Journal of Universal Computer Science, vol. 15, no. 4 (2009), 722-741
submitted: 30/6/08, accepted: 27/2/09, appeared: 28/2/09 © J.UCS

PDE-PEDA: A New Pareto-Based Multi-objective Optimization Algorithm

Xuesong Wang
(School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou, Jiangsu, China
The Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation Chinese Academy of Sciences, Beijing, China
wangxuesongcumt@163.com)

Minglin Hao
(School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou, Jiangsu, China
haofamous@163.com)

Yuhu Cheng
(School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou, Jiangsu, China
chengyuhu@163.com)

Ruhai Lei
(School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou, Jiangsu, China
lrh1960@vip.sina.com)

Abstract: Differential evolution (DE) algorithm puts emphasis particularly on imitating the microscopic behavior of individuals, while estimation of distribution algorithm (EDA) tries to estimate the probabilistic distribution of the entire population. DE and EDA can be extended to multi-objective optimization problems by using a Pareto-based approach, called Pareto DE (PDE) and Pareto EDA (PEDA) respectively. In this study, we describe a novel combination of PDE and PEDA (PDE-PEDA) for multi-objective optimization problems by taking advantage of the global searching ability of PEDA and the local optimizing ability of PDE, which can, effectively, maintain the balance between exploration and exploitation. The basic idea is that the offspring population of PDE-PEDA is composed of two parts, one part of the trial solution generated originates from PDE and the other part is sampled in the search space from the constructed probabilistic distribution model of PEDA. A scaling factor Pr used to balance contributions of PDE and PEDA can be adjusted in an on-line manner using a simulated annealing method. At an early evolutionary stage, a larger Pr should be adopted to ensure PEDA is used more frequently, whereas at later stage, a smaller Pr should be adopted to ensure that offspring is generated more often using PDE. The hybrid algorithm is evaluated on a set of benchmark problems and the experimental results show that PDE-PEDA outperforms the NSGA-II and PDE algorithms.

Keywords: Pareto estimation of distribution algorithm, Pareto differential evolution, multi-objective optimization, offspring generation scheme
Categories: G.1.6, I.1.2, I.2.8

1 Introduction

Problems that have two or more objectives to be simultaneously optimized are common in real-world situations. Such problems are called multi-objective optimization problems (MOPs) and were originally studied in the context of economics [Coello, 1999]. The importance of these multi-objective problems was soon made evident and eventually became an important topic within the field of operations research [Ngatchou, 2005]. Often this task becomes challenging due to the inherent conflicting nature of the objectives to be optimized. The multi-objective optimization problem is an extension of single-objective (scalar) optimization (SOP). Compared to MOP, SOP is much easier to solve, because there is only one solution. In the case of MOP, there may not exist one solution that is best or a global optimum with respect to all objectives. The solution of a MOP is usually a set of acceptable trade-off optimal solutions. This solution set is called a Pareto set [Coello, 1999]. There are, at present, two classes of approaches for solving multi-objective optimization problems [Coello, 1999], [Ngatchou, 2005], [Konak, 2006]. One class consists of the traditional algorithms, including the multi-objective weighted method, the layered optimization method, the constraint method, the global criterion method and the goal programming method. The basic idea of the traditional algorithm is to convert the multi-objective optimization problem to a single objective optimization problem. This conversion is usually done by aggregating all objectives in a weighted function, or simply transforming all but one of the objectives into constraints. The merit of the traditional algorithm is that we can apply existing single-objective optimization algorithms to solve MOP directly. But this approach has several limitations [Ngatchou, 2005]: 1) it requires a priori knowledge about the relative importance of the objectives and the limits on the objectives that are converted into constraints; 2) it cannot find multiple solutions in a single run, thereby requiring it to be applied as often as the number of desired Pareto optimal solutions; 3) trade-off between objectives cannot be easily evaluated and 4) the solution may not be attainable unless the search space is convex.

The second class of approach to solve multi-objective optimization problems are evolutionary algorithms. In recent years, many different intelligent optimization methods such as genetic algorithm (GA), evolutionary strategy (ES) and particle swarm optimization (PSO) have been successfully developed and applied in many domains such as machine learning, process control, economic prediction and engineering optimization [Smith, 2005], [Priandoko, 2006], [Ngom, 2006], [Esmin, 2002], [Krohling, 2006]. Therefore, the application of these optimization methods in solving MOPs has already become a popular research topic. Because these stochastic optimization methods are highly distributed and parallel, they can obtain different compromising solutions in the population simultaneously at each run. In the past few years a considerable amount of interest has been shown in multi-objective evolutionary algorithm (MOEA) and a number of different MOEAs have been suggested, such as NPGA [Horn, 1994], NSGA [Srinivas, 1994] and its improved version NSGA-II [Deb, 2002], SPEA [Zitzler, 1999] and its improved version SPEA2 [Zitzle, 2001], PAES [Knowles, 1999], MOPSO [Coello, 2002] and others. Compared with traditional algorithms, MOEAs are more suitable for solving MOPs for the following reasons [Coello, 2007]: 1) multiple solutions can be found in a single run of a MOEA; 2) a good spread of the nondominated solutions can be obtained and 3) a MOEA is less susceptible to the shape or continuity of the Pareto-optimal front.

The differential evolution (DE) algorithm is one of the most successful evolutionary algorithms (EAs) developed by Storn and Price for optimization problems over continuous domains [Storn, 1997]. In DE, the value of each variable in the chromosome is represented by a real number. Therefore, this method of real number representation is much more practical than a conventional GA. In addition, the mutation operation used by DE has a perfect fine-tuning function, whereas GA does not. Many successful cases have been reported in the literature on applying differential evolution to solving various practical problems [Bhat, 2006], [Kim, 2007], [Nobakhti, 2008]. It has been proven that DE has some special characteristics of good convergence, high efficiency, concise concept and being ease of understand in the field of optimization problems [Wong, 2005]. Recently, the success of DE in the optimization of single-objective optimization problems is the motivation of extending the basic idea in a multi-objective optimization context by using a Pareto-based approach, called the Pareto DE (PDE) [Madavan, 2002], [Xue, 2003], [Robic, 2005].

Like GA, DE algorithm is a genetics-based optimization method, that is, it acquires solutions by carrying out genetics operators such as crossover and mutation for each individual in the population. Its emphasis is particularly on imitating the microscopic behavior of individuals. Therefore, DE algorithm has good local optimization ability and poor global searching ability. In the last few years, a class of novel optimization algorithm, called estimation of distribution algorithm (EDA), has become a favorite topic in the field of evolutionary computation and an efficient approach for various practical engineering problems [Inza, 2000], [Sagarna, 2005], [Simionescu, 2006]. With EDA an entirely new paradigm of evolutionary computation has been introduced, which is a combination of statistical learning theory and stochastic optimization algorithm without using conventional evolutionary operators such as crossover and mutation [Larranaga, 2002]. EDA tries to estimate the probabilistic distribution of an entire population or to describe its evolutionary trend directly from a macroscopic point of view. Therefore, EDA has good global searching ability and poor local optimization ability.

