

Model Accuracy for Hierarchical Problems

Hossein Karshenas, Amin Nikanjam, B. Hoda Helmi, Adel T. Rahmani

SCOMAS Laboratory, Department of Computer Engineering
Iran University of Science and Technology (IUST)

Tehran, Iran

ho_karshenas@comp.iust.ac.ir, {nikanjam, helmi, rahmani}@iust.ac.ir

Abstract— Estimation of distribution algorithms, especially those using Bayesian network as their probabilistic model, have been able to solve many challenging optimization problems, including the class of hierarchical problems, competently. Since model-building constitute an important part of these algorithms, finding ways to improve the quality of the models built during optimization is very beneficial. This in turn requires mechanisms to evaluate the quality of the models, as each problem has a large space of possible models. The efforts in this field are mainly concentrated on single-level problems, due to complex structure of hierarchical problems which makes them hard to treat. In order to extend model analysis to hierarchical problems, a model evaluation algorithm is proposed in this paper which can be applied to different problems. The results of applying the algorithm to two common hierarchical problems are also mentioned and described.

Keywords— model accuracy; template matrix; evaluation metrics; hierarchical problems; EDAs

I. INTRODUCTION

As a successful Evolutionary algorithm, Genetic algorithms with their simple mechanism have been able to solve many difficult real world problems. But there are certain types of problems that pose great challenge to these algorithms. Additively separable problems [7 and 11] are one class of these problems that deceive the algorithm to the local optima. In order to solve these challenging problems the notion of competent genetic algorithms have been proposed [9]. These algorithms are designed to solve complicated problems quickly, accurately and reliably.

The building-block hypothesis asserts that for successful problem solving, the genetic algorithm should be able to identify, preserve and combine the building-blocks of optimum solutions effectively. It turns out that as the genetic operators, especially recombination, employed in the genetic algorithm do not regard the building blocks, they will probably disrupt them. In order to overcome this problem and as a solution to design a competent genetic algorithm, the dependencies or linkages between the problem variables can be identified and be taken into consideration when dealing with solution strings.

Linkage learning, as it is known in the genetic algorithm community, is itself a challenging task which several methods have been proposed for. Estimation of distribution algorithms (also known as probabilistic model-building genetic algorithms) as a type of genetic algorithms and a solution to

linkage learning, exploit a probabilistic model to capture and store the interdependencies between problem variables. In a typical estimation of distribution algorithm the recombination operator of the traditional genetic algorithm, which is responsible for generating new solutions to the problem, is replaced by learning a probabilistic model from promising solutions of the problem and then sampling this model to generate new solutions.

A. Bayesian Optimization Algorithm

Since the probabilistic model used in estimation of distribution algorithms directly affects the behavior and efficiency of these algorithms, the choice of the probabilistic model should be taken carefully. Bayesian network with capability to represent several multivariate dependencies among problem variables is one of the probabilistic models that have made several complex problems solvable. Bayesian Optimization Algorithm (BOA) [10], Estimation of Bayesian Network Algorithm (EBNA) [2] and Learning Factorized Distribution Algorithm (LFDA) [6] are examples of the algorithms that use Bayesian network as their probabilistic model. Since the workings of these algorithms are almost similar, in this paper we will only consider BOA.

Although the use of powerful probabilistic models such as Bayesian network has helped estimation of distribution algorithms to solve complex problems that were considered challenging for traditional genetic algorithms, but the great computational complexity of model-building in these algorithms has created a further challenge for their scalability. Thus in recent years many efforts are made to enhance the efficiency of these algorithms. The accuracy of the models used for guiding the search in the space of all possible solutions to the problem is one of the issues that have attracted many attentions. Model accuracy is not only important because a good model can help generating better solutions, but also because a model with good accuracy can reveal the structure of the underlying problem, the point that is very important in unknown real world problems.

B. Previous Works

Hauschild et al. [3] have analyzed the probabilistic models built by BOA for two different types of problems. They tried to trace the model-building process over different generations and investigate how different dependencies are added to the network. For evaluating the accuracy of the models the notion of perfect model is introduced. Perfect model is a model

consisting of all necessary dependencies without having any spurious or unnecessary dependencies.

