

# A Real-Coded Multi-objective Estimation of Distribution Algorithm

XIAOPING ZHONG, WEIJI LI  
School of Aeronautics  
Northwestern Polytechnical University  
No.127 West Youyi Road, Xi'an  
CHINA

*Abstract:* - A new real-coded multi-objective estimation of distribution algorithm (RCMEDA) for optimization problems with continuous variables is developed. Decision tree is used for discretization to encode conditional dependencies among variables in RCMEDA, i.e. decision-tree-based probabilistic model is used. By building and sampling the probabilistic models, the algorithm reproduces the genetic information of the next generation. Incorporating this reproduction mechanism together with the ranking method and the truncated selection, RCMEDA can approximate the Pareto front. And polynomial mutation operator is used in order to enhance exploration and maintain diversities in the populations. Furthermore, RCMEDA adopts a procedure to eliminate a solution with smallest crowding distance at a time in the truncated selection, so that it can obtain a well spread solution set. The performance of the proposed algorithm is evaluated on four test problems and metrics from literature. Simulation results show that the proposed approach is competitive with NSGA-II and RCMEDA is a general and effective method for multi-objective optimization.

*Key-Words:* - Real-coded multi-objective estimation of distribution algorithm, decision tree, probabilistic model, truncated selection, polynomial mutation operator, multi-objective optimization

## 1 Introduction

The estimation of distribution algorithms (EDAs) [1,2,3] have attracted an increasing attention over recently years. EDAs adopt probabilistic models of underlying problem structure and sampling techniques to reproduce offspring, instead of recombination and mutation operators used in standard evolutionary algorithms. By probabilistic model building and sampling, it is able to avoid the disruption of important building blocks and to solve non-linear or even deceptive problems with a considerable degree of epistasis [4]. Furthermore, the prior information about the problem can be incorporated into these probabilistic models to guide and accelerate the optimization process, but it is not essential.

Many EDAs have been proposed to address discrete and continuous multi-objective optimization problems [4,5,6,7,8]. For continuous variables, the naive MIDEA [8] uses clustering techniques to divide the promising solutions into linear clusters and for each cluster one Gaussian network is used. But clustering does not allow proper mixing of building blocks between clusters [9]. Očenášek presented AMBOA in which decision trees are used to discretize continuous variables, i.e. decision-tree-based probabilistic model is used to encode

conditional dependencies among continuous variables for single objective optimization [10]. And the lack for mixing of building blocks is overcome. AMBOA performs comparably to an effective optimization algorithm CMA-ES [12]. Therefore we extend AMBOA and suggest a real-coded multi-objective estimation of distribution algorithm.

The rest of the paper is organized as follows. In section 2, we discuss Bayesian optimization algorithm with decision tree. In Section3, the proposed real-coded estimation of distribution algorithm is presented in detail. Section 4 demonstrates the applicability of this approach on four test problems and compares the results to NSGA-II [13]. Finally, the paper is concluded with the remarks in section 5.

## 2 Bayesian Optimization Algorithm

The Bayesian optimization algorithm (BOA) uses Bayesian networks (BNs) to model promising solutions, and then samples new solutions from them. Bayesian network is a general probabilistic model for discrete variables. It is able to encode any dependencies among a finite set of variables. BN is often denoted as  $B = (S, \theta)$ , where  $S$  is the structure part and  $\theta$  the parameters for  $S$ . The

structure  $S$  is a directed acyclic graph (DAG) that represents a set of conditional dependencies. Each node in  $S$  corresponds to a variable. A variable is denoted as upper letter, such as  $X_1$ . And its value is denoted as lower upper, such as  $x_1$ . In  $B = (S, \theta)$ ,  $\theta$  is a finite set of parameters for the local conditional probabilities with  $S$ .

**2.1 BOA with Decision Tree Models**

BN is suitable for discrete variables. For continuous domains, decision tree (DT) can be used for discretization to encode the conditional dependencies. For each variable  $X_i$ , a decision tree is built. In DT for  $X_i$ , the split nodes cut the domain of the parent variables  $Pa(X_i)$  into parts, where the variable  $X_i$  and other variables seem to be mutually independent. And on each leaf,  $X_i$  follows Gaussian kernel distributions. Thus, this decomposition is captured globally by the Bayesian network model with decision trees and the Gaussian kernel distributions are used locally to approximate the values in each leaf node. The following is an illustrative of DT for a continuous variable  $X_1$ , where  $X_2, X_5, X_6$  are parents of  $X_1$  and  $f(X_1)$  follows Gaussian kernel distribution.

