PARETO BAYESIAN OPTIMIZATION ALGORITHM FOR THE MULTIOBJECTIVE 0/1 KNAPSACK PROBLEM

Josef Schwarz
Jiří Očenášek

Brno University of Technology
Faculty of Engineering and Computer Science
Department of Computer Science and Engineering
CZ - 61266 Brno, Bozetechova 2

Tel.: 420 5 41141210, fax: 41141270, e-mail: schwarz@dcse.fee.vutbr.cz,
Tel.: 420 5 41141283, fax: 41141270, e-mail: ocenasek@dcse.fee.vutbr.cz

Abstract: *This paper deals with the utilizing of the Bayesian optimization algorithm (BOA) for the Pareto bi-criteria optimization of the 0/1 knapsack problem. The main attention is focused on the incorporation of the Pareto optimality concept into classical structure of the BOA algorithm. We have modified the standard algorithm BOA for one criterion optimization utilizing the known niching techniques to find the Pareto optimal set. The experiments are focused mainly on the bi-criteria optimization because of the visualization simplicity but it can be extended to multiobjective optimization, too.*

Key Words: *Knapsack problem, multiobjective optimization, Pareto set, evolutionary algorithms, Bayesian optimization algorithm, niching techniques.*

1 Introduction

Practical problems are often characterized by several non-commensurable and often competing objectives. While in the case of single-objective optimization the optimal solution is simply distinguishable, this is not true for multiobjective optimization. The standard approach to solve this difficulty lies in finding all possible trade-offs among the multiple, competing objectives. These solutions are optimal, nondominated, in that there are no other solutions superior in all objectives. These so called Pareto optimal solutions lie on the Pareto optimal front. There are many papers that present various approaches to find Pareto optimal front almost based on the classical evolutionary algorithms. The Niched Pareto Genetic Algorithm (NPGA) combines tournament selection and the concept of Pareto dominance [1]. A wide review of basic approaches and the specification of original Pareto evolutionary algorithms include the dissertations [2], [3] where the last one describes the original Strength Pareto Evolutionary Algorithm (SPEA). An interesting approach using nondominated sorting in genetic algorithm (NSGA) is published in [4].

The classical genetic algorithms (GA) have the common disadvantage - the necessity of the setting the parameters like crossover, mutation and selection rate and the choice of suitable type of genetic operators. That is why we have analyzed and used one of the Estimation of Distribution Algorithms (EDAs) called probabilistic model-building genetic algorithms, too. The crossover and mutation operators used in standard GA are replaced by probability estimation and sampling techniques. We have focused on the Bayesian optimization algorithm published in the basic issue in [5], [6]. Recently we have published our experience with this algorithm in [7], [8] where single criterion and bi-criteria optimization of hypergraph bisectioning was described. In this paper we have focused on the bi-objective optimization of knapsack problem which belongs to the well known test benchmarks.

2 Multiobjective optimization

A general multiobjective optimization/maximization problem (MOP) can be described as a vector function $\boldsymbol{f}$ that maps a tuple of n parameters to a tuple of m objectives [3]:

$$\begin{aligned} &\max && \boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})) \\ &\text{subject to} && \boldsymbol{g}(\boldsymbol{x}) = (g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), \ldots, g_k(\boldsymbol{x})) <= \boldsymbol{0} \\ &\text{subject to} && \boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \boldsymbol{X} \\ & && \boldsymbol{y} = (y_1, y_2, \ldots, y_m) \in \boldsymbol{Y}, \end{aligned} \tag{1}$$

where $\boldsymbol{x}$ is called decision vector, $\boldsymbol{X}$ is the parameter space, $\boldsymbol{y}$ is the objective vector, $\boldsymbol{Y}$ is the objective space and the constraint vector $\boldsymbol{g}(\boldsymbol{x}) <= \boldsymbol{0}$ determines the set of feasible solutions/set $\boldsymbol{X}_f$.

The set of solutions of MOP includes all decision vectors for which the corresponding objective vectors can not be improved in any dimension without degradation in another - these vectors are called Pareto optimal set. The idea of Pareto optimality is based on the Pareto dominance.

