proposed by Storn and Price [14] for optimization problems over a continuous domain. DE is similar to  $(\mu, \lambda)$  evolution strategy in which mutation plays the key role. For any selected individual,  $p_i$ , that undergoes mutation, the mutation operator is represented as follows:

$$p'_i = \gamma \cdot p_{best} + (1 - \gamma)p_i + F \cdot \sum_{k=1}^K (p_{k_s} - p_{k_b})$$

where  $p_{best}$  is the best individual in parent population,  $\gamma \in [0, 1]$  represents greediness of the operator, and  $K$  is the number of differentials used to generate the perturbation,  $F$  is the factor that scales the perturbation,  $p_{k_s}$  and  $p_{k_b}$  are randomly selected mutually distinct individuals in the parent population, and  $p'_i$  is the offspring. The basic idea of DE is to adapt the search step along the evolutionary process. At the beginning of evolution, the perturbation is big since parent individuals are far away to each other. As the evolutionary process proceeds to the final stage, the population converges to a small region and the perturbation becomes small. As a result, the adaptive search step allows the evolution algorithm to perform global search with a large search step at the beginning of evolutionary process and refine the population with a small search step at the end. The selection operator in DE selects the better of the parent and the offspring by comparing their fitness values:

$$p_i^{(t+1)} = \begin{cases} p_i^{(t)} & \text{if } \Phi(p_i^{(t)}) > \Phi(p_i^{(t)}) \\ p'_i & \text{otherwise} \end{cases}$$

In this paper, the DE concept is scaled to the discrete domain, and to the multi-objective optimization problem. In the basic differential evolution and its subsequent variants, DE allows mutation toward both the best individual and random perturbations. This is realized by forcing the individual to move in the direction of differential vector between the best individual and itself, and adding perturbation of differential vector among randomly selected individuals from the parent population. In a discrete domain, the decision variable is an  $n$  dimensional vector variable  $\mathbf{x} = [x_1, x_2, \dots, x_n]$ ,  $x_i \in \Omega_i$ , where  $\Omega_i$ 's are a set of discrete vectors. In many situations, such as in various planning problem involved

in supply manufacturing, the elements of the vector,  $\Omega_i$ , are integer index, which have no physical meaning. In such situations, the differential vectors in traditional DE have no ordered physical interpretation. However, the main concept of DE, which is directing the individuals to current best solutions with adaptive perturbations, can be implemented in another way. In our discrete DE, this concept is realized by introducing three probabilities: *greedy probability*  $p_g$ , *mutation probability*  $p_m$ , and *crossover probability*  $p_c$ . The decision variable,  $\mathbf{x}$ , is represented in the evolutionary algorithm using a gene vector of length  $n$  that is the same as the real decision vector. The value in each allele position is as the corresponding value of the real decision vector. With this representation, the allele  $j$  of offspring of any individual  $p_i$  can be obtained in the reproduction operator as follows:

$$p'_{ij} = \begin{cases} p_{best_j} & \text{if } p_g < rand() \\ \Omega_{j_{rand}} & \text{elseif } p_g + p_m < rand() \\ p_{a_j} & \text{elseif } p_g + p_m + p_c < rand() \\ p_{ij} & \text{else} \end{cases}$$

where  $rand()$  is a random number between 0 and 1,  $\Omega_{j_{rand}}$  is a random selected value from  $\Omega_j$  containing all possible values for allele  $j$ ,  $p_a$  is a randomly selected individual from parent population that is distinct with  $p_i$ . For a single objective optimization problem, the  $p_{best}$  can be easily identified by choosing the individual with highest fitness value. In this way, the offspring reproduction operator introduced above captures the DE concept. The reproduction of each offspring is guided by the best individual by reproducing some of its gene information. The mutation part can be regarded as a constant small perturbation for the offspring generation. Reproducing gene information from other parent individuals is the adaptive perturbation that varies during the evolutionary process. At the beginning of the evolutionary process, this perturbation is large due to less similarity of the population, and small at the end due to more similarity of the population. This reproduction mechanism is as shown in Figure 2.