An efficient optimization algorithm should make use both of local information from solutions found so far and global information about the search space. Local solutions found so far can be helpful in exploiting information, while global information can guide the search for exploring promising areas. The search in EDA is mainly based on global information, while DE exploits information on distance and direction, which is a type of local information. Like GA and DE, EDA can be extended to multi-objective optimization problems by using a Pareto-based approach, called Pareto EDA (PEDA). Therefore, this study mainly investigates the combination of PDE and PEDA for MOPs to improve the performance of both algorithms. The proposed hybrid multi-objective optimization algorithm PDE-PEDA can maintain the balance between exploration and exploitation effectively.

The remainder of our presentation is organized as follows. Section 2 provides a brief introduction to MOP. The algorithm steps and the offspring generation scheme of the novel PDE-PEDA algorithm are described in detail in section 3. Extensive experimental studies are conducted in section 4, including a comparative study of the proposed algorithm with PDE and the well-known NSGA-II algorithms on a number of benchmark problems. The last section presents some concluding remarks.

2 Multi-objective Optimization Problem

Consider a general multi-objective minimization problem as presented below:

$$\begin{cases} \min Y = f(x) = \{f_1(x), f_2(x), \cdots, f_R(x)\} \\ \text{s.t. } \Gamma(x) = \{\Gamma_1(x), \Gamma_2(x), \cdots, \Gamma_l(x)\} \leq 0 \\ \quad\;\; \rho(x) = \{\rho_1(x), \rho_2(x), \cdots, \rho_p(x)\} = 0 \end{cases} \tag{1}$$

where, $x = (x_1, x_2, \cdots, x_D)$ is a vector of D-dimensional decision variables, Y is a vector of R-dimensional objective functions, $\Gamma(x)$ and $\rho(x)$ are inequality and equality constraints respectively.

The inherent nature of multi-objective optimization problems is that objective functions are competitive and conflicting. Therefore the global optimization for multi-objective optimization problems is not unique. In MOPs, a number of conflicting objective functions are to be optimized simultaneously. An ideal solution, at which each objective function gets its optimal value, usually does not exist due to the conflicting nature of the objective functions. Thus, a different definition of optimality is required. The solution of a MOP is associated with the definition of a Pareto optimal solution.

A Pareto optimal solution is a key concept in multi-objective optimization problems. This concept formulated by Vilfredo Pareto is defined by Tan et al. in [Tan, 2002]: for decision vector $x^* \in R^D$, if there does not exist any other decision vectors $x \in R^D$ that can make the inequation $f_r(x) \leq f_r(x^*)$, $r = 1,2,\cdots,R$ be true but exists at least one $r_0 \in \{1,2,\cdots,R\}$ that can make inequation $f_{r_0}(x) < f_{r_0}(x^*)$ be true, then the decision vector x^* is called one Pareto optimal solution vector of the MOP.

In general, multi-objective optimization problems have many Pareto optimal solution vectors, so the main task to solve multi-objective optimization problems is to find a collection of Pareto optimal solution vectors that form the Pareto set.

3 PDE-PEDA: A Multi-objective Optimization Algorithm

3.1 Algorithm steps of PDE-PEDA

For the sake of simplicity, without loss of any generality, we consider the following unconstrained multi-objective minimization problem:

$$\min Y = f(x) = \{f_1(x), f_2(x), \cdots, f_R(x)\} \tag{2}$$

In many cases, the constrained optimization problem (1) can be handled as an unconstrained multi-objective optimization problem (2) by using penalty functions. It should be especially noted that the proposed PDE-PEDA algorithm is also suitable for constrained MOPs even though unconstrained MOPs have been used in our investigation.

The algorithm steps of PDE-PEDA for solving multi-objective optimization problems are in the following described in detail.

Step 1: Initialization. An initial population is chosen randomly and should uniformly cover the entire solution space based on the consideration of the requirement of population diversity. For example, an initial population can be represented as the form $x_{j,G} = (x_{j1,G}, \cdots, x_{ji,G}, \cdots, x_{jD,G})$, where $i = 1,2,\cdots,D$, $j = 1,2,\cdots,NP$, NP is the population size, $G = 1,2,\cdots,G_{\max}$ is the generation number and $G_{\max}$ is the maximum generation. As a rule, we will assume a uniform probabilistic distribution for all random decisions unless otherwise stated. Differential evolution can encode real parameters as floating-point numbers, which makes DE well suited for real parameter optimization. Therefore, chromosomes are encoded by real numbers as follows:

$$x_{ji} = low_{ji} + rand \times (high_{ji} - low_{ji}) \tag{3}$$

where $high_{ji}$ and low_{ji} denote the upper and lower bounds of each chromosome respectively and $rand$ is a random number chosen from the range [0, 1].

Step 2: Selection. The Pareto-based evolutionary algorithm developed here differs from the basic algorithm primarily in the selection procedure used to select subsequent generations of the population. Low-quality solutions (individuals) are removed from the population, while high-quality individuals are reproduced. We have opted to use the nondominated sorting and ranking selection procedure developed by Deb et al. in [Deb, 2002] to select nondominated individuals to constitute a nondominated solution set.

Step 3: Modeling. A probabilistic distribution model is built, based on statistical information extracted from the solutions in this nondominated solution set according to the PEDA.

Step 4: Offspring generation. Suppose $0 \le p_r \le 1$, the EDA is used to generate a new individual when $RAND < p_r$, i.e., the sample of a new individual in the search space given the constructed probabilistic distribution model of PEDA. On the other hand, when $RAND > p_r$, the DE algorithm is used to generate a new individual, i.e., crossover and mutation operators are applied to the parent individuals to generate a new individual. $RAND$ is also a uniform random number chosen from the range [0, 1].

The scaling factor p_r is used to control the contributions of PEDA and PDE to offspring generation or to balance the use of global and local information. All individuals in the new population are sampled from the probabilistic model of PEDA when $p_r = 1$, while all individuals in the new population are generated from the

PDE when $p_r = 0$. At an early evolutionary stage, a larger p_r should be adopted to ensure the dominant function of PEDA and to enhance its global searching ability. PEDA directs PDE to search along the Pareto front. The scaling factor should be reduced during the evolutionary process to force PDE to take up the dominant function gradually and to ensure the solutions converge to a true Pareto front. Therefore, the scaling factor p_r can be adjusted in an on-line manner using a simulated annealing method.

$$\begin{cases} p_r^0 = p_r^{\max} \\ p_r^{G+1} = p_r^{\min} + \beta(p_r^G - p_r^{\min}) \end{cases} \tag{4}$$

where $p_r^{\max}$ and $p_r^{\min}$ are the upper and lower bounds of the scaling factor and $0 \le \beta \le 1$ is an annealing factor.

