

# Parallel BMDA with Probability Model Migration

Jiří Jaroš, and Josef Schwarz, *Member, IEEE*

**Abstract**—The paper presents a new concept of parallel bivariate marginal distribution algorithm using the stepping stone based model of communication with the unidirectional ring topology. The traditional migration of individuals is compared with a newly proposed technique of probability model migration. The idea of the new xBMDA algorithms is to modify the learning of classical probability model (applied in the sequential BMDA [1]). In the first strategy, the adaptive learning of the resident probability model is used. The evaluation of pair dependency, using Pearson's chi-square statistics is influenced by the relevant immigrant pair dependency according to the quality of resident and immigrant subpopulation. In the second proposed strategy, the evaluation metric is applied for the diploid mode of the aggregated resident and immigrant subpopulation. Experimental results show that the proposed adaptive BMDA outperforms the traditional concept of individual migration.

## I. INTRODUCTION

THE concept of traditional parallel genetic algorithm (PGA) is well known. The population is divided into a few subpopulations or demes, and each of these demes evolves separately on different processors. Exchange of information among subpopulations is possible via a migration operator. In this context, the term island model is ordinarily used. On each island the population is free to converge toward different optima. The migration operator is supposed to mix good features that emerge locally in the different demes.

Many topologies can be defined to connect the demes, but the most common models are the island model and the stepping stones model. In the basic island model, migration can occur between any subpopulations, whereas in the stepping stone model demes are disposed on a ring and migration is restricted to neighboring demes. In [2] the theory is published providing rational decisions for the proper setting of control parameters. An interesting survey of PGA is published in [3]. An effective technique for the massive parallelization of compact GA was published in [4]. An extremely prestigious PGA which is capable to solve

Manuscript received March 31, 2007. This work was partially supported by the Grant Agency of the Czech Republic under No.~102/07/0850 Design and hardware implementation of a patent-invention machine and the Research Plan No. MSM 0021630528 - Security-Oriented Research in Information Technology.

Jiří Jaroš is with the Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic (e-mail: jarosjir@fit.vutbr.cz).

Josef Schwarz is with the Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic (e-mail: schwarz@fit.vutbr.cz).

billion-variable optimization problems was recently published in [5].

This paper concerns the application of the stepping stone model (for simplicity will use the term island-based model) for bivariate marginal distribution algorithm BMDA. This new approach using probability model migration is conceptually different from the traditional parallel genetic algorithms with migration of individuals/solutions and also from the EDAs using parallel building of pseudo-sequential probabilistic models.

The paper is organized as follows: Section II introduces the basic concept of EDA algorithm and current techniques used in the parallelization of the EDA algorithms. In Section III the sequential BMDA is described including the factorization and graphical representation of probability model. Section IV presents the motivation and a new idea of learning probability model using a concept of probability model migration. Experimental results are shown in Section V, Section VI concludes the paper.

## II. TRADITIONAL EDAS

EDAs belong to the advanced evolutionary algorithms based on the estimation and sampling of graphical probabilistic models [6-13]. They do not suffer from the disruption of building blocks known from the theory of standard genetic algorithms. In Fig. 1 the canonical sequential EDA is described.

```
Set  $t \leftarrow 0$ ;  
Generate initial population  $D(0)$ ;  
While termination criteria is false do  
begin  
  Select a set of promising solution  
   $D^s(t)$ ;  
  Construct a new probability model  $M$   
  from  $D^s(t)$  using chosen metric;  
  Sample offspring  $O(t)$  from  $M$ ;  
  Evaluate  $O(t)$ ;  
  Create  $D(t+1)$  as a subset of  $O(t) \cup$   
   $D(t)$  with cardinality  $N$ ;  
   $t \leftarrow t + 1$ ;  
end
```

Figure 1. The pseudo code of canonical EDA

EDAs often surpass classical EAs in the number of required fitness function evaluations. However, the absolute execution time is still limiting factor which determines the size of practically tractable problems. Referring to Fig. 1 the most time consuming task is the estimation of probability

model. Most papers on EDAs concentrate on parallel construction and sampling of probabilistic models. The well-known algorithm employing parallel construction of Bayesian network is EBNA algorithm designed for MIMD architecture designed both for MPI and POSIX threads, published in [14, 15, 16]. In [17] the theory of population sizing and timing to convergence is published.

A new idea of the multideme parallel estimation of distribution algorithm (PEDAs) based on PBIL algorithm was published in [18]. In [19] mixtures of distribution with Bayesian inference are discussed. Parallel learning of belief networks in large domains is investigated in [20]. Using the concept of PBIL algorithm [21, 22, 23] the classical phenomenon of migration in island based EAs was carried over into probability distribution of EDAs. A new approach of probability vector crossover was implemented with very good performance. Another approach is published in [24] where on UMDA platform the convex combination of univariate probability models is investigated for various network topologies (ring, star etc.). Further enhancement of this concept is described in [25] where the local search methods are used to identify which part of the immigrant model can improve the resident model.

In our paper we propose a new concept of island based BMDA algorithm with ring topology based on the combination of two adjacent bivariate probability models.

### III. SEQUENTIAL BMDA

The well known representative of bivariate EDAs is the Bivariate Marginal Distribution Algorithm (BMDA) proposed by Pelikan and Mühlenbein [1, 23]. This algorithm uses a factorization of the joint probability distribution that exhibits second-order dependencies.

