

Effect of Model Complexity for Estimation of Distribution Algorithm in NK Landscapes

Rung-Tzuo Liaw and Chuan-Kang Ting

Department of Computer Science and Information Engineering and

Advanced Institute of Manufacturing with High-tech Innovations

National Chung Cheng University

Chia-Yi 621, Taiwan

Email: {lrt101p, ckting}@cs.ccu.edu.tw

Abstract—Evolutionary algorithms (EAs) have been widely proved to be effective in solving complex problems. Estimation of distribution algorithm (EDA) is an emerging EA, which manipulates probability models instead of genes for evolution. EDA creates probability models based on the promising solutions in the population and generates offspring by sampling from these models. The model complexity is a key factor in the performance of EDA. Complex models can express the relations among variables more accurately than simple models. However, for some problems with strong interaction among variables, building a complex model for all the relations becomes unrealistic and impractical due to its high computational cost and requirement for a large population size. This study aims to understand the behaviors of EDAs with different model complexities in NK landscapes. Specifically, this study compares the solution quality and convergence speed of univariate marginal distribution algorithm (UMDA), bivariate marginal distribution algorithm (BMDA), and estimation of Bayesian network (EBNA) in the NK landscapes with different parameter settings. The comparative results reveal that high complexity does not imply high performance: Simple model such as UMDA and BMDA can outperform complex model like EBNA on the tested NK landscape problems. The results also show that BMDA achieves a stable high probability of generating the best solution and satisfactory solution quality; by contrast, the probability for EBNA drastically declines after some generations.

Index Terms—Estimation of distribution algorithm, model complexity, NK landscapes, probability model.

I. INTRODUCTION

Evolutionary algorithms (EAs) have widely proved to be an effective way to resolve complex and large-scale problems. One of the most famous EAs is the genetic algorithm (GA) proposed by Holland in 1970's [1], which introduces the concept of nature selection from Darwin to evolve candidate solutions through selection and variation operators. GA has achieved a lot of successes on search and optimization problems [2]. Nevertheless, some problems are hard for GA since the fitness is involved with the interaction of several genes. These genes are viewed to be linked or form a building block. To deal with this issue, some approaches based on the concept of exploiting the information in the population are proposed. Estimation of distribution algorithm (EDA) is a famous family of these EAs. Algorithm 1 presents the general framework of EDA. Like most EAs, EDA is a stochastic, population based optimization method [3], [4], [5]; however, it differs from other EAs in several places. The major difference lies in the

variation operators: For example, GA performs crossover and mutation to generate new individuals, whereas EDA selects a set of promising solutions to build probability models and then generate new solutions from these models through sampling.

A variety of EDAs have been proposed. According to [4], EDAs can be classified by the graphical model. A graphical model consists of graph structure and local parameters. The graph structure indicates the inference relations among random variables, where each variable has a local parameter for its distribution. Depending on the graph structure, EDAs are classified into univariate model, bivariate model (also known as tree-based model), and multivariate model (see Fig. 1).

The univariate model is the simplest probability model utilized in EDAs. This model assumes that all random variables are mutually independent. The EDA of this class includes population based incremental learning (PBIL) [6], compact genetic algorithm (cGA) [7], and univariate marginal distribution algorithm (UMDA) [8]. For the univariate model, the joint probability density function of a solution in the search space can be represented by the following factorization of random variables.

$$\mathcal{P}(\mathbf{x}) = \prod_{i=1}^n \mathcal{P}(x_i). \quad (1)$$

Beyond the assumption of mutual independence, the bivariate model considers conditional independence between random variables. The graph structure is then subject to the learning algorithm and may form a chain [9], a tree [10], or a forest [11]. Let $x_{pa(i)}$ denote the parent variable of random variable x_i . Equation (2) shows the factorization of conditional joint density function for solutions with bivariate model.

$$\mathcal{P}(\mathbf{x}) = \prod_{i=1}^n \mathcal{P}(x_i | x_{pa(i)}). \quad (2)$$

The multivariate model is a more complex model in which the graph structure is often represented as directed acyclic graph (DAG). Some multivariate EDAs based on different learning methods for the graph structure or local parameters have been proposed, e.g., Bayesian optimization algorithm (BOA) [12], hierarchical Bayesian optimization algorithm (hBOA) [13], estimation of Bayesian network (EBNA) [14],

and estimation of Gaussian network (EGNA) [15]. Equation (3) gives the factorization of probability distribution for solutions with multivariate model, where pa_i denotes the instantiation of x_i 's parent set.

$$\mathcal{P}(\mathbf{x}) = \prod_{i=1}^n \mathcal{P}(x_i | pa_i). \quad (3)$$

Complex graph structures are commonly believed to have a high ability of correctly describing the inference relations among variables and therefore to achieve good solution quality. However, Ceberio et al. [3] indicated that EDAs with complex model fail to outperform those with simple model on some permutation-based problems such as TSP, QAP, and LOP.

In this paper, we analyze the behavior of EDAs with different model complexities in NK landscapes. This study compares the solution quality and convergence speed of three EDAs. Furthermore, we investigate the probability of generating the known best solution. The remainder of this paper is organized as follows. Section II describes the NK landscape problem. Section III recapitulates the studies on model analysis of EDA and introduces the three adopted EDAs. Section IV presents the experimental results and discussion. Finally, in Section V we draw the conclusions and recommend some directions for future work.