The role of common selection and replacement strategies on the quality of the models and the performance of the BOA is being analyzed by Lima et al. [5]. They have considered correct building-block partitions (elsewhere called module [1]) and computed the ratio of these partitions with correct and incorrect linkages. Their analysis show that while tournament selection in combination with restricted tournament replacement can result in better performance of the algorithm, but when model quality is of more importance one should use truncation selection.

Model structural accuracy (MSA) is used as a measure for evaluating model quality in [4]. This metric is the ratio of correct linkages over total linkages encoded in the model. The authors have used this metric to measure model overfitting in BOA. They have investigated how modifying the scoring metric used for model-building can help to increase model accuracy.

Although all of these and other related works have given valuable knowledge about model-building accuracy in EDAs using Bayesian Network, the class of problems investigated are restricted to single level decomposable problems. In this paper we try to extend the experiments to the class of hierarchical problems, where problem structure is organized in more than one level. In order to evaluate model accuracy for hierarchical problems a new evaluation algorithm is proposed and is tested on both single level and multi level problems.

II. PROPOSED METHOD

A. Template Matrices

Evaluating model accuracy, especially structural accuracy should be done with respect to a reference. When we are building a probabilistic model, e.g. Bayesian network, our reference to evaluate the network is the population of

promising solutions. If the model-building process of the algorithm is accurate enough, then the result will be an exact model according to the population of promising solutions used during model-building. However this exact model may not correctly represent the problem structure. Therefore another reference achieved from the problem structure should be used to evaluate the model. Perfect models are such references that can be used to determine how close the built model is to the nature of the problem at hand.

A typical perfect model for a problem is represented by addressing possible correct dependencies available to be added to the model, and incorrect dependencies that should be avoided. But when considering hierarchical models, this way of representation will not help. In a hierarchical problem, especially fully hierarchical problems, almost all problem variables can have dependencies with each other. Thus the number of incorrect dependencies cannot be used to detect model accuracy. On the other hand, certainly not all models for a hierarchical problem are perfect or even exact. Therefore another method for representing perfect models of hierarchical problems should be used. One clue for achieving this model is the multi-level structure of hierarchical problems. In other words the dependencies in a hierarchical problem can be clustered in to different groups according to their level of occurrence. Hence a perfect model for a hierarchical problem should be able to distinguish between linkages in different levels of hierarchy.

In the method proposed in this paper, the dependencies are ranked according to their level, and each level's dependencies are traversed separately. The reference model used here is called template matrix since it is used as a template guide for the possible linkages that can be added to network in different levels. In this matrix forbidden links are shown with zero, level one links with 1, level two links with 2, etc. Fig. 1 shows typical template matrices for two hierarchical problems.

