A Problem Knowledge - Based Evolutionary Algorithm KBOA for Hypergraph Bisectioning

Josef SCHWARZ, Jiří OČENÁŠEK
Brno University of Technology, Faculty of Engineering and Computer Science
Department of Computer Science and Engineering
CZ - 61266 Brno, Božetěchova 2
e-mail: schwarz@dcse.fee.vutbr.cz, ocenasek@dcse.fee.vutbr.cz

Abstract. This paper is an experimental study on an utilization of additional knowledge about the decomposition problem to be solved. We have demonstrated this approach on the hypergraph bisectioning that can serve as a model of system decomposition in common, data base decomposition etc. We have focused on the extension of the Bayesian Optimization Algorithm BOA. The extension of the original BOA algorithm is based on the usage of a prior information about the hypergraph structure. This knowledge is used for both setting initial Bayesian network and the initial population using injection of clusters to improve the convergence of the decomposition process. The behaviour of our version KBOA is tested on the set of benchmarks, such as grid and random geometric graphs as well as real hypergraphs.

1 Introduction

The decomposition of the system such as communication networks, complex data base system and VLSI layout are usual tasks to be solved. These tasks can be often formalized as a hypergraph partitioning into k partitions – it is a well known problem of graph theory. This paper deals with the case of 2-way partitioning called bisectioning. If necessary the k-way partitioning can be realized by a series of the hypegraph bisectioning.

Many heuristics are used to solve this NP-complete problem. We can refer to the well known paper [1], where a good overview of the efficiency of known local search techniques as well as simulated annealing is presented. A very good survey about netlist partitioning techniques was done in [2], where the geometric representation, combinatorial formulations, move-based and clustering approaches are discussed. The interesting hybrid genetic algorithm is described in [3]. The authors have used a simple genetic algorithm (GA) [4], with reordering of graph nodes to receive shorter schemata/building blocks so as to prevent the disruption of schemata mainly of large defining length.

The problem of schemata disruption has been intensively studied in GA community during few last years. As a result a new class of promising approaches based on an estimation of the joint distribution of promising solutions (EDA) was proposed in [5], [6] and recently new ideas in [7], [8].

We focus in this paper on an extension of Bayesian Optimization Algorithm BOA [5], [9], [10] based on the problem knowledge to speed up the convergence of optimization process during hypergraph bisectioning.

The paper is organized as follows. In chapter 2 and 3 the problem formulation is declared and cost function is defined. Chapter 4 presents the original BOA algorithm which allows to understand the ideas of knowledge based approach presented in chapter 5. In chapter 6 benchmark tests are specified and experimental results are discussed. In chapter 7 the conclusions and research activities for the future are summarized.

2 Problem formulation

More formal, the particular bisectioning problem can be defined as follows: Let us assume a hypergraph $H=(V,E)$, with $n = |V|$ nodes and $m = |E|$ edges. We are supposed to find such a bisection $(V1,V2)$ of V into equal sized parts ($\lceil V1 \rceil = \lceil V2 \rceil$, $V= V1 \cup V2$) that minimizes the number of hyperedges which have nodes in different set $V1, V2$. The set of external hyperedges can be labelled as $E_{cut}(V1,V2)$. The primary objective function/cost of the hypergraph bisectioning is the number of external hyperedges, shortly called cut size:

$$c(V_1,V_2) = \lceil E_{cut}(V_1,V_2) \rceil = \lceil \{e \in E \mid e \cap V_1 \neq 0,\ e \cap V_2 \neq 0\} \rceil \qquad (1)$$

This is the case of strongly balanced bisectioning. Next, we define the set $M(v)$ of hyperedges incident to node v and the set $N(v)$ of nodes that are neighbours of node v:

$$M(v) = \{e \in E \mid v \in e\} \qquad (2)$$

Using the previous equation the cost can be expressed as

$$c(V1,V2) = \lceil \cup_{v \in V_1} M(v) \rceil + \lceil \cup_{v \in V_2} M(v) \rceil - m \qquad (3)$$

The immediate/first neighbours of a node v:

$$N(v) = \{u \mid \exists e \ \ v, u \in e, v \neq u\} \qquad (4)$$

The degree of node v is defined as $D(v) = \lceil N(v) \rceil$.