For any two decision vectors ***a, b*** it holds

$\boldsymbol{a} \succ \boldsymbol{b}$ (***a*** *dominates* ***b***) *iff* $f(\boldsymbol{a})>f(\boldsymbol{b})$,
$\boldsymbol{a} \succ= \boldsymbol{b}$ (***a*** *weakly dominates* ***b***) *iff* $f(\boldsymbol{a})>=f(\boldsymbol{b})$,
$\boldsymbol{a} \sim \boldsymbol{b}$ (***a*** *is indifferent to* ***b***) *iff* ***a, b*** *are not comparable*

A decision vector ***a*** dominates decision vector ***b*** $(\boldsymbol{a} \succ \boldsymbol{b})$ *iff* $f_i(\boldsymbol{a}) \geq f_i(\boldsymbol{b})$ *for* $i=1,2,..,m$ with $f_i(\boldsymbol{a}) > f_i(\boldsymbol{b})$ for at least one ***i.*** The vector ***a*** is called Pareto optimal if there is no vector ***b*** which dominates vector ***a*** in parameter space ***X.***

In objective space the set of nondominated solutions lies on a surface known as Pareto optimal front. The goal of the optimization is to find a representative sampling of solutions along the Pareto optimal front. The way how to do it lies in keeping the diversity using some of the niching techniques. Standard BOA is able to find mostly one optimal solution at the end of the optimization process, when the whole population is saturated by phenotype-identical individuals.

3 Problem specification

Generally, the 0/1 knapsack problem consists of set of items, weight and profit associated with each item, and an upper bound of the capacity of the knapsack. The task is to find a subset of items which maximizes the sum of the profits in the subset, yet all selected items fit into the knapsack so as the total weight does not exceed the given capacity. This single objective problem can be extended to multiobjective problem by allowing more than one knapsack. Formally, the multiobjective 0/1 knapsack problem is defined in the following way: Given a set of n items and a set of m knapsacks, with

$p_{i,j}$ profit of item j according to knapsack i

$w_{i,j}$ weight of item j according to knapsack i

c_i capacity of knapsack i

find a vector $\boldsymbol{x} = (\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_n}) \in \{0,1\}^n$, such that $x_j = 1$ iff item j is selected and

$$\boldsymbol{f(x)}=(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_m(\boldsymbol{x})) \text{ is maximum, where} \tag{2}$$

$$f_i(\boldsymbol{x}) = \sum_{j=1}^{n} p_{i,j} * x_j \tag{3}$$

and for which the constraint is fulfilled

$$\forall i \in \{1,2\ldots,m\}: \sum_{j=1}^{n} w_{i,j} * x_j \leq c_i \tag{4}$$

The complexity of the problem solved depends on the values of knapsack capacity. According to [3] we used the knapsack capacities stated by the equation:

$$c_i = 0.5 \sum_{j=1}^{n} w_{i,j} \tag{5}$$

The encoding of solution into chromosome is realized by binary string of the length n. To satisfy the constraints (4) it is necessary to use repair mechanism on the generated offspring to be feasible one. The repair algorithm removes items from the solution step by step until the capacity constraints are fulfilled. The order in which the items are deleted is determined by the maximum profit/weight ratio per item; the maximum profit/weight ratio q_j is given by the equation

$$q_j = \max_{j=1}^{m} \frac{p_{i,j}}{w_{i,j}} \tag{6}$$

The items are considered in increasing order of the q_j, i.e., item with the lowest profit per weight unit is removed first. This mechanism respects the capacity constraints while decreasing the overall profit as little as possible.

4 Pareto optimal BOA

In our Pareto BOA algorithm we replaced the fitness assignment and replacement step of standard BOA by the diversity-preserving niching method based on the promising Pareto technique utilizing a new strength criterion for the evaluation process [3]. The following specification describes the whole reproduction process of our algorithm. Let us note that although we solved bi-objective optimization, our algorithm is able to solve m-objective optimization problems.

Our Pareto BOA algorithm can be described by the following steps:

Step 1: ***Initialization:*** Generate an initial population P_0 of size N randomly.

