Engineering

Industrial & Management Engineering fields

Okayama University *Year* 2003

Estimation of Bayesian network algorithm with GA searching for better network structure

Hisashi Handa
Okayama University

Osamu Katai
Kyoto University

IEEE Int. Conf. Neural Networks & Signal Processing
Nanjing, China, December 14-17, 2003

ESTIMATION OF BAYESIAN NETWORK ALGORITHM WITH GA SEARCHING FOR BETTER NETWORK STRUCTURE

*Hisashi Handa**
*Osamu Katai***

*Okayama University
Tsushima-naka 3-1-1
Okayama, 700-8530, JAPAN
**Kyoto University
Yoshida-Hommachi
Kyoto, 651-8501, JAPAN

ABSTRACT

Estimation of Bayesian Network Algorithms which adopt Bayesian Networks as the probabilistic model were one of the most sophisticated algorithms in the Estimation of Distribution Algorithms. However the estimation of Bayesian Network is key topic of this algorithm, conventional EBNAs adopt greedy searches to search for better network structures. In this paper, we propose a new EBNA which adopts Genetic Algorithm to search the structure of Bayesian Network. In order to reduce the computational complexity of estimating better network structures, we elaborates the fitness function of the GA module, based upon the synchronicity of specific pattern in the selected individuals. Several computational simulations on multidimensional knapsack problems show us the effectiveness of the proposed method.

1. INTRODUCTION

Estimation of Distribution Algorithms are a promising approach in the EC literature. Instead of crossover and mutation operations in the conventional GAs, the EDAs employ the probabilistic model estimated from selected individuals to generate the next generation. Such probabilistic models for EDAs are widely devised. EBNAs, which adopt Bayesian Networks as the probabilistic model, were one of the most sophisticated algorithms in the EDAs. In general, in order to generate the Bayesian Network from selected individuals, the score + search algorithm is used in the EBNAs. However the scoring methods are proposed in various, the search method is basically a kind of greedy algorithms because of the NP property of searching better network structure of Bayesian Networks.

In this paper, we propose a new EBNA algorithm which has two populations: first one is ordinal EDA population, namely, it searches for good solutions in the given problem. The other one is a GA population whose individuals indicate partial network structures for the first population. In order to reduce the computational complexity of estimating better network structures, we elaborates the fitness function of the GA module, based upon the synchronicity of specific pattern in the selected individuals.

In next section, we will explain conventional EDAs briefly. Then, we explain the proposed method in Section 3. In Section 4, several computer simulations are examined and confirm us effectiveness of our approaches, and finally, this paper is concluded.

2. ESTIMATION OF BAYESIAN NETWORK ALGORITHM

2.1. General Framework of EDAs

The Estimation of Distribution Algorithms are a class of evolutionary algorithms which adopt probabilistic models to reproduce the genetic information of the next generation, instead of conventional crossover and mutation operations. The probabilistic model is represented by conditional probability distributions for each variable (locus). This probabilistic model is estimated from the genetic information of selected individuals in the current generation. Hence, the pseudo-code of EDAs can be written as Fig. 1, where D_l, D_{l-1}^{Se}, and $p_l(\mathbf{x})$ indicate the set of individuals at lth generation, the set of selected individuals at $l-1$th generation, and estimated probabilistic model at lth generation, respectively [1]. The representation and estimation methods of the probabilistic model are devised by each algorithm. The following subsections will overview some EDAs. For a more thorough overview, see [1].

0-7803-7702-8/03/ $17.00 ©2003 IEEE

```
Procedure Estimation of Distribution Algorithm
begin
  initialize D_0
  evaluate D_0
  until Stopping criteria is hold
    D_{l-1}^{Se} ← Select N individuals from D_{l-1}
    p_l(x) ← Estimate the prob. model from D_{l-1}^{Se}
    D_l^{Se} ← Sampling M individuals from p_l(x)
  end
end
```

Fig. 1. Pseudo code of Estimation of Distribution Algorithms

```
Procedure EBNA with GA
begin
  initialize D_0
  evaluate D_0
  until Stopping criteria is hold
    D_{l-1}^{Se} ← Select N individuals from D_{l-1}
    E_l ← GA searches for good partial network
          structure from D_{l-1}^{Se}
    S_l ← Build network structure from E_l
    p_l(x) ← Constitute the BN from D_{l-1}^{Se}, S_l
    D_l^{Se} ← Sampling M individuals from p_l(x)
  end
end
```