An example of the hypergraph with 8 nodes and 5 hyperedges is shown in Fig.1. The bisection of the hypergraph resulted in $V_1=\{v_1, v_2, v_7, v_4\}$, $V_2 = \{v_5, v_6, v_3, v_8\}$. The external hyperedges $E_{cut}(V_1,V_2) = \{e_3, e_4\}$, cut size $c(V_1,V_2) = 2$.

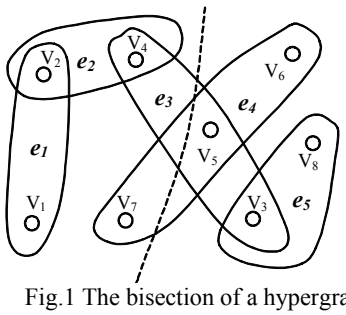


Fig.1 The bisection of a hypergraph

3 Solution encoding

For the BOA algorithm the following ordinary encoding of the solution is used:

Table 1. The ordinary solution encoding for bisection of hypergraph.

Chromosome/String		**Meaning**
gene value	**1 1 0 1 0 0 1 0**	Partition number
locus value	1 2 3 4 5 6 7 8	Hypergraph nodes

Each solution of the bisection is represented by a chromosome, which is a binary string $X=(x_1, x_2, \ldots, x_n)$. The gene value x_i represents the partition number, the index of locus specifies the node in the hypergraph. It is possible to express the cost function using the string encoding. In case of hypergraph bisectioning the cost function can be expressed using the equation (3):

$$c(V1,V2) = \Big/ \bigcup_{\forall x_i=1} M(i) \Big/ + \Big/ \bigcup_{\forall x_i=0} M(i) \Big/ - m, \quad \text{for } i=1,\ldots,n \qquad (5)$$

In case we define a dummy node v_0 with $M(v_0)=\varnothing$, we can transform the last equation into:

$$c(V1,V2) = \Big/ \bigcup M(i\ x_i) \Big/ + \Big/ \bigcup M(i\ (1-x_i)) \Big/ - m, \quad \text{for } i=1,\ldots,n\ , \qquad (6)$$

where $M(i)$ is a set of hyperedges incident of node i.

4 BOA algorithm

In the simple genetic algorithms the standard crossover and mutation operators for offspring generation are used. In the last few years there has been a growing interest in the field of Estimation of Distribution Algorithms (EDAs) also called probabilistic model-building genetic algorithms, where crossover and mutation operators are replaced by probability estimation and sampling techniques. They use statistical information contained in the set of selected parents to detect gene dependencies. The estimated probability model is used to generate new promising solutions according to this distribution. The process can be described as follows:

Generate initial population of size N (randomly);
While *termination criteria is false* **do**
begin
Select parent population of S individuals according to a selection method (S≤N);
Estimate the distribution of the selected parents;
Generate new offspring (according to the estimated model);
Replace some individuals in current population by generated offspring;
end

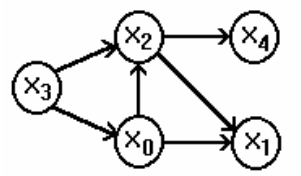
There are various probability distribution models with different complexity.

In BOA (Bayesian Optimization Algorithm) [5] Bayesian network (BN) is used to encode the structure of a problem. Each gene in the chromosome is treated as a variable and represented by a node in the dependency graph. For each variable X_i it is defined a set of variables Π_{X_i} called parents it depends on, so the distribution of individuals is encoded as

$$p(X) = \prod_{i=0}^{n-1} p\left(X_i \mid \Pi_{X_i}\right) \qquad (7)$$

Generally, the existence of oriented edge from X_j to X_i in the network implies that the variable X_j belongs to the set Π_{X_i}. To reduce the space of possible networks, the number of incoming edges into each node is limited to k. According to results in [10] we have limited the parameter k in our experiments to be three.