Step 5: Boundary treatment. Because the search space is limited, it is essential to ensure that parameter values lie inside their allowed range $(low_{ji},\ high_{ji})$. A simple boundary treatment equation is shown as follows:

$$x_{ji} = \begin{cases} x_{ji} + Rand \times (high_{ji} - x_{ji}), & \text{if } \ x_{ji} > high_{ji} \\ x_{ji} - Rand \times (x_{ji} - low_{ji}), & \text{if } \ x_{ji} < low_{ji} \end{cases} \tag{5}$$

where $Rand$ is a random number chosen from the range [0, 1].

Step 6: Combine the new population with the existing parent population. Note that the size of the combined population is $2 * NP$.

Step 7: Selection. Evaluate each individual in the combined population and carry out the nondominated sorting and ranking selection, developed by Deb et al. in [Deb, 2002] to select nondominated individuals to constitute a nondominated solution set.

Basic differential evolution algorithm usually adopts the following method to update each individual in the population. It determines whether the fitness value of the new individual is superior to that of the parent individual. If not, the parent individual is preserved for the next generation. Otherwise, the new individual will replace its parent in the next generation. For MOPs, the new individual that is not dominant or is dominated by its parent is possibly superior to the other new individuals or parent. If we adopt the above basic individual updating method, the obtained optimization information is easily lost, which will affect the evolutionary result. Therefore, all the parent individuals are combined with the new individuals generated from PDE and PEDA to form a combined population with size $2 * NP$.

Step 8: Stopping condition. If the stopping condition is met, stop. Otherwise return to Step 3. The commonly used stopping condition is to set a maximum number of evolutionary generations.

3.2 Offspring generation scheme of PDE

A differential evolution algorithm is conceptually simple and possesses good convergence properties that have been demonstrated in a variety of applications. It is a genetics-based approach in continuous search spaces [Storn, 1997]. Differential evolution can be extended to multi-objective optimization problems by using a Pareto-based approach. The Pareto-based DE algorithm, developed here differs from the basic algorithm used to select subsequent generations of the population in the selection procedure. Therefore, a candidate individual in PDE can be generated according to the following mutation and crossover operations, commonly used in a basic DE.

3.2.1 Mutation

The main difference between DE and other evolutionary algorithms is the implementation of a mutation operation. The mutation operation of DE applies vector differentials between the existing population members for determining both the degree and direction of perturbation.

Mutation is primarily responsible for keeping a population robust and for searching new territory. Instead of using a predefined probability to carry out mutation operation on genes, differential evolution is self-adjusting since it deduces the perturbations from the distances between the vectors. The inherent ‘self-adjusting’ of DE works as follows: as the population converges to an optimum, any randomly chosen difference vector will become smaller in size. Eventually when all members converge to a single solution, the difference vector will be zero and the mutation operator will be disabled altogether. Differential evolution generates a perturbed vector $W_{j,G}$, corresponding to the parameter vector $x_{j,G}$, using a mutation scheme. New perturbed vectors are generated by adding the weighted difference to another vector.

In [Storn, 1997], the authors developed a set of mutation schemes that allow a large number of options, depending on the nature of the problem, which are called DE/rand/1, DE/best/1, DE/rand/2, DE/best/2 and DE/rand-to-best/1. We have used scheme DE/rand/1.

Scheme DE/rand/1

In this scheme, for each vector $x_{j,G}$, a perturbed vector is generated according to equation (6).

$$W_{j,G} = x_{r_1,G} + F(x_{r_2,G} - x_{r_3,G}) \tag{6}$$

where $r_1, r_2, r_3 \in [1, NP]$ are integers chosen randomly and $r_1 \neq r_2 \neq r_3 \neq j$. The mutation factor $F \in [0, 2]$ is a real number, which controls the amplification of the differential variation. It should be noted that there is no relationship between parameter vectors $x_{r_1,G}$ and $x_{j,G}$.

It can be seen from equation (6) that, unlike in a genetic algorithm, in DE the degree of mutation is derived from a difference vector that is calculated using members of the current populations. Figure 1 illustrates the generation process of a perturbed vector defined by equation (6) within a solution space.

3.2.2 Crossover

Once the perturbed vector is created, it will undergo a crossover operation to increase the population diversity and avoid falling into a local minimum. According to the jth population vector $x_{j,G}$ and its corresponding perturbed vector $W_{j,G}$, the crossover creates a new trial vector $U_{j,G}$ with the rule

$$U_{j,G} = (u_{j1,G}, u_{j2,G}, \cdots, u_{jD,G}) \tag{7}$$

$$u_{ji,G} = \begin{cases} w_{ji,G}, & \text{if } \mathrm{rand}b(i) \le CR \quad \text{or} \quad i = \mathrm{mbr}(j) \\ x_{ji,G}, & \text{if } \mathrm{rand}b(i) > CR \quad \text{and} \quad i \ne \mathrm{mbr}(j) \end{cases} \tag{8}$$

where $\mathrm{rand}b(i)$ is a randomly chosen real number within the range [0, 1]. Index $\mathrm{mbr}(j)$ is a randomly chosen integer within the range [1, D] and is responsible for the trial vector to contain at least one parameter from the perturbed vector. CR is a crossover factor within the range [0, 1] and is the probability to create parameters of the trial vector from the perturbed vector.

The idea of crossover operation is illustrated in figure 2 for D=6 and $\mathrm{mbr}(j) = 2$.

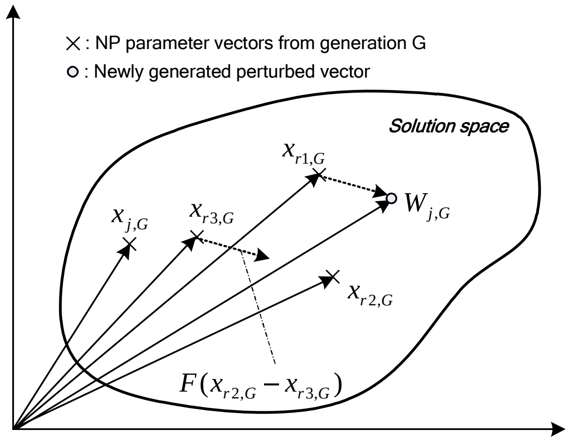


Figure 1: Sketch map of scheme DE/rand/1

3.3 Offspring generation scheme of PEDA

Estimation of distribution algorithm, also called a probabilistic model-building genetic algorithm (PMBGA) [Pelikan, 1999], is a new area of evolutionary computation. In EDA there is neither a crossover nor a mutation operator. It directly extracts global statistical information about the search space from the search so far and builds a probabilistic model of promising solutions. Then sampling this probabilistic model generates offspring. In such a way, the relationships between the variables involved in the problem domain are explicitly and effectively exploited. EDA can be extended to multi-objective optimization problems by using a Pareto-based approach. The Pareto-based EDA developed here differs from the basic algorithm primarily in the selection procedure used to select subsequent generations of the population. Therefore, a candidate individual in PEDA can be generated according to the following modeling and sampling operations commonly used in a basic EDA.