B. Extracted metrics

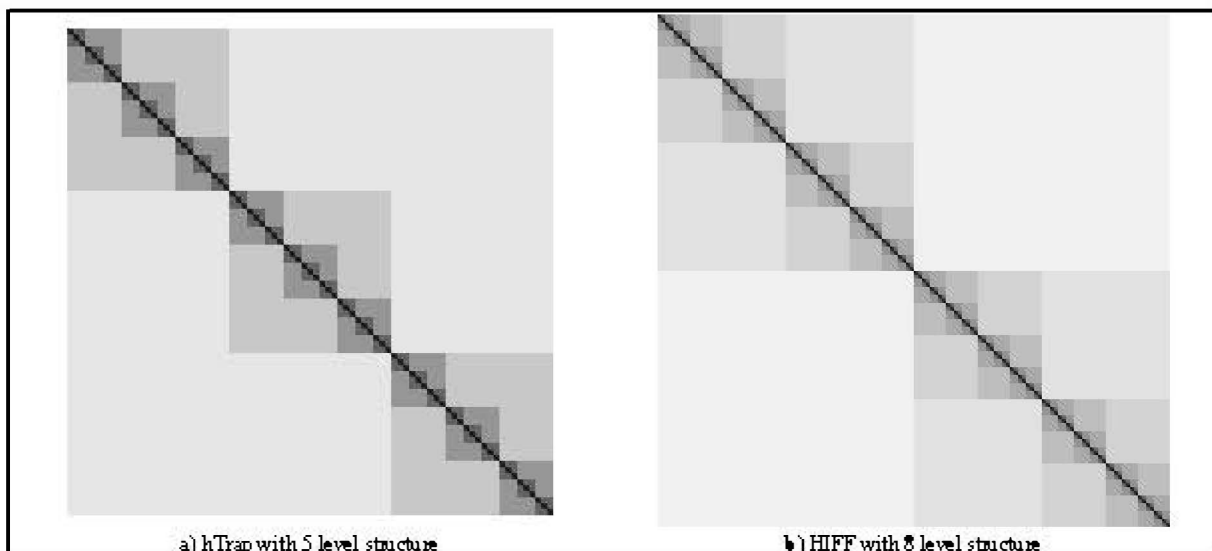


Figure 1. Typical template matrices for two hierarchical problems

Given the template matrix for a hierarchical problem, the constructed model for this problem can be evaluated. The model evaluator algorithm starts from the first level of the hierarchy and matches each link in the model with its corresponding entry in the template matrix and through a series of phases it decides about the link. In order to have a quantity about the quality of the model several indicators or metrics are computed during model evaluation. The terms used for describing these metrics are widely employed in the machine learning terminology. *False positive rate* is the ratio of false positive links to total number of links added to model. *False negative rate* is the ratio of false negative links to total number of possible links which were not added to the model. The way that false positive or false negative links are computed will be described in the next sub-section.

As mentioned earlier in a typical hierarchical problem all variables can be connected to each other. As a result false positive links can barely be identified. However a good model is required to show the actual structure of a problem with as less number of links as possible. Thus another metric is used to show how perfect a given model is. *Excess links rate* is the ratio of excess links (which is different from spurious links) to total number of links added to the model. Actually this metric is in direct correspondence with false positive rate metric. False positive rate is computed for the links in the first level of the hierarchy while excess links rate is computed for the links in the higher levels of the hierarchy.

C. Template based Model Structure Evaluator algorithm

In order to evaluate a hierarchical model the links in the model should be traversed hierarchically. The Template based Model Structure Evaluator (TMSE) algorithm proposed in this paper, works exactly in this way. Fig. 2 shows the pseudo code of this algorithm.

```

TMSE Algorithm (Template_Model: tm, Input_Model: im)
Begin
For each combination of variables i and j repeat
  If tm[i][j] = 0
    If im[i][j] = 0
      Increase number of true negative links
    Else //im[i][j] = 1
      Increase number of false positive links
      Mark i, j as connected
  Else if tm[i][j] > 0
    If im[i][j] = 0
      If i, j are already connected in this level
        Increase number of true negative links
      Else // i, j are not connected in this level
        Increase number of false negative links
    Else //im[i][j] = 1
      If i, j are already connected in this level
        Increase number of false positive links
      Or
        Increase number of excess links
      Else // i, j are not connected in this level
        Increase number of true positive links
        Mark i, j as connected
End

```

Figure 2. Template based Model Structure Evaluator (TMSE) algorithm

The algorithm is divided into two main phases to allow its compatibility for model evaluation of single-level problems as well as hierarchical problems. In the first phase, lower level links are processed, their validity is determined and any forbidden link added to the model is regarded as a false positive link. A module [1] formation procedure maintains the way variables are connected while processing the links. This step is especially essential since it will help to have a temporal ordering of how different blocks of the problem are constructed along the hierarchy.

In the second phase a similar mechanism is applied to all higher level links in a level by level order. If the link between two variables is allowed according to the reference template matrix, it is checked whether or not they are connected in this level. If already connected, this link is marked as an excess link otherwise a true positive link. In this way of link processing the order in which the links are processed will be important. But since we are only interested in the number of excess or false positive links and not on their place of occurrence, this order of visiting the links will not affect the algorithm.

This link processing is performed level by level until all links added to the model are processed. In addition to different metrics computed, the algorithm can depict different modules formed in each level of the hierarchy according to the links present in the model.

III. EXPERIMENTS

In order to examine the capabilities of the algorithm in model evaluation, we have used two popular hierarchical problems introduced in this field [8 and 11]: HIFF and hTrap. HIFF is one of the first hierarchical problems that challenge ordinary genetic algorithms by deceiving any attempts that try to solve the problem as a single level problem [11]. Claimed to be the ultimate challenge by its introducers, hTrap is based on the $m-k$ trap functions [8].