EDAs are also population based algorithm but unlike GAs the new population is generated by sampling the recognized probability model.

Let us denote:

$D = (X^0, X^1, \dots, X^{n-1})$  with  $X \in D$ , is the population of strings /solutions/individuals,

$\mathbf{X} = (X_0, X_1, \dots, X_{n-1})$  is a string/solution of length  $n$  with  $X_i$  as a variable,

$x = (x_0, x_1, \dots, x_{n-1})$  is a string/solution with  $x_i$  as a possible instantiation of variable  $X_i$ ,  $x_i \in \{0,1\}$ ,

$p(\mathbf{X}) = p(X_0, X_1, \dots, X_{n-1})$  denotes the  $n$  dimensional probability distribution,

$p(x_0, x_1, \dots, x_{n-1}) = p(X_0 = x_0, X_1 = x_1, \dots, X_{n-1} = x_{n-1})$  denotes a probability of a concrete  $n$  dimensional vector.

The probabilistic model used in BMDA can be formalized by  $M = (G, \theta)$ , where  $G$  is dependency graph and  $\theta = (\theta_0, \theta_2, \dots, \theta_{n-1})$  is a set of parameters which are estimated by local conditional or marginal probability for each node/variable of the dependency graph.

A greedy algorithm for building dependency graphs is used. At the beginning the root node is selected and

subsequently the nodes with maximum dependency value are searched among the remaining nodes and joined. These pairwise dependencies in BMDA are discovered by Pearson's chi-square statistics:

$$\chi_{i,j}^2 = N \left( \sum_{\forall x_i \in \text{Dom}(X_i)} \sum_{\forall x_j \in \text{Dom}(X_j)} \frac{m^2(x_i, x_j)}{m(x_i)m(x_j)} - 1 \right) \quad (1)$$

where  $N$  is the size of parent population and  $m(x_i, x_j)$ ,  $m(x_i)$  resp.  $m(x_j)$  denote the number of individuals in the parent population with concrete values of  $x_i$  and/or  $x_j$ . These values are stored in the contingency tables. From the theoretical point of view this metric can be seen as statistical testing of hypothesis – for example binary genes  $X_i$  and  $X_j$  are considered to be independent at 95 percent confidence level if  $\chi_{i,j}^2 < 3.84$ . Like COMIT, BMDA also uses a variant of minimum spanning tree technique to learn a model. However, during the tree construction, if none of the remaining variables can be “rooted” to existing tree, BMDA starts to form additional tree from remaining variables. The final probability distribution is thus a forest distribution (a set of mutually independent dependency trees):

$$p(\mathbf{X}) = \prod_{X_r \in R} p(X_r) \prod_{X_i \in V \setminus R} p(X_i | X_{j(i)}) \quad (2)$$

where  $V$  is the set of nodes of dependency tree,  $R$  is the set of root nodes and  $X_{j(i)}$  denotes the parent node of  $X_i$ . Given the tree dependence structure, the univariate marginal probability distributions are estimated from the promising/parent population:

$$p(X_i = 1) = \frac{m(X_i = 1)}{N} \quad (3)$$

and the bivariate conditional probability distributions  $p(X_i | X_{j(i)})$  are estimated as

$$p(x_i | x_{j(i)}) = \frac{m(x_i, x_{j(i)})}{m(x_{j(i)})} \quad (4)$$

For example, the joint probability distribution for the dependency graph in Fig. 2 can be expressed by the factorization:

a)  $p(\mathbf{X}) = p(X_4) p(X_3 | X_4) p(X_2 | X_3) p(X_1 | X_2) p(X_0 | X_1)$

b)  $p(\mathbf{X}) = p(X_2) p(X_3 | X_2) p(X_0 | X_4) p(X_4) p(X_1 | X_4)$

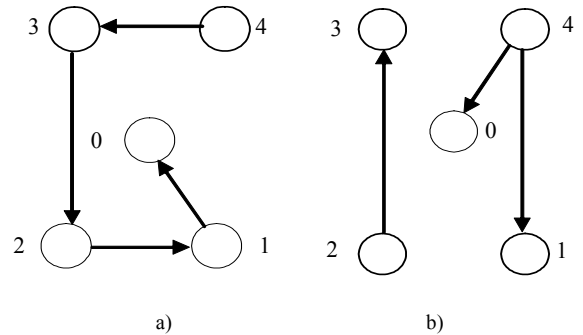


Figure 2. Example of dependency graph for: a) COMIT, b) BMDA

#### IV. ISLAND-BASED BMDA

The principal motivation for the proposal of a new concept of BMDA parallelization is to discover the efficiency of the transfer of probabilistic parameters in comparison with the traditional transfer of individuals. The main goal is to obtain better solution quality. The present approaches recently published use a simpler probability model only (PBIL, UMDA), see [18, 24, 25].

With the concordance of the theoretical conclusion shown in [1] and on the basis of experimental works done in [14], we used the island-based communication model with unidirectional ring topology with synchronization. We simulated this system on single processor computer. It is evident that we can decomposed the migration process in the ring loop into pairwise interactions of two adjacent islands - one of them is the resident island specified by resident probabilistic model and the second one is the immigrant island whose probabilistic model is transferred to a new resident model using a predefined migration rate.