Step 2: ***Fitness assignment:*** Evaluate the initial population.

Step 3: ***Selection:*** Select the parent population as the best part of current population by 50% truncation selection.

Step 4: ***Model construction:*** Estimate the distribution of the selected parents using Bayesian network construction.

Step 5: ***Offspring generation:*** Generate new offspring (according to the distribution associated to the Bayesian network).

Step 6: ***Nondominated set detection and fitness assignment:*** Current population and offspring are joined, nondominated solutions are found, evaluated and stored at the top of the new population. Then dominated offspring and parents are evaluated separately.

Step 7: ***Replacement*:** The new population is completed by offspring and the best part of current population, so the worst individuals from current population are canceled to keep the size of the population constant.

Step 8: ***Termination:*** If predefined number of generations N_g is reached or stopping criterion is satisfied then the last Pareto front is saved, else go to Step 3.

The most important part of our Pareto algorithm described above is the procedure (step 6) for detection of nondominated (current Pareto front) and dominated solutions and sophisticated fitness calculation. The procedure for current nondominated and dominated set detection is described in following steps:

1. For each individual X in the population P compute the vector of the objective functions
$$\overline{f}(X) = (f_1(X), f_2(X), \dots, f_m(X)) \tag{7}$$
2. Detect subset of nondominated solutions
$$\overline{P} = \{X_j \mid X_j \in P \wedge \nexists X_i \in P : X_i \succ X_j\} \tag{8}$$
3. For each nondominated solution X_j from $\overline{P}$ compute its strength value as
$$s(X_j) = \frac{\left|\{X_i \mid X_i \in P \wedge X_j \succ X_i\}\right|}{|P| + 1} \tag{9}$$
The fitness for nondominated solutions is equal to the reverse of the strength value
$$f'(X_j) = 1/s(X_j) \tag{10}$$
4. For each dominated solution X_i from P determine the fitness as
$$f'(X_i) = 1\Big/\left(1 + \sum_{X_j} s(X_j)\right), \tag{11}$$

where $X_j \in \overline{P} \wedge X_j \succ X_i$. In the original approach [3] all individuals dominated by the same nondominated individuals have equal fitness. We proposed an extension by adding a term $c.r(X_i)/(|P|+1)$ into the denominator (11), where $r(X_i)$ is the number of individuals from P (not only from nondominated solutions) which dominate X_i and coefficient c is set to very small number, for example 0.0001. This term is used to distinguish the importance of individuals in the same "niche" (niche is an area in the objective space dominated by the same part of Pareto front).

This type of fitness evaluation has the following advantages:

- For all nondominated individuals $f'(X_i) \geq 1$, for dominated individuals holds $f'(X_i) < 1$. If we use the replace-worst strategy, implicit Pareto elitism is included.
- Individuals from Pareto front dominated smaller set of individuals receive higher fitness, so the evolution is guided towards the less-explored search space.
- Individuals having more neighbours in their „niche" are more penalized due to the higher $s(X_j)$ value of associated nondominated solution.
- Individuals dominated by smaller number of nondominated individuals are more preferred.

5 Experimental results

5.1 Test benchmarks

In order to be able to compare the performance of our algorithm and other evolutionary algorithms with known results we used two knapsack benchmarks specified by 100 (Kn100) and 250 items (Kn250) published on the web site *[http://www.tik.ee.ethz.ch/~zitzler/testdata.html#fileformat]*. Let us note that the uncorrelated profits $p_{i,j}$ and weights $w_{i,j}$ were chosen, where $p_{i,j}$ and $w_{i,j}$ are random integers in the interval <10,100>. We have compared our results with presented results obtained by two evolutionary algorithms SPEA [3] and NSGA[4]. These two algorithms represent the well working evolutionary multiobjective algorithms.

5.2 Experiments and results

All experiments were run on Sun Enterprise 450 machine (4 CPUs, 4 GB RAM), in future we consider the utilizing the cluster of Sun Ultra 5 workstations.