The BOA used in the experiments is the one equipped with diversity preserving mechanism, namely RTR, and decision tree representation for the Bayesian network. This combination for BOA is also referred to as hierarchical BOA (hBOA) [8]. The algorithm is required to find the solution string for the problem or reach 200 generations to terminate. For each instance of each problem, BOA is run 30 times to determine the appropriate population with the bisection algorithm. This also permits statistical analysis to be more accurate.

Fig. 3-a and 5-a show two real models derived by BOA for two instances of aforementioned hierarchical problems. The TMSE algorithm gets such models as input and according to a template matrix, evaluates the model. It can be seen that the algorithm actually clusters the linkages in the model into different groups depending on their level and therefore it can decide which edges are excess in each group. The false positive and false negative rates are negligible for these models so they are not included here.

Fig. 3-b and 5-b depict the module formation during model evaluation by TMSE algorithm. When more than one edge connects to modules, some kind of excess information is supplied.

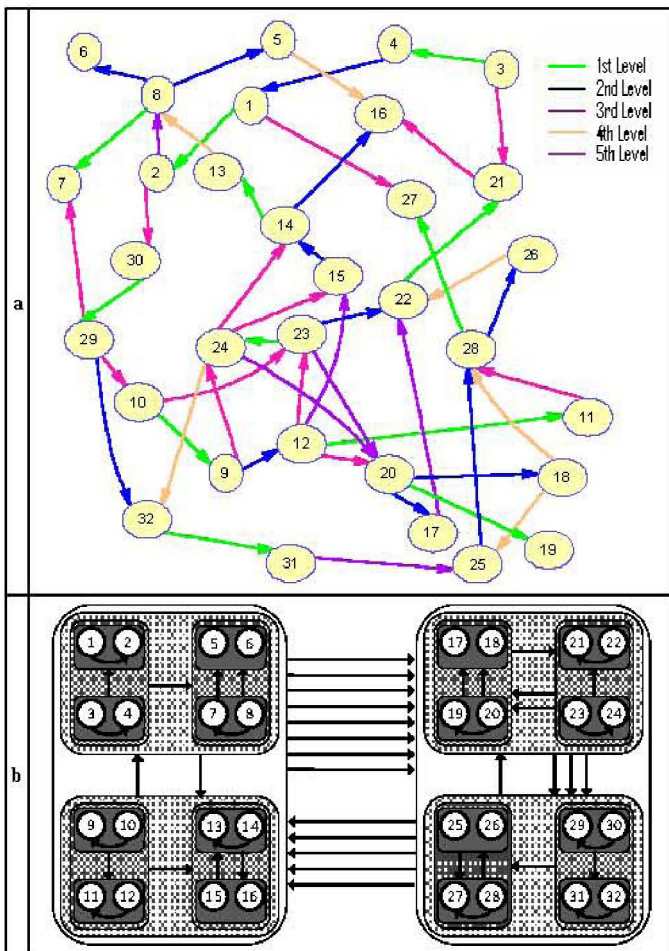


Figure 3. Model evaluation by TMSE algorithm for HIFF problem of size 32: a) Edge clustering into different levels, b) Module formation process.

The direction of edges can also be taken into account which allows linkages in different directions to exist between two different modules. However since in this context directions of the linkages are important, links in different directions are not treated separately. When an edge connects two modules that are already connected indirectly in the level under consideration, this edge is not considered as an excess link. For example all three edges in Fig. 4 are permitted.

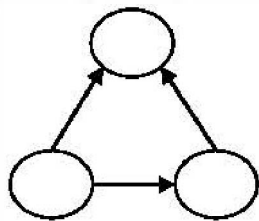


Figure 4. A possible edge connection between three modules of a problem.

Fig. 6 shows the accuracy of models built for the hierarchical problems with different sizes (and different number of levels). Here two versions of hTrap problem are used [7]. The computed accuracy is averaged over all runs (i.e. 30 runs) performed for solving the problem. It can be seen that these models have an accuracy of about 60 to 75 percent with regard to excess link rate. This level of accuracy is not very bad

since the model learning process is completely blind to the problem structure and also it is only working with a subset of possible solutions of the problem.