II. NK LANDSCAPES

In 1989, Kauffman and Weinberger [16] proposed the NK model and NK landscape problem. The NK model can be regarded as a system composed of n parts, each of which corresponds to an element with several states and each part has interaction with other k parts. This model provides a family of benchmarks that can be easily tuned by adjusting parameters n and k , standing for the size and connectivity of the model respectively. In addition, the landscape of NK model with the same size n is changed whenever the connectivity k is changed. Restated, the connectivity k determines the ruggedness of the landscape. For instance, the landscape becomes linear for $k = 0$ and is most rugged for $k = n - 1$.

The NK landscape problem is a real-valued optimization problem whose objective is to maximize the sum of evaluation results from each part in the NK model. The main objective function of the problem consists of n sub-functions. The evaluation value of each sub-function is subject to the $k + 1$ corresponding elements. Let X and $x \in \{0, \dots, p - 1\}$ be a discrete random variable with p possible values and its instantiation, respectively. Similarly, let $\mathbf{X} = (X_1, \dots, X_n)$ denote a random vector and \mathbf{x} stand for its instantiation. Given an NK model $NK(n, k)$, where n denotes the dimension and $0 \leq k \leq n - 1$ represents the degree of connectivity, the problem can be formulated as an optimization problem

$$\operatorname{argmax}_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n f_i(x_i, \dots, x_{i+k}).$$

The value of each sub-function $f_i(\cdot)$ is predefined and generated according to an arbitrary distribution L ; therefore, we

Algorithm 1 General framework of EDA

```

1  Population initialization
2  Evaluation
3  while (Termination criterion is not met) do
4      Selection from population
5      Update model using selected individuals
6      Sampling new individuals from model
7      Evaluation new individuals
8      Replacement
9  end while

```

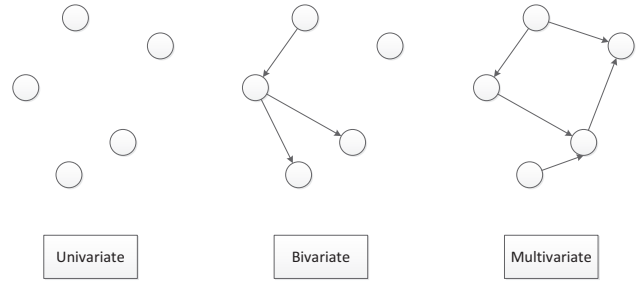


Figure 1: Graphical structures of probability models used in EDAs.

need to sample p^{k+1} values from L to determine the evaluation of a sub-function for each state, where totally np^{k+1} values will be sampled. For simplicity of analysis, \mathbf{X} is usually considered as a binary random vector and the distribution L is a uniform distribution between 0 and 1. In the experiments, we will use $p = 2$ and $L = U[0, 1)$, where the value corresponding to each state in each sub-function $f_i(\cdot)$ will be randomly sampled from L .

III. METHODOLOGY

The probability model plays an important role in EDAs since all the new individuals are sampled from it. Several studies are focused on the analysis of probability model for EDAs. Echegoyen et al. [17] analyzed the limitation on estimation of Bayesian network (EBNA) through the additively decomposable functions (ADFs) and found that the required population size will increase exponentially with the degree of interactions among sub-functions. They also pointed out that the number of edges in the learned structure through EBNA will be bounded in a small range when the number of sub-function increases. Furthermore, Echegoyen et al. [18], [19] conducted a quantitative analysis to understand the internal behavior of EBNA by comparing the probabilities of sampling the optimum solution and the most probable solution (MPS) between success and failure trials. They utilized three different structures of Bayesian network, a learned structure, a given structure corresponding to the problem, and a tree-based structure, to observe the influence on the solution quality. The results show that reducing the information accuracy in the probability model leads to the decrease in the probability of

Algorithm 2 UMDA

```

1   $D_0 \leftarrow$  Random sampling
2   $D_0 \leftarrow$  Evaluation
3   $t \leftarrow 0$ 
4  while (Termination criterion is not met) do
5     $D_t^s \leftarrow$  Selection( $D_t$ )
6     $\mathcal{P}_t^c(\mathbf{X}) \leftarrow$  Update model( $D_t^s$ )
7     $D_{t+1} \leftarrow$  Sampling( $\mathcal{P}_t^c(\mathbf{X})$ )
8     $D_{t+1} \leftarrow$  Evaluation
9     $D_{t+1} \leftarrow$  Survival( $D_t \cup D_{t+1}$ )
10    $t \leftarrow t + 1$ 
11 end while

```

generating the optimum solution. Lima et al. [20] investigated the model accuracy using the model structural accuracy (MSA) determined by the ratio between the number of correct edges to the total edges in the Bayesian network.

In this paper, we adopt the NK landscape problem to analyze the behavior of EDAs in terms of model complexity. Three EDAs with different graph structures are considered: univariate marginal distribution algorithm (UMDA) [8], bivariate marginal distribution algorithm (BMDA) [11], and estimation of Bayesian network algorithm (EBNA) [14], corresponding to the univariate model, bivariate model and multivariate model, respectively. More details about the three algorithms are given below.