We focused on the problem of combining the immigrant model with the model in the resident island. In general, the modification of resident model by immigrant model can be formalized by [21, 23]:

$$M_R = \beta M_R \circ (1-\beta)M_I, \quad (5)$$

where operator  $\circ$  can be e. g. sum operator and the coefficient  $\beta$  in the range  $\langle 0, 1 \rangle$  specifies the influence of the immigrant model. We have proposed two types of model combination:

- A) mixed learning of dependency graph,
- B) adaptive learning of dependency graph.

##### A. Mixed Learning of Dependency Graphs

In this variant, both subpopulations in resident and immigrant islands are treated as one subpopulation with a diploid structure. During the consecutive building of the resident dependency graph  $G_R$  the new node is added to the graph resulting in addition a new edge between node pairs  $(x_i, x_j)$  according the resulting Pearson's chi-square statistics:

$$\chi_{i,j}^2 = \circ(\chi_{i_R,j_R}^2, \chi_{i_I,j_R}^2, \chi_{i_R,j_I}^2, \chi_{i_I,j_I}^2) \quad (6)$$

It means that the search procedure for a new edge of  $G_R$  examines not only inner node pairs of the resident island and the inner node pairs of the immigrant island but also mixed node pairs, where each node of the pair belongs to different islands. We tested two operators  $\circ$ : a) random choice (*random*), b) maximum value (*max*). More promising solutions were achieved by max operator:

$$\chi_{i,j}^2 = \max(\chi_{i_R,j_R}^2, \chi_{i_I,j_R}^2, \chi_{i_R,j_I}^2, \chi_{i_I,j_I}^2) \quad (7)$$

##### B. Adaptive Learning of Dependency Graphs

This variant uses the adaptive learning of both parts of the probabilistic model – the dependence graph  $G_R$  and the parameter set  $\Theta_R$ . Equation 5 is thus specified by the following equations:

$$\chi_{i,j}^2 = \beta \chi_{i_R,j_R}^2 + (1-\beta) \chi_{i_I,j_I}^2 \quad (8)$$

$$\Theta'_R = \beta \Theta_R + (1-\beta)\Theta_I \quad (9)$$

The learning of the dependency graph by (8) and (9) does not utilize mixed node pairs. In addition, the adaptation coefficient  $\beta$  is used as:

$$\beta = \begin{cases} \frac{F_R}{F_I + F_R} & \text{if } F_I \geq F_R \\ 0.9 & \text{otherwise} \end{cases} \quad (10)$$

where  $F_R$  represents the mean fitness value of the resident subpopulation and  $F_I$  represents the mean fitness value of the immigrant subpopulation. In Fig. 3 the pseudocode of the model combination is shown.

```

Procedure(Output:  $M'_R$ , Input:  $SubPop_I, SubPop_R$ )
Define  $n$  as the problem size;
Calculate  $F_R$  for the resident subpopulation;
Calculate  $F_I$  for the immigrant subpopulation;
Calculate  $\beta$ :
 $\beta = \begin{cases} \frac{F_R}{F_I + F_R} & \text{if } F_I \geq F_R \\ 0.9 & \text{otherwise} \end{cases}$ 
For  $i=0$  to  $n-1$  do begin
  For  $j=0$  to  $n-1$  do begin
    Calculate  $\chi_{i_R,j_R}^2, \chi_{i_I,j_I}^2$ ;
    Store in Chi_Table[ $i,j$ ]:  $\chi_{i,j}^2 = \beta \chi_{i_R,j_R}^2 + (1-\beta) \chi_{i_I,j_I}^2$ 
  end
end
Build the new dependency graphs  $G'_R$  according Chi_Table;
Calculate set of the parameters:  $\Theta_R(G'_R), \Theta_I(G'_R)$ ;
Learning of the parameters:  $\Theta'_R = \beta \Theta_R + (1-\beta)\Theta_I$ 
Compose new resident model:  $M'_R = (G'_R, \Theta'_R)$ 
Sample the adapted model  $M'_R$ ;
Replacement of  $SubPop_R$ ;

```

Figure 3. Adaptive model learning

#### V. EXPERIMENTAL WORKS

In our experiments, we compared six different variants of the BMDA algorithm. The first group consists of three versions using the probability model combinations:

- 1) aBMDA, with adaptive learning of dependency graph.
- 2) mBMDA, with mixed learning of dependency graph with *max* operator.
- 3) rBMDA, with mixed learning of dependency graph with *random* operator.

These three parallel BMDA algorithms works with 8 island subpopulations, each consisting of 256 individuals as a portion of the full population with 2048 individuals.

The second group used for the comparison includes 3 classical variants of BMDA:

- 4) iBMDA, with the migration of individuals.
- 5) sBMDA, sequential BMDA, with full population of 2048 individuals (as the whole eight-island model).
- 6) oBMDA - sequential BMDA with reduced population consisting of 256 individuals (as in case of one island).

Now, we have used fixed subpopulation size for the whole range of problem size. We have not wittingly used the possibility of the adaptation of the subpopulation size according to problem size as discussed in [17]. Our goal was to compare namely the parallel adaptive aBMDA version with traditional iBMDA version under limited resources (subpopulation size). The value of the population size for sBMDA is set to 2048 derived partially from our experience and from the experimental results published in [17] for the 3-Deceptive problem.