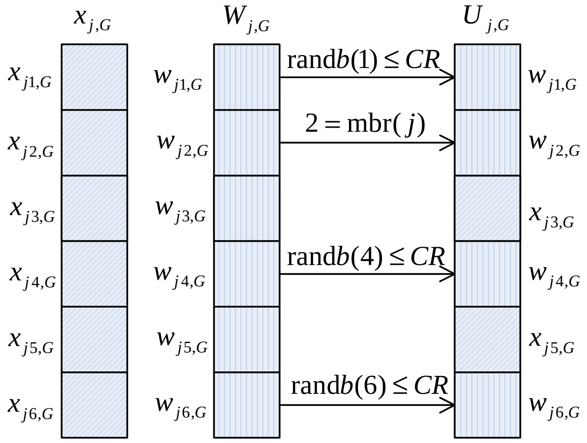
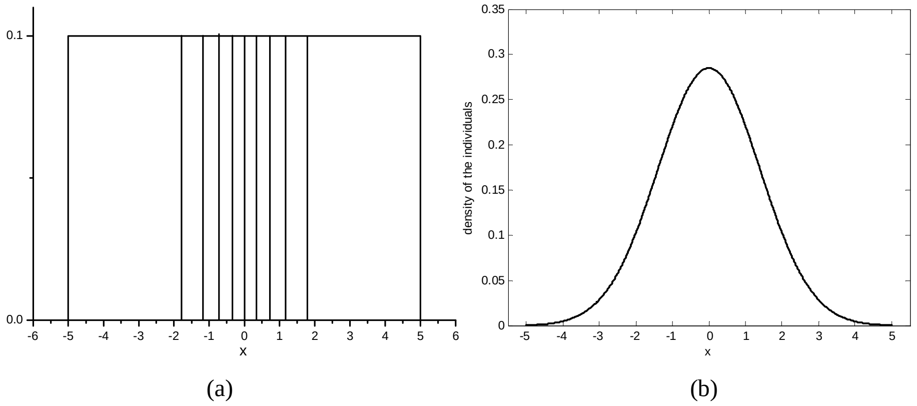


Figure 2: Illustration of the crossover operation

3.3.1 Building probabilistic model

A major issue in EDA is how to build a probabilistic distribution model. The estimation of the joint probabilistic distribution associated with the database containing the selected individuals is not an easy task. Generally speaking, in EDA for the global continuous optimization problem, the probabilistic model can be a Gaussian distribution [Quinlan, 1986], a Gaussian model with a diagonal covariance matrix (GM/DCM) [Quinlan, 1993], a Gaussian mixture [Brieman, 1984], or a histogram [Tsutsui, 2001]. Among these popular probabilistic models, the histogram is the most straightforward method to estimate probabilistic density.

There are two types of marginal histogram models: fixed-width histogram (FWH) and fixed-height histogram (FHH) [Tsutsui, 2001]. In our study, we assumed that the selected set of individuals obey the fixed-height histogram model. In a FHH model, we divide the search space $[a_i, b_i]$ of each variable x_i into H bins (subintervals). Each bin has the same height, which means that each bin contains the same number of individuals. Then the range of the width of each bin can be calculated according to the number of individuals. The important characteristic of FHH is that the bins in dense regions are narrower and thus the accuracy of the model in important regions increases. In the context of evolutionary algorithms, the width of bins around high peaks decreases as more individuals are located in these areas. Since the probability of generating a new individual from each bin in the FHH is the same, we expect both the density as well as the accuracy to improve in promising regions of the search space. Figure 3(a) gives an example of the FHH for a population which has a normal distribution $N(0,1.4)$ for variable x_i in the range [-5, 5] as shown in figure 3(b). Ten bins were used. In this example, more individuals were sampled around $x_i = 0$ because the bin width around $x_i = 0$ is narrower and therefore there are more bins for the same area.



(a) (b)

Figure 3: Example of the FHH of normal density

3.3.2 Sampling

A new individual can be generated as follows: first, select stochastically a bin h_s ($s = 1,2,\cdots,H$) with a probability of $1/H$. Then generate a number from the selected bin with uniform distribution. This process is repeated until all individuals are obtained. It should be noted that each bin has the same selection probability, i.e., $1/H$.

4 Experimental Results

In this section, PDE-PEDA is tested on a set of commonly used multi-objective minimization problems in order to evaluate its performance and compared with a PDE and one of the state-of-the art NSGA-II. These benchmark problems are collected from the published literature and include various relevant features such as low dimensional and high dimensional decision spaces, convex and nonconvex Pareto optimal fronts, continuous and discontinuous fronts, symmetric and asymmetric fronts, etc. Many of these features may cause difficulties in a multi-objective optimization algorithm. Table 1 shows eight test problems used in this study. It should be noted that all objective functions are to be minimized.