Example of Bayesian network for 5 variables:



The corresponding joint probability distribution is

$$p(X) = p(X_3) \cdot p(X_0 \mid X_3) \cdot p(X_2 \mid X_0, X_3) \cdot p(X_1 \mid X_0, X_2) \cdot p(X_4 \mid X_2) \quad (8)$$

The Bayesian Dirichlet metric (BD) [11] is used to measure the quality of the network:

$$p(D, B \mid \xi) = p(B \mid \xi) \prod_{i=0}^{n-1} \prod_{\pi_{X_i}} \frac{m'(\pi_{X_i})!}{(m'(\pi_{X_i}) + m(\pi_{X_i}))!} \prod_{x_i} \frac{(m'(x_i, \pi_{X_i}) + m(x_i, \pi_{X_i}))!}{m'(x_i, \pi_{X_i})!} \quad (9)$$

where ξ is the prior information about the problem, D is the population of parents and $p(B \mid \xi)$ is the prior probability of the network B. The product over x_i runs over all instances of the variable X_i and the product over π_{X_i} runs over all instances of the set of its parents Π_{X_i}. The term $m(\pi_{X_i})$ stands for the number of occurrences of the tuple π_{X_i} in the population D and $m(x_i, \pi_{X_i})$ stands for the number of individuals in D having both X_i set to x_i as well as Π_{X_i} set to π_{X_i}. The prior knowledge on the probability distribution is represented by numbers $m'(\pi_{X_i})$ and $m'(x_i, \pi_{X_i})$ representing the expected $m(\pi_{X_i})$ and $m(x_i, \pi_{X_i})$ values. The detailed description of the prior parameters is in the next section.

If no prior information is available, the simpler K2 metrics can be used. It is a special case of BD metrics having all prior counts $m'(x_i, \pi_{X_i})$ set to 1 and all prior counts of parent configurations $m'(\pi_{X_i})$ set to number of possible values on i-th position (equal to 2 for binary chromosome).

Many algorithms can be used to build up the network. The optimal search is NP-hard, thus in the implementation a simple greedy algorithm was used with only one edge addition in each step. The algorithm starts with an empty network B and the edge giving the highest increase of metric value is then added to the network B. This process is repeated until no more addition is possible.

After network is constructed, new instances/solutions are generated. First, the variables/nodes of BN are ordered in the topological order and each iteration, the nodes whose parents are already calculated are generated using the conditional probabilities. This is repeated until all the variables are generated.

5 Knowledge based KBOA algorithm

For many combinatorial problems the information about the structure of the problem is available. If the information is complete, the dependence graph can be constructed in-hand by the expert. This method is used in the FDA (Factorized Distribution Algorithm) [6]. In our case, the structure of the problem is given by the list of edges between hypergraph nodes but the decomposition of the problem in the sense of FDA is not simple because the variables of the cost functions are overlapping. We can only expect that adjacent nodes in the hypergraph are dependent but we have no exact knowledge about the degree of independence between non-adjacent nodes.

The purpose of our approach is to use modified BOA algorithm to learn the structure of the problem together with the partial (local) information about the linkage to improve the performance of that algorithm. Three possible approaches were tested:

- The first one was based on the prior information about the problem in the BD metric. The term $p(B|\xi)$ represents the prior probability of the network B. We use a simple assignment $p(B|\xi) = c\kappa^{\delta}$, where c is a normalization constant, $\kappa \in (0,1]$ is a penalty constant factor and δ is the number of edges in the final Bayesian network having no match in the hypergraph to be bisected. The greedy algorithm that constructs the dependency graph using only one edge addition in each step, prefers edges existing between adjacent nodes of hypergraph (local information about the problem) with the setting δ to be zero. When no such edge exists, a penalization is used by setting δ=1.
- The prior knowledge about the optimized problem is also represented by numbers $m'(\pi_{X_i})$ and $m'(x_i,\pi_{X_i})$. It can be shown, that the values of $m'(\pi_{X_i})$ and $m'(x_i,\pi_{X_i})$ express our belief in the accuracy of the values $m(\pi_{X_i})$ and $m(x_i,\pi_{X_i})$. When these values increase, the BD metric will tend more towards the prior assignment of distribution, when those numbers decrease the influence of sampled values $m(\pi_{X_i})$ and $m(x_i,\pi_{X_i})$ becomes more significant. Our experiments with this type of information were not too promising, so we used the original uniform assignment.
- The standard BOA uses the random set of solutions in the initial population. We have tested the usage of the initial population based upon the knowledge of the hypergraph structure. Initial solutions are affected by injection of clusters of predefined size. The detection of such a cluster in a hypergraph is shown in the Fig. 2:

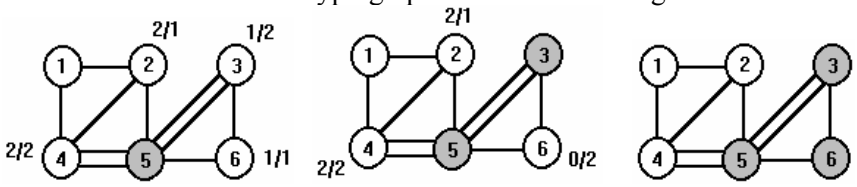


Fig.2. Clustering technique

First, a random node is selected to be the seed of the cluster. Then the effect of addition of each neighbouring node is expressed by the number of external and internal hyperedges incident of this node. Each step the node having minimal ratio external/internal edges is selected. When maximum size of cluster is reached or all neighbours are selected, next cluster is created. The cluster injection serves as a source of low-order building blocks.

6 Experimental results

6.1 Test graphs

The four types of graph structures are used:

1. Regular graphs *Gridn.c* with grid structure [3], [10] where the notion *n* specifies the number of nodes, *c* specifies the minimal cut size. As an example a bisection of graph *Grid100.2* is represented in Fig.3a having a 2-edge bottle-neck in the dashed cut line.
2. Random geometric graphs *Un.d* [1], [3]. Random geometric graph on *n* vertices is placed in the unit square and coordinates of its nodes are chosen uniformly.

There exists an edge between two vertices if their Euclidean distance is l or less, where the expected vertex degree is specified by $d = n\pi l^2$.

3. Caterpillar graphs *CATk_n* [3], with *k* articulations, six legs for each articulation, and *n* nodes, see Fig. 3c with *k=3* and *n=21*. This type of graph serves as a hard benchmark.

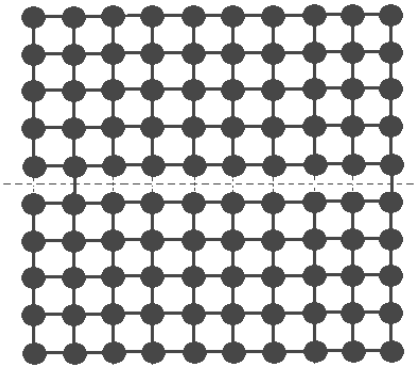


Fig.3a Grid graph

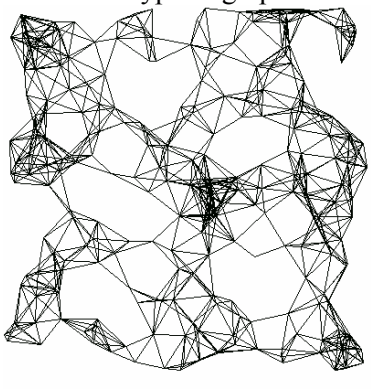


Fig.3b Geometric random graph

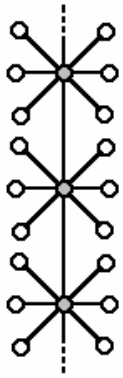


Fig.3c Caterpillar graph

4. Hypergraphs representing real circuits labelled by *ICn* [12]. The global optima is not known. The hypergraphs can be also specified by pair nodes/edges*: IC67/138, IC151/419, IC116/329, Fract149/147.* The structure of these circuits can be characterized as a random logic.

6.2 Results of the hypergraph and graph bisectioning

We have performed various experiments to demonstrate the efficiency of the developed algorithm KBOA comparing to the original BOA algorithm.