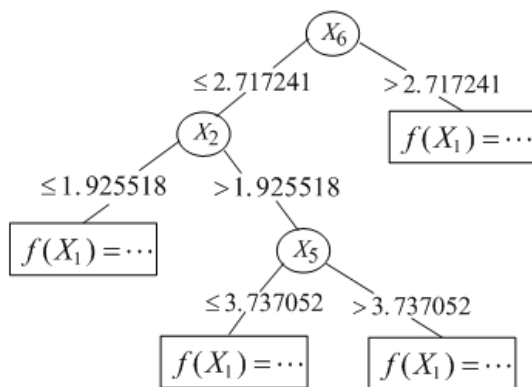


Fig.1 An example of DT for continuous variable

**2.2 DT Metric**

A scoring metric is a measure of how well a probabilistic model models the promising solutions. To measure the quality of probabilistic model with decision trees, a Bayesian scoring metric, calling DT metric, is derived from the well-known Bayesian-Dirichlet metric by Očenášek [11]. It is

$$Gain(X_i, X_j) = \frac{\left[ \Gamma\left(\sum_{x_j} \sum_{x_i} (m(x_i, x_j) + 1)\right) \right] \cdot \left[ \prod_{x_j} \prod_{x_i} \Gamma(m(x_i, x_j) + 1) \right]}{\left[ \prod_{x_j} \Gamma\left(\sum_{x_i} (m(x_i, x_j) + 1)\right) \right] \cdot \left[ \prod_{x_i} \Gamma\left(\sum_{x_j} (m(x_i, x_j) + 1)\right) \right]} \tag{1}$$

Where,  $X_j$  is a splitting node of  $X_i$ ,  $m(x_i, x_j)$  denotes the number of solutions in each splitting leaf and  $\Gamma$  stands for the Gamma function. With this scoring metric, a DT grows by splitting its leaves. A leaf is not split further if the logarithm of its metric is not greater than zero.

As a result of model building, a set of decision trees is obtained. The conditional dependencies between variables are acyclic, so there is a topological order of these variables. According the topological order, probabilistic logic sampling [11] can be used to generate the values for the variables.

**3 Real-Coded Multi-objective Estimation of Distribution Algorithm**

The above-motioned decision tree is used for discretization and the decision-tree-based probabilistic model is used to encode conditional dependencies among variables in RCMEDA. The solutions selected to build such models are based on the fitness that is a combination of the rank and crowding distance of each solution. There are two ways for the combination. One is to add the crowding distance to the rank the other is a hierarchical way. In fact, the rank and the crowding distance are two different traits for a solution, so the addition will blur the concept and leads to inaccurate result. As for the hierarchical way, the comparison is based on rank first. If a tie occurs, it will be broken by considering the crowding distance. The one with greater crowding distance wins. Here, the hierarchical way is used.

Along with non-dominated sorting and crowding distance estimation, truncated selection is performed based on the fitness. After the proper number of solutions is chosen, those solutions are used to construct a probabilistic model in which a decision tree is built for each variable. And a Gaussian kernel model is used to capture the local distribution of each tree leaf. Offspring are sampled from these models by probabilistic logic sampling. In addition, polynomial mutation is performed on the new solutions because of the exploratory capability it could give to the algorithm. In order to overcome the deficiency in adapting the variance of the search distribution, a variance adaptation mechanism is used in RCMEDA. For multi-objective optimization problems with constraints, constrained-dominance [13] is adopted as

constraint handling approach in the process of ranking.

### 3.1 Non-dominated sorting and truncated selection

The fast non-dominated sorting approach, crowding distance estimation and truncated selection are incorporated in RCMEDA to pick out a set of best solutions. We assign rank 1 to solutions in the first front, rank 2 to those in the second front, and so on. The lower front a solution belongs to, the better it is. After identifying all the non-dominated solutions (those in the first front) in the population, they are copied to a mating pool. And crowding distance of each solution in the mating pool is estimated in objective space. Then the truncated selection is done based on crowding distances of solutions.

In case of non-dominated solutions exceeds the population size, some solutions with the smaller values of crowding distance are removed. In this context, only the most crowded solution is removed. This process continues until the number of left solutions is equal to the population size. Note that the crowding distance of the adjoining solutions should be re-estimated each time a solution is left out. For simplicity it can be re-estimated for all the remained solutions of this rank in the mating pool in the implementation.