A. UMDA

The univariate marginal distribution algorithm (UMDA) is proposed by Mühlenbein and Paaß [8]. As aforementioned, UMDA assumes that random variables are mutually independent. Recapturing the factorization of joint probability density function in (1), the probability $\mathcal{P}_t(X_i = x_i)$ of each random variable at time t is calculated by the frequency of x_i in the position i among the selected individuals, i.e.,

$$\mathcal{P}_t(X_i = x_i) = \frac{N(X_i = x_i)}{|D_t^s|},$$

where D_t^s represents the set of selected individuals at time t and function $N(X_i = x_i)$ gives the number of x_i existing in D_t^s . Note that the situation $N(X_i = x_i) = 0$, i.e., x_i not existing in D_t^s , causes the probability $\mathcal{P}_t(X_i = x_i) = 0$. Since new individuals are sampled from the probability model, such a zero probability in position i will cause the loss of diversity and consequently the premature convergence. The Laplace correction [5] resolves this problem by adding one to the number of occurrences for each possible x_i value. Hence,

$$\mathcal{P}_t(X_i = x_i) = \frac{N(X_i = x_i) + 1}{|D_t^s| + p},$$

where p is the number of possible values of x_i , e.g., $p = 2$ for binary strings.

To balance the information accumulated from the past population to the current, a learning rate α is introduced to

Algorithm 3 Structure learning of BMDA

```

1   $A \leftarrow V \leftarrow \{1, \dots, n\}$ ,  $E \leftarrow \phi$ 
2   $R \leftarrow$  arbitrary node  $i$  from  $A$ 
3   $A \leftarrow A \setminus i$ 
4  if  $A = \phi$  then goto 9
5  if  $\mathcal{X}_{i,j}^2 < 3.84 \forall i \in A$  and  $\forall j \in V \setminus A$  then goto 2
6   $i \leftarrow \operatorname{argmax} \mathcal{X}_{i,j}^2 \forall i \in A$  and  $\forall j \in V \setminus A$ 
7   $E \leftarrow E \cup e(i, j)$ 
8  goto 3
9  End

```

Algorithm 4 BMDA

```

1   $D_0 \leftarrow$  Random sampling
2   $D_0 \leftarrow$  Evaluation
3   $t \leftarrow 0$ 
4  while (Termination criterion is not met) do
5     $D_t^s \leftarrow$  Selection( $D_t$ )
6     $M \leftarrow$  Structure learning( $D_t^s$ )
7     $\mathcal{P}_t(\mathbf{X}) \leftarrow (M, D_t^s)$ 
8     $O \leftarrow$  Ancestral order( $M$ )
9     $D_{t+1} \leftarrow$  Sampling( $O, \mathcal{P}_t(\mathbf{X})$ )
10    $D_{t+1} \leftarrow$  Evaluation
11    $D_{t+1} \leftarrow$  Survival( $D_t \cup D_{t+1}$ )
12    $t \leftarrow t + 1$ 
13 end while

```

the calculation of probability:

$$\mathcal{P}_t^c(X_i = x_i) = \alpha \mathcal{P}_t(X_i = x_i) + (1 - \alpha) \mathcal{P}_{t-1}^c(X_i = x_i)$$

and $\mathcal{P}_0^c(X_i = x_i) = \mathcal{P}_0(X_i = x_i)$. In UMDA, the sampling order does not matter due to its assumption of mutual independence. Algorithm 2 presents the procedure of UMDA.

B. BMDA

Bivariate marginal distribution algorithm (BMDA) is an EDA with bivariate model proposed by Pelikan and Mühlenbein [11]. Unlike UMDA, there are two stages in constructing the probability model in BMDA, including the graph structure learning and the conditional probability calculation. As mentioned in Section I, an arc $e(i, j)$ from node i to node j represents an inference relation between them. To establish the graph structure, BMDA checks the Pearson's chi-square statistics between different random variables, which express the independence of the two variables. Let $\mathcal{X}_{i,j}^2$ indicate the chi-square value of variables i and j . Two variables are viewed as independent if $\mathcal{X}_{i,j}^2 < 3.84$ at 95% confidence level. Equation (4) shows the formula for calculating the Pearson's chi-square, where $\mathcal{P}(x_i)$ denotes the frequency of $X_i = x_i$ and $\mathcal{P}(x_i, x_j)$ stands for the frequency of $(X_i = x_i) \cap (X_j = x_j)$, both with Laplace correction as aforementioned.

$$\mathcal{X}_{i,j}^2 = \sum_{x_i, x_j} \frac{(|D_t^s| \mathcal{P}(x_i, x_j) - |D_t^s| \mathcal{P}(x_i) \mathcal{P}(x_j))^2}{|D_t^s| \mathcal{P}(x_i) \mathcal{P}(x_j)} \quad (4)$$

