Multi-objective differential evolution – algorithm, convergence analysis, and applications

Feng Xue
General Electric
Global Research Center
1 Research Circle
Niskayuna, NY 12309 USA
xue@research.ge.com

Arthur C. Sanderson
Dept. of Electrical, Computer and Systems Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180 USA
sandea@rpi.edu

Robert J. Graves
Thayer School of Engineering
Dartmouth College
Hanover, NH 03755 USA
robert.j.graves@dartmouth.edu

***Abstract*— The revival of multi-objective optimization (MOO) is mostly due to the recent development of evolutionary multi-objective optimization that allows the generation of the whole Pareto optimal front. Several evolutionary algorithms have been developed for this purpose. This paper focuses on the recent development of differential evolution (DE) algorithms for the multi-objective optimization purposes. Although there are a few other papers on the extension of DE concept to the MOO domain, this paper is intended to provide an overall picture of one specific multi-objective differential evolution (MODE) algorithm. In the MODE, the DE concept for the continuous single-objective optimization is extended to MOO for both continuous and discrete problems (C-MODE and D-MODE, respectively). The MODE is modeled in the context of Markov framework and global random search. Convergence properties are developed for both C-MODE and D-MODE. In particular, a set of parameter-setting guidelines for the C-MODE is derived based on the mathematical analysis. An application of the D-MODE to the planning of design, supply, and manufacturing resources in product development is also reported in this paper.**

I. Introduction

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):

Definition 1 (MOOP): Multi-objective optimization problem (MOOP).

$$\min Z(\mathbf{x}) = \begin{bmatrix} z_1(\mathbf{x}) \\ z_2(\mathbf{x}) \\ \vdots \\ z_k(\mathbf{x}) \end{bmatrix} \tag{1}$$

where $\mathbf{x} \in \chi$, and $\chi = \{\mathbf{x} | h(\mathbf{x}) = 0, g(\mathbf{x}) \leq 0\}$ defines the feasible solutions subset of $\Re^n$; Z, h, g are vector functions among which $Z : \Re^n \rightarrow \Re^k$, $h : \Re^n \rightarrow \Re^{m_1}$, $g : \Re^n \rightarrow \Re^{m_2}$; k is the number of objectives, m_1 and m_2 are the number of equality and inequality constraints, respectively.

In the recent years, evolutionary algorithms have been explored to obtain approximations to the Pareto optimal solutions for multi-objective optimization problems [1]–[4]. Several algorithms based on the evolutionary mechanism have been proposed [5]–[8]. For a thorough review on multi-objective evolutionary algorithms and their applications, readers are suggested to refer [3], [9], [10]. The main theme of this paper is give a summary of the recent development in the multi-objective differential evolution algorithm. The purpose is to link the algorithm, the convergence analysis, and the applications together to give an overall picture. The details of these different aspects will be referred to individual papers. To serve this purpose, the new algorithm for multi-objective optimization is elaborated and a theoretical framework for its convergence is also developed. The planning of design, supply, and manufacturing in the product development is formulated in the framework of multi-objective optimization and the MODE algorithm is applied to solving the problem.

The rest of the paper is organized in the following: section II introduces the MODE algorithms in both continuous and discrete domains and simulation results based on benchmark functions; section III and IV are the modeling and convergence analysis of the MODE in discrete and continuous domains, respectively; section V is the application of MODE to the optimal planning involved in product development; the last section is the conclusion and future research work.

II. Multi-Objective Differential Evolution

A. Differential Evolution

In this section, we introduce the MODE algorithm [11]–[13].

Differential Evolution (DE) is a type of evolutionary algorithm proposed by Storn and Price [14] for optimization problems in a continuous domain. The main motivation of this approach is to adapt the search steps while the evolutionary algorithm is proceeding. It has been empirically shown that this strategy turns out to be very effective in terms of convergence accuracy and speed. Joshi and Sanderson [15], [16] applied this approach to multisensor fusion using minimal representation criteria. Many successful stories have been reported in the literature on applying differential evolution to solving practical problems. A web site[1] maintained by Storn records the whole development of differential evolution and its applications since its origination. The reader can also refer

[1] http://www.icsi.berkeley.edu/ storn/

0-7803-9363-5/05/$20.00 ©2005 IEEE.

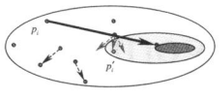

Fig. 1. Illustrative example of differential evolution for single objective in a 2-dimensional decision space. The darker area indicates better fitness value. The thick solid arrow represents the differential vector, and dashed arrows represent the perturbation vectors. The individual p_i creates its offspring p_i' after the reproduction operation.

to a repository of DE bibliography maintained by Lampinen[2] for many other published algorithms and applications.

The success of DE in the optimization of single objective optimization problems is the motivation of extending this basic idea to multi-objective optimization context. Recently, Abbass el al [17] and Madavan [18] independently studied the extension of differential evolution to multi-objective optimization problems in the continuous domain, though applying different approaches.