If the number of non-dominated solutions is less than the population size, it is necessary to pick up all the dominated solutions in the second front. And they are copied to the mating pool. If it is not enough yet, those in the following front (the third front) are identified and added to the mating pool, too. This procedure is repeated until the number of solutions in mating pool is not less than the population size. Then the above elimination procedure is performed on the dominated solutions on the highest front in the mating pool until the appropriate number of solutions is left.

### 3.2 Variance adaptation

If the kernel width of Gaussian distributions decreases fast, offspring will be very similar to its parent, which leads to hard generation of better solutions. To prevent variance from premature shrinking, an overall scaling factor,  $\eta$ , is used to control the kernel width of the marginal distributions adaptively. We find that the variance adaptation formula for single objective optimization in [10] is not good for multi-objective optimization problems. Based on experiments with multi-objective

optimization problems, the modifications of  $\eta$  becomes

$$\eta' = \eta^{(t)} \alpha^{0.4N_{succ}} \alpha^{0.6N_{fail}D/(p-1)} \quad (2)$$

$$\eta^{(t+1)} = \begin{cases} \eta' e^{0.7pt/t_{max}} & \text{if } t < 0.5t_{max} \\ \eta' e^{0.3pt/t_{max}} & \text{otherwise} \end{cases} \quad (3)$$

where  $N_{succ}$  denotes the number of new solutions being selected into the next generation,  $N_{fail}$  represents the number left,  $t$  is the current generation,  $t_{max}$  is the maximum generation,  $p = 0.05 + 0.3/\sqrt{m}$ ,  $m$  is the number of design variables,  $\alpha = \exp(4/N_{sam})$  and  $N_{sam}$  is the size of new solutions. Then each kernel width  $\sigma$  is set to its product with  $\eta^{(t+1)}$ .

### 3.3 Structure of RCMEDA

The flow of RCMEDA is described as follows:

- (1) Set population size  $N_{pop}$ , the promising solutions size  $N_{par}$ , the offspring size  $N_{sam}$ , mutation probability  $P_m$ , the distribution index for mutation  $\eta_m$ , the maximum iteration number  $t_{max}$ , and generate the initial population  $P(t)$  randomly, set  $t=0$ ;
- (2) Select  $N_{par}$  promising solutions from  $P(t)$  and form a set  $S(t)$ ;
- (3) Build a decision-tree-based probabilistic model  $B$  using  $S(t)$  by DT metric;
- (4) Sample  $N_{sam}$  new solutions from  $B$ , and the new solutions form  $O(t)$ ;
- (5) Perform polynomial mutation on solutions in  $O(t)$  with probability  $P_m$ ;
- (6) Create a new population  $P(t+1)$  by selecting  $N_{pop}$  solutions from the merge of  $P(t)$  and  $O(t)$  using ranking method and truncated selection described in section 3.1. And Modify the variance according to section 3.2, let  $t=t+1$ ;
- (7) If the termination criteria are not met, go to (2). Note that if  $S(t)$  and  $P(t)$  have the same size, then step (2) is skipped and  $S(t)$  is the same as  $P(t)$ . The algorithm was implemented in C++ and can be requested from the authors via email.

## 4 Numerical experiments and results

The performance of RCMEDA is compared to NSGA-II (real-coded) on four test problems using the convergence metric  $\gamma$  and diversity metric  $\Delta$  [13]. And 500 uniformly spaced solutions are chosen from the true Pareto front. The two algorithms are run for a maximum of 250 generations with a population size 100. In NSGA-II, the crossover probability is 0.9 and the mutation probability is  $1/n$  (where  $n$  is the number of design variables). The distribution index for

crossover operators is 20 and mutation operators 20. For RCMEDA, 100 promising solutions are selected as parents used to build a probabilistic model and 100 solutions are sampled on each generation. The distribution index for mutation operators is the same as it is in NSGA-II, 20. With these parameters setting, the two algorithms have the same function evaluations after 250 generations.