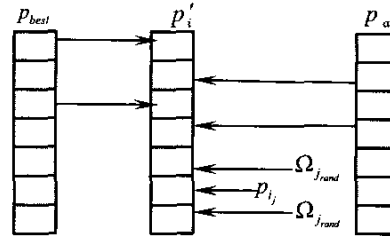


Figure 2: Discrete DE operator to produce offspring. The individual under operation gets some gene information from the "best" individual, and some from a selected individual from current parent population, and some from a random generation.

### 3. Multi-Objective Evolutionary Algorithm

Since the emergence of multi-objective evolutionary algorithms, there have been many variants. The representatives are multi-objective genetic algorithm (MOGA) due to Fonseca and Fleming [4], nondominated sorting genetic algorithm (NSGA) due to Srinivas and Deb [13], and niched Pareto genetic algorithm (NPGA) due to Horn et al. [8]. More recently, Zitzler and Thiele [18] proposed the strength Pareto evolutionary algorithm (SPEA), which has an external repository of global Pareto solutions with continuously update of this repository. This deterministic way itself is a complementary part of randomness of non-elitist evolutionary algorithms such as NSGA to keep the convergence to global Pareto solutions without having an effect on the stochastic properties of the evolutionary algorithm, though it is not the situation in SPEA.

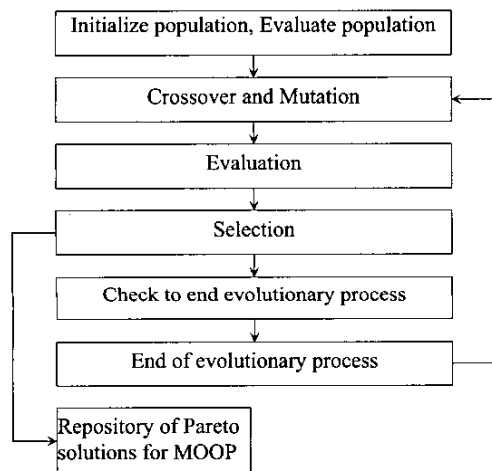


Figure 3: Flowchart of revised NSGA with Pareto solutions repository

The structure of a revised NSGA with this external repository population is shown in Figure 3. In NSGA, the non-dominated individuals are assigned rank 1 and removing them from contention, then a new set of non-dominated individuals are ranked 2, and so forth until all of the individuals are assigned a rank. A so called dummy fitness value is assigned each rank, with rank 1 having the highest fitness value. The same sharing techniques as used by Srinivas and Deb [13] is used here. After the selection of each generation, the new population is put into the repository and update the global Pareto solutions. The repository also crowds out solutions with close objective values in all dimensions.

### 4. Multi-Objective Differential Evolution

The traditional DE is extended to solve discrete problems by introducing the operator as described in section 2. It can also be scaled to discrete multi-objective optimization problem with careful design of selection of best individuals for production operator. Abbas et al [1] and Madavan [10] independently studied the extension of differential evolution to multi-objective optimization problem in continuous domain. As mentioned above, the best individual used in the production operator can easily be identified by choosing the individual with highest fitness value. However, in a multi-objective domain, the purpose of evolutionary algorithm is to identify a set of solutions, the so called Pareto optimal solutions. In this proposed discrete multi-objective differential evolution (D-MODE), a Pareto-based approach is introduced to implement the selection of the best individual for the production operator. At a certain generation of evolutionary algorithm, the population is sorted into several ranks. This is illustrated in the objective space for a bi-objective problem as shown in Figure 4. For any individual in the population, a set of non-dominated individuals,  $D_i$ , that dominates this individual can be identified. In the reproduction operator for a dominated solution in the parent population, the  $p_{best}$ , is chosen randomly from the set  $D_i$ . If the individual is already a non-dominated individual in the parent solution, the  $p_{best}$  will be itself. For a particular case as shown in Figure 4, all of the circled individuals would be the set of  $D_i$ , one of which would be the  $p_{best}$  for production operator of the bold solution  $x$ .