In DE, every individual p_i in the parent population generates offspring by using the following reproduction operator:

$$p_i' = \gamma \cdot p_{best} + (1-\gamma) \cdot p_i + F \cdot \sum_{k=1}^{K} (p_{i_a^k} - p_{i_b^k}) \quad (2)$$

where p_{best} is the best individual in the parent population, γ represents greediness of the operator, and K is the number of perturbation vectors, F is the scale factor of the perturbation, $p_{i_a^k}$ and $p_{i_b^k}$ are randomly selected mutually distinct individual pairs in the parent population, and p_i' is the offspring that is generated; γ, K, and F are the parameters associated with the algorithm. The DE approach is illustrated schematically in Figure 1. The selection operator chooses the survivor of the parent and the corresponding offspring.

The basic idea of DE is to adapt the search step inherently along the evolutionary process in a manner that trades off exploitation and exploration. The scale of the perturbation vectors is roughly proportional to the extent of the population diversity. At the beginning of the evolution, the perturbation is large since parental 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 benefits the evolution algorithm by performing global search with a large perturbation step at the beginning of the evolutionary process and refine the population with a small search step at the end.

B. Multi-objective differential evolution

In this section, we describe the MODE algorithm in both continuous and discrete domain. The MODE approach has three major components, those are reproduction, Pareto-based evaluation, and selection. These components will be described in detail in this section.

[2]http://www2.lut.fi/ jlampine/debiblio.htm

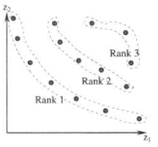

Fig. 2. The non-dominated sorting method assigns the first rank to the non-dominated individuals in the population and the second rank to the non-dominated individuals in the rest of the unlabelled population. This process continues until all the individuals are assigned ranks

1) Pareto-based evaluation: Fitness assignment to the individuals in a evolving population of a multi-objective evolutionary algorithm is rather different than the single-objective counterpart. There are different approaches to evaluating individual fitness of a population, all of which can be regarded as the variants of the non-dominated sorting schema suggested by Goldberg [19]. This Pareto-based rank assignment is employed in a way to assign ranks to individuals based on their Pareto dominance relation to other individuals in the same population. At first, the whole population is considered and all the non-dominated solutions are assigned rank 1, which indicates they have the highest fitness values in the population. At the next step, the already labeled solutions are removed from the population and only the rest subset population is considered and the non-dominated solutions in that subset population are identified and assigned rank 2, which indicates they have the next highest fitness values. The above process continues until every individual in the population is assigned a fitness value. This process is illustrated in Figure 2 where the whole population is divided into multiple ranks with all individuals in each rank assigned the same fitness values.

2) Pareto-based reproduction operator: In order to mimic the reproduction operator in the DE approach described above, we need to define two kinds of vectors: the differential vector and the perturbation vectors for each individual in the population for its reproduction of offspring. In the proposed MODE, a Pareto-based approach is introduced to implement the selection of the best individual to define the differential vector for the reproduction operation of an individual. As described in the Pareto-based evaluation procedure above, the whole population is divided into multiple ranks. The algorithm can identify the set non-dominated solutions (those assigned as first rank) of the population at each generation of the evolutionary process. In order to apply the reproduction operation to an individual, p_i , we need to examine whether the individual is dominated or not. If this is a dominated individual, a subset non-dominated individuals, D_i , that dominates this individual can be identified. A "best" solution, p_{best} , is chosen randomly from the set D_i. The vector defined between p_i and p_{best} becomes the differential vector for the reproduction operation of the individual p_i. If the individual is already a non-dominated individual, the p_{best} will be the individual itself. In this case, the differential vector becomes 0 and only perturbation vectors play effect.

The perturbation vectors are defined by randomly chosen distinct pairs of individuals from the parent population. Thus,

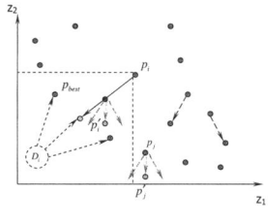

Fig. 3. In order to realize the reproduction operator of a dominated individual in a generation, a set individuals in the first rank that dominate this individual are identified and the differential vector is defined as the vector between one random solution in that set and individual under consideration; a non-dominated individual employs only the perturbation part of the reproduction operator

each allele of an individual will undertake the following procedure with certain reproduction probability p_r (reader can regard an individual as a vector, in which the operation applies to each component of the vector):

$$p'_i = \begin{cases} p_i + F \cdot \sum_{k=1}^{K} \left(p_{i_a^k} - p_{i_b^k}\right) & p_i \text{ is non-dominated} \\ \gamma \cdot p_{best} + (1-\gamma) \cdot p_i + & \\ F \cdot \sum_{k=1}^{K} \left(p_{i_a^k} - p_{i_b^k}\right) & p_i \text{ is dominated} \end{cases} \tag{3}$$

where p_{best} is the best individual in the Pareto sense chosen from the parent population, $\gamma \in [0,1]$ represents greediness of the operator, and K is the number of perturbation vectors, F is the factor that scales the perturbation, $p_{i_a^k}$ and $p_{i_b^k}$ are randomly selected mutually distinct individuals in the parent population, and p'_i is the offspring. This approach is illustrated in the objective space for a bi-objective problem as shown in Figure 3. In this particular case, the dominant set, D_i, of individual p_i contains three individuals, one of which would be the p_{best} for the reproduction operator of individual p_i, and the two dashed arrows are the perturbation vectors. For an individual p_j, which is a non-dominated solution, only the perturbation part is applied. These perturbation vectors are possibly different from the previous ones since they are defined independently for each individual. It should be noted that the operation is applied to decision variables in the decision space though this concept is shown in the objective space in Figure 3 for the convenience of illustration.