Fig. 2. Pseudo code of EBNA with GA

2.2. EBNA

Like BOA and LFDA [2], [3], the EBNA (Estimation of Bayesian Networks Algorithms) adopts Bayesian Network (BN) as the probabilistic model, which is proposed by Larrañaga *et al.* [1]. They proposed several kinds of EBNAs, such as EBNA_{PC}, EBNA_{K2+pen}, EBNA_{BIC}, and so on. Here, we introduce only EBNA_{BIC} used in our experiments. EBNA_{BIC} searches for the better structure of BN by using search+score method. In the case of the EBNA_{BIC}, scoring is achieved by penalized maximum likelihood $BIC(S, D)$ for a given structure S and a dataset D, called Bayesian Information Criteria, denoted by the following equation:

$$BIC(S,D) = \sum_{i=i}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk}\log\frac{N_{ijk}}{N_{ij}} - \frac{1}{2}\log N\sum_{i=1}^{n} q_i(r_i-1), \quad (1)$$

where the structure S is represented by Direct Acyclic Graphs, n is the number of variables of the Bayesian Network, r_i is the number of different values that variable X_i can take, q_i is the number of different values that the parent variables of X_i in the structure S can take, N_{ij} is the number of individuals in D in which the parent variables of variable X_i take their j^{th} value, and N_{ijk} is the number of individuals in D in which variable X_i takes its k^{th} value and the parent variables of the variable i take their j^{th} value [1]. As the search method in the EBNA_{BIC}, an arc-based local search is adopted due to the NP property of searching the best structure for BNs.

3. EBNA with GA

3.1. Framework

In this paper, we propose a new EBNA algorithm which has two populations: first one is ordinal EDA population, namely, it searches for good solutions in the given problem. The other one is a GA population whose individuals indicate partial network structures for the first population. In this paper, we define "good" partial network structure as a set of variables whose values are frequently occurred in the selected individuals in the EDA population. The procedure of the proposed method is described in Fig. 2. As described in this figure, GA search is carried out in every generation. In the case of EBNA_{BIC}, the computational cost of the calculations of the BIC score, which is used to measure of the effectiveness of network structures, is quite expensive. Hence, we introduce naive fitness evaluation method which evaluates how effective the partial structures (GA individuals) are. The following subsections describe the coding method, the fitness evaluation, and how to build Bayesian Network from evolved population, respectively.

3.2. Individual Representation

The GA module incorporated in the proposed method searches for good partial network structures. In order to represent partial network, each individual in the second population consists of 1 integer part and N bits string part, where N is the number of variables in the given problems, as described in Fig. 3. The former part denotes antecedent node in the Bayesian Network, and the latter part shows which nodes are precedent variables for the antecedent node indicated by the former part. Hence, if an allele in the latter part is 1, corresponding node is activated as precedent variable for the antecedent variable.

3.3. Fitness Evaluation

In order to calculate the fitness for GA individuals, we first constitute the conditional probabilities table (CPT) of all couples of variables from the selected individuals D_{l-1}^{Se}. Suppose that the probability such that a variable X_i has a value x_i is represented by $P(X_i = x_i)$, and the conditional probability such that

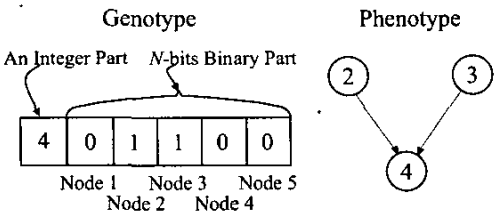

Fig. 3. Coding method for the GA module

a variable X_i has a value x_i if a variable X_j has a value x_j is denoted by $P(X_i = x_i|X_j = x_j)$. Moreover, let the most observed value at an antecedent node X_a of GA individual in the selected individuals in the EDA population be x_a^* and let value for each precedent variable j, which has the highest conditional to $X_a = x_a^*$, be x_j^{a*}, i.e.,

$$x_j^{a*} = \underset{x_j \in X_j}{\operatorname{argmax}} P(X_a = x_a^*|X_j = x_j).$$

Then, the fitness $F_{GA}$of an individual in the GA population is calculated as

$$F_{GA} = \sum_{X_j \in n_p} \log(P(X_a = x_a^*|X_j = x_j^{a*})P(X_j = x_j^{a*})).$$

In comparison with BIC score, this fitness measure might be insufficient to build general Bayesian Networks. However, in the case of applying the Bayesian Networks to solve optimization problems, we consider the fitness measure is sufficient and effective since (1) the EDA population tends to converge certain better solutions and (2) computational effort of this fitness is lower than the BIC score in equation (1).