Problem	D	Variable bounds	Objective functions
SCH	1	$\left[-10^3, 10^3\right]$	$f_1(x) = x^2$ $f_2(x) = (x-2)^2$
FON	3	$\left[-4, 4\right]$	$f_1(x) = 1 - \exp(-\sum_{i=1}^{D}(x_i - \frac{1}{\sqrt{3}})^2)$ $f_2(x) = 1 - \exp(-\sum_{i=1}^{D}(x_i + \frac{1}{\sqrt{3}})^2)$
KUR	3	$\left[-5, 5\right]$	$f_1(x) = \sum_{i=1}^{D-1}(-10\exp(-0.2\sqrt{x_i^2 + x_{i+1}^2}))$ $f_2(x) = \sum_{i=1}^{D}(\lvert x_i\rvert^{0.8} + 5\sin x_i^3)$
ZDT1	30	$\left[0, 1\right]$	$f_1(x) = x_1$ $f_2(x) = g(x)(1 - \sqrt{x_1 / g(x)})$ $g(x) = 1 + 9(\sum_{i=2}^{D} x_i) \big/ (D-1)$
ZDT2	30	$\left[0, 1\right]$	$f_1(x) = x_1$ $f_2(x) = g(x)(1 - (x_1 / g(x))^2)$ $g(x) = 1 + 9(\sum_{i=2}^{D} x_i) \big/ (D-1)$
ZDT3	30	$\left[0, 1\right]$	$f_1(x) = x_1$ $f_2(x) = g(x)(1 - \sqrt{\frac{x_1}{g(x)}} - \frac{x_1}{g(x)}\sin(10\pi x_1))$ $g(x) = 1 + 9(\sum_{i=2}^{D} x_i) \big/ (D-1)$
ZDT4	10	$x_1 \in \left[0, 1\right],$ $x_i \in \left[-5, 5\right],$ $i = 2, \cdots, D$	$f_1(x) = x_1$ $f_2(x) = g(x)(1 - \sqrt{x_1 / g(x)})$ $g(x) = 1 + 10(D-1)$ $+ \sum_{i=2}^{D}(x_i^2 - 10\cos(4\pi x_i))$
ZDT6	10	$\left[0, 1\right]$	$f_1(x) = 1 - \exp(-4x_1)\sin^6(6\pi x_1)$ $f_2(x) = g(x)(1 - (f_1(x) / g(x))^2)$ $g(x) = 1 + 9(\sum_{i=2}^{D} x_i \big/ (D-1))^{0.25}$

Table 1: Benchmark problems used in this study

In order to make a fair comparison, for NSGA-II, PDE and PDE-PEDA a real-coded mode was adopted and their common parameters were set the same. For all benchmark problems, a population of size 100 was used and each experiment carried out to 250 generations. For the real-coded NSGA-II, the simulated binary crossover (SBX) and polynomial mutation operators with distribution indices $\eta_c = 20$ and $\eta_m = 20$ were used in [Deb, 2002]. For PDE and PDE-PEDA, we chose a reasonable set of values and did not make any effort in finding the best parameter settings for different problems. The detailed parameter settings are shown in Table 2.

The optimization goal of multi-objective optimization algorithms is: 1) to minimize the distance between the generated and the true Pareto front and 2) to maintain a good diversity in solutions of the Pareto optimal set. The performance of the algorithms is evaluated with respect to the following metrics. The details of these metrics may be found in [Deb, 2002] or [Khare, 2003].

1) Metric γ measures the extent of convergence to a known set of Pareto optimal solutions. The smaller the value of γ, the better the convergence toward a Pareto optimal front.

2) Metric Δ measures the extent of spread achieved among the obtained solutions. The smaller the value of Δ, the better the diversity.

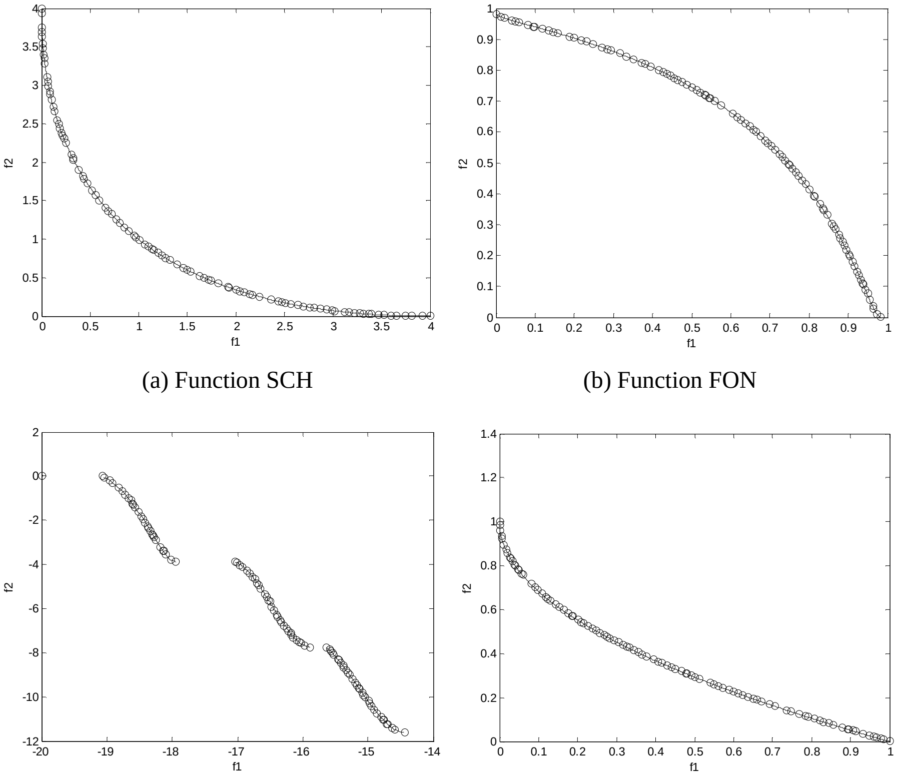
Algorithm	η_c	η_m	F	CR	NP	$G_{\max}$	$p_r^{\max}$	$p_r^{\min}$	β	H
NSGA-II	20	20	/	/	100	250	/	/	/	/
PDE	/	/	0.3	0.3	100	250	/	/	/	/
PDE-PDEA	/	/	0.3	0.3	100	250	0.9	0.2	0.95	20

Table 2: Parameter settings

Algorithm	**SCH**	**FON**	**KUR**	**ZDT1**
NSGA-II	0.003391	0.001931	0.028964	0.033482
	0	0	0.000018	0.004750
PDE	0.001600	0.000715	0.027312	0.001187
	1.50000e-8	2.27821e-8	2.80982e-6	9.82142e-9
PDE-PEDA	**0.0015917**	**0.000697**	**0.023050**	**0.000971**
	1.35606e-8	9.03766e-9	6.84544e-5	3.92606e-9
Algorithm	**ZDT2**	**ZDT3**	**ZDT4**	**ZDT6**
NSGA-II	0.072391	0.114500	**0.513053**	0.296564
	0.031689	0.007940	0.118460	0.013135
PDE	0.001325	0.001425	42.22191	0.012050
	5.64285e-8	1.59285e-8	2.088529	7.0634e-5
PDE-PEDA	**0.001188**	**0.001363**	0.651008	**0.005227**
	5.37550e-8	5.53571e-9	0.227134	2.2220e-5

Table 3: Mean (first row) and variance (second row) of the convergence metric γ

Figure 4 shows one of the resultant nondominated solutions in 10 runs corresponding to all the studied problems using PDE-PEDA, where solid lines and hollow circles, respectively, denote the true Pareto front and the obtained nondominated solutions.These figures demonstrate the abilities of PDE-PEDA in converging to the true front and in finding diverse solutions in the front. The problem ZDT4 has 21^9 different local Pareto-optimal fronts in the search space, of which only one corresponds to the global Pareto-optimal front. Figure 4.(g) shows that PDE-PEDA get stuck at different local Pareto-optimal sets. It can be seen from these fugures, in both aspects of convergence and distribution of solutions, PDE-PEDA peformed well on all benchmark problems except on ZDT4.