In the proposed MODE, all the parent and the children (one parental individual generates one child) are put together to compete for entering into the next generation based on their ranks and the crowd distance metrics.

C. Discrete Multi-Objective Differential Evolution

In a discrete domain, the decision variable usually can be represented by a multi-dimensional vector variable $\mathbf{x} = [x_1, x_2, \cdots, x_n]$, $x_i \in \chi_i, \forall i \in [1,n]$, where χ_i is a set of discrete values that delimits the possible value that the corresponding x_i can have. In many situations, such as in various planning problem involved in supply manufacturing, the elements of the set, χ_i, are integer index, which themselves have no physical meaning. In such situations, the differential vectors in traditional DE have no 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 perturbation probability p_p to apply probabilistic actions to individuals in the evolving population. 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 represents the corresponding value of the real decision vector. From the traditional DE, we know that its key component is the reproduction operator. This reproduction operator will be realized in a probabilistic sense in the discrete DE. With the representation mentioned above, the allele j of offspring of any individual p_i can be obtained in the reproduction operator as follows:

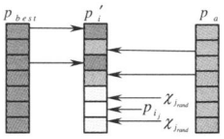

Fig. 4. Discrete DE operator to produce offspring. The individual under operation gets partial gene information from the "best" individual, and partial from a selected individual from current parent population, and partial from a random generation

$$p'_{i_j} = \begin{cases} p_{best_j} & \text{else if } p_g < rand() \\ \chi_{j_{rand}} & \text{else if } p_g + p_m < rand() \\ p_{a_j} & \text{else if } p_g + p_m + p_p < rand() \\ p_{i_j} & \text{else} \end{cases} \tag{4}$$

where $rand()$ is a random number between 0 and 1, $\chi_{j_{rand}}$ is a random selected value from χ_j containing all possible values for allele j, p_a is a randomly selected individual from parent population that is distinct with p_i. 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 small perturbation to introduce random information to the population. Reproducing gene information from other parent individuals is the adaptive perturbation that varies during the evolutionary process. This reproduction operator is illustrated in Figure 4. As long as the reproduction concept is defined, we can use the same Pareto based approach as in the C-MODE section to identify the "best" solution and perform the operator. All other operators follow the way introduced in the previous sections.

D. Simulation results

The proposed MODE was applied to solving a suite of benchmark functions proposed in Zitzler et al [20]. These benchmark functions are carefully designed to represent different families of difficulties to multi-objective evolutionary algorithms. This suite of functions consists of six functions, one of which is discrete and represents a deceptive problem. The other 5 continuous functions ($\mathcal{T}_{1-5}$, $\mathcal{T}_5$ is $\mathcal{T}_6$ in [20]) are

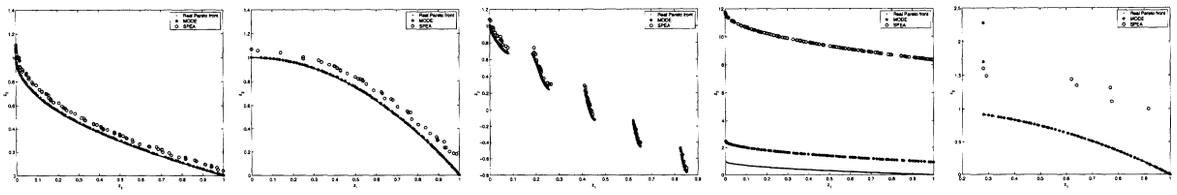

Fig. 5. Typical computed non-dominated front obtained from a single run of MODE and SPEA for each of the five test functions versus the real Pareto front in objective space. The curve in each plot is the real Pareto front, the stars are those obtained using MODE, and the circles are those obtained using SPEA.

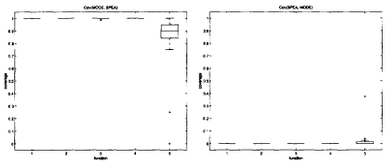

Fig. 6. The left panel is the Box plot for $\mathcal{C}(MODE, SPEA)$ while the right panel is the Box plot for $\mathcal{C}(SPEA, MODE)$.

chosen as the test bed for the continuous MODE, which have been used extensively in the literature as benchmark functions.

In all of the experiments, the maximal generation and population size are set the same as those in the experiments in Zitzler et al [20]. Thus the number of objective function evaluations is roughly the same, which indicates that the experiments conducted in this chapter are based on comparable computational resources. Typical results from one run are shown in Figure 5.

In order to the compare the performance between the proposed MODE and the SPEA from the dominance perspective view, the coverage metric is calculated using the non-dominated solution set from each of the 30 runs. 30 coverage values are calculated for each benchmark function. These values are plotted using the Box plot to see the statistical difference. It is shown in Figure 6 that nearly all solutions obtained from SPEA are dominated by the solutions obtained from MODE in all the independent runs for each of the benchmark functions.