Algorithm 5 EBNA

```

1   $D_0 \leftarrow$  Random sampling
2   $D_0 \leftarrow$  Evaluation
3   $t \leftarrow 0$ 
4  while (Termination criterion is not met) do
5     $D_t^s \leftarrow$  Selection( $D_t$ )
6     $M \leftarrow$  Hill climbing( $M, D_t^s$ )
7     $\theta \leftarrow (M, D_t^s)$ 
8     $O \leftarrow$  Ancestral order( $M$ )
9     $D_{t+1} \leftarrow$  Sampling( $O, \theta$ )
10    $D_{t+1} \leftarrow$  Evaluation
11    $D_{t+1} \leftarrow$  Survival( $D_t \cup D_{t+1}$ )
12    $t \leftarrow t + 1$ 
13 end while

```

Algorithm 3 describes a greedy method for the dependency graph with maximal sum of chi-square. This method outputs a structure E containing multiple trees and a node set R containing the root nodes of each tree. After construction of graph structure, we need to calculate the frequency for sampling. For the root nodes, the calculation is the same as UMDA; as for other nodes, it needs to calculate the conditional probability. Further, BMDA uses the probabilistic logic sampling algorithm (PLS) [21] to find the ancestral order and sample new individuals depending on this order. For example, arc $e(i, j)$ represents that node i precedes node j ; therefore, the variable X_i must be sampled before variable X_j . Algorithm 4 displays the procedure of BMDA.

C. EBNA

Etzeberria and Larrañaga [14] proposed an EDA based on Bayesian network called estimation of Bayesian network algorithm (EBNA), which consists of a graph structure M and local parameters θ_i for each random variable X_i . Like BMDA, EBNA needs to learn a graph structure and then calculate the local parameters θ_i for sampling. To learn the graph structure of Bayesian network, EBNA uses a metric called Bayesian information criterion (BIC):

$$\text{BIC}(M, D_t^s) = \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log \frac{N_{ijk} + 1}{N_{ij} + r_i} - \frac{\log N}{2} \sum_{i=1}^n (r_i - 1) q_i \quad (5)$$

where q_i denotes the number of instantiations for the set of parents of random variable X_i and r_i denotes the number of instantiations of random variable X_i . Let $X(k)$ be the k -th instantiation of random variable X and $pa_i(j)$ be the j -th instantiation of the set of parents of random variable X_i . The equations $X_i = X_i(k)$ and $Pa_i = pa_i(j)$ mean that the random variable X_i is in its k -th instantiation and the set of parents of random variable X_i is in its j -th instantiation, respectively.

The first term in (5) represents the likelihood of the given structure M according to the set of individuals D_t^s with

Table I: Parameter setting

Parameter	Value	
Problem size (n)	10	30
Generation	5000	15000
Population Size (m)	100	1000
Selection	Truncation of $m/2$	
Sampling	Probabilistic Logic Sampling	
Replacement	$\mu + \lambda$	

Laplace correction, where N_{ijk} indicates the number of individuals satisfying $X_i = X_i(k)$ and $Pa_i = pa_i(j)$. The second term in (5) gives a penalty for complex models. The objective of structure learning is to maximize the BIC; thus, it needs to maximize the likelihood as well as minimize the penalty. In other words, the aim is to find a structure with maximum likelihood as simple as possible. However, to find a Bayesian network with maximum likelihood is an NP-complete problem. As a suggested in [14], we use the randomized hill climbing (HC) to build the graph structure M from an arcless topology with one edge addition or deletion neighborhood function.

For sampling a new individual, we need to calculate the local parameters θ_{ijk} for all $j \in S_{Pa_i}$ and $k \in S_{X_i}$ by

$$\theta_{ijk} = \mathcal{P}(X_i = X_i(k) | Pa_i = pa_i(j)) = \frac{N_{ijk} + 1}{N_{ij} + r_i}, \quad (6)$$

where S_{Pa_i} is the collection of parent instantiations of random variable X_i and S_{X_i} is the set of all possible values of X_i . An individual can thus be sampled through the PLS determined by the graph structure M and the local parameters θ_i . Algorithm 5 shows the process of EBNA.

IV. RESULTS

This study investigates the performance of EDAs on the NK landscape problems of $n \in \{10, 30\}$ and $k \in \{2, 6\}$. Table I lists the parameter setting for UMDA, BMDA and EBNA. To observe the influence of model complexity on the solution quality, we check three EBNA versions, to wit, EBNA₂, EBNA₄, EBNA₆, which restrict the degrees of in and out order in the Bayesian network to 2, 4, and 6, respectively. The maximal degree of order for the test EBNA is set to 6 since Echegoyen et al. [17] indicated, as the number of sub-function increases, the maximum degree of order is within the range from 3 to 6. To examine the behavior of each algorithm, we record the best known solution over all trials and trace the variation in the probability of generating the best solution [19]. All experiments are conducted on Intel core i7-920 machines with 30 trials.

