Enhancement of Multiobjective Hierarchical Bayesian Optimization Algorithm using Sporadic Model Building

Wiesław Piasecki[*]

Information Security Division, Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

This paper describes and analyzes the efficiency enhancement of Multiobjective hierarchical Bayesian Optimization Algorithm (mohBOA) by using Sporadic Model Building (SMB). Firstly, Multiobjective hierarchical Bayesian Optimization Algorithm is shortly described. Secondly, sporadic model building is presented. Using sporadic model building, the structure of a probabilistic model is updated once every few iterations, whereas in the remaining iterations only model parameters (conditional and marginal probabilities) are updated. Since the time of learning the structure of a model is much longer than the time of updating model parameters, sporadic model building decreases the total time complexity of model building. The results of experiments show that the theoretical predictions about using sporadic model building to the enhancement of mohBOA are true. Finally, short discussion about the results of experiments is added.

1. Introduction

In the real world there are many problems that fall within the scope of *multiobjective* (or *multicriteria*) *optimization problems* (MOPs), consisting of several incommensurable and often conflicting objectives [1]. Without loss of generality, a general MOP can be formulated as follows [2]:

$$\text{minimize } z = f(y) = \left(f_1(y), f_2(y), ..., f_M(y) \right) \tag{1}$$

$$\text{subject to } e(y) = \left(e_1(y), e_2(y), ..., e_J(y) \right) \geq 0, \tag{2}$$

where $y = (y_1, y_2, ..., y_n) \in \Omega$, $z = (z_1, z_2, ..., z_M) \in \Lambda$. The decision vector is represented here by y, Ω denotes the decision space, z is the objective vector, and Λ stands for the objective space. The set of decision vectors **y** that satisfy the constraint $e(y) \geq 0$ is defined to be the feasible set Ω_f, and the image of Ω_f in the objective space is defined to be the feasible region Λ_f.

---

[*]*e-mail address*: w.piasecki@pollub.pl

There are two basic approaches to the solution of multiobjective optimization problems [3]. The first one assumes the analysis of a multiobjective problem as a single problem where the objective consists of the weighed sum of all objects. The second one involves finding the true Pareto front, which is defined as such a set of solutions that the improvement for one objective entails the impairment of these solutions quality for any other objective.

There are many algorithms to solve multiobjective optimization problems. One of them is Multiobjective Hierarchical Bayesian Optimization Algorithm. It combines the hierarchical BOA with the nondominated crowding assignment of NSGA-II as well as with clustering in the objective space. Thus, decomposable multiobjective problems can be solved in low-order polynomial time, whereas other comparable algorithms do not have adequate scalability and become intractable for small problems [3].

The purpose of this paper is to present the results produced by the efficiency enhancement of mohBOA with Sporadic Model Building. The results indicate that Sporadic Model Building leads to the significant speedup of model building that decreases the asymptotic complexity of model building in mohBOA.

The paper starts with the description of multiobjective hierarchical Bayesian Optimization Algorithm. Section 3 describes Sporadic Model Building. Section 4 describes experiments and presents experimental results. Finally, section 5 summarizes and concludes the paper.

2. Multiobjective Hierarchical Bayesian Optimization Algorithm (mohBOA)

MohBOA combines hBOA, Nondominated Sorting Genetic Algorithm II (NSGA II) and clustering in the objective space. Pseudo code of mohBOA is presented in Figure 1.

Firstly, the initial population is generated at random and evaluated. Similarly the hBOA [4,5] and other EDAs each iteration starts by selecting promising solutions. However, instead of standard selection methods, mohBOA first uses the nondominated crowding assignment of NSGA-II to rank candidate solutions and assign them crowding distances. The ranks and distances are used to compare candidates during selection. There are two widely used selection strategies: tournament selection and truncation selection. The first one, connected with restricted tournament replacement, is recommended if we concern the overall performance (the number of function evaluations). But using truncation selection instead of tournament is substantially better for the purpose of having accurate structural linkage [6]. Selection operator, used in NSGA-II and mohBOA, successfully maintains diversity in the Pareto-optimal front [7]. After selection, K-means clustering is used to divide parents' population into the

specific number of clusters. Although K-means clustering permits empty clusters, they are not taken into consideration in the recombination phase. The Bayesian network with local structure is built for each cluster separately and sampled to generate the same number of new candidate solutions in each cluster. This allows even coverage of the entire front.

```
multiobjective-hBOA(N, k, objectives)
t := 0;
generate initial population P(0) of size N;
evaluate P(0);
while (not done) {
    rank members of P(t) using nondominated crowding;
    select S(t) from P(t) based on the ranking;
    cluster S(t) into k clusters;
    build Bayesian network with local structures
    for each cluster;
        create O(t) of size N by sampling the model
    for each cluster to generate N/k solutions;
        evaluate O(t);
    combine O(t) and P(t) to create P(t+1);
    t:=t+1;
}
```