There are few discrete benchmark functions in the literature, though 0/1 knapsack problems are used very often as discrete test cases [8]. It is convenient to construct a set of discrete benchmark functions based on a set of continuous functions by discretizing each dimension of the decision variable vector such that $\mathbf{x} = \{x_1, x_2, \cdots, x_n\}$, each component of $\mathbf{x}$, $x_i \in \chi_i$. χ_i is a set that contains all the values that x_i may take.The testing functions constructed in this way can be easily scaled by increasing or decreasing the number of discrete values that a decision variable can take. The more intervals for each component, the larger the search space.

We discretize the decision variable range into 50 equal intervals, which means we have 51 discrete values for each of the decision variables. The discrete testing functions constructed have decision spaces of $n = 30$, the feasible values for each component x_i belongs to a set, χ_i. Since the dimension of the decision space is 30 and each dimension has 51 alternatives, the whole search space for each test function is very huge, and contains 51^{30} possible solutions. The properties associated with the functions in the continuous space are largely kept in the discrete space. For the convenience of descriptions, these functions are called $\mathcal{DT}_{1-5}$ corresponding to the continuous functions $\mathcal{T}_{1-5}$ described above.

For these discrete functions, the MODE is again performed with the same parameter settings as before using a population size of $N = 200$, greedy probability $p_g = 0.3$, perturbation probability $p_p = 0.2$, mutation probability $p_m = 0.1$, and maximal generations of $G_{max} = 250$. The MODE is performed with the crowd distance $\sigma_{crowd} = 0.001$. We run 30 independent trials with random seeds for each of these test functions. A randomly selected run from the 30 runs for each of these discrete functions is chosen to be plotted in the Figure 7. The D-MODE performs very well for such complex discrete problems. The detailed performance analysis [13] is omitted due to space limitation.

III. Markov model of D-MODE

In this section, we concisely summarize the theoretical analysis of D-MODE in the framework of Markov chain without detailed proofs.

The Markov process has been successfully applied to modeling single-objective evolutionary algorithms in the past such as Rudolph [21], [22], Suzuki [23], Fogel [24]. In most evolutionary algorithm implementations, the evolutionary operators are performed based only on the state of the current population without considering any previous populations along the evolutionary process. Such a way of evolving the population makes the Markov process an appropriate model to describe the evolutionary process. The convergence analysis of multi-objective evolutionary algorithm based on Markov process has been studied in [25].

In this section, the D-MODE is modeled in the framework of Markov process. We studied the Markov model for the D-MODE with finite population size. Two situations are considered: one with a population large enough to contain all the Pareto optimal solutions while the other is the opposite.

We consider the problems with discrete search space $\chi \subseteq \mathbb{Z}^n$, in which there are totally $|\chi|$ possible solutions. The restriction of the search space to a subset of $\mathbb{Z}^n$ can be relaxed and the Markov model can apply to other more general discrete spaces of dimension n as well. The decision variable of the multi-objective optimization problem can be encoded as a gene of length n in the evolutionary algorithm.

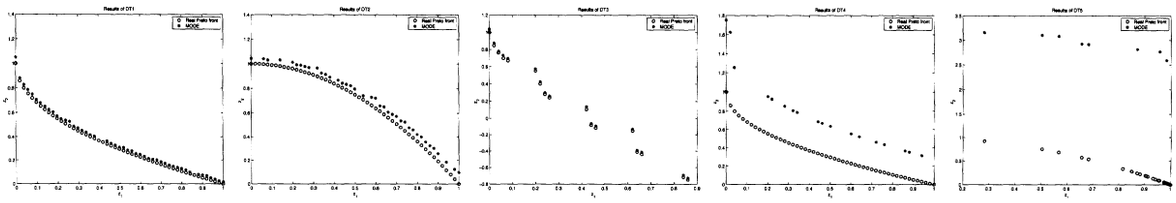

Fig. 7. Computed non-dominated front from a randomly selected run for each of the complex discrete benchmark functions ($\mathcal{DT}_{1-5}$ with 50 discrete intervals for each decision dimension). The circles are the Pareto optimal front while the stars are the computed non-dominated front

Let $\Lambda^* := \{\mathbf{x}^* | \mathbf{x}^* \in \chi, \nexists \mathbf{x} \in \chi : Z(\mathbf{x}) \preceq Z(\mathbf{x}^*)\}$ be the Pareto optimal solution set. The population size of D-MODE is denoted by N. Assuming that we can order the solutions in the whole search space χ in a certain way and each solution is assigned a number as its label ranging from 1 to $|\chi|$. The solutions in the search space are distinguished by their labels $i \in \{1, \cdots, |\chi|\}$. We use a tuple $(y_1, y_2, \cdots, y_{|\chi|})$ to represent the state of a population, where y_i is a nonnegative integer to represent the number of solutions labeled as i that presents in the population with summation of all the y_i equal to N. In addition, the D-MODE operator restricts the presence of duplicated solutions such that $y_i \in 0, 1, \forall i = 1, 2, \cdots, |\chi|$. Therefore, the size of the Markov state space is the total number ways of drawing N balls from a set of $|\chi|$ distinct balls, which is $|S| = C(|\chi|, N)$