3.4. Building Bayesian Networks

After GA search is finished, partial networks, i.e., survived individuals in the final GA population, are assembled into a network structure. The procedure of building a Bayesian Network is as follows: First, Network structure S is set to be $\oslash$. Next, GA population are sorted in accordance with their fitness. For each individual j in this sorted population, if the partial network represented by the individual j is consistent with the network structure S, then the partial network is added into the network structure S. Here, "consistent situation" against existent network structure is (1) antecedent variable does not have any path to precedent variable in S, moreover, (2) none of precedent variables have any path to the other precedent variables in S.

4. COMPUTATIONAL SIMULATIONS

4.1. Parameters and Problem Settings

In all the experiment whose results will be shown in the next subsection, we compare three kinds of evolutionary computations: Steady-State GA (SSGA) [5], Estimation Bayesian Network Algorithm (EBNA$_{BIC}$), and the proposed method. The GA parameters of SSGA, i.e., the population size, the mutation probability, and the crossover probability are set to be 100, 0.05, and 1.0, respectively. The population size and the size of selected population of the EBNA$_{BIC}$ are set to be 3000 and 1000, respectively. Those of the proposed method are the same as the EBNA$_{BIC}$.

This paper applies the proposed method to large-scaled combinatorial problems (multidimensional knapsack problems). In general, multidimensional knapsack problems are formulated as follow:

$$\text{maximize} \quad \sum_{j=1}^{n} c_j x_j, \tag{2}$$

$$\text{subject to} \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad (i = 1, \ldots, m). \tag{3}$$

Here, c_j, a_{ij} and b_i are positive values, and x_j is a binary value which takes 0 or 1.

In order to apply SSGA, EBNA$_{BIC}$, and the proposed method to the above multidimensional knapsack problems, we adopt the following fitness function in a similar manner to that used by Goldberg *et al.*[6]:

$$Fitness = O - C_{penalty} \times (\sum_{j=1}^{n} e_j)^2$$

where O denotes the objective value of the given problem, i.e., the maximum value in equation (1), $C_{penalty}$ is a predefined coefficient for the penalty term, and e_j indicates the quantity of constraint violation in each inequality constraint in equation (3).

This paper uses the problems generated randomly as the problem instances of the multidimensional knapsack problems. Randomly Generated Multidimensional Knapsack problems (RGMK) are generated by the manner described by Sakawa *el al.* [7], and are described as follows: In equations (2) and (3), the coefficients c_j, a_{ij} are randomly set in the interval $[0, 999]$ by selecting them with uniform distribution. Moreover, the coefficients b_i's are set by using a random variable $\gamma \in [10, 20]$ and the following equation:

$$b_i = \gamma \sum_{j=1}^{n} a_{ij}, \quad (i = 1, \ldots, m).$$

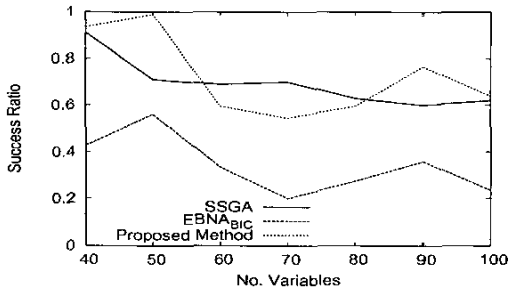

Fig. 4. Experimental results on the Randomly Generated Multidimensional Knapsack Problems; success ratio