**4.1 Test problems**

Four benchmark problems ZDT4, ZDT6, FON and OSY from [13] are used to test the performance of RCMEDA. These problems are

ZDT4:  
 $f_1(\mathbf{x}) = x_1$   
 $f_2(\mathbf{x}) = g(\mathbf{x})[1 - \sqrt{x_1 / g(\mathbf{x})}]$   
 $g(\mathbf{x}) = 1 + 10(n - 1) + \sum_{i=2}^n (x_i^2 - 10 \cos(4\pi x_i))$   
 $x_1 \in [0, 1], x_i \in [-5, 5], i = 2, \dots, 10$

ZDT6:  
 $f_1(\mathbf{x}) = 1 - \exp(-4x_1) \sin^6(6\pi x_1)$   
 $f_2(\mathbf{x}) = g(\mathbf{x})[1 - (f_1(\mathbf{x}) / g(\mathbf{x}))^2]$

$g(\mathbf{x}) = 1 + 9 \left[ \left( \sum_{i=2}^n x_i \right) / (n - 1) \right]^{0.25}$   
 $x_i \in [0, 1], i = 1, \dots, 10$

FON:  
 $f_1(\mathbf{x}) = 1 - \exp(-\sum_{i=1}^5 (x_i - 1 / \sqrt{5})^2)$   
 $f_2(\mathbf{x}) = 1 - \exp(-\sum_{i=1}^5 (x_i + 1 / \sqrt{5})^2)$   
 $x_i \in [-4, 4], i = 1, \dots, 5$

OSY:  
 $f_1(\mathbf{x}) = -25(x_1 - 2)^2 - (x_2 - 2)^2 - (x_3 - 1)^2 - (x_4 - 4)^2 - (x_5 - 1)^2$

$f_2(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2$   
 $g_1(\mathbf{x}) = 2 - x_1 - x_2 \leq 0$   
 $g_2(\mathbf{x}) = x_1 + x_2 - 6 \leq 0$   
 $g_3(\mathbf{x}) = x_2 - x_1 - 2 \leq 0$   
 $g_4(\mathbf{x}) = x_1 - 3x_2 - 2 \leq 0$   
 $g_5(\mathbf{x}) = (x_3 - 3)^2 + x_4 - 4 \leq 0$   
 $g_6(\mathbf{x}) = 4 - (x_5 - 3)^2 - x_6 \leq 0$   
 $x_1 \in [0, 10], x_2 \in [0, 10], x_3 \in [1, 5], x_4 \in [0, 6],$   
 $x_5 \in [1, 5], x_6 \in [0, 10]$

**4.2 Simulation results and discussion**

The results are from 30 independent runs of the two

algorithms. Each experiment starts from a randomly generated population. Table 1 shows the mean and variance of the convergence metric  $\gamma$  obtained by RCMEDA and NSGA-II on the four test problems. For brevity,  $\gamma(R)$  is used to denote this metric of the non-dominated sets obtained by RCMEDA and  $\gamma(N)$  denotes that of NSGA-II. And  $\Delta(R)$  stands for the diversity metric of non-dominated sets obtained by RCMEDA,  $\Delta(N)$  denotes that of NSGA-II.

Table 1 Mean and variance of  $\gamma$

Convergence metric $\gamma$		ZDT4	ZDT6
$\gamma(R)$	Mean	0.002013	0.019145
	Variance	0.000004	0.000069
$\gamma(N)$	Mean	0.004711	0.008966
	Variance	0.000006	0.000001
		FON	OSY
$\gamma(R)$	Mean	0.002108	0.454643
	Variance	0	0.032137
$\gamma(N)$	Mean	0.003780	0.344955
	Variance	0	0.016811

Table 2 shows the mean and variance of the diversity metric  $\Delta$  obtained by RCMEDA and NSGA-II on the four test problems

Table 2 Mean and variance of  $\Delta$

Diversity metric $\Delta$		ZDT4	ZDT6
$\Delta(R)$	Mean	0.103513	0.191394
	Variance	0.013463	0.012254
$\Delta(N)$	Mean	0.013809	0.022902
	Variance	0.000061	0.000015
		FON	OSY
$\Delta(R)$	Mean	0.021759	0.058320
	Variance	0.000046	0.004323
$\Delta(N)$	Mean	0.007672	0.090751
	Variance	0.000025	0.003517

From table 1 and table 2, we can see that RCMEDA performs better on all ZDT4 and FON while NSGA-II shows better on ZDT6 and OSY in terms of convergence metric  $\gamma$ . With regard to diversity metric  $\Delta$ , RCMEDA gets better spread of non-dominated solutions only on OSY.

We perform additional experiments by increasing the number of maximum generation to 500 with other parameters fixed. Table 3 and table 4 show the convergence and diversity metric respectively.