In all BMDA variants, truncation-based selection strategy was used, i.e. all individuals were ordered by their fitness value and the better half was used for model building. The truncation-based replacement strategy was also used for the replacement operator, i.e. the new generated solutions (offspring) replace the worse half of the subpopulation. The probabilistic model is built in each generation. Frequency of the model migration or individual migration was even - once per five generations. In case of the individual migration algorithm the elitism is used, that is, 13 best individuals of the immigrant subpopulation (i.e. about of  $k=5$  percent of the subpopulation) replace the worse individuals of resident subpopulation. First stop condition was met after 500 generations; the second condition was activated if there is no improvement in the interval of 50 generations.

#### A. Specification of Benchmarks

For our experimental study four well known benchmarks with various complexity and known global optimum were used. The OneMax and TwoMax problems served as the basic benchmarks for the testing of the basic performance. The Quadratic problem represents the adequate benchmark that should be solvable just by any BMDA algorithm. The 3-Deceptive task belongs to the hard deceptive benchmark for BMDA and is often used for the testing of BOA algorithms.

$$\text{OneMax: } f_{\text{OneMax}}(x) = \sum_{i=0}^{n-1} x_i$$

$$\text{TwoMax: } f_{\text{TwoMax}}(x) = \left| \sum_{i=0}^{n-1} x_i - \frac{n}{2} \right| + \frac{n}{2}$$

$$\text{Quadratic: } f_{\text{Quadratic}}(x) = \sum_{i=0}^{\frac{n}{2}-1} f_2(x_{\pi(2i)}, x_{\pi(2i+1)})$$

$$\text{where } f_2(u, v) = 0.9 - 0.9(u + v) + 1.9uv$$

$$\text{3-Deceptive: } f_{\text{3-Deceptive}}(x) = \sum_{i=0}^{\frac{n}{3}-1} f_3(x_{\pi(3i)} + x_{\pi(3i+1)} + x_{\pi(3i+2)})$$

$$\text{where } f_3(u) = \begin{cases} 0.9 & \text{if } u = 0 \\ 0.8 & \text{if } u = 1 \\ 0 & \text{if } u = 2 \\ 1 & \text{otherwise} \end{cases}$$

The four mentioned objective functions were used to form fitness functions (FF) without additional modification. We have tested 6 variants of BMDA using 30 independent runs. To have baseline to island based versions we first tested the classic sequential BMDA (sBMDA) with ordinary

population of 2048 individuals and the classical sequential BMDA with reduced population (oBMDA).

The first metric is represented by the often used success rate of the global optimum discovery. The second metric is calculated as the average value of the last generation fitness function (FF) over 30 runs. The third metric is computed as the mean value of the last generation number of correctly discovered buildings blocks (BBs) over 30 runs. These metrics/statistics are discussed in the next Sections.

#### B. OneMax Problem

The sBMDA, aBMDA and rBMDA algorithms succeeded in the whole range of the problem size, see Fig. 4. The mBMDA version was able to reliably solve this problem up to 280 variables. Behind this limit its success rate steadily decreases and beyond 420 variables it completely fails. Classical iBMDA version produces comparative result only up to 260 variables. The rapid drop follows after this threshold. oBMDA was very significantly outperformed with all algorithms.

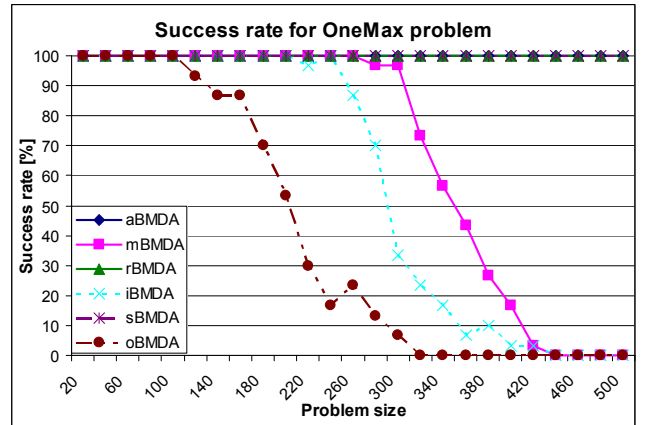


Figure 4. Success rate for OneMax problem

#### C. TwoMax Problem

In case of TwoMax problem, see Fig. 5, the results of the tested algorithms are similar to the results achieved for OneMax problem.

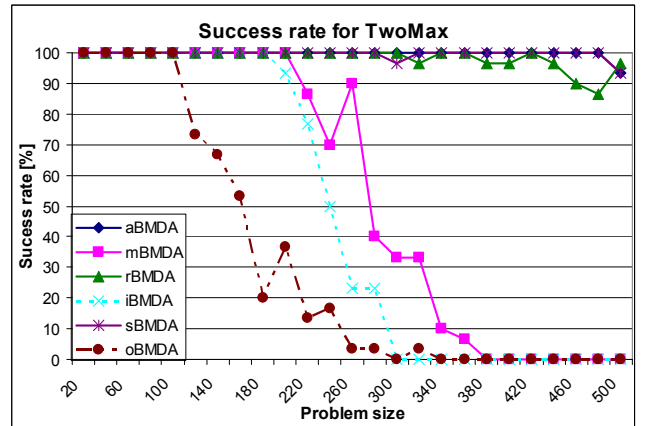


Figure 5. Success rate for TwoMax problem

The aBMDA version outperformed all other versions and achieved the same results as sBMDA. The drop of success rate for the iBMDA version with individual migration is stronger than in the case of OneMax problem.