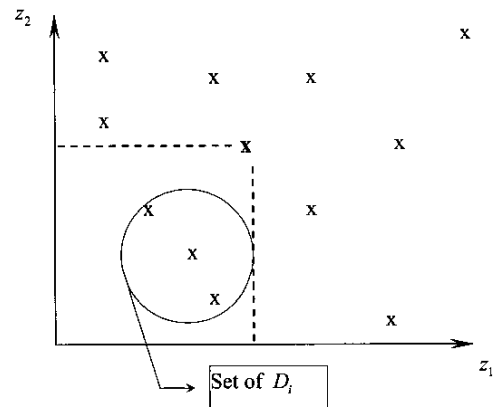


Figure 4: In order to realize the reproduction operator of a dominated individual in current generation, those individuals in first rank that dominate this individual are identified

In order to apply to multi-objective optimization, the Pareto-based fitness assignment and selection of NSGA-II introduced by Deb [3] is incorporated. The NSGA-II algorithm incorporates both an elite-preserving and an

explicit diversity-preserving strategy. The population is sorted as in the NSGA. Instead of computing the niche count to add a penalty to the individuals crowded in a small region, the individuals within each non-dominated front that reside in the least crowded region in that front are assigned a higher rank. A crowding distance metric for a particular individual is obtained by calculating the summation of normalized distance along each objective between the individual and the surrounding individuals within the same non-dominated front. Such a crowding distance metric is used to estimate the density of solutions around such particular individual.

The original NSGA-II applies a  $(\mu + \lambda)$  selection strategy. The individuals are first compared using Pareto rank, if the Pareto rank ties, the crowd distance metric is compared to fill the population of next generation. This strong elitism strategy, however, does not produce good results in our experiments. The authors [2] also point out the importance to keep diversity among non-dominated fronts to allow individuals in lower rank to enter the next generation. In the proposed D-MODE, there is another parameter  $\sigma_{crowd}$  to specify how close the solution is to its surrounding solutions in objective space in order to reduce its fitness. In fact, this parameter will prevent very similar individuals from entering next generation, which might lead to premature convergence.

## 5. Experimental Results

The proposed algorithm is applied to the design, supplier, manufacturing planning problem using design and supplier data from a real commercial electronic circuit board product, and data from three commercial manufacturing facilities.

In the experiment with seven modules, nine suppliers for each module, and four to six contract manufacturers, the total number of possible solution is  $O(10^7)$ . Both the revised NSGA and the proposed D-MODE are applied to find the Pareto set based on criteria of total product time and cost. Various experiments were conducted to simulate the different manufacturability, for instance, the available manufacturers might not have surface mount lines or mixture lines or through-hole lines, or they might have only surface mount lines or all three of them. Due to space limitation here, we only show the experimental results obtained from two of them, where there are only surface mount lines or no surface mount lines, denoted by Only-SM and No-SM, respectively. For a problem of this size, it is possible to identify the real Pareto set using exhaustive search, although it takes a long time to do so. In order to evaluate the performance, both revised NSGA and D-MODE were applied to this problem, and the real Pareto solutions for each possible situation are identified using exhaustive search for comparison. For such a bi-criteria problem the easiest way to compare the computed

results with the real Pareto solution is to plot the real Pareto solutions and the Pareto solutions obtained by the evolutionary algorithm in the two dimensional objective plane.

