

On the Effectiveness of Distributions Estimated by Probabilistic Model Building

Chung-Yao Chuang
Ying-ping Chen

NCLab Report No. NCL-TR-2008001
February 2008

Natural Computing Laboratory (NCLab)
Department of Computer Science
National Chiao Tung University
329 Engineering Building C
1001 Ta Hsueh Road
HsinChu City 300, TAIWAN
<http://nclab.tw/>

On the Effectiveness of Distributions Estimated by Probabilistic Model Building

Chung-Yao Chuang and Ying-ping Chen
Department of Computer Science
National Chiao Tung University
HsinChu City 300, Taiwan
{cychuang, ypchen}@nclab.tw

February 02, 2008

Abstract

Estimation of distribution algorithms (EDAs) are a class of evolutionary algorithms that capture the likely structure of promising solutions by explicitly building a probabilistic model and utilize the built model to guide the further search. It is presumed that EDAs can detect the structure of the problem by recognizing the regularities of the promising solutions. However, in certain situations, EDAs are unable to discover the entire structure of the problem because the set of promising solutions on which the model is built contains insufficient information for some parts of the problem and renders EDAs incapable of accurate model building. In this work, we firstly propose a general concept that the effectiveness of probabilistic models should be evaluated and verified in EDAs. Based on the concept, we design a practical approach which utilizes a reserved set of individuals to inspect the built model for the fragments that may be inconsistent with the actual problem structure. Furthermore, we provide an implementation of the designed approach on the extended compact genetic algorithm (ECGA) and conduct numerical experiments. The results indicate that the proposed concept can significantly assist ECGA to handle problems of different scalings.

1 Introduction

Genetic algorithms (GAs) are search methods based on the paradigm of natural evolution, in which, species of creatures tend to adapt to their living environments by mutation and inheritance of useful traits. Genetic algorithms mimic this mechanism by introducing artificial selections and operators to identify and recombine partial solutions. By properly growing and mixing good partial solutions, which are often referred to as building blocks (BBs), GAs are capable of solving problems efficiently. The ability to implicitly process a large number of partial solutions has been recognized as an important computational power source. According to the Schema theorem [1], short, low-order, and highly fit substrings increase their share to be combined, and also as stated in the building block hypothesis [2], GAs implicitly decompose a problem into sub-problems by processing building blocks. The decompositional bias is a good strategy for tackling many real-world problems, because many real-world problems can be reliably solved by combining the pieces of promising solutions in the form of problem decomposition.

However, proper growth and mixing of building blocks are not always achieved. GA in its simplest form employing fixed representations and problem-independent recombination operators often breaks the promising partial solutions while performing crossovers. Such a situation leads to the vanishing of crucial building blocks and thus the converging to local optima. In order to overcome the building block disruption problem, various techniques have been proposed. In

this study, we focus on one line of such efforts which are often called the estimation of distribution algorithms (EDAs) [3]. These methods construct probabilistic models of promising solutions and utilize the built models to generate new solutions. Early EDAs, such as the population-based incremental learning (PBIL) [4] and the compact genetic algorithm (cGA) [5], assume no interaction between variables. Subsequent studies start from capturing pairwise interactions, such as mutual-information-maximizing input clustering (MIMIC) [6], Baluja’s dependency tree approach [7], and the bivariate marginal distribution algorithm (BMDA) [8], to modeling multivariate interactions, such as the extended compact genetic algorithm (ECGA) [9], the Bayesian optimization algorithm (BOA) [10], the estimation of Bayesian network algorithm (EBNA) [11], the factorized distribution algorithm (FDA) [12], and the learning version of FDA (LFDA) [13]. With the reasoning of dependencies among variables by building probabilistic models, these approaches can capture the problem structure and thus avoid the disruption of partial solutions.

While EDAs can provide good performance on problem optimization, most of the studies focus on the problems composed of subproblems of equal importance. For the real-world applications, it is often the case that some parts of the problem are more important than other parts. This situation poses two types of difficulties. Firstly, because the population processing is statistical in nature, the disparate scaling can cause inaccurate processing of less salient building blocks [14, 15]. Moreover, because the low salience oftentimes causes the given building block to be processed at a later time compared to those of higher salience, the delay on the timeline may result in the allele drifting. Some other previous studies on this topic include the explicit role of scale in a systematic experimental setting [16], a theoretical model on convergence behavior of exponentially scaled problems [17], and an extension to larger building blocks [18]. In this study, we make a further attempt to enhance the flexibility and applicability of EDAs such that EDAs can perform well on problems of different scalings. Particularly, we propose a general concept to evaluate and verify the effectiveness of built probabilistic models, design a practical approach to inspect the built model for inconsistent parts, and implement the proposed approach on the extended compact genetic algorithm.

In the next section, we will explore the interaction between the scaling difficulties and the probabilistic model building. More specifically, we will take a look at how scaling difficulties shadow the ability of EDAs to recognize building blocks and propose the concept and approach to resolve such a situation. In section 3, an implementation of the proposed concept on the extended compact genetic algorithm is described in detail. Section 4 presents the empirical results, followed by the discussion and observations on the results in Section 5. Finally, section 6 concludes this paper.

2 Effective Distributions

The primary ability of EDAs in dealing with the building block disruption problem comes from the explicitly modeling of promising solutions by building probabilistic models. The model construction algorithms, though differ in their representative power, capture the likely structures of good solutions by processing the population-wise statistics collected from the selected solutions. By reasoning the dependencies among different parts of the problem and the possible formations of good solutions, reliable mixing and growing of building blocks can be achieved. As noted in [9], learning a good probability distribution is equivalent to learning linkage, where linkage refers to the dependencies among variables or equivalently the decomposition of the problem.