#### D. Quadratic Problem

To achieve global solution for this problem, the second order statistics is necessary. This benchmark is thus perfectly suitable for testing and comparing all BMDA variants. In Fig. 6, the success rate for all compared algorithms can be seen. The best results were reached by sBMDA that succeeded in the nearly whole range of the problem sizes. The similar behavior can be observed even for aBMDA version that achieved 100 percent success rate up to 260 variables. mBMDA and rBMDA versions are not so efficient as the sBMDA and aBMDA, see the drop of success rate for about 120 variables. The worst results from all parallel versions were produced by classical iBMDA version.

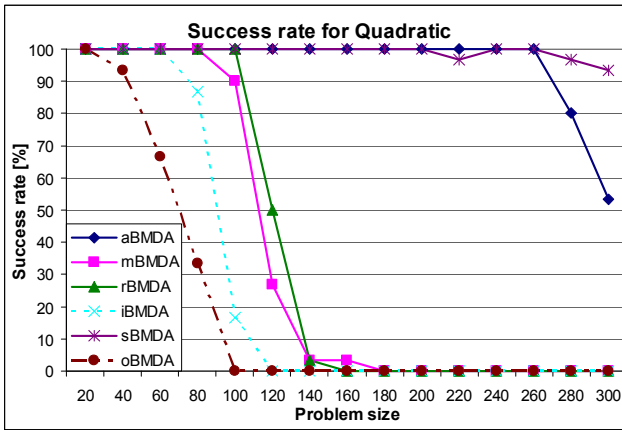


Figure 6. Success rate for Quadratic problem

Besides the success rate metric the mean±std statistics of the fitness function are presented in Table 1. It is evident that aBMDA version and sBMDA version provide the same results up to 260 variables. For higher number of variables sBMDA achieves better results. The highest value achieved for each problem size is written in bold. In Table 2, the mean±std statistics for the number of correctly recognized building blocks (BBs) are shown.

#### E. 3-Deceptive Problem

The problem was investigated for the variable range from 21 to 120, see Fig. 7. For higher number of variables, the drop of success rate is significant for all proposed algorithms. It is caused by rather high complexity of the 3-Deceptive problem that requires a more complex model and also larger population size for efficient performance. The best success rate was gained by aBMDA version. Very similar values were also gained by sBMDA. On the other hand, the worst results were obtained by rBMDA and iBMDA. The quality of the mBMDA version is somewhere in between.

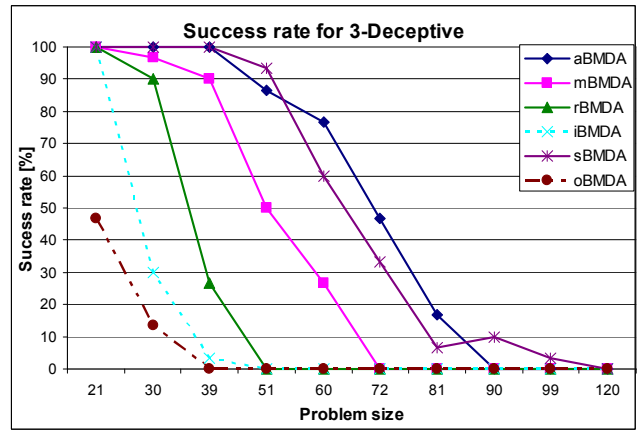


Figure 7. Success rate for 3-Deceptive problem

In Table 3 the mean±std statistics of the fitness function are presented. The best results were obtained by the aBMDA version and by sBMDA. Very good results were also achieved by mBMDA. The worst mean fitness values were gained by oBMDA algorithm followed by iBMDA. In Table 4 the mean value and standard deviation of the discovered BBs are presented. Adaptive aBMDA version proves significant correlation between the mean value of fitness and the mean value of BBs. For the case of the 99-variable problem the mean number of BBs is 26.3 which is 80 percent of total 33 blocks. Note, that iBMDA discovered only 11.9 BBs (36 percent). It is interesting comparison with the experimental results published for BOA algorithm in [17], where the achieved number of building blocks (BBs) for 99 variables and the population size estimated to 250 equals to 25 percent.

#### F. Survey of resulting statistics

The statistics including classic mean±std values of fitness function (FF) was processed only for two harder problems – the Quadratic problem in Tables 1, 3-Deceptive problem in Table 3. The best values are written in bold. From Table 1, it is evident that for Quadratic problem aBMDA and sBMDA have reached the global optima up to 260 variables and outperformed the other variants of algorithms. In the case of 3-Deceptive problem aBMDA outperformed all other algorithms besides the sBMDA that is better for the problem size exceeding 90 variables. Note, that for the 120-variable problem the mean value of FF in case of aBMDA equals to 38.6 which is near to global optimum represented by value 40. In Table 2 and Table 4 the statistics results for BBs are shown for Quadratic and 3-Deceptive problems. From Table 2 it is evident, that in case of Quadratic problem, aBMDA found all BBs up to 260 variables while iBMDA was successful up to 30 variables. In case of 3-Deceptive problems, see Table 4, aBMDA outperformed iBMDA in the whole range of problem size. Note that aBMDA achieved approximately two time higher mean value of BBs for the problem size exceeding 60 variables.