For both of revised NSGA and D-MODE, the same population size of 200 is used and the same maximal generations of 200 are evolved. The real Pareto front and the computed Pareto front is plotted in the same plane as shown in Figure 5, 6, 7, 8, 9, 10. In all of these figures, a cross indicates a real Pareto solution, while a diamond indicates a computed solution. In Figure 5, computed results obtained using revised NSGA after 200 generations are plotted along with the real Pareto solution for Only-SM experiment; while the computed solutions for the same experiment using D-MODE after 100 and 200 generations are plotted in Figure 6 and Figure 7 respectively. In Figure 8, computed results obtained using revised NSGA after 200 generations are plotted along with the real Pareto solution for No-SM experiment; while the computed solutions for the same experiment using D-MODE after 100 and 200 generations are plotted in Figure 9 and Figure 10 respectively. For the Only-SM experiment, there are totally 21 Pareto solutions. The D-MODE finds 20 of them after 200 generations and 16 of them after 100 generations; while the revised NSGA finds only 14 of them after 200 generations. For the No-SM experiment, there are totally 18 Pareto solutions. The D-MODE finds 17 of them after 200 generations and 9 of them after 100 generations; while the revised NSGA only finds 10 of them after 200 generations. It can be seen that the results obtained using D-MODE are much better than those obtained using revised NSGA in terms of the number of Pareto solutions found and the convergence speed. The results obtained using D-MODE after 100 generations can even compete with the results obtained using revised NSGA after 200 generations.

It is interesting to note that the different performance of revised NSGA from No-SM to Only-SM experiment in terms of roughly fitting of the real Pareto front. It seems that this goodness of fitting is affected by the nature of the optimization problem. When the real Pareto front is roughly evenly distributed as in the No-SM experiment, the revised NSGA can identify an approximate Pareto front, though not real Pareto front, to roughly represent the trade-off nature among the multiple objectives of the optimization problem. In contrast, when the real Pareto front like the one of the Only-SM experiment does not possess roughly distributed solutions, the revised NSGA neglects a large part of solutions resulting in a bad representation of the trade-off nature of the optimization problem. Since the representation of trade-off nature is so important in decision making, this speculation poses an open question on choosing those continuous benchmark functions that always have smooth shape as test beds,

which might have amenable properties for evolutionary algorithms to identify the Pareto front.

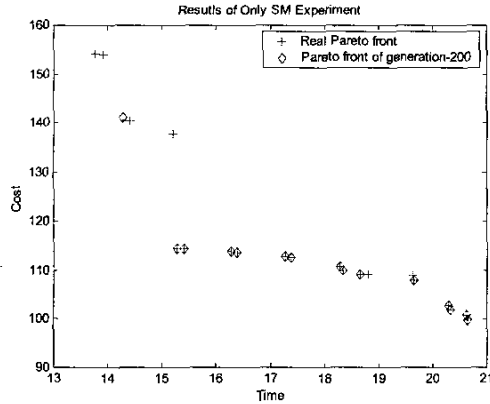


Figure 5: Pareto solutions obtained after 200 generations using revised NSGA and the real Pareto front for Only-SM

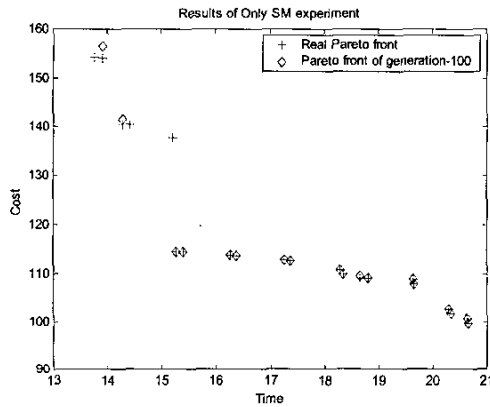


Figure 6: Pareto solutions obtained after 100 generations using D-MODE and the real Pareto front for Only-SM