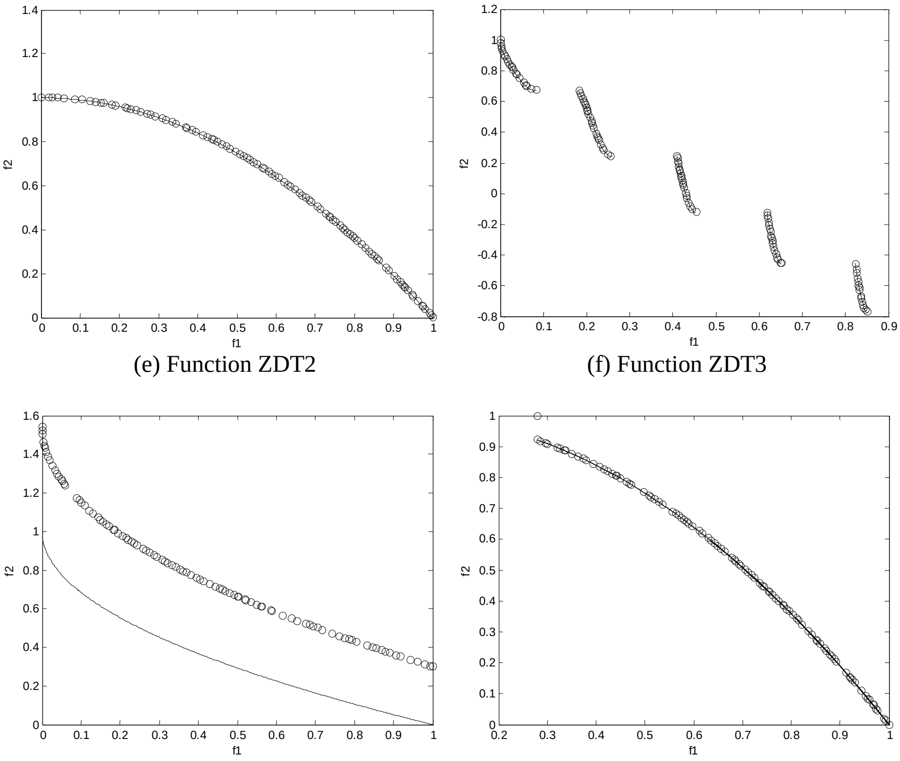


(a) Function SCH

(b) Function FON

(c) Function KUR

(d) Function ZDT1



(e) Function ZDT2

(f) Function ZDT3

(g) Function ZDT4

(h) Function ZDT6

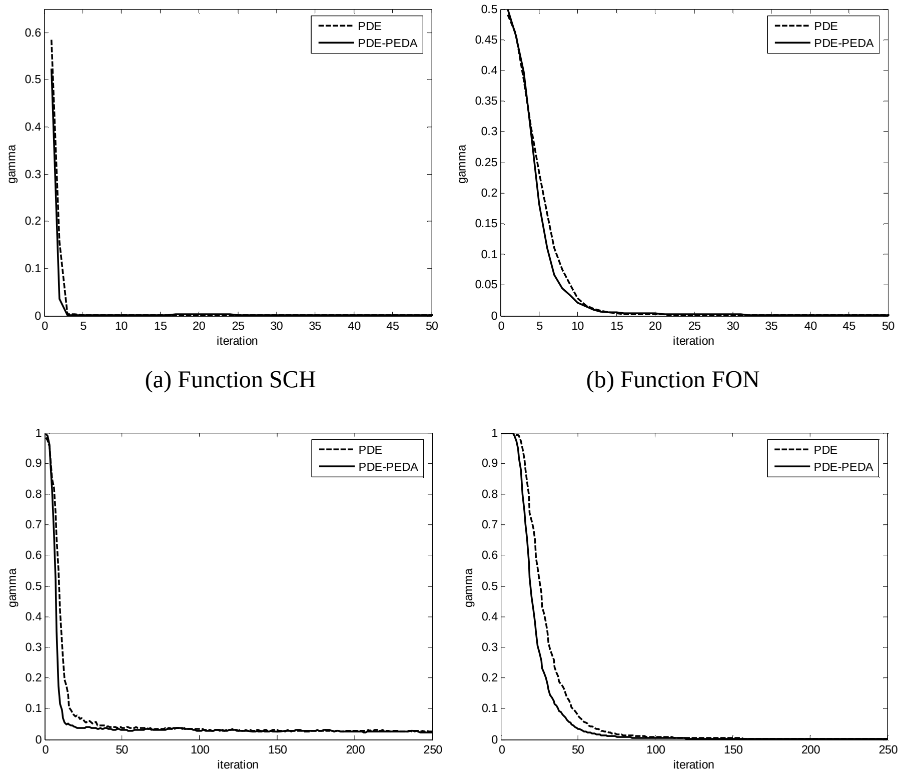
Figure 4: Nondominated solutions with PDE-PEDA on Benchmark functions

Algorithm	**SCH**	**FON**	**KUR**	**ZDT1**
NSGA-II	0.477899	**0.378065**	0.411477	0.390307
	0.003471	0.000639	0.000992	0.001876
PDE	0.410946	0.401087	0.407138	0.374037
	2.39182e-3	6.81966e-4	3.08166e-4	1.38130e-3
PDE-PEDA	**0.407875**	0.389124	**0.406625**	**0.351025**
	1.09088e-3	1.07227e-3	1.54365e-4	8.23222e-4
Algorithm	**ZDT2**	**ZDT3**	**ZDT4**	**ZDT6**
NSGA-II	0.430776	0.738540	0.702612	0.668025
	0.004721	0.019706	0.064648	0.009923
PDE	0.383337	0.571987	0.575167	0.945390
	1.69575e-3	1.02328e-3	0.175642	0.142226
PDE-PEDA	**0.341925**	**0.525325**	**0.392650**	**0.582955**
	5.07730e-4	6.62290e-4	7.3551e-4	0.058701

Table 4: Mean (first row) and variance (second row) of the diversity metric Δ

Table 4 shows the mean and variance of the diversity metric Δ obtained in 10 independent runs using real-coded NSGA-II, PDE and PDE-PEDA. We show two rows for each benchmark problem. The first row presents the mean of Δ in 10 runs and the second row shows its variance. For each benchmark problem, the smallest value of the mean of Δ is shown in bold numbers. As can be clearly seen from Table 4, PDE-PEDA is able to find a better spread of solutions on all benchmark problems except on FON, where NSGA-II found better diversity.