In Fig. 1 the history of evolution process for the benchmark Kn100 is depicted. The 1st, 25th, 50th and 100th generation of one run of Pareto optimal BOA is shown. In experiment we set the population size N=4000, but only 500 of the randomly chosen individuals/solutions are shown in the graph. The X-coordinate of each point equals to the function value $f_1(x)$ and the Y-coordinate equals to the function $f_2(x)$. This experiment shows the dynamism of the evolution process and the creation of Pareto front.

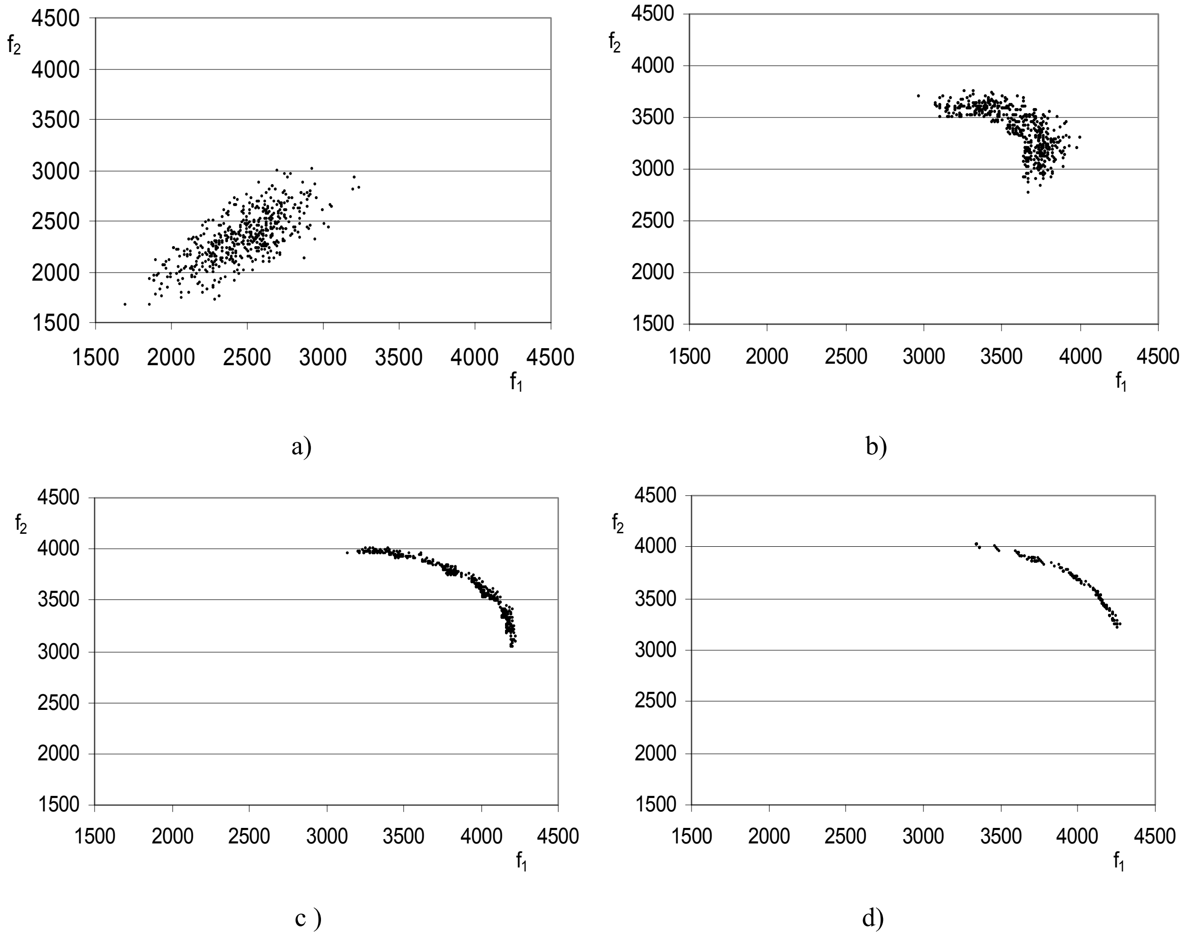

Fig.1. Distribution of the solutions for Kn100 benchmark plotted in the a) 1st generation, b) 25th generation, c) 50 th generation, d) 100 th generation.