A. Markov model with $N \geq |\Lambda^*|$

Since the population size is large enough to contain all the Pareto optimal solutions, we can divide the state space, S, of the Markov process into a series of subspace $S_0, S_1, S_2, \cdots, S_{|\Lambda^*|}$, where the subscript of a particular subset represents the number of Pareto optimal solutions that a population contains if the population takes any state from that subset. For instance, any population state in the subset S_0 contains no solutions from the Pareto optimal set Λ^*; on the other hand, any population state in the subset $S_{|\Lambda^*|}$ contains the whole non-dominated set Λ^* (we know the population size is large enough to contain all these solutions).

Proposition 1: The Markov chain of the D-MODE without selection but with positive mutation probability is irreducible with a positive transition matrix.

Proposition 2: The Markov chain of the D-MODE with selection and positive mutation probability is reducible. The probability transition matrix has all (or partial) states in the subspace $S_{|\Lambda^*|}$ being recurrent and other states being transient.

The above propositions assure that the D-MODE converges to the Pareto optimal set with probability one given that $N \geq |\Lambda^*|$, which is a direct result from reducible Markov chain and that the states containing all the Pareto solutions are the recurrent states.

B. Markov model with $N < |\Lambda^*|$

When the population size is less than the number of Pareto optimal solutions, the transition matrix of the Markov process becomes problem dependent. In general, the population states that consist of N solutions all from Λ^* cannot be a recurrent class. The only exception is that any N distinct solutions from the Λ^* dominate all the solutions from $\chi \setminus \Lambda^*$. In this case, these solutions compose a recurrent class and the Markov process stays in this class once it enters. Even in this particular case, the Markov process visits every state in this recurrent class infinitely often [26] as it proceeds, but we are not able get the full set Λ^* simply because the limited population size.

It is convenient to look at the D-MODE without selection, for which the Markov chain is irreducible with a positive transition matrix. This can assure that the Pareto optimal solutions can be visited infinitely often. We have the following proposition to assure convergence by having a repository external population.

Proposition 3: The D-MODE can obtain all the Pareto optimal solutions with probability one if there is an external population to store any non-dominated solutions found up to the current generation.

The idea is that any population state can be accessed within finite number of generations whatever initial state the Markov process starts with. If all visited Pareto optimal solutions are stored externally, we can guarantee all the Pareto optimal solutions can be obtained in finite number of generations.

IV. Stochastic model of C-MODE

The section is adopted partially from [27];

A. Global convergence analysis

The convergence properties of C-MODE can be studied in a similar manner to [28] by studying the convergence base on global random search. An *ϵ-efficient* solution can be defined as $\Lambda_\epsilon = \{\mathbf{x} : \mathbf{x} \in \chi, Z(\{\prec \mathbf{x}\}) \subseteq N_\epsilon(Z(\mathbf{x}))\}$. In another words, an *$\epsilon$-efficient* solution is a solution that all its dominating solutions are located within its *ϵ-neighborhood* in the criterion space. The *ϵ-neighborhood* of a point, $\mathbf{z}$, in the space $\mathbb{R}^k$ is defined as $N_\epsilon(\mathbf{z}) = \{\mathbf{z}' : |\mathbf{z}' - \mathbf{z}| \leq \epsilon\}$. The convex cone is defined as $V = \{\mathbf{v} : \mathbf{v} \in \mathbb{R}^k, \mathbf{v} \leqslant \mathbf{0}\}$. we denote the set of solutions that dominate a particular solution, $\mathbf{x}$, by $\{\prec \mathbf{x}\} = \{\mathbf{x}' : \mathbf{x}' \in \chi, Z(\mathbf{x}') \in (Z(\mathbf{x}) + V)\}$.

If we approximate the reproduction operator by component-wise Gaussian perturbation, the probability that an individual in the parental population generates an offspring belonging to the set Λ_ϵ via the evolutionary operators is just an integral of a certain Gaussian probability density function over the compact region Λ_ϵ. This probability is positive since the Λ_ϵ region is compact and measurable. Therefore, there exists a

probability $p_t > 0$ at each generation at which the parental population generates an offspring population with all its individuals located within the Λ_ϵ region via the reproduction operators. These p_t's do not need to be the same from generation to another. However, there is a lower bound to this probability corresponding to the population containing solutions with largest distance to the Λ_ϵ. This positive probability of reaching the Λ_ϵ region lead to the following preposition:

Proposition 4: Let the MOOP $(\chi, Z(.))$ be regular. Then

$$\lim_{t\to\infty} Pr(\mathcal{P}^t \subseteq \Lambda_\epsilon) = 1, \forall \epsilon > 0$$

The proposition states that the whole population converges to the Λ_ϵ region with probability one as evolution proceeds.