Figure 2 shows the anytime behavior and the logarithm of probability of test EDAs on the NK landscape problems with fixed problem size $n = 10$ and different k values. The results indicate that, as k increases, the convergence speed decreases for all the test EDAs; that is, EDAs generally require more generations if the interaction between variables is intensified. Additionally, UMDA and BMDA converge faster than EBNA

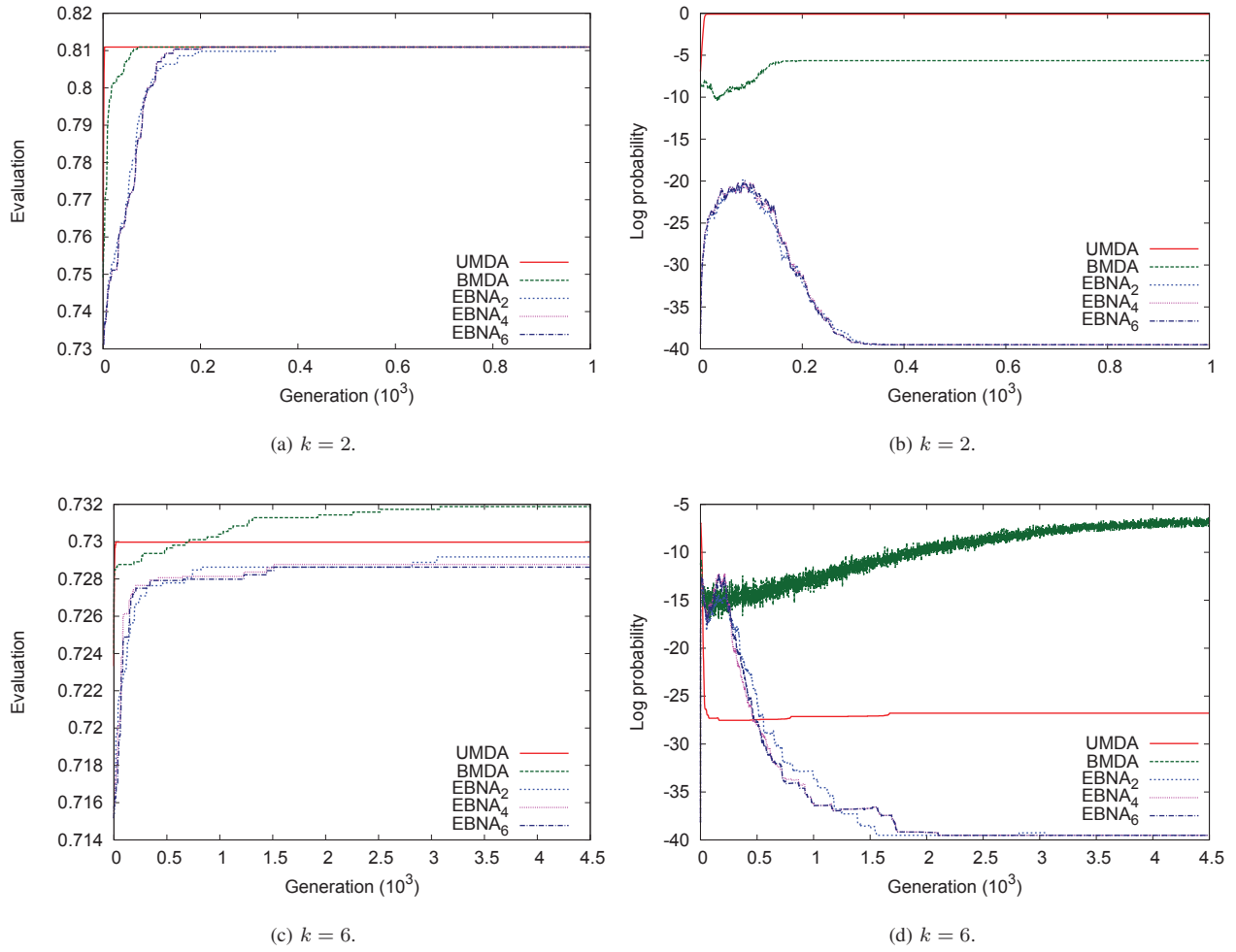


Figure 2: Anytime behavior (left column) and log probability of generating the best known solution (right column) of UMDA, BMDA, and EBNA_s on the NK landscape problems with $n = 10$.

does. The experimental results demonstrate that UMDA outperforms others in both solution quality and convergence speed for $k = 2$, while BMDA does for $k = 6$. According to the logarithm of probability, EBNA_s with different degrees of order behave similarly. Furthermore, UMDA converges to the best known solution very fast for $k = 2$, but suffers from premature convergence for $k = 6$. BMDA achieves the best solution quality and the highest probability of sampling the best known solution for $k = 6$.

This study further looks into the anytime behavior of the test EDAs on the NK landscape problems with a larger problem size $n = 30$. Since the problem size is associated with the number of sub-functions, a larger problem size brings about a harder problem. According to Fig. 3, UMDA performs well for $k = 2$ but fails to find the best known solution for $k = 6$ due to premature convergence. By contrast, BMDA converges slowly for $k = 2$ and yet outperforms others for $k = 6$. For the EBNA_s with different orders, there is no significant difference in solution quality and convergence speed. As k

increases, EBNA_s converge very slowly in comparison with the UMDA and BMDA. In view of the logarithmic probability of sampling the best known solution, the low probability of UMDA corresponds to its poor solution quality, which implies the vulnerability of UMDA to local optima. On the other hand, BMDA achieves a stable high probability and satisfactory solution quality. Compared to the stable probabilities of UMDA and BMDA, the probability for EBNA_s varies with the number of generations; in particular, the probability drastically declines after some generations.