Fig. 1. Pseudocode of the multiobjective hBOA [3]

The population of new candidate solutions is then incorporated into the original population employing two methods [3]:
1) the elitist replacement based on the non-dominated crowding of NSGA-II, and
2) the restricted tournament replacement (RTR) [8] based on the non-dominated crowding.

Restricted tournament replacement is used to ensure useful diversity maintenance [8]. The run is terminated when the termination criteria are met.

3. Sporadic Model Building

Learning the Bayesian network with local structures comprises two steps [9]:
1) learning the structure
2) learning the parameters (conditional probabilities).

Learning the structure of the Bayesian with decision trees can be bounded by $O(kn^2N)$, where k is the order of subproblems, n is the number of decision variables in the problem and N is the population size [4]. However, learning the parameters can be computed in $O(knN)$ time steps [10]. Thus, building the

network structure is the most expensive part of hBOA. The structures in the consequent iterations of hBOA are usually very similar. Sporadic model building (SMB) uses these facts by building the structure of the Bayesian network once in every few iterations, while in the remaining iterations only the parameters are updated for the structure used in the previous iteration. Important observations regarding the use of sporadic model building for solving nearly decomposable and hierarchical problems by hBOA were made by Pelikan, Sastry, and Goldberg [10]. They noticed that 'sporadic model building leads to significant improvements of asymptotic time complexity of hBOA model building and the optimal speedup of model building grows with problem size' while 'the number of evaluations until convergence insignificantly increases' and 'the asymptotic complexity of hBOA with respect to the number of evaluations until convergence remains a low-order polynomial'.

4. Experiments

This section presents and discusses experimental results. First, test problems are described and the experimental methodology is discussed. Next, the results are presented and discussed.

The primary purpose of our experiments is to analyze and compare the speedup and the performance of mohBOA with sporadic building block as well as without it. The other purpose of our research involves finding and determining the relationship between the structure-building interval and the number of clusters used by KMeans. The test problem consists in solving 2-D function.

Problem 1: Onemax-zeromax

The first test problem was proposed by Pelikan [3]. It consists of two fitness functions: *onemax* and *zeromax*. Onemax function is defined as the sum of ones in input binary string X:

$$onemax(X) = \sum_{i=1}^{n} X_i \ . \tag{3}$$

Zeromax function is defined as the sum of zeros in input binary string X:

$$zeromax(X) = \sum_{i=1}^{n} (1 - X_i) \tag{4}$$

Onemax and *zeromax* are conflicting objectives; any modification that increases one objective reduces the other one. The Pareto-optimal front for an n-bit Onemax – zeromax consists of (n+1) solutions with unique values of the two objectives.

Problem 2: dec-3 – invdec-3

The first fitness function is concatenated 3 bit deceptive trap and the second one is inverse concatenated 3 bit deceptive trap. The task is finding all optimal Pareto front.

Deceptive 3-bit trap function (dec-3) was proposed by Deb & Goldberg [11]. The input string is first divided into independent groups of 3 bits each. This division is constant during calculation and it is unknown to the algorithm. For each group of 3 bits, dec-3 function is applied. The fitness value is equal to the sum of all values of dec-3 function. Deceptive 3-bit function is defined as follows [10]:

$$dec(u) = \begin{cases} 1 & if\ u = 3 \\ 0 & if\ u = 2 \\ 0.8 & if\ u = 1 \\ 0.9 & if\ u = 0 \end{cases} \tag{5}$$

where u is the number of ones in the input string of 3 bits. The aim is to maximize the function.