In order to evaluate the convergence speed of algorithms, the changes of the values of γ along with each generation are presented in figure 5 on all the studied problems using PDE and PDE-PEDA, where solid and dashed lines denote PDE-PEDA and PDE respectively. It can be seen from figure 5 that, compared with PDE, the convergence speed of PDE-PEDA on these MOPs are improved to different degrees.

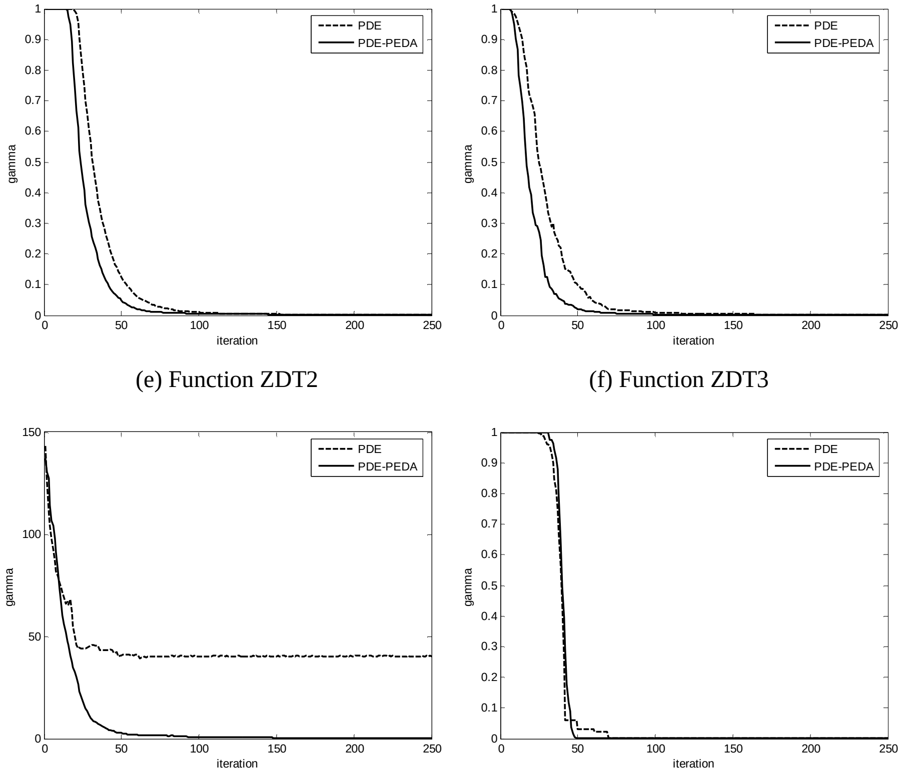


(a) Function SCH

(b) Function FON

(c) Function KUR

(d) Function ZDT1



(e) Function ZDT2

(f) Function ZDT3

(g) Function ZDT4

(h) Function ZDT6

Figure 5: Metric γ along with generation on Benchmark functions

In summary, our experimental results show that PDE-PEDA, compared with NSGA-II and PDE algorithms, has better convergence performance. In addition, because estimation of distribution algorithm with good global search property is introduced into the Pareto-based differential evolution, PDE-PEDA not only can exploit in depth but also can explore in width, which makes for effective improvements in both the diversity of the obtained nondominated solutions and the convergence speed.

5 Conclusions

Many real-world problems involve the simultaneous optimization of various noncommensurable and conflicting objectives that are difficult, if not impossible, to solve without the aid of powerful optimization algorithms. In recent years, many different stochastic optimization methods such as GA, ES, PSO and DE have been successful developed and applied to solve MOPs by using a Pareto-based approach. While it has been shown that these biologically inspired heuristics offer better performances over classical optimization approaches in complex MOPs, they are plagued by their own limitations in exploration ability.

In our study, a novel hybrid optimization algorithm PDE-PEDA is presented for solving MOPs by introducing estimation of distribution algorithm, with good global exploration ability into Pareto-based DE. PEDA is used to guide PDE to search along the Pareto front, while PDE is used to ensure that solutions converge to the true Pareto front. PDE-PEDA combines local information obtained by PDE with global information extracted by PEDA to create offspring. The offspring of PDE-PEDA is composed of two parts, one part generated originates from PDE and the other part is sampled from the constructed probabilistic distribution model of PEDA. In order to maintain the balance between exploration and exploitation effectively, we designed a scaling factor that can be adjusted in an on-line manner using a simulated annealing method. At an early evolutionary stage, a larger scaling factor should be adopted to ensure PEDA is used more frequently, whereas at later stage, a smaller scaling factor should be used to ensure the offspring are generated more often, using PDE. Our experimental results on several benchmark problems show that PDE-PEDA, compared with NSGA-II and PDE algorithms, can find many Pareto optimal solutions distributed onto the Pareto front and can improve convergence speed effectively. It may be noted that although the proposed algorithm has been demonstrated for unconstrained optimal problems with only two objective functions, it can be used for constrained problems with more than two objective functions.

For future research, we will investigate the possibility to relate the scaling factor with performance metrics such as convergence or diversity metrics in order to make it possible for the algorithm to choose the PDE and the PEDA in a more suitable way.

Acknowledgements

This work was supported by grants from the National Nature Science Foundation of China (60804022), the Nature Science Foundation of Jiangsu Province (BK2008126), the Specialized Research Fundation for the Doctoral Program of Higher Education of China (20070290537, 200802901506), the National Science Foundation for Post-doctoral Scientists of China (20070411064), the Science Foundation for Post-doctoral Scientists of Jiangsu Province (0601033B) and the Scientific and Technological Foundation of China University of Mining and Technology (0C080302).

References

[Bhat, 06] Bhat, T.R., Venkataramani, D., Ravi, V., Murty, C.V.S.: An improved differential evolution method for efficient parameter estimation in biofilter modeling, Biochemical Engineering Journal, vol 28, no 2, pp 167-176, 2006.

[Brieman, 84] Brieman, L., Friedman, J., Olshen, R., Stone, C.: Classification of regression trees, Wadsworth Inc., 1984.

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

[Coello, 2002] Coello, C.A.C, Lechuga, M.S.: MOPSO: a proposal for multiple objective particle swarm optimization, Proceedings of the IEEE International Conference on Evolutionary Computation, Honolulu, HI, USA, pp 1051-1056, 2002.

[Coello, 2007] Coello, C.A.C, Aquirre, A.H., Zitzler, E.: Evolutionary multi-objective optimization, European Journal of Operational Research, vol 181, no 3, pp 1617-1619, 2007.

[Deb, 2002] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation, vol 6, no 2, pp 182-197, 2002.