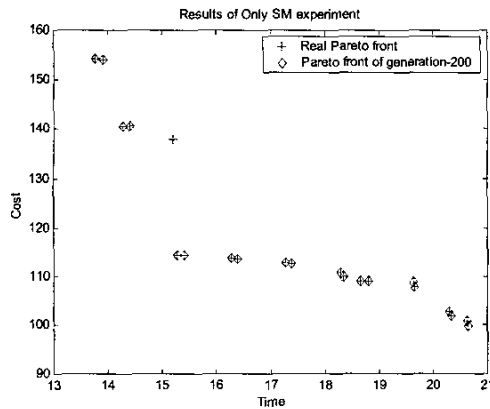


Figure 7: Pareto solutions obtained after 200 generations using D-MODE and the real Pareto front for Only-SM

## 6. Conclusions

Single objective evolutionary algorithms have been extensively applied to various practical problems. However, in many practical problems, there often are multiple objectives which cannot be optimized simultaneously. In this situation, it is essential to identify the trade-off solutions, i.e., the Pareto front, to facilitate the final decision. Multi-objective evolutionary algorithms have been developed to find such Pareto front. In this paper, the concept of differential evolution, which is well-known in single-objective continuous domain for its fast convergence and adaptive parameter setting, is extended to the discrete domain by introducing greedy probability, mutation probability, and crossover probability. Moreover, this concept is extended to the discrete multi-objective optimization problem. The preliminary testing of the proposed multi-objective differential evolution on an integrated design, supplier, manufacturing planning problem shows that this D-MODE is very effective in terms of convergence and the capability to identify Pareto solutions. The experimental results show that this D-MODE has much better performance compared with a revised NSGA, though further experiments need to be conducted to compare with more recent multi-objective evolutionary algorithms such as NSGA-II and SPGA. It is also noted that the experimental results in this paper pose an interesting question on choosing benchmark functions as test beds for multi-objective evolutionary algorithms.

## Acknowledgements

This work has been conducted in the Electronics Agile Manufacturing Research Institute (EAMRI) at Rensselaer Polytechnic Institute. The EAMRI is partially funded by grant number DMI-#0121902 by National Science Foundation.

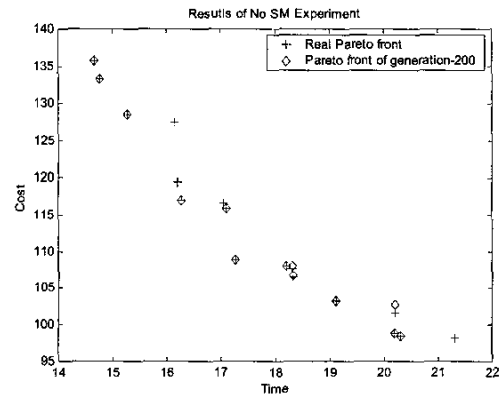


Figure 8: Pareto solutions obtained after 200 generations using revised NSGA and the real Pareto front for No-SM

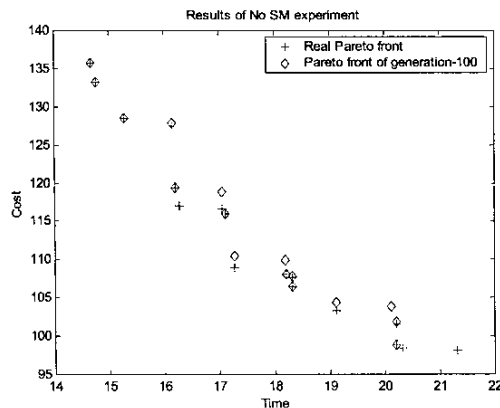


Figure 9: Pareto solutions obtained after 100 generations using D-MODE and the real Pareto front for No-SM