Next, we examine the statistical significance using the Kruskal-Wallis test, a non-parametric analysis of variance suggested by García et al. [22]. All statistical analyses are under significance level $\alpha = 0.05$. Tables II and III summarize the statistical results, where the label Chi-square denotes the Chi-square statistics in the Kruskal-Wallis test and the term $\text{Pr} > \text{Chi-square}$ indicates a significance difference among all test methods if the value is smaller than α . The results exhibit that there is no significant difference among the test

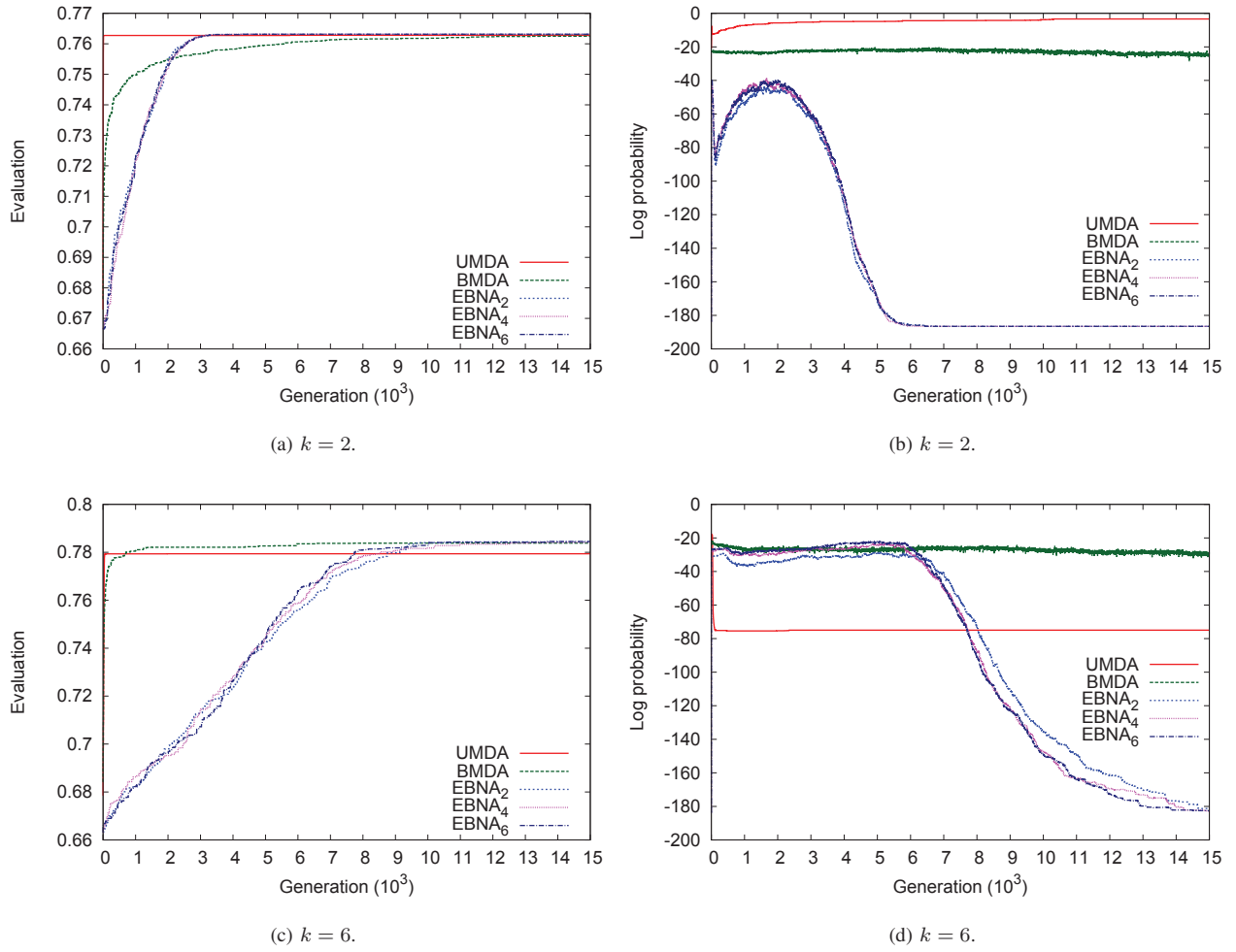


Figure 3: Anytime behavior (left column) and log probability of generating the best known solution (right column) of UMDA, BMDA, and EBNA_s on the NK landscape problems with $n = 30$.

EDAs for small n and k . However, as n or k increases, the significance emerges: BMDA outperforms other methods on the problems with large size or high connectivity (e.g. NK(10, 6) and NK(30, 6)). Similar to the results of anytime behavior, the three EBNA_s have insignificant difference in the solution quality. This outcome reveals that EBNA is insensitive to the order of degrees for the graphical structure in terms of solution quality.

According to the above results on the NK landscape problems, EDAs with low model complexity like UMDA render good optimizers for simple small problem, namely, smaller problem size and low interaction between variables. Nonetheless, for more complex problems, such EDAs may suffer from premature convergence. BMDA, on the other hand, can nicely resolve the complex problems (large size and strong interaction) in terms of solution quality and convergence speed. Surprisingly, EDAs with even higher model complexity like EBNA may not perform satisfactorily on the complex problems, as shown in our experiments. Additionally, EBNA_s

converge slower than UMDA and BMDA. In general, it shows that a simple model is not necessarily worse than a complex one for EDAs: The bivariate model outperforms the univariate and multivariate models in both solution quality and convergence speed on the NK landscape problem.