In Fig.2a there is the comparison of the final Pareto front produced by our Pareto BOA algorithm and by the SPEA [3] and NSGA [4] algorithms for the case of Kn100 benchmark and in Fig. 2b for the case of Kn250 benchmark. We performed 5 independent runs and constructed the final Pareto front from the 5 particular Pareto fronts. From Fig. 2a it is evident that for Kn100 the Pareto solutions produced by our Pareto BOA in the middle part of Pareto front are slightly better than the Pareto solutions produced by SPEA and NSGA. What is more important – our Pareto BOA produces more solutions in the Pareto front margins. In Fig. 2b we see that for Kn250 the difference between Pareto fronts is more expressive – our Pareto BOA outperforms the SPEA and NSGA. We used the following setting for our algorithm: for the Kn100 we set the population size *N* to 2000, for the case of the Kn250 the population size N equals to 5000. The number of generations used is about 300. The computation time is about 15 minutes for the Kn100. In case of the Kn250 the time was 2 hours for N=5000 and about 40 minutes for N=2000. To reduce the computation time we consider the implementation of parallel version of the Pareto BOA algorithm. It would be also possible to shorten the computational time by decreasing the size of population, but we wanted to keep those predefined values used in our previous experiments (e.g. in the case of hypergraph partitioning [7],[8]).

In the context of algorithm comparison an important question arises: What measure should be used to express the quality of the results so that the various evolutionary algorithms can be compared in a meaningful way. In [3] two measures are described. One of them denoted as *S* represents the size of the objective space covered, the second measure denoted *C* represents the coverage of two sets according to their dominance. We preferred in our comparison the topology/shape of the Pareto fronts.