### G. Result Discussion

In the first experiment, two groups of algorithms are compared: A) island-based BMDAs with probabilistic model learning in three modes, B) sequential sBMDA version with the full population size, reduced sequential oBMDA version and the traditional island-based iBMDA with individual migration. Both aBMDA and sBMDA versions are capable to find global optima with 100 percent success rate up to 500 variables in case of OneMax and TwoMax problems and up to 260 variables in case of Quadratic problems. For difficult problems, like 3-Deceptive, the algorithms lack the ability repeatedly to find the optimal solution for problem size larger than 39.

Both mBMDA and rBMDA versions provide worse results than aBMDA but better ones than iBMDA. It is evident that aBMDA is effective optimization tool outperforming iBMDA version based on the traditional individual migration. From these points of view the range of solvable problem size is at least two times larger in case of aBMDA version.

The time complexity of all algorithms measured by the number of generation is comparable. For example, in case of Quadratic problem with 60 variables the average computational time is about 20 generations, see Fig. 8. Note, that oBMDA was able to find the global optima for this instance of Quadratic problem only in 66 percent of the 30 runs.

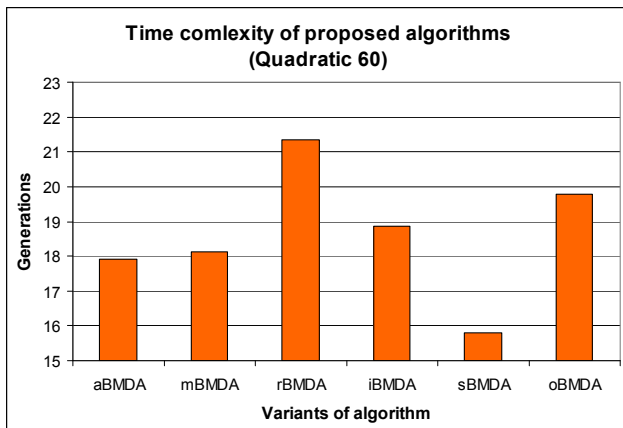


Figure 8. Time complexity for Quadratic problem

In real parallel implementation of aBMDA, it will be necessary to transport all contingency tables from the immigrant node to the resident node. The spatial complexity of these tables is  $4n^2$  where  $n$  is the cardinality of the solved problem. In contrast to the model migration, the individual migration works with the spatial complexity  $nkN$ , where  $kN$  is the number of migrating individuals. Because the communication overhead in modern interconnection networks depends more strongly on start up latency of communication than on a transported message size, we can consider that the communication overhead will be nearly the

same. If we used asynchronous type of migration [26] the communication overhead can be simply overlapped.

### VI. CONCLUSIONS

In the paper we have proposed new island-based xBMDA algorithms with probability model migration. We have tested two basic strategies for combining probability distributions. Both of them work with graphical interpretation of the dependency graphs. The first strategy implemented in mBMDA and rBMDA uses the concept of mixed pairwise dependencies with two distinct evaluation operators. In the second strategy applied in aBMDA the only internal pairs of nodes of resident and immigrant subpopulation are investigated, but in addition the adaptive learning based on the quality of resident and immigrant subpopulation is used. Experimental results confirm our expectation that migration of the probabilistic model with adaptation can significantly obtain better results than the traditional concept of individual migration. The sequential sBMDA with full population size of 2048 individuals provides competitive results compared with the aBMDA but the time complexity of aBMDA version can be significantly reduce on the platform of multiprocessor system.

The future work will be focused on more sophisticated testing of aBMDA algorithm and further modifications of learning techniques with limited size of parameter transfer. We also aim to parallelize the Bayesian Optimization Algorithm (BOA) using a modified concept of probabilistic model migration.

### REFERENCES

- [1] M. Pelikan, and H. Mühlenbein, "The bivariate marginal distribution algorithm," in *Advances in Soft Computing - Engineering Design and Manufacturing*, pp. 521-535, London, 1999. Springer-Verlag.
- [2] E. Cantu-Paz, *Efficient and Accurate Parallel genetic algorithm*. Kluwer Academic Publishers, 2000.
- [3] E. Alba, and J. M. Troya, *A Survey of Parallel Distributed Genetic Algorithms*. Complexity, 4: 303-346, 1999.
- [4] F. G. Lobo, C. F. Lima, and H. Martires, "Massive parallelization of the compact genetic algorithm," in R. Ribeiro et al., editors, *Proceedings of the International Conference on Adaptive and Natural Computing Algorithms (ICANNGA-2005)*, pages 530-533. Springer, 2005.
- [5] D. E. Goldberg, K. Sastry 1, and Xavier Llorà, *Toward routine billion-variable optimization using genetic algorithms*. Complexity 12: 27-29, 2007.
- [6] M. Pelikan, D. E. Goldberg, and E. Cantu-Paz, *Linkage problem, distribution estimation, and Bayesian networks*. IlliGAL Report No. 98013, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, Illinois, 1998.
- [7] P. Larrañaga, and J. A. Lozano, *Estimation of Distribution Algorithms*. Kluwer Academic Publishers, London 2002, ISBN 0-7923-7466-5.
- [8] P. A. N. Bosman, and D. Thierens, *An Algorithmic Framework For Density Estimation Based Evolutionary Algorithms*. Technical report UU-CS-1999-46, Utrecht University, 1999.
- [9] P. A. N. Bosman, and D. Thierens, "Linkage information processing in distribution estimation algorithms", in *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*. Volume 1, pp. 60-67, Orlando, Florida, 1999. Morgan Kaufmann Publishers, San Francisco, CA.