B. Population evolution and operator effects

To facilitate the mathematical analysis, we assume that the population is initialized by sampling from a Gaussian distribution with a mean μ^0 and a covariance matrix $\mathbf{\Sigma}^0$, that is, any individual in the initial population $\mathbf{X}^0 \sim N(\mu^0, \mathbf{\Sigma}^0)$. The analysis of the C-MODE without the selection operator can be performed with rather simple mathematical analysis on the differential and perturbation operators.

Proposition 5: If the initial population is Gaussian distributed and contains the Pareto optimal set Λ^*, the subsequent populations generated by the C-MODE without selection are also Gaussian distributed and the population mean converges to the center of the Pareto optimal set Λ^*.

This proposition can be deducted from the way of C-MODE reproduction operator. The individuals at generation $t+1$ are generated by: $X^{t+1} = \gamma X^* + (1-\gamma)X^t + F \cdot \sum_{i=1}^{K}(X_{a_i} - X_{b_i})$, where X^t, X_{a_i}, X_{b_i} are individuals from the population at generation t while X^* is a randomly chosen solution in the Pareto optimal set Λ^*. In this way, $X^{t+1} \sim N(\gamma X^* + (1-\gamma)\mu^t, (2KF^2 + (1-\gamma)^2) \cdot \Sigma^t)$.

A model that is closer to the practical application is that the best solutions found by the C-MODE evolve as the algorithm proceeds. Let X^{t*} denotes the an non-dominated solution identified up to generation t while the whole non-dominated solution set is denoted as Λ^t. Similarly, we can regard X^{t*} as a random variable defined on the probability support Λ^t and a specific solution from Λ^t is used to define the differential vector for the whole population. We can have $X^{t+1} = \gamma X^{t*} + (1-\gamma)X^t + F \cdot \sum_{i=1}^{K}(X_{a_i} - X_{b_i})$. We can reformulate the equation by introducing a difference variable $D^t = X^{t*} - X^*$ such that $X^{t+1} = \gamma(X^* + D^t) + (1-\gamma)X^t + F \cdot \sum_{i=1}^{K}(X_{a_i} - X_{b_i})$. Let's define a sequence $q^i = \frac{|E(D^{i-1})|}{|E(D^i)|}, i = 1, \cdots, t$. We can find the maximal value of this sequence $q = \max_{i=1}^{t} q^i$. We can then express the population mean at generation t approximately as $E(X^t) = \gamma \sum_{i=0}^{t-1}(1-\gamma)^{t-i-1}E(X^*) + \gamma \sum_{i=0}^{t-1}(1-\gamma)^{t-i-1}E(D^i) + (1-\gamma)^t\mu^0$.

The mathematical results developed in this section reveal guideline for parameter setting of C-MODE. Two conditions are introduced based on the mathematical analysis to guide the parameter setting of C-MODE:

- $2KF^2 + (1-\gamma)^2 > 1$
- $(1-\gamma)q < 1$

Detailed simulation analysis is conducted in [27] confirming the mathematical results. In practice, we want to set the differential factor and perturbation factor such that they combine to make $2KF^2 + (1-\gamma)^2 > 1$ to allow that the C-MODE has enough capability to explore the search space. On the other hand, we want to set the differential factor γ a relatively large value to loose the restriction on q.

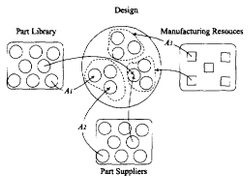

Fig. 8. Structure of the design-supply-manufacturing planning decision problem. Lines with arrowheads indicate assignments. Dashed lines indicate aggregates.

V. Design-Supply-Manufacturing Planning Application

The advances in information technologies are driving fundamental changes in 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, the development of products is based on the creation of strategic and dynamic partnerships among suppliers, contract manufacturers, and customers. Decision makers need to take actions on building and maintaining this partnership and on alternative choices in the realization of product. However, as the numbers of the distinct entities increase, the complexity of forming efficient partnerships grows exponentially; it becomes more difficult to make ideal assignments with respect to multiple criteria including cost, lead time, and quality of the product. It becomes necessary to develop efficient decision-making systems that can automate significant portions of the overall decision task.

A. General DSM planning problem

A model of the integrated design, supplier and manufacturing planning for modular products is shown in Figure 8 and described in [29], [30]. This planning problem consists of three assignment problems (A_1, A_2, A_3). The assignment problem A_1 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 A_2 is the assignment of suppliers (from a list of available suppliers) who will supply the parts in a design, and the assignment problem A_3 is the assignment of designs to available manufacturing resources. Each of these assignments contributes to overall product cost, realization time, and quality with nonlinear (cannot be evaluated as weighted sums) effects on these measures.

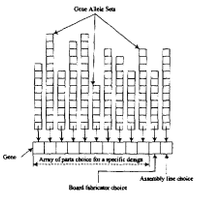

Fig. 9. Evolutionary representation of the PCBA problem

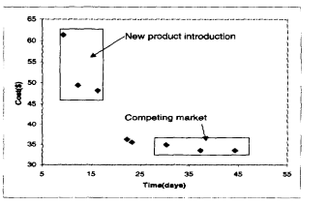

Fig. 10. Decision making based on optimal plans obtained by MODE regarding product cost and lead time criteria.