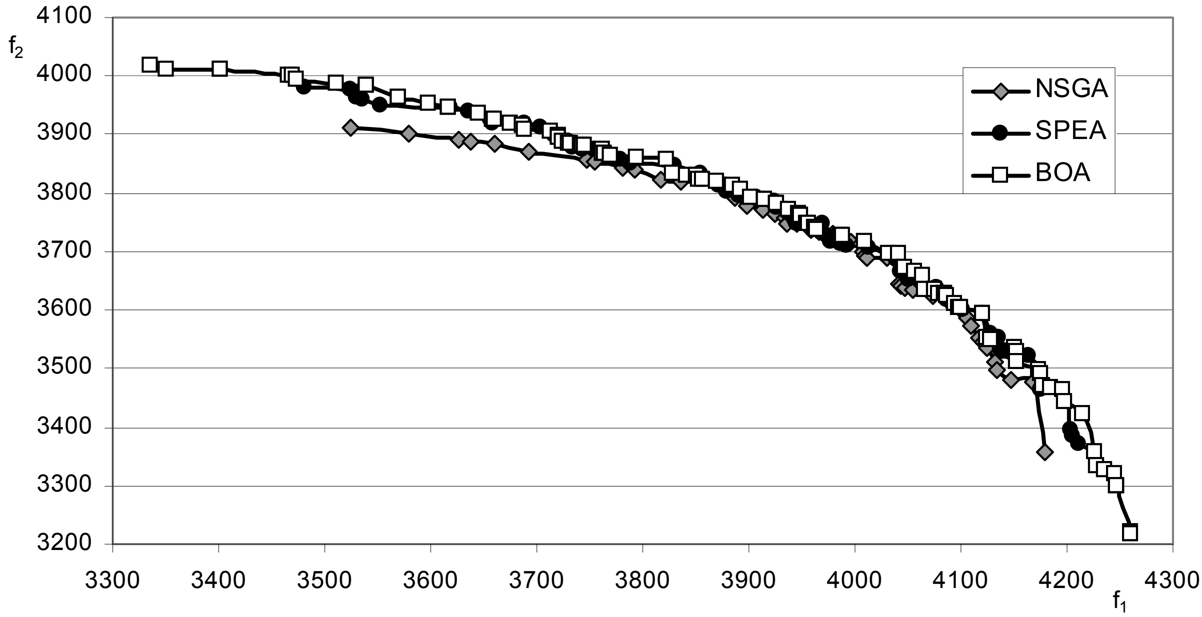

Fig. 2a Final Pareto fronts for Kn100, population size N=2000

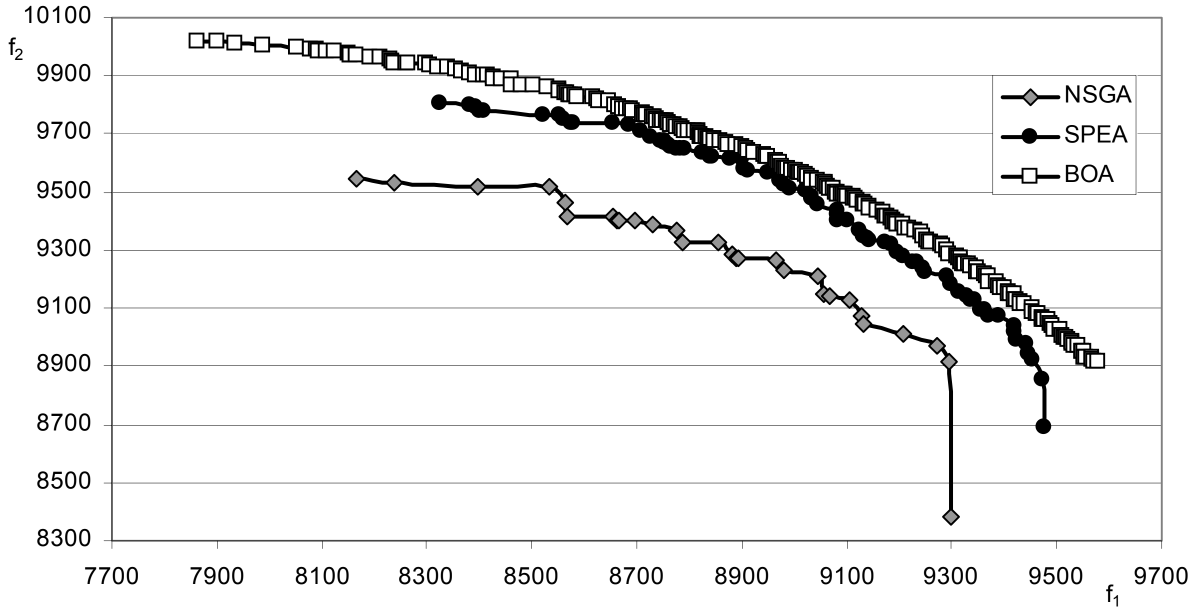

Fig. 2b Final Pareto fronts for Kn250, population size N=5000

6 Parallel Pareto BOA

In [9], [10] we proposed the Distributed Bayesian Optimization Algorithm. It uses a cluster of workstations as a computing platform to speedup the evolution process. Let's note that in the distributed environment the whole population P is split into several parts, each part P_k is being generated and evaluated by different processor. This approach can be extended to the Pareto BOA. We propose the following modification of the procedure for Pareto detection and fitness assignment:

First, each processor will compute the vector of objective functions for all individuals from its part P_k of population P. Then, each processor detects its local set of nondominated solutions $\overline{P}_k$ as

$$\overline{P}_k = \left\{X_j \mid X_j \in P_k \wedge \overline{\exists} X_i \in P_k : X_i \succ X_j\right\} \tag{12}$$

and the master processor creates the global nondominated set $\overline{P}$ from the union of local nondominated sets $\overline{P}_k$ as

$$\overline{P} = \left\{X_j \mid X_j \in \bigcup_k \overline{P}_k \wedge \overline{\exists} X_i \in \bigcup_k \overline{P}_k : X_i \succ X_j\right\} \tag{13}$$

The strength values for nondominated solutions from $\overline{P}$ can be obtained as the sum of local strength values computed in parallel by all processors:

$$s(X_j) = \frac{\sum_k \left| \left\{ X_i \mid X_i \in P_k \wedge X_j \succ X_i \right\} \right|}{|P| + 1} \tag{14}$$

After that all nondominated solutions and their strength values are known, so each processor is able to compute the Pareto fitness for all individuals from its part of population according to equations (10) and (11).