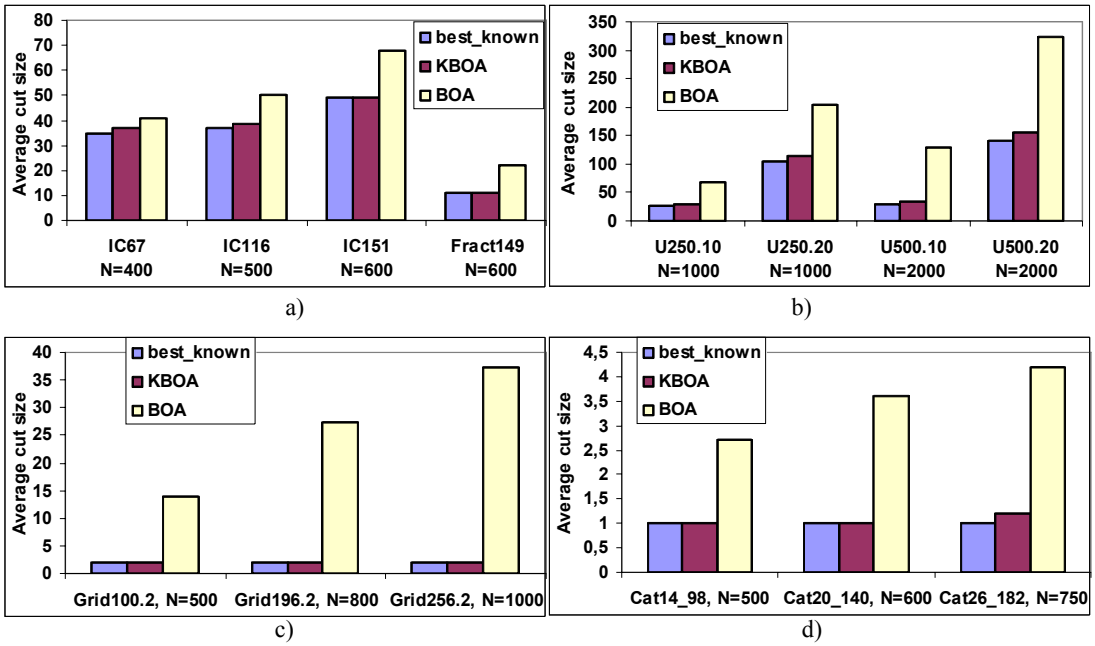


Fig.4 Average and best known cut size for BOA and KBOA a) on real hypergraphs, b) on geometric random graphs c) on grid graphs, d) on caterpillar graphs

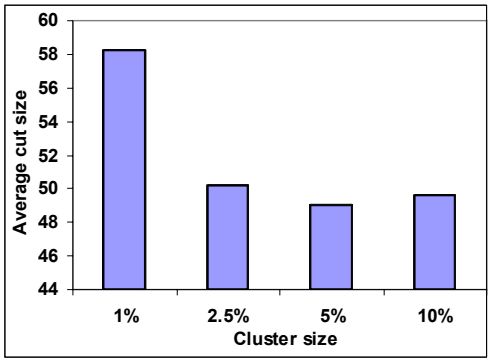


Fig. 5 Cluster size versus average cut size for IC151

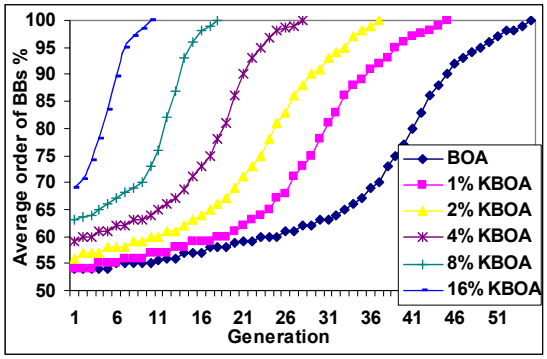


Fig. 6 Evolution of building blocks

In Fig. 4 the comparison of BOA and KBOA is shown for four groups of graphs. The best known solutions are determined by KBOA itself or well known hMetis program[13]. We performed ten independent runs for KBOA and BOA on each of the 14 test graphs of various type and size. The population size is the same for BOA and KBOA and it is set to the value for which the KBOA was successful for all ten runs for the case of grid graphs. Generally, for other types of graphs the population size was increased linearly with the size of the problem (as suggested in [8]). The KBOA outperforms BOA in all tests of hypergraph bisectioning. The time consumption for both algorithms is comparable due to equal population size. Let us note that BOA algorithm is theoretically capable to reach the similar cut size as KBOA but with about five times greater population size (for our test graphs) and thus with a rapid increase of computation time.

In Fig. 5 the performance of KBOA algorithm on the hypergraph IC151 for different size of injected clusters is shown. The size of clusters is expressed as a percentage of the number of hypergraph nodes. The proper value about 5% is used in all our trials. In the case of the smaller cluster injection, the growth of proper building blocks is weak and the opposite case of the injection of too large clusters leads into local optima.