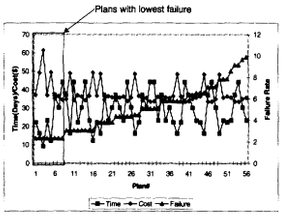

Fig. 11. Optimal plans obtained by MODE regarding product cost, lead time, and quality criteria. The plans are sorted by the quality criteria. In this example, those plans covered by the square are those plans with lowest failure rate.

The objectives considered in the optimization of this planning problem are total cost, total lead time involved in the whole product development, and the quality and reliability of the final product: $C^T = C\left(\bigcup_{p,s,m} \phi_{ps}, \omega_{pm}\right), T^T = T\left(\bigcup_{p,s,m} \phi_{ps}, \omega_{pm}\right), R^T = R\left(\bigcup_{p,s,m} \phi_{ps}, \omega_{pm}\right)$, where $C(\cdot)$ and $T(\cdot)$ are rather complicated models to evaluate the final cost and lead time with various decisions for a particular design as input. They can be further decomposed into such aspects as parts, overhead, and manufacturing. $R(\cdot)$ is a model of the reliability of a particular selection of parts and the corresponding suppliers and manufacturing resources.

B. A printed circuit board assembly example

We consider a planning problem from the printed circuit board assembly (PCBA) as an example. There are three stages to produce a PCB assembly. First, a collection of parts and their suppliers are selected to satisfy the design specifications along with a specification of the layout and the interconnections of the parts on a PCB. Next, a printed circuit board fabricator needs to be selected, which involves the generation of board fabrication characteristics according to design and part choices. Finally, a manufacturing facility needs to be selected that can assemble the product with the chosen parts and circuit board. A *virtual design* for this PCBA planning has an internal representation as shown in Figure 9. This internal data structure is an array of n integers called a genome. Each of the first $n-2$ integer slots (genes) encodes a discrete choice from the set of available parts/suppliers for a part-equivalent class. The gene at position $n-1$ encodes the choice for a PCB fabricator, and the gene at the position n encodes the choice for a PCB assembly line. A set of model-based applications are used to produce the objective values for a given *virtual design*.

We take a PCBA case from a commercial company as an example. In this example, there are $O(10^6)$ number of plans in the search space. With the same problem, we performed a two-stage optimization to investigate the value of optimizing multiple objectives simultaneously. Initially, a cost-time optimization is considered; while all three objectives are optimized in the second stage. In Figure 10, the computed non-dominated plans obtained from the cost-time optimization are plotted in the cost-time space. In this case, the MODE generates 8 Pareto optimal plans. We notice that there are fairly large variations among each dimension in the solution set. The whole tradeoff structure between cost and lead time provided by MODE optimization allows the decision makers to know the overall picture, their current position in this tradeoff structure, and what will the next position if the company needs to change their tradeoff strategies.

In Figure 11, the computed non-dominated plans obtained from cost-time-quality optimization are plotted when all three criteria are considered. There are more non-dominated solutions (56 optimal solutions) generated for this example since more criteria are considered. In this case with more than two criteria, the tradeoff structure is examined by reducing the solution size to look at the lowest failure rate level. Therefore multiple plans have failure rates at the lowest possible level of 2.35. If we plot this subset of solutions in the cost-time space and compare the optimal solutions obtained from the MODE cost-time optimization, the subset of solutions with this lowest failure rate is only slightly worse in the cost dimension (while maintaining the same lead time). Further investigation reveals that the solutions obtained from the cost-time optimization have large variation in their quality index. As the plot in Figure 12 reveals, the plans from cost-time-quality optimization make the product stay in the lowest possible failure rate by incurring only a small increase of product cost. This valuable information cannot be discovered by optimization considering the cost and time criteria only.

The example considered above demonstrates the importance of considering multiple criteria at the same time. The tradeoff information among these multiple criteria captured by performing multi-objective optimization can reveal potential values where optimization with fewer criteria cannot reach. For more details, the reader is referred to [31].

VI. Conclusion and Future Research Work

In this paper, we provide an overall summary of the recent development in multi-objective differential evolution (MODE) and its applications. The main features of the MODE are

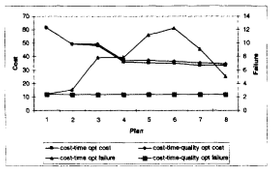

Fig. 12. Extra quality information revealed by MODE in the cost-time-quality optimization. The crosses and diamonds are the failure rate and cost of the highest quality subset solutions obtained from cost-time-quality optimization; while triangles and squares are the failure rate and cost of the solutions obtained from the cost-time optimization.

summarized and simulation analysis is conducted for both C-MODE and D-MODE. The simulation results are compared with those from the SPEA. The results show that the C-MODE outperforms the SPEA in all the benchmark cases. In addition, the D-MODE also performs very well in those very complex discrete benchmark cases.