The inverse 3-bit deceptive trap function (invdec-3) is similar to dec-3 function. The difference lies in its definition:

$$invdec(u) = \begin{cases} 0.9 & if\ u = 3 \\ 0.8 & if\ u = 2 \\ 0 & if\ u = 1 \\ 1 & if\ u = 0 \end{cases} \tag{6}$$

The objective is to maximize these two functions. It can be seen that dec-3 and inverse dec-3 are conflicting objectives. For the binary string length n the number of Pareto optimal solution is equal to $2^{n/3}$.
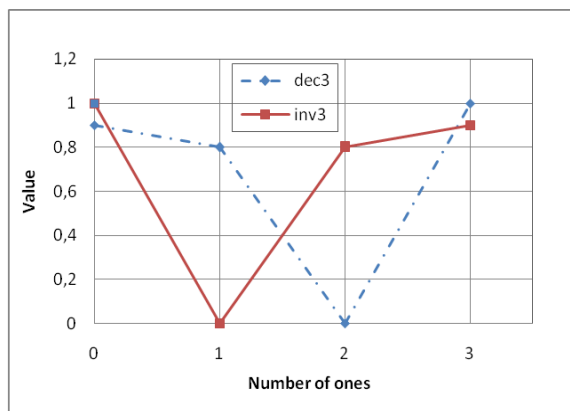

Fig. 2. Dec-3 and inverse dec-3 for one 3-bit block

The Pareto optimal solution is composed of blocks size 3 of all ones or all zeros. Figure 2 presents two Pareto optimal solutions for the binary string of size 3. The first one consists of all ones and the second one is made up of all zeros.

Binary tournament selection was used to select promising solutions. As a replacement strategy, RTR is used where the window size w is set as suggested in [12] as $w = max \{n, N/20\}$ where *n* is the number of variables in the problem and *N* is the population size.

For each problem size and problem type, bisection was used to determine a minimum population size in order to find all solutions consisting of the Pareto-optimal front (solutions with the same values of both objectives are considered equivalent) in 5 out of 5 independent runs.

5. Results

This section presents the experiment results of test problems used to test sporadic model building in mohBOA.

We found that the number of structure-building interval has influence on the overall numbers of mohBOA fitness evaluations regardless of a problem type. The experiment results show that the total fitness evaluation increases with the size of structure-building interval. However, the time expended on calculating additional fitness function evaluations is recompensed by the time saved on building probabilistic model. For example, we get speedup about $n^{0.15}$ for 5c where the model is built once in every 6 generations. We found that the speedup is smaller than in hBOA with sporadic model building due to mohBOA use non-dominated sorting and K-Means algorithms. The computational complexity of non-dominated sorting is higher than the complexity of K-means and is equal to $O(MN^2)$ [13] where M is the number of objectives and N is the population size. If we compare it with the complexity of building probabilistic model $O(kn^2N)$ [7], we find out that the model building takes more time than non-dominated sorting. Assuming that the critical population size is $N_{crit} = O(2^k n^{1.05})$ [4] and substituting N_{crit} for *N,* we obtain $O(M2^{2k}n^{2.1})$ for the non-dominated sorting and $O(2k^2n^{3.05})$ for Bayesian model building, where n is the number of bits, M is the number of objectives and k is the maximum order of the subproblems in the problem decomposition.

In Figure 3 for the problem size 10, we can observe clear distinction in the total number of fitness evaluations for different size of structure-building intervals. These irregularities are caused by the wrong proportion of clusters number used in KMeans and minimal population size.
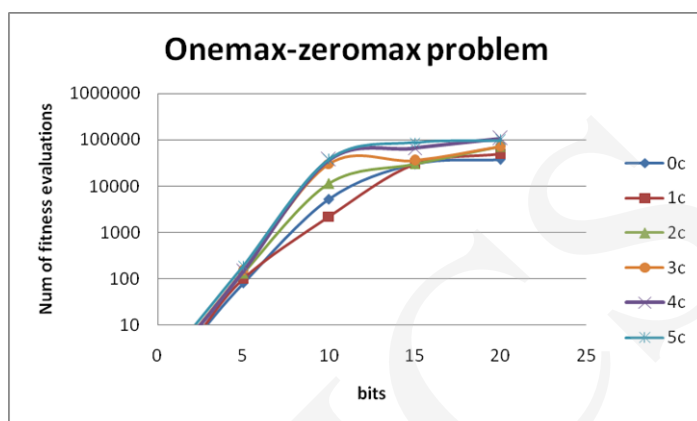
Fig. 3. The influence of sporadic model-building interval on the total number of fitness evaluations
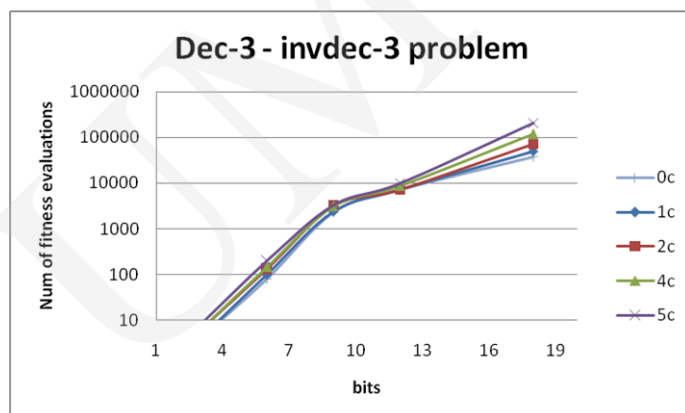