V. CONCLUSIONS

The model complexity is a key factor in the performance of EDA. Complex models can express the relations among variables more accurately than simple models. However, for some problems with strong interaction among variables, building a complex model for all the relations becomes unrealistic and impractical due to its high computational cost and requirement for large population size. In this study, we take the EDAs based on three different models, namely, univariate, bivariate and multivariate models, to observe their behaviors in terms of solution quality and convergence speed in NK landscapes. This study further tracks the variation in the logarithmic probability of generating the best known solution to examine

Table II: Kruskal-Wallis non-parametric analysis of variance with multiple comparison post-hoc analysis for the NK landscape problems with $n = 10$ and $k \in \{2, 6\}$. All methods are tested under 30 trials and the significance level $\alpha = 0.05$.

Wilcoxon Scores (Rank Sums) with Null Hypothesis $E[SS]_{H_0} = 2265.0$ and $Stdev_{H_0} = 0.0$			Kruskal-Wallis analysis with multiple comparisons (No significance difference within the same group)			
METHOD	SS	Mean Score	df	4	METHOD	Group Ranking
UMDA	2265.0	75.50	Chi-square	0.0000	UMDA	A
BMDA	2265.0	75.50	Pr > Chi-square	1.0000	BMDA	A
EBNA ₂	2265.0	75.50			EBNA ₂	A
EBNA ₄	2265.0	75.50			EBNA ₄	A
EBNA ₆	2265.0	75.50			EBNA ₆	A

(a) $k = 2$

Wilcoxon Scores (Rank Sums) with Null Hypothesis $E[SS]_{H_0} = 2265.0$ and $Stdev_{H_0} = 188.8943$			Kruskal-Wallis analysis with multiple comparisons (No significance difference within the same group)			
METHOD	SS	Mean Score	df	4	METHOD	Group Ranking
UMDA	2235.0	74.5000	Chi-square	20.4827	UMDA	B
BMDA	3093.0	103.1000	Pr > Chi-square	0.0004	BMDA	A
EBNA ₂	2090.0	69.6667			EBNA ₂	B
EBNA ₄	1988.0	66.2667			EBNA ₄	B
EBNA ₆	1919.0	63.9667			EBNA ₆	B

(b) $k = 6$

Table III: Kruskal-Wallis non-parametric analysis of variance with multiple comparison post-hoc analysis for NK landscapes problems with $n = 30$ and $k \in \{2, 6\}$. All methods are tested under 30 trials and the significance level $\alpha = 0.05$.

Wilcoxon Scores (Rank Sums) with Null Hypothesis $E[SS]_{H_0} = 2265.0$ and $Stdev_{H_0} = 87.5961$			Kruskal-Wallis analysis with multiple comparisons (No significance difference within the same group)			
METHOD	SS	Mean Score	df	4	METHOD	Group Ranking
UMDA	2104.00	70.13	Chi-square	14.6103	UMDA	A
BMDA	2021.00	67.37	Pr > Chi-square	0.0056	BMDA	A
EBNA ₂	2400.00	80.00			EBNA ₂	A
EBNA ₄	2400.00	80.00			EBNA ₄	A
EBNA ₆	2400.00	80.00			EBNA ₆	A

(a) $k = 2$

Wilcoxon Scores (Rank Sums) with Null Hypothesis $E[SS]_{H_0} = 2265.0$ and $Stdev_{H_0} = 204.7660$			Kruskal-Wallis analysis with multiple comparisons (No significance difference within the same group)			
METHOD	SS	Mean Score	df	4	METHOD	Group Ranking
UMDA	1046.50	34.8833	Chi-square	37.2876	UMDA	B
BMDA	2685.00	89.5000	Pr > Chi-square	<0.0001	BMDA	A
EBNA ₂	2305.50	76.8500			EBNA ₂	A
EBNA ₄	2690.00	89.6667			EBNA ₄	A
EBNA ₆	2598.00	86.6000			EBNA ₆	A

(b) $k = 6$

the convergence of the probability model. For comparing the solution quality among EDAs, we adopt the Kruskal-Wallis non-parametric analysis of variance to evaluate the statistical significance.

On the NK landscapes problem with low interaction and small problem size, we found that UMDA can perform well in solution quality and convergence speed. As the problem complexity increases, UMDA suffers from premature convergence and gets stuck on local optima. Meanwhile, BMDA and EBNA can solve more complex problems; in particular, the former can converge much faster than the latter. Additionally, BMDA outperforms other methods on the problems with large size or high connectivity. In general, the study reveals that a simple model is not necessarily worse than a complex one for EDAs on the NK landscape problem. Furthermore, we adopted different degrees of order in the Bayesian network to observe the influence of model complexity. The results show that the three EBNA have insignificant difference in the solution quality. That is, EBNA are insensitive to the order of graphical model. This outcome infers that a simple graphical model is good enough for EBNA to tackle the NK landscape problem.