IV. CONCLUSION

Bayesian optimization algorithm which uses Bayesian network as its probabilistic model is one of the most successful estimation of distribution algorithms. A special version of this algorithm which is equipped with diversity preserving mechanism and a proper network representation type, has shown to be able to solve a complex class of problems known as hierarchical problems.

An important issue in this type of algorithms is how perfect they can build the model representing the problem structure. This is important not only because many enhancement techniques for this algorithms assume their dependability but also because if acquired correctly, these models can represent the underlying structure of previously unknown problems. While some success in evaluating model accuracy for single-level problems have been achieved, but no work is done to evaluate the quality of the models built for hierarchical problems due to their complicated, multi-level structure.

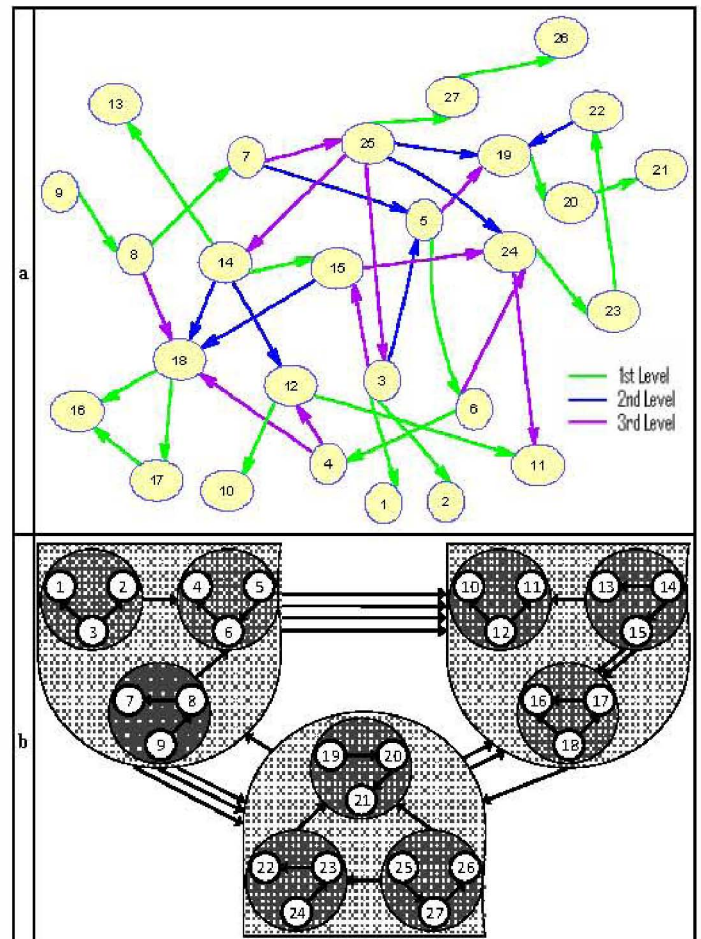


Figure 5. Model evaluation by TMSE algorithm for hTrap problem of size 27: a) Edge clustering into different levels, b) Module formation process.