- [10] P. A. N. Bosman, and D. Thierens, "Continuous iterated density estimation evolutionary algorithms within the IDEA framework," in *Proceedings of the Optimization by Building and Using Probabilistic Models OBUPM workshop*. Genetic and Evolutionary Computation Conference GECCO-2000, pp. 197-200, 2000.
- [11] D. Heckerman, D. Geiger, and M. Chickering, *Learning Bayesian networks: The combination of knowledge and statistical data*. Technical Report MSR-TR-94-09, Microsoft Research, Redmond, WA, 1994.
- [12] M. Pelikan, D. E. Goldberg, and F. Lobo, *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.
- [13] M. Pelikan, K. Sastry, and D. E. Goldberg, *Evolutionary Algorithms + Graphical Models = Scalable Black-Box Optimization*. IlliGAL Report No. 2001029, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, Illinois, 2001.
- [14] A. Mendiburu-Alberro, "Parallel implementation of estimation of Distribution Algorithms based on probabilistic graphical models. Application to chemical calibration models," PhD thesis, the University of Bascue Country, Donostia-San Sebastian, January 2006.
- [15] J. Ocenasek, "Parallel Estimation of Distribution Algorithms," PhD. Thesis, Faculty of Information Technology, Brno University of Technology, Brno, Czech Rep., pp. 1-154, 2002.
- [16] J. Ocenasek, J. Schwarz, and M. Pelikan, "Design of Multithreaded Estimation of Distribution Algorithms," in: *Genetic and Evolutionary Computation Conference - GECCO 2003*. Springer Verlag: Berlin, pp. 1247-1258, 2003.
- [17] M. Pelikan, K. Sastry, and E. Cantú-Paz, *Bayesian Optimization Algorithm, Population Sizing and Time to Convergence*. IlliGAL Report No. 2000001, Illinois Genetic Algorithm Laboratory, University of Illinois at Urbana-Champaign, 13 pages, 2000.
- [18] C. W. Ahn, D. E. Goldberg, and R. S. Ramakrishna, Multiple-Deme Parallel Estimation of Distribution Algorithms: Basic Framework and Application. IlliGAL Report No. 2003016, May, 2003.
- [19] J. M. Marin, K. Mengersen, and C. P. Robert, "Bayesian Modelling and Inference on Mixtures of Distribution", in *Handbook of Statistics*, 25, 16, 459-507.
- [20] Y. Xiang, and T. Chu, "Parallel Learning of Belief Networks in Large and Difficult Domains," in *Data Mining and Knowledge Discovery*, vol.3, pp. 315-339, 1999.
- [21] S. Baluja, Population-Based Incremental Learning: A method for Integrating Genetic Search Based Function Optimization and Competitive Learning. Technical report CMU-CS-94-163, Carnegie Mellon University, 1994.
- [22] M. Höhfeld, and G. Rudolph, "Towards a theory of population-based incremental learning", in *Proceedings of the 4th International Conference on Evolutionary Computation*, pp. 1-5, IEEE Press, 1997.
- [23] H. Mühlenbein, "The equation for response to selection and its use for prediction," in *Evolutionary Computation 5(3)*, pp. 303-346, 1997.
- [24] L. delaOssa, J. A. Gámez, and J. M. Puerta, "Migration of probability models instead of individuals: an alternative when applying the island model to edas," in *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242 of LNCS, pages 242–252. Springer, 2004.
- [25] L. delaOssa, J. A. Gámez, and J. M. Puerta, "Improving model combination through local search in parallel univariate EDAs," *Evolutionary Computation*, Vol. 2, pp. 1426-1433.
- [26] S. C. Lin, W. F. Punch, and E. D. Goodman, "Coarse-grain parallel genetic algorithms: categorization and a new approach," in *Sixth IEEE Conference on Parallel and Distributed Processing*, pp 28-37, IEEE Press, Piscataway, NJ, 1994.

TABLE 1. STATISTICS RESULTS (MEAN±STD OF FF) FOR QUADRATIC PROBLEM

Algorithm	Problem size							
	60	80	100	120	140	260	280	300
1. aBMDA	<b>30.0±0.00</b>	<b>40.0±0.00</b>	<b>50.0±0.00</b>	<b>60.0±0.00</b>	<b>70.0±0.00</b>	<b>130.0±0.00</b>	139.9±0.24	149.7±0.46
2. mBMDA	<b>30.0±0.00</b>	<b>40.0±0.00</b>	49.9±0.03	59.9±0.80	69.7±0.10	128.2±0.26	137.9±0.27	147.5±0.44
3. rBMDA	<b>30.0±0.00</b>	<b>40.0±0.00</b>	<b>50.0±0.00</b>	59.9±0.09	69.7±0.13	127.4±0.24	137.2±0.32	146.7±0.37
4. iBMDA	<b>30.0±0.00</b>	39.9±0.30	49.8±0.80	59.7±0.13	67.7±3.14	127.1±0.37	136.5±0.30	146.1±0.27
5. sBMDA	<b>30.0±0.00</b>	<b>40.0±0.00</b>	<b>50.0±0.00</b>	<b>60.0±0.00</b>	<b>70.0±0.00</b>	<b>130.0±0.00</b>	<b>139.9±0.02</b>	<b>149.9±0.02</b>
6. oBMDA	29.9±0.06	39.8±0.10	49.6±0.14	59.3±0.24	69.1±0.25	126.3±0.57	135.8±0.54	145.3±0.64
Global optimum	30	40	50	60	70	130	140	150