In Fig.6 the growth of average value of building blocks order (ABB) in the population is depicted. This experiment is done for grid graph Grid100.2 with cluster size as a parameter. For the experiment the population size is set to N=300-2400 for KBOA and to N=2400 for BOA to get the global optimum for all sizes of cluster. Because two symmetrical optimal solutions exist, we separate all the chromozomes into two groups according to their similarity to each of the possible solution and the calculation of ABB based on the Hamming distance is summarized. It is clear that for small cluster size and BOA the average order of BBs in the first generation is about 50% which is typical for initial random population. It is evident a strong influence of the cluster size on the speed up of the evolution. Let us note that in the 1% KBOA the cluster injection is not used and it is affected only by the phenomenon of prior probability (penalization) of the Bayesian network (see chapter 5) and only this phenomenon differentiates 1% KBOA from the BOA.

7 Conclusions and future works

In this paper we presented KBOA algorithm as a promising enhancement of the original version of probabilistic model-building genetic algorithm BOA [8], [9], [10]. The both algorithms use Bayesian network to model multivariate data as a mean of estimation of distribution of promising solutions. Although the BOA is powerful tool for solving hard optimization problems inclusive class of deceptive problems there are open questions to be solved e. g. the time complexity for very large problems, the setting of population size and parameter k (the complexity parameter of BN) for current problem, etc.

Our goal was to incorporate a problem knowledge into the whole process of BN construction. Firstly we applied the prior probability of the Bayesian network expressed by the term $p(B \mid \xi) = c\kappa^{\delta}$. The essence of this approach lies in the penalization of edges of BN having no match in the hypergraph. The influence of this phenomenon can be recognized in Fig.6 on the comparison of 1%KBOA and BOA as a slight increase of convergence speed for 1% KBOA. The concept of cluster injection into the initial population detected on the hypergraph structure seems to be a really promising tool for enhancement of the population genotype. This phenomenon leads to the meaningful reduction of cost function and population size. This approach can be used for another optimization problems but with particular problem knowledge.

Future activities will be directed towards an efficient construction of the dependency graphs using the techniques of parallel processing and on a hybridizing of KBOA algorithm using local improvement to enhance genotype not only in the initial population but also during the evolution.

Acknowledgements

This research has been carried out under the financial support of the Research intention No. CEZ: J22/98: 262200012 – "Research in information and control systems" and it was also supported by the Grant Agency of Czech Republic grant No. 102/98/0552 "Research and Application of Heterogeneous Models".

References

[1] Johnson, D. S., Aragon, C., McGeoch, L., & Schevon, C.: Optimization by Simulated Annealing: An experimental Evaluation, Part 1, Graph Partitioning, Operations Research, vol.37, pp. 865-892, 1989.

[2] Alpert, C. J., Kahng, A. B.: Recent Directions in Netlist Partitioning: A Survey, Integration: The VLSI Journal 19 (1995), pp. 1-81.

[3] Bui, T. N., Moon, B. R.: Genetic Algorithm and Graph Partitioning. IEEE Transactions on Computers, Vol.45, No.7, July 1996, pp. 841-855.

[4] Goldberg, D. E.: Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.

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

[6] Muehlenbein, H., Rodriguez, A. O.: Schemata Distributions and Graphical Models in Evolutionary Optimization. GMD Forschungs Zentrum Informationstechnik, 53754-St. Augustin, 1998, pp.1-21.

[7] Pelikan, M., Goldberg, D. E., & Lobo, F.: A Survey of Optimization by Building and Using Probabilistic Model, Illigal Report 99018, September 1999, pp. 1-12.

[8] Pelikan, M., Goldberg, D. E., & Cantú-Paz, E.: Bayesian optimization Algorithm, Population Sizing, and Time to Convergence, Illigal Report 200001, January 2000, pp. 1-13.

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

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

[11] Heckerman, D., Geiger, D., & Chickering, M. (1994). Learning Bayesian Networks: The combination of Knowledge and Statistical Data (Technical Report MSR-TR-94-09, Redmont, WA: Microsoft Research, 1995, pp. 1-53.

[12] A benchmark set, University of California, Los Angeles, VLSI CAD Laboratory, http://vlsicad.cs.ucla.edu/~cheese/benchmarks.html.

[13] Karypis, G., Kumar, V.: Hmetis - A Hypergraph Partitioning Package, version 1.5.3, University of Minnesota, Department of Computer Science&ENGINEERING, Army HPC Research Center Minneapolis, http://www-users.cs.umn.edu/~karypis/metis/hmetis/download.shtml.