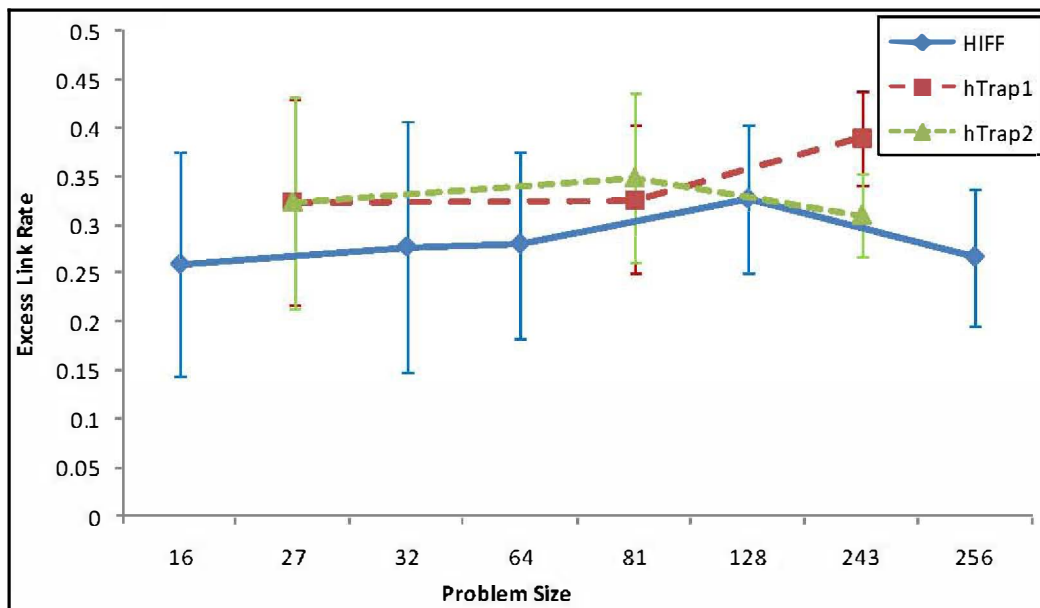


Figure 6. Model accuracy with respect to excess link rate for different sizes of HIFF and two versions of hTrap problem. The results are averaged over 30 different runs of BOA.

To evaluate models built for hierarchical problems a model evaluation algorithm is introduced in this paper, called TMSE algorithm. This algorithm uses a special kind of perfect models represented by a template matrix and evaluates the given model level by level. At least three metrics can be extracted by this algorithm which can depict different aspects of model accuracy. The metric that is of most relevance to hierarchical problems is the rate of excess links added to the model during model construction. This mechanism of model evaluation is visually depicted for two real models derived for common hierarchical problems. The computed model accuracy for different sizes of these problems is also shown.

REFERENCES

- [1] E.D. de Jong, D. Thierens, and R.A. Watson, "Hierarchical Genetic Algorithms", *LECTURE NOTES IN COMPUTER SCIENCE*, 2004, pp. 232-241.
- [2] C. Echegoyen, J.A. Lozano, R. Santana, and P. Larranaga, "Exact Bayesian network learning in estimation of distribution algorithms", *In Proceedings of IEEE Congress on Evolutionary Computation (CEC 2007)*, 2007, pp. 1051-1058.
- [3] M. Hauschild, M. Pelikan, C.F. Lima, and K. Sastry, "Analyzing probabilistic models in hierarchical BOA on traps and spin glasses", *In Proceedings of 9th Genetic and Evolutionary Computation Conference (GECCO-2007)*: ACM Press New York, NY, USA, 2007, pp. 523-530.
- [4] C.F. Lima, F.G. Lobo, and M. Pelikan, "From mating pool distributions to model overfitting", *In Proceedings of Genetic and Evolutionary Computation Conference (GECCO-2008)*: ACM New York, NY, USA, 2008, pp. 431-438.
- [5] C.F. Lima, M. Pelikan, D.E. Goldberg, F.G. Lobo, K. Sastry, and M. Hauschild, "Influence of selection and replacement strategies on linkage learning in BOA", *In Proceedings of IEEE Congress on Evolutionary Computation (CEC 2007)*, 2007, pp. 1083-1090.
- [6] H. Muhlenbein and T. Mahnig, "FDA-A scalable evolutionary algorithm for the optimization of additively decomposed functions", *Evolutionary Computation*, 7(4), 1999, pp. 353-376.
- [7] M. Pelikan, Hierarchical Bayesian optimization algorithm, Springer, 2005.
- [8] M. Pelikan and D.E. Goldberg, "A hierarchy machine: Learning to optimize from nature and humans", *Complexity*, 8(5), 2003, pp. 36-45.
- [9] M. Pelikan and D.E. Goldberg, "Escaping Hierarchical Traps with Competent Genetic Algorithms", *In Proceedings of Genetic and Evolutionary Computation Conference (GECCO-2001)*: Morgan Kaufmann, 2001, pp. 511-518.
- [10] M. Pelikan, D.E. Goldberg, and E. Cantu-Paz, "BOA: The Bayesian optimization algorithm", *In Proceedings of Genetic and Evolutionary Computation Conference (GECCO-99)*, 1999, pp. 525-532.
- [11] R.A. Watson, G.S. Hornby, and J.B. Pollack, "Modeling Building-Block Interdependency", *In Proceedings of Parallel Problem Solving from Nature (PPSN)*, 1998, pp. 97-108.