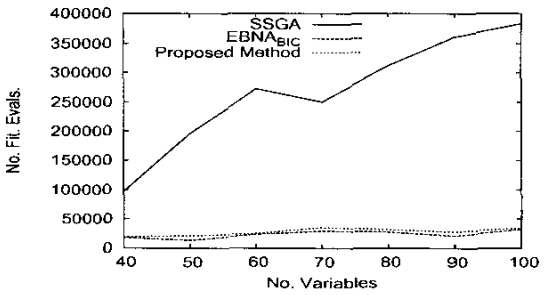

Fig. 5. Experimental results on the Randomly Generated Multidimensional Knapsack Problems; the number of fitness evaluations

4.2. Experimental Results

We compare the proposed method with SSGA and EBNA$_{BIC}$ on the RGMK problems. Their results are depicted in Fig. 4 and Fig. 5. The x axes in these figure indicate the number of variables. The y axis in Fig. 4 is the success ratio, namely, a ratio the number of finding optimal solutions to the total number of experiments. On the other hand, the y axis in Fig. 5. indicates the averaged number of fitness evaluations until finding an optimal solution in the case of success experiments, i.e., this index is not taking account the experiments such that algorithms converges into local optima. In these experiments, the optimal solutions of RGMK are searched by lp_solve in advance. These experiments are carried out until the number of fitness evaluations is equal to ten million (RGMK). In RGMK, 10 problem instances are generated for each variable size, and 10 experiments are carried out for each problem instance.

The success ratio of the proposed method is dramatically improved that of EBNA$_{BIC}$ but is similar to that of SSGA. Since the difference between the proposed method and EBNA$_{BIC}$ is only how to generate Bayesian Networks, we can see GA module incorporated in the proposed method works well. On the other hand, the number of fitness evaluations of EBNA$_{BIC}$ and the proposed method are less than SSGA's. In the case of applying to multidimensional knapsack problems, since the calculation of the fitness has no so heavy computational cost and the estimating Bayesian Network takes a large amount of computational effort, this difference does not affect computational time. However, for problems which have the heavier computational cost of fitness, EBNA$_{BIC}$ and the proposed method would solve such problems faster than SSGA. In spite of the use of GA module in the proposed method, the computational time of estimating BN is shorter than EBNA$_{BIC}$ in the case of high-dimensional problems. It is due to the difference of the computational complexity between the fitness evaluation method in the proposed method and the calculation of BIC score.

5. CONCLUSION

In this paper, we proposed a new EBNA which adopts Genetic Algorithm to search the structure of Bayesian Network. In order to reduce the computational complexity of estimating better network structures, we elaborated the fitness function of the GA module, based upon the synchronicity of specific pattern in the selected individuals. Simulation results tell us that (1) the proposed method improve the success ratio of finding optimal solution in comparison with EBNA$_{BIC}$, and (2) the proposed method searches for optimal solutions faster. Future work is (1) application to another kind combinatorial problems and (2) analysis of the estimated Bayesian Networks.

6. REFERENCES

[1] P. Larrañaga and J. A. Lozano Editors, *Estimation of Distribution Algorithms*, Kluwer Academic Publishers, 2002.

[2] M. Pelikan *et al.*, "BOA: The Bayesian optimization algorithm", *Proceedings of the Genetic and Evolutionary Computation Conference*, Vol.1, pp. 525–532, 1999.

[3] H. Mühlenbein and T. Mahnig, "FDA - a scalable evolutionary algorithms for the optimization of additively decomposed functions", *Evolutionary Computation*, Vol.7, No.4, pp. 353–376, 1999.

[4] E. Bengoetxea *et al.*, "Learning and simulation of Bayesian networks applied to inexact graph matching", *Pattern Recognition*, Vol.35, No.12, pp. 2867–2880, 2002.

[5] G. Syswerda, "A Study of Reproduction in Generational and Steady-State Genetic Algorithms", *Foundations of Genetic Algorithms*, G. J. E. Rawlins Editors, Morgan Kaufman, pp. 94–101, 1990.

[6] D. E. Goldberg and R. E. Smith, "Nonstationary Function Optimization using Genetic Algorithms with Dominance and Diploidy", *Proc. of 2nd ICGA*, eds. J. J. Grefenstette, pp. 59–68, 1987.

[7] M. Sakawa *et al.*, "Genetic Algorithms with Double Strings for Multidimensional Integer Knapsack Problems" (in Japanese), *J. of Japan Society for Fuzzy Theory and Systems*, Vol.12, pp. 562–569, 2000.