Table 3 Mean and variance of  $\gamma$

Convergence metric $\gamma$		ZDT4	ZDT6
$\gamma(R)$	Mean	0.001126	0.003974
	Variance	0	0
$\gamma(N)$	Mean	0.001658	0.004200
	Variance	0	0
		FON	OSY
$\gamma(R)$	Mean	0.002175	0.414480
	Variance	0	0.029617

$\gamma(N)$	Mean	0.003735	0.282066
	Variance	0	0.019050

Table 4 Mean and variance of  $\Delta$

Diversity metric $\Delta$		ZDT4	ZDT6
$\Delta(R)$	Mean	0.001881	0.002076
	Variance	0.000002	0.000001
$\Delta(N)$	Mean	0.005072	0.008363
	Variance	0.000006	0.000013
		FON	OSY
$\Delta(R)$	Mean	0.015502	0.042904
	Variance	0.000014	0.002455
$\Delta(N)$	Mean	0.005014	0.070678
	Variance	0.000017	0.002632

It can be seen that RCMEDA converges better in ZDT4, ZDT6 and FON after 500 generations. And the variance in 30 runs is very small (less than 1e-6) except in OSY. RCMEDA gets better spread of non-dominated solutions on ZDT4, ZDT6 and OSY. The results reveal that RCMEDA converges a little slower than NSGA-II. But it can approximate the true Pareto-optimal front very closely after some generations, say 500.

The non-dominated solutions obtained in a certain run on ZDT4, ZDT6 and OSY are shown in figures 2, 3 and 4 respectively. There are axial translations for clarity in these three figures. It can be seen that the four obtained sets are very close to the known Pareto front. And the two obtained by RCMEDA are more evenly scattered.

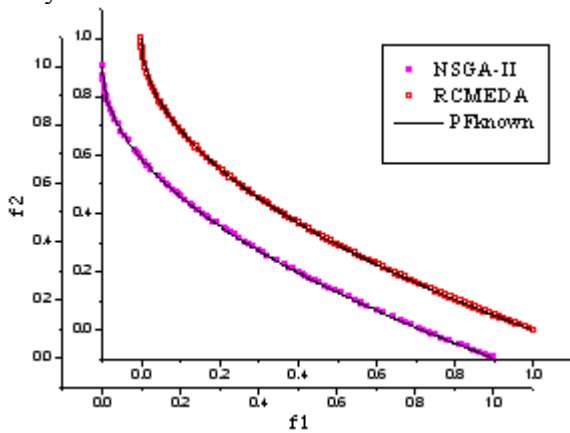


Fig.2 Non-dominated solutions on ZDT4

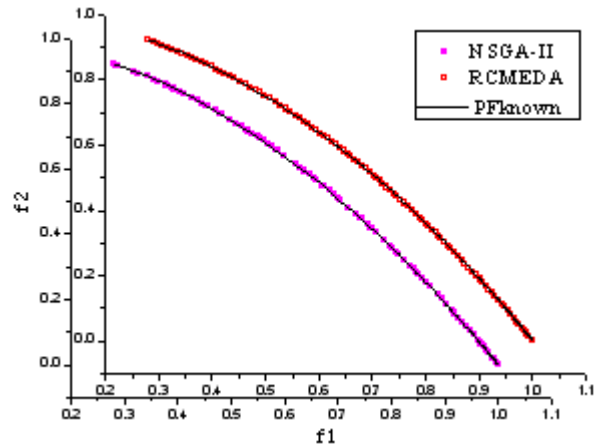


Fig.3 Non-dominated solutions on ZDT6

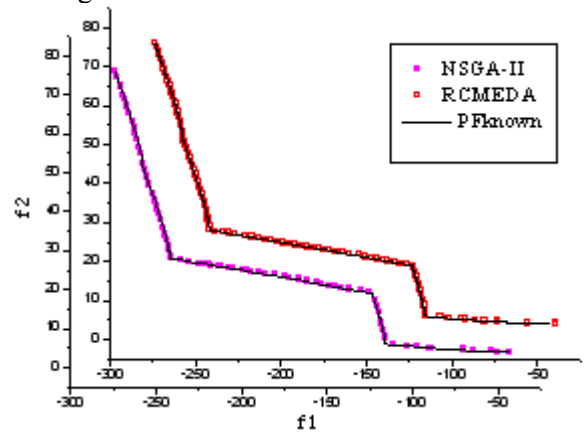


Fig.4 Non-dominated solutions on OSY

RCMEDA needs more CPU time than NSGA-II because of the decision-tree-based probabilistic model building and sampling (for the generation of offspring) on each generation which requests much more time than the simulated binary crossover in NSGA-II. For a single run on Pentium-4-2.66GHz PC with 512M memory, the time cost is given in the following table for ZDT4 and OSY