The D-MODE is studied in the Markov framework, while the C-MODE is studied in the context the global random search and population evolution based on Gaussian initial population assumption. The convergence analysis of the D-MODE to the true Pareto solution set with probability one is derived. For the C-MODE, the convergence of the population to be true Pareto solutions is concluded, though the distribution of the population is unknown in general. Detailed modeling of the population evolution of the C-MODE is conducted based on the assumption of Gaussian initial population. The mathematical analysis also provides parameter setting guidelines to the C-MODE. In order to provide the readers with a complete picture, an application of the MODE to the design-supply-manufacturing planning involved in product development is included. In this application, we want to emphasize the value of having multi-objective optimization as part of the overall decision making process to reveal potential values. In the future research work, we plan to apply the MODE to other real-world problems. Comparison of the MODE with other algorithms is also interesting to investigate.

ACKNOWLEDGMENT

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 from National Science Foundation. Dr. Sanderson was supported in part by grant #IIS 0329837 from the National Science Foundation.

REFERENCES

[1] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: formulation, discussion, and generalization," in *Proc. of the Fifth Int. Conf. on Genetic Algorithms*, 1993, pp. 416–423.

[2] ——, "An overview of evolutionary algorithms in multiobjejctive optimization," *Evolutionary Computation*, vol. 3, no. 1, pp. 1–16, 1995.

[3] C. Coello, "A comprehensive survey of evolutionary-based multiobjective optimization techniques," *International Journal of Knowledge and Information Systems*, vol. 1, no. 3, pp. 269–308, 1999.

[4] D. Van Veldhuizen and G. Lamont, "Multiobjective evolutionary algorithms: analyzing the state-of-the-art," *Evolutionary Computation*, vol. 8, no. 2, pp. 125–147, 2000.

[5] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1995.

[6] J. Horn, N. Nafpliotis, and D. Goldberg, "A niched pareto genetic algorithm for multiobjective optimization," in *Proc. of the First IEEE Conf. on Evolutionary Computation*, 1994, pp. 82–87.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Trans. of Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[8] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Trans. of Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[9] C. Coello, D. Van Veldhuizen, and G. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, 2002.

[10] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley, 2001.

[11] F. Xue, A. Sanderson, and R. Graves, "Multi-objective differential evolution and its application to enterprise planning," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2003.

[12] ——, "Pareto-based multi-objective differential evolution," in *Proc. of IEEE Congress on Evolutionary Computation*, 2003.

[13] F. Xue, "Multi-objective differential evolution: Theory and applications," Ph.D. dissertation, Rensselaer Polytechnic Institute, 2004.

[14] R. Storn and K. Price, "Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces," International Computer Science Institute, Berkeley, Tech. Rep. TR-95-012, 1995.

[15] R. Joshi and A. Sanderson, "Minimal representation multisensor fusion using differential evolution," *IEEE Trans. on Systems, Man, and Cybernetics, Part A*, vol. 29, no. 1, pp. 63–76, 1999.

[16] ——, *Multisensor Fusion: A Minimal Representation Framework*. World Scientific, 1999.

[17] H. A. Abbass, R. Sarker, and C. Newton, "Pde: A pareto-frontier differential evolution approach for multi-objective optimization problems," in *Proc. of IEEE Congress on Evolutionary Computation*, 2001, pp. 971–978.

[18] N. K. Madavan, "Multiobjective optimization using a pareto differential evolution approach," in *Proc. of IEEE Congress on Evolutionary Computation*, 2002, pp. 1145–1150.

[19] D. Goldberg, *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, 1989.

[20] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

[21] G. Rudolph, "Convergence analysis of canonical genetic algorithms," *IEEE Transaction on Neural Networks*, vol. 5, no. 1, pp. 96–101, 1994.

[22] ——, "Convergence of evolutionary algorithms in general search space," in *Proc. of IEEE Int. Conf. on Evolutionary Computation*, 1996, pp. 50–54.

[23] J. Suzuki, "A markov chain analysis on simple genetic algorithm," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 25, no. 4, pp. 655–659, 1995.

[24] D. Fogel, "Asymptotic convergence properties of genetic algorithms and evolutionary programming: analysis and experiments," *Cybernetics and Systems*, vol. 25, no. 3, pp. 389–407, 1994.

[25] G. Rudolph and A. A., "Convergence properties of some multi-objective evolutionary algorithms," in *Proc. of IEEE Congress on Evolutionary Computation*, 2000, pp. 1010–1016.

[26] G. Lawler, *Introduction to stochastic processes*. Chapman & Hall, 1995.

[27] F. Xue, A. Sanderson, and R. Graves, "Modeling and convergence analysis of a continuous multi-objective differential evolution algorithm," in *IEEE Congress on Evolutionary Computation*, 2005.

[28] T. Hanne, "On the convergence of multiobjective evolutionary algorithms," *European Journal of Operational Research*, no. 117, 1999.

[29] R. Subbu and A. Sanderson, *Network-based distributed planning using coevolutionary algorithms*. World Scientific, 2004.

[30] ——, "Modeling and convergence analysis of distributed coevolutionary algorithms," *IEEE Tran. of Systems, Man and Cybernetics, Part B*, vol. 34, no. 2, pp. 806–822, 2004.

[31] F. Xue, A. Sanderson, and R. Graves, "Multi-objective evolutionary decision support for design-supplier-manufacturing planning," in *IEEE Conference on Automation Science and Engineering*, 2005.