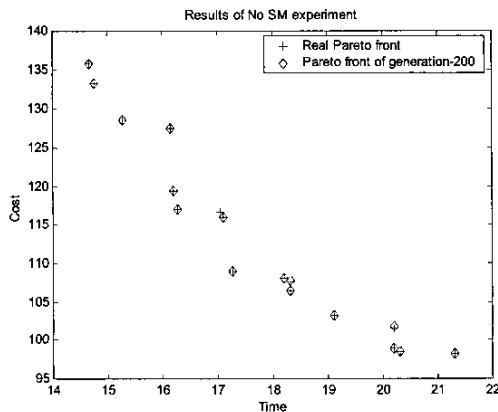


Figure 10: Pareto solutions obtained after 200 generations using D-MODE and the real Pareto front for No-SM

## Reference:

- [1] Abbass, H A, Saker, R, and Newton C, "PDE: A Pareto-frontier differential evolution approach for multi-objective optimization problems", *Proc. of IEEE Congress on Evolutionary Computation*, pp971-978, 2001.
- [2] Deb, K and Goel, T "Controlled elitist non-dominated sorting genetic algorithms for better convergence," *Proc. of the First Int. Conf. on Evolutionary Multi-Criterion Optimization*, pp67-81, 2001
- [3] Deb, K, Pratap, A, Agarwal, S, Meyarivan, T, "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Trans. on Evolutionary Computation*, vol. 6, No. 2, pp182-197, 2002.
- [4] Fonseca C M and Fleming, P J, "Genetic algorithms for multiobjective optimization: formulation, discussion, and generalization", *Genetic Algorithms: Proc. of the Fifth Int. Conf.*, Forrest S ed., pp416-423, 1993.
- [5] Fourman, M P, "Compaction of symbolic layout using genetic algorithms" In Grefenstette JJ ed., *Proc. of the First Int. Conf. on Genetic Algorithms*, pp141-153, 1985.
- [6] Goldberg, D E, *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley, 1989.
- [7] Hajela, P, and Lin, C-Y, "Genetic search strategies in multicriterion optimal design", *Structural Optimization*, 4, pp99-107, 1992.
- [8] Horn J, Nafpitiolis N and Goldberg D, "A niched Pareto genetic algorithm for multiobjective optimization", *Proc. of the First IEEE Conf. on Evolutionary Computation*, pp82-87, 1994.
- [9] Kursawe, F, "A variant of evolution strategies for vector optimization, In Schwefel H-P and R. Manner eds., vol. 496 of *Lecture notes in computer science*, pp193-197, 1991.
- [10] Madavan N K, "Multiobjective optimization using a Pareto differential evolution approach", *Proc. of IEEE Congress on Evolutionary Computation*, pp1145-1150, 2002.
- [11] Schaffer J D, "Multiple objective optimization with vector evaluated genetic algorithm", In Grefenstette J J ed., *Genetic Algorithms and Their Applications: Proc. of the First Int. Conf. on Genetic Algorithms*, pp93-100, 1985.
- [12] Schaffer J D and Grefenstette J J, Multi-objective learning via genetic algorithms, *Proc. of the Ninth Int. Joint Conf. on Artificial Intelligence*, pp593-595, 1985.
- [13] Srinivas, N and Deb, K, "Multiobjective optimization using nondominated sorting in genetic algorithms", *Evolutionary Computation*, 2(3), pp221-248, 1995.
- [14] Storn, R and Price, K, "Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces", Technical Report TR-95-012, International Computer Science Institute, Berkeley.
- [15] Subbu, R, *Network-based distributed planning using coevolutionary algorithms*, Ph.D thesis, Rensselaer Polytechnic Institute, 2001.
- [16] Subbu, R, Hocaoglu, C, and Sanderson, A C, "A virtual design environment using evolutionary agents", In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1998.
- [17] Subbu, R, Sanderson, A C, Hocaoglu, C, and Graves, R J, "Distributed virtual design environment using intelligent agent architecture", In *Proc. of the Industrial Engineering Research Conf.*, 1998.
- [18] Zitzler, E and Thiele, L, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach", *IEEE Trans. on Evolutionary Computation*, vol. 3, No. 4, pp257-271, 1999.