[Esmin, 2002] Esmin, A.A.A., Aoki, A.R., Lambert-Torres, G.: Particle swarm optimization for fuzzy membership functions optimization, Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Yasmine Hammamet, Tunisia, pp 106-111, 2002.

[Horn, 1994] Horn, J., Nafpliotis, N., Goldberg, D. E.: A niched Pareto genetic algorithm for multiobjective optimization, Proceedings of the 1st IEEE International Conference on Evolutionary Computation, Orlando, FL, USA, pp 82-87, 1994.

[Inza, 2000] Inza, I., Larranaga, P., Etxeberria, R., Sierra, B.: Feature subset selection by Bayesian network-based optimization, Artificial Intelligence, vol 123, no 1-2, pp 157-184, 2000.

[Khare, 2003] Khare, V., Yao, X., Deb, K.: Performance scaling of multi-objective evolutionary algorithms, Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization, Faro, Portugal, pp 376-390, 2003.

[Kim, 2007] Kim, H.K., Chong, J.K., Park, K.Y., Lowther, D.A.: Differential evolution strategy for constrained global optimization and application to practical engineering problems, IEEE Transactions on Magnetics, vol 43, no 4, pp 1565-1568, 2007.

[Knowles, 1999] Knowles, J.D., Corne, D.W.: The pareto archived evolutionary strategy: a new baseline algorithm for multiobjective optimization, Proceedings of the IEEE International Conference on Evolutionary Computation, Washington, DC, USA, pp 98-105, 1999.

[Konak, 2006] Konak, A., David, W.C., Smith, A.E.: Multi-objective optimization using genetic algorithms: a tutorial, Reliability Engineering and System Safety, vol 91, no 9, pp 992-1007, 2006.

[Krohling, 2006] Krohling, R.A., Coelho, L. D.S.: Coevolutionary particle swarm optimization using gaussian distribution for solving constrained optimization problems, IEEE Transactions on Systems, Man and Cybernetics Part B: Cybernetics, vol 36, no 6, pp 1407-1416, 2006.

[Larranaga, 2002] Larranaga, P., Lozano, J.A.: Estimation of distribution algorithms: a new tool for evolutionary computation, Boston: Kluwer Academic Publishers, 2002.

[Madavan, 2002] Madavan, N. K.: Multiobjective optimization using a Pareto differential evolution approach, Proceedings of the IEEE International Conference on Evolutionary Computation, Honolulu, HI, USA, pp 1145-1150, 2002.

[Ngatchou, 2005] Ngatchou, P., Zarei, A., El-Sharkawi, M.A.: Pareto multi objective optimization, Proceedings of the 13th International Conference on Intelligent Systems Applications to Power System, Arlington, VA, United States, pp 84-91, 2005.

[Ngom, 2006] Ngom, A., Simovici, D.A., Stojmenovic, I.: Evolutionary strategy for learning multiple-valued logic functions, Journal of Multiple-Valued Logic and Soft Computing, vol 12, no 5-6, pp 459-489, 2006.

[Nobakhti, 2008] Nobakhti, A., Wang, H.: A simple self-adaptive Differential Evolution algorithm with application on the ALSTOM gasifier, Applied Soft Computing Journal, vol 8, no 1, pp 350-370, 2008.

[Pelikan, 1999] Pelikan, M., Goldberg, D.E., Lobo, F.: A survey of optimization by building and using probabilistic models, IlliGAL Report No. 99018, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, Illinois, 1999.

[Priandoko, 2006] Priandoko, B., Ahmad, A.S., Nazaruddin, Y.Y., Sasongko, D.: Modeling and control of a binary distillation unit using genetic algorithm based optimization in nonlinear predictive control, Proceedings of the 6th Asian Control Conference, Bali, Indonesia, pp 871-876, 2006.

[Quinlan, 1986] Quinlan, J.R.: Induction of decision trees, Machine Learning, no 1, pp 81-106, 1986.

[Quinlan, 1993] Quinlan, J.R.: C4.5: Programs for machine learning, Morgan Kaufmann, 1993.

[Robic, 2005] Robic, T., Filipic, B.: DEMO: differential evolution for multiobjective optimization, Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization, Guanajuato, Mexico, pp 520-533, 2005.

[Sagarna, 2005] Sagarna, R., Lozano, J.: On the performance of estimation of distribution algorithms applied to software testing, Applied Artificial Intelligence, vol 19, no 5, pp 457-489, 2005.

[Simionescu, 2006] Simionescu, P.A., Beale, D.G., Dozier, G.V.: Teeth-number synthesis of a multispeed planetary transmission using an estimation of distribution algorithm, Journal of Mechanical Design, vol 128, no 1, pp 108-115, 2006.

[Smith, 2005] Smith, M.G., Bull, L.: Genetic programming with a genetic algorithm for feature construction and selection, Genetic Programming and Evolvable Machines, vol 6, no 3, pp 265-281, 2005.

[Srinivas, 1994] Srinivas, N., Deb, K.: Multi-objective function optimization using non-dominated sorting genetic algorithms, Evolutionary Computation, vol 2, no 3, pp 221-248, 1994.

[Storn, 1997] Storn, R., Price, K.: Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous space, Journal of Global Optimization, vol 11, no 4, pp 341-359, 1997.

[Tan, 2002] Tan, K.C., Lee, T.H., Khor, E.F.: Evolutionary algorithms for multi-objective optimization: performance assessments and comparisons, Artificial Intelligence Review, vol 17, no 4, pp 253-290, 2002.

[Tsutsui, 2001] Tsutsui, S., Pelikan, M., Goldberg, D.E.: Evolutionary algorithm using marginal histogram models in continuous domain, Proceedings of the Genetic and Evolutionary Computation Conference Workshop, San Francisco, CA, pp 230-233, 2001.

[Tsutsui, 2006] Tsutsui, S., Pelikan, M., Ghosh, A.: Edge histogram based sampling with local search for solving permutation problems, The International Journal of Hybrid Intelligent Systems, vol 3, no 1, pp 11-22, 2006.

[Wong, 2005] Wong, K. P., Dong, Z. Y.: Differential evolution, an alternative approach to evolutionary algorithm, Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems, Arlington, VA, United States, pp 73-83, 2005.

[Xue, 2003] Xue, F., Sanderson, A.C., Graves, R.J.: Pareto-based multi-objective differential evolution, Proceedings of the IEEE International Conference on Evolutionary Computation, Canberra, ACT, Australia, pp 862-869, 2003.

[Zitzler, 1999] Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach, IEEE Transactions on Evolutionary Computation, vol 3, no 4, pp 257-271, 1999.

[Zitzler, 2001] Zitzler, E., Thiele, L.: SPEA2: improving the strength pareto evolutionary algorithm for multiobjective optimization, Research Report, May, 2001.