TABLE 2. STATISTICS RESULTS (MEAN±STD OF BBS) FOR QUADRATIC PROBLEM

Algorithm	Problem size							
	60	80	100	120	140	260	280	300
1. aBMDA	<b>30.0±0.00</b>	<b>40.0±0.00</b>	<b>50.0±0.00</b>	<b>60.0±0.00</b>	<b>70.0±0.00</b>	<b>130±0.00</b>	129±2.45	147±4.64
2. mBMDA	<b>30.0±0.00</b>	<b>40.0±0.00</b>	49.9±0.34	58.9±0.84	67.0±1.06	112±2.55	119±2.71	124±4.40
3. rBMDA	<b>30.0±0.00</b>	<b>40.0±0.00</b>	50.0±0.00	59.2±0.93	66.7±1.27	104±2.43	112±3.18	117±3.36
4. iBMDA	<b>30.0±0.00</b>	39.8±0.37	48.7±0.78	56.7±1.26	64.5±1.36	101±3.71	105±3.27	111±2.77
5. sBMDA	<b>30.0±0.00</b>	<b>40.0±0.00</b>	<b>50.0±0.00</b>	<b>60.0±0.00</b>	<b>70.0±0.00</b>	<b>130±0.00</b>	<b>140±0.00</b>	<b>149.9±0.25</b>
6. oBMDA	29.6±0.61	38.8±1.02	46.5±1.41	53.8±2.48	60.8±2.38	93.4±5.86	98.0±5.15	104±4.93
Global optimum	30	40	50	60	70	130	140	150

TABLE 3. STATISTICS RESULTS (MEAN±STD OF FF) FOR 3-DECEPTIVE PROBLEM

Algorithm	Problem size									
	21	30	39	51	60	72	81	90	99	120
1. aBMDA	<b>7.00±0.00</b>	<b>10.0±0.00</b>	<b>13.0±0.00</b>	16.9±0.04	<b>19.9±0.06</b>	23.9±0.15	26.8±0.15	29.5±0.20	32.4±0.35	38.6±0.29
2. mBMDA	<b>7.00±0.00</b>	9.99±0.02	12.9±0.03	16.9±0.12	19.8±0.12	23.6±0.17	26.4±0.23	29.0±0.31	31.7±0.26	38.0±0.28
3. rBMDA	<b>7.00±0.00</b>	9.99±0.03	12.8±0.09	16.5±0.13	19.2±0.19	22.7±0.13	25.4±0.1	28.1±0.17	30.7±0.15	37.0±0.23
4. iBMDA	<b>7.00±0.00</b>	9.92±0.06	12.7±0.11	16.35±0.09	19.1±0.13	22.7±0.14	25.4±0.16	28.1±0.15	30.8±0.15	37.1±0.14
5. sBMDA	<b>7.00±0.00</b>	<b>10.0±0.00</b>	<b>13.0±0.00</b>	<b>16.9±0.03</b>	<b>19.9±0.06</b>	<b>23.9±0.09</b>	<b>26.8±0.1</b>	<b>29.7±0.16</b>	<b>32.6±0.18</b>	<b>39.2±0.2</b>
6. oBMDA	6.90±0.05	9.75±0.12	12.1±1.99	16.1±0.16	18.8±0.22	22.4±0.24	25.1±0.23	28.6±0.57	30.5±0.22	38.6±0.59
Global optimum	7	10	13	17	20	24	27	30	33	40

TABLE 4. STATISTICS RESULTS (MEAN±STD OF BBS) FOR 3-DECEPTIVE PROBLEM

Algorithm	Problem size									
	21	30	39	51	60	72	81	90	99	120
1. aBMDA	<b>7.00±0.00</b>	<b>10.0±0.00</b>	<b>13.0±0.00</b>	16.8±0.45	<b>19.7±0.59</b>	<b>23.1±1.02</b>	25.0±1.51	25.8±2.08	26.3±2.99	26.2±3.72
2. mBMDA	<b>7.00±0.00</b>	9.97±0.18	12.9±0.30	16.3±0.74	18.4±1.20	20.3±1.48	21.6±2.24	21.9±2.56	21.2±3.12	21.4±2.72
3. rBMDA	<b>7.00±0.00</b>	9.90±0.30	11.7±0.96	12.3±1.35	12.8±1.93	12.7±1.36	11.8±1.48	11.1±1.45	10.9±1.74	10.8±1.83
4. iBMDA	<b>7.00±0.00</b>	9.23±0.56	10.6±1.11	10.6±0.92	11.3±1.49	11.5±1.56	11.5±1.62	11.3±1.41	11.9±1.83	11.6±1.80
5. sBMDA	<b>7.00±0.00</b>	<b>10.0±0.00</b>	<b>13.0±0.00</b>	<b>16.9±0.39</b>	19.6±0.56	23.0±0.93	<b>25.3±0.98</b>	<b>27.8±1.61</b>	<b>29.4±1.85</b>	<b>32.1±2.09</b>
6. oBMDA	6.47±0.50	7.60±1.11	7.33±1.72	8.27±1.67	7.23±2.39	8.23±2.29	7.92±2.39	8.50±2.26	8.67±2.95	9.03±1.74
Global optimum	7	10	13	17	20	24	27	30	33	40