Fig. 4. For the dec-3 – invert dec3 problem the overall number of evaluations increase,
due to problem difficulty

During our research we also found out that the strong dependency exists between the number of clusters and the finding of all Pareto optimal solutions. We have to note that the number of clusters determines the Bayesian networks and their parameters, since each cluster determines subspace of individuals that is used to create the Bayesian network and calculate its parameters. If the population size is too small, then the number of individuals in each cluster is not big enough to find the whole true Pareto front. This happens because the Bayesian network contains insufficient statistical information (linkage information) to create good candidate solutions. For this reason algorithm falls into a trap. If the number of individuals in each cluster is sufficient, then an adequate Bayesian model can be built and good candidate solution can be made.

Experiment results show that the probabilistic structure has not necessarily to be updated in each iteration; only its parameters should be updated in each iteration. This gives algorithm the speedup in building probabilistic model. However, the speedup is not as high as it was in hBOA with sporadic model building.

Conclusions

This paper presented and analyzed an efficiency enhancement technique called sporadic model building and its effect in speedup model building in mohBOA. Sporadic model building gives the capability of updating only the parameters of the structure, while the probabilistic model is not updated.

References:

[1] Zitzler E., Laumanns M., and Thiele L., *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*. In Proceedings of Evolutionary Methods for Design, Optimization, and Control, (2002) 95.

[2] Ahn C.W., *Advances in Evolutionary Algorithms Theory, Design and Practice*. Springer, Berlin, (2006).

[3] Pelikan M., Sastry K., Goldberg D.E., *Multiobjective hBOA, Clustering, and Scalability*, IlliGAL Report, 2005005 (2005).

[4] Pelikan, M., *Hierarchical Bayesian Optimization Algorithm: Toward a New Generation of Evolutionary Algorithms*. Springer-Verlag, (2005).

[5] Pelikan, M., Goldberg, D.E., Cant´u-Paz, E., *BOA: The Bayesian Optimization Algorithm*. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99), I, Also IlliGAL Report, 99003 (1999) 525.

[6] Lima C.F., Pelikan M., Goldberg D.E. Sastry K., Lobo F.G., Hauschild M., *Influence of Selection and Replacement Strategies on Linkage Learning in BOA*. IlliGAL Report, 2007013 (2007).

[7] Khan N., Goldberg D., Pelikan M., *Multi-objective Bayesian Optimization Algorithm*, IlliGAL Report, 2002009 (2002).

[8] Harik, G.R., *Finding Multimodal Solutions Using Restricted Tournament Selection*. Proceedings of the International Conference on Genetic Algorithms (ICGA-95), (1995) 24.

[9] Heckerman, D., Geiger, D., Chickering, D.M., *Learning Bayesian Networks: The Combination of Knowledge and Statistical Data.* Technical Report MSR-TR-94-09, Microsoft Research, Redmond, WA., (1994).

[10] Pelikan M., Sastry K., Goldberg D.E., *Sporadic Model Building for Efficiency Enhancement of hBOA*. IlliGAL Report, 2005026 (2005).

[11] Deb, K., Goldberg, D.E., *Sufficient Conditions for Deceptive and Easy Binary Functions*. Annals of Mathematics and Artificial Intelligence, 10 (1994) 385.

[12] Pelikan, M., Goldberg, D.E., Lobo F., *A Survey of Optimization by Building and Using Probabilistic Models*. Computational Optimization and Applications, 21(1), 5-20. Also IlliGAL Report, 99018 (2002).

[13] Deb K., Pratap A., Sammer A., Meyarivan T., *A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II*, IEEE Trans. on Evolutionary Computation, 6(2) 2002 182.