Some directions remain for future work. First, other problems or larger instance size should be further considered in performance comparison. Second, our study compares three EDAs: UMDA, BMDA and EBNA. More EDAs and other models can be applied in the comparison. In addition, our work suggests that a simple model is not necessarily worse than a complex one. Given that efficiency is an important issue in the design of algorithms, combination of simple models is a promise alternative to a single complex model used in EDA.

ACKNOWLEDGMENT

This study was supported by the Ministry of Education, Taiwan.

REFERENCES

- [1] J. Holland, *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- [2] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.
- [3] J. Ceberio, E. Irurozki, A. Mendiburu, and J. A. Lozano, "A review on estimation of distribution algorithms in permutation-based combinatorial optimization problems," *Progress in Artificial Intelligence*, vol. 1, pp. 103–117, 2012.
- [4] M. Hauschild and M. Pelikan, "An introduction and survey of estimation of distribution algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 3, pp. 111–128, 2011.
- [5] P. Larrañaga and J. A. Lozano, *Estimation of distribution algorithms: A new tool for evolutionary computation*. Boston / Dordrecht / London: Kluwer Academic Publishers, 2002.

- [6] S. Baluja, "Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning," Pittsburgh, PA: Carnegie Mellon University, Tech. Rep. CMU-CS-94-163, 1994.
- [7] G. R. Harik, F. G. Lobo, and D. E. Goldberg, "The compact genetic algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 287–297, 1999.
- [8] H. Mühlenbein and G. Paaß, "From recombination of genes to the estimation of distributions I. binary parameters," in *Parallel Problem Solving from Nature - PPSN IV*, ser. Lecture Notes in Computer Science, H. Voigt, W. Ebeling, I. Rechenberg, and H. Schwefel, Eds. Berlin, Germany: Springer-Verlag, 1996, vol. 1141, pp. 178–187.
- [9] J. S. De Bonet, C. L. Isbell, and P. Viola, "Mimic: Finding optima by estimating probability densities," *Advances in neural information processing systems*, vol. 9, pp. 424–430, 1997.
- [10] S. Baluja and S. Davies, "Using optimal dependency-trees for combinatorial optimization: learning the structure of the search space," in *Proceedings of the 14th international conference on machine learning*, 1997, pp. 30–38.
- [11] M. Pelikan and H. Mühlenbein, "The bivariate marginal distribution algorithm," in *Advances in Soft Computing - Engineering Design and Manufacturing*, R. Roy, T. Furuhashi, and P. K. Chawdhry, Eds. Berlin, Germany: Springer-Verlag, 1999, pp. 521–535.
- [12] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz, "BOA: the Bayesian optimization algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference*, vol. 1, 1999, pp. 525–532.
- [13] M. Pelikan, "Hierarchical Bayesian optimization algorithm," in *Hierarchical Bayesian Optimization Algorithm*, ser. Studies in Fuzziness and Soft Computing. Berlin, Germany: Springer-Verlag, 2005, vol. 170, pp. 105–129.
- [14] R. Etxeberria and P. Larrañaga, "Global optimization using Bayesian networks," in *Proceedings of the Second Symposium on Artificial Intelligence*, A. Ochoa, M. R. Soto, and R. Santana, Eds., 1999, pp. 151–173.
- [15] P. Larrañaga, R. Etxeberria, J. A. Lozano, and J. M. Peña, "Optimization in continuous domains by learning and simulation of Gaussian networks," in *Workshop Proceedings of the Genetic and Evolutionary Computation Conference*, 2000, pp. 201–204.
- [16] S. A. Kauffman and E. D. Weinberger, "The NK model of rugged fitness landscapes and its application to maturation of the immune response," *Journal of Theoretical Biology*, vol. 141, no. 2, pp. 211–245, 1989.
- [17] C. Echegoyen, Q. Zhang, A. Mendiburu, R. Santana, and J. A. Lozano, "On the limits of effectiveness in estimation of distribution algorithms," in *Proceedings of IEEE Congress on Evolutionary Computation*, 2011, pp. 1573–1580.
- [18] C. Echegoyen, A. Mendiburu, R. Santana, and J. Lozano, "Analyzing the probability of the optimum in EDAs based on Bayesian networks," in *Proceedings of IEEE Congress on Evolutionary Computation*, 2009, pp. 1652–1659.
- [19] C. Echegoyen, A. Mendiburu, R. Santana, and J. A. Lozano, "Toward understanding EDAs based on Bayesian networks through a quantitative analysis," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 99, pp. 173–189, 2011.
- [20] C. F. Lima, G. F. Lobo, M. Pelikan, and D. E. Goldberg, "Model accuracy in the Bayesian optimization algorithm," *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, vol. 15, no. 7, pp. 1351–1371, 2011.
- [21] M. Henrion, "Propagating uncertainty in Bayesian networks by probabilistic logic sampling," in *Proceedings of Uncertainty in Artificial Intelligence 2 Annual Conference on Uncertainty in Artificial Intelligence (UAI-86)*, Amsterdam, NL, 1986, pp. 149–163.
- [22] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization," *Journal of Heuristics*, vol. 15, no. 6, pp. 617–644, 2009.