7 Conclusions

We have implemented bi-objective Pareto BOA algorithm for two multiple 0/1 knapsack problems Kn100 and Kn250. For this purpose we have modified the evaluation phase of the single BOA algorithm [5], [7], [8] using the concept of a strength criterion published in the SPEA algorithm [3]. Let us note that the SPEA is a modern multiobjective optimization algorithm which outperforms a wide range of classical methods on many problems. That is why we compare the performance of our Pareto BOA algorithm mainly with the performance of the SPEA algorithm. The Pareto solutions produced by our algorithm are uniformly distributed along the Pareto front which is more global than the Pareto fronts obtained by NSGA and SPEA algorithms.

But many problems remain to be solved, namely the greater computational complexity. The next possible improvement lies also in more sophisticated niching technique, modification of replacement phase of the algorithm and the introduction of problem knowledge into optimization process.

To reduce the computational complexity we proposed the idea of the parallelization of Pareto BOA including the decomposition and detection of the Pareto front. This approach is an extension of Distributed Bayesian Optimization Algorithm [9], [10] based on the parallelization of Bayesian network construction. The future work will be focused on the implementation of Parallel Pareto BOA algorithm for the cluster of SUN workstations.

Acknowledgement

This research has been carried out under the financial support of the Czech Ministry of Education – FRVŠ No. 0171/2001 "The parallelization of Bayesian Optimization Algorithm" and the Research intention No. CEZ: J22/98: 262200012 – "Research in information and control systems".

References

[1] Horn, J., Nafpliotis, N., Goldberg, D.E.: *A Niched Pareto Genetic Algorithm for Multiobjective Optimization,* In Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, Vol. 1, pages 82-87, Piscataway, New Jersey, June 1994. IEEE Service Center.

[2] Coello, C. A.: *An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design.* PhD thesis, Department of Computer Science, Tulane University, New Orleans, LA, April 1996.

[3] Zitzler, E.: *Evolutionary Algorithms for Multiobjective Optimization*: Methods and Applications PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, November 1999.

[4] Srinivas, N., Deb, K.: *Multiobjective Optimization using Nondominated Sorting in Genetic Algorithm*. Evolutionary Computation, Vol.2, No. 3, 1994, pp. 221-248.

[5] Pelikan, M.: *A Simple Implementation of Bayesian Optimization Algorithm in C++(Version1.0)*. Illigal Report 99011, February 1999, pp. 1-16.

[6] Pelikan, M., Goldberg, D. E., & Cantú-Paz, E.: *Linkage Problem, Distribution Estimation, and Bayesian Networks.* IlliGal Report No. 98013, November 1998, pp. 1-25.

[7] Schwarz, J., Očenášek, J..: *Experimental Study: Hypergraph Partitioning Based on the Simple and Advanced Genetic Algorithm BMDA and BOA,* Proceedings of the Mendel'99 Conference, Brno University of Technology, Faculty of Mechanical Engineering, Brno, 1999, pp. 124-130, ISBN 80-214-1131-7.

[8] Schwarz, J., Očenášek, J.: *Evolutionary Multiobjective Bayesian Optimization Algorithm: Experimental Study.* Proceedings of the MOSIS'01, 2001, pp. 101-108, ISBN 80-85988-57-7.

[9] Očenášek, J., Schwarz, J.: *The Distributed Bayesian Optimization Algorithm for Combinatorial Optimization*, EUROGEN 2001 - Evolutionary Methods for Design, Optimisation and Control with Applications to Industrial Problems, Athens, Greece, September 19-21st, 2001, accepted paper to be published.

[10] Očenášek, J., Schwarz, J.: *The Parallel Bayesian Optimization Algorithm*, Proceedings of the European Symposium on Computational Intelligence, Physica-Verlag, Košice, Slovak Republic, 2000, pp. 61-67, ISBN 3-7908-1322-2.