Table 5 Time cost of NSGA-II and RCMEDA

Algorithm	ZDT4		OSY	
	250	500	250	500
NSGA-II	1.1	2.0	1.0	1.8
RCMEDA	13.0	24.8	8.5	16.4

In the above table, 250 and 500 are the generations, and the other digits are the time cost in seconds. Although RCMEDA is more time consuming, this shortcoming can be negligible in engineering applications where the CPU time is mainly consumed by function evaluations.

### 5 Summary and conclusions

Here we have presented a real-coded multi-objective estimation of distribution algorithm for optimization problems with continuous variables. The algorithm uses decision tree for discretization and decision-

tree-based probabilistic model to encode conditional dependencies among variables. And new solutions are sampled from the probabilistic model on each generation.

The proposed algorithm is applied to four test problems. The simulation results reveal that RCMEDA converges a little slower and needs more CPU time than NSGA-II. But it can approximate the true Pareto-optimal front very closely and get a uniformly spread of non-dominated solution set. And these results show that the RCMEDA is competitive with NSGA-II. It is an effective and robust multi-objective optimization algorithm.

*References:*

[1] M. Pelikan, D. E. Goldberg, and Erick Cantú-Paz, BOA: The Bayesian Optimization Algorithm, Illinois Genetic Algorithms Laboratory, Tech. Rep. 99003, 1999.

[2] P. Larrañaga, J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Optimization*, Kluwer Academic Publishers, Boston, 2001.

[3] Q. Zhang, J. Sun, E. Tsang and J. Ford, Hybrid Estimation of Distribution Algorithm for Global Optimisation, *Engineering Computations*, Vol.21, No.1, 2004, pp. 91-107.

[4] M. Laumanns, and J. Očenášek, Bayesian Optimization Algorithms for Multi-objective Optimization, Lecture Notes in Computer Science: Parallel Problem Solving from Nature - PPSN VII, Granada, Spain, 2002, pp. 298-307.

[5] J. Schwarz, and J. Očenášek, Pareto Bayesian Optimization Algorithm for the Multiobjective 0/1 Knapsack Problem, Proceedings of the 7th International Mendel Conference on Soft Computing, Brno University of Technology, Brno, 2001, pp. 131-136.

[6] N. Khan, D. E. Goldberg, and M. Pelikan, Multi-objective Bayesian Optimization Algorithm, Illinois Genetic Algorithms Laboratory, Tech. Rep. 2002009, 2002.

[7] Q. Zhang, A. Zhou and Y. Jin, RM-MEDA: A Regularity Model Based Multiobjective Estimation of Distribution Algorithm, *IEEE Transaction on Evolutionary Computation*, in press, 2007.

[8] D. Thierens, and P. A. N. Bosman, Multi-objective Mixture-based Iterated Density Estimation Evolutionary Algorithms, Proceedings of the GECCO-2001 Genetic and Evolutionary Computation Conference, San Francisco, California, 2001, pp. 663-670.

[9] J. Očenášek, and J. Schwarz, Estimation Distribution Algorithm for Mixed Continuous-discrete Optimization Problems, 2nd Euro-International symposium on computational intelligence, Kosice, Slovakia, 2002, pp. 227-232.

[10] J. Očenášek, S. Kern, N. Hansen, and P. Koumoutsakos, A Mixed Bayesian Optimization Algorithm with Variance Adaptation, Parallel Problem Solving from Nature - PPSN VIII, Berlin, 2004, pp. 352-361.

[11] J. Očenášek, Parallel Estimation of Distribution Algorithms, PhD dissertation, Brno University of Technology, 2002.

[12] N. Hansen, S. D. Muller, and P. Koumoutsakos, Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES), *Evolutionary Computation*, Vol. 11 No. 1, 2003, pp.1-18.

[13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, A Fast and Elistist Multi-objective Genetic Algorithm: NSGA-II, *IEEE Transaction on Evolutionary Computation*, Vol. 6, No.2, 2002, pp. 181-197.