It is presumed that EDAs can detect linkage by recognizing building blocks. However, in this study, we argue that in some cases, accurate and complete linkage information can not be acquired by distribution estimation because the selected set of solutions on which the model is built contains insufficient information on the less salient parts of the problem. For example,

Generation	Marginal Product Model	Effective Partial Model
1	[1 2 3 4] [6 11 14] [5 8 12] [7 9 13] [10 15 16]	[1 2 3 4]
2	[1] [2] [3] [4] [5 6 7 8] [9 12 13] [10 15 16] [11] [14]	[5 6 7 8]
3	[1] [2] [3] [4] [5] [6] [7] [8] [9 10 11 12] [13 15 16] [14]	[9 10 11 12]
4	[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13 14 15 16]	[13 14 15 16]

Table 1: Marginal product models built by ECGA in solving an exponentially scaled problem. The variables are denoted by their index numbers. Each group of variables represents a model in which a marginal distribution resides. The variables with converged alleles are crossed out.

consider a k -bit trap function,

$$f_{trap_k}(s_1 s_2 \cdots s_k) = trap_k(u), \text{ where } u = \sum_{i=1}^k s_i$$

$$= \begin{cases} k, & \text{if } u = k, \\ k - 1 - u, & \text{otherwise.} \end{cases},$$

where u is the number of ones in the string $s_1 s_2 \cdots s_k$. Suppose that we are handling a 16-bit maximization problem,

$$f(s_1 s_2 \cdots s_{16}) = \sum_{i=0}^3 (10^{3-i} f_{trap_4}(s_{4i+1} s_{4i+2} s_{4i+3} s_{4i+4})),$$

where $s_1 s_2 \cdots s_{16}$ is a solution string. We choose ECGA [9], which uses a class of multivariate probabilistic models called marginal product models (MPMs), to tackle this problem. By observing subsequent generations of the optimization process, a series of models built by ECGA can be shown in Table 1. The variables are denoted by their index numbers. Each group of variables represents a marginal model in which a marginal distribution resides. The variables with converged alleles are crossed out.

It can be observed that the models shown in Table 1 are only partially correct. More specifically, at each generation, only the most salient building block of which the alleles have not converged is modeled correctly. The phenomenon is caused by the fact that some parts of the problem contribute much more than all others in combine do. If a part of the problem is significantly more important than others, this part of the solution solely determines the chance whether or not the solution will be selected. As a consequence, only the most salient building block can provide sufficient information to be modeled correctly since the model searching is performed based on the selected solutions. The rest parts of the model are simply the result of less salient partial solutions “hitchhiking” on those more salient building blocks.

From the example, we can see that not all building blocks can be detected from a given set of selected solutions by probabilistic model building. Model building algorithms cannot “see” the entire structure of the problem from the selected set of solutions when disparate scalings among different building blocks prevents the complete linkage information from being included in the selected individuals. In this work, we will refer this concept as *linkage sensibility*, and those problem structures that can be identified properly using the given set of solutions are called *sensible linkage*. Based on the notion, we can re-examine EDAs on the building block disruption problem. It is clear that building block disruption still exists in the insensible portion because that part of the problem cannot be modeled properly. Although the example in Table 1 is an extreme case that each subproblem is exponentially scaled, in real-world problems, it is often the case that the subproblems are weighted differently and the linkage might be partially sensible. In addition to building block disruption, the random drifting of the less salient parts of the problem

makes the situation worse. Whereas the problems with these difficulties are often handled with EDAs by increasing population sizes, in this study, we propose the idea that it is possible to distinguish sensible linkage from insensible linkage instead of using larger populations.

The idea of sensible linkage can be closely mapped into another notion called *effective distributions*. By effective distributions, we mean that the solution quality can be reliably advanced by sampling the distributions. The essential condition for effective distributions is the consistency with building blocks. If it is possible to extract effective distributions from the entire built probabilistic model, we can perform partial sampling based on only the effective distributions and leave the rest parts of the solutions unchanged. As a result, the diversity is maintained, and we are free from building block disruption and random drifting. For the aforementioned example of the 16-bit problem, if it is possible to identify the partial models which are really built on the sensible linkage, such as [1 2 3 4] at the first generation and [5 6 7 8] at the second generation (Table 1), we can sample only the corresponding, effect marginal distributions. More specifically, at the first generation, for each solution string, we re-sample only $s_1s_2s_3s_4$ according to the marginal distribution and keep $s_5s_6 \cdots s_{16}$ unchanged. At the second generation, we re-sample only $s_5s_6s_7s_8$ according to the marginal distribution and keep $s_9s_{10} \cdots s_{16}$ unchanged, where $s_1s_2s_3s_4$ are converged. In this way, the EDA users do not have to resort to increasing population sizes for the difficulties caused by disparate scalings.

The thought leaves us one complication: the identification of effective distributions. Direct identification of effective distributions may not be an easy task if not impossible. If there exists a good way to identify effective distributions, we can just employ it in the model building process, and all the difficulties disappear. Therefore, it may be wise to adopt the complementary approach—to identify the distributions that are *not* likely to be effective. If we can identify the ineffective distributions, we can bypass them and sample only the other distributions to approximate the effect of identifying effective distributions. The basic idea is that we split the entire population into two populations and use only one population for building the probabilistic model. We use the other population to collect statistics for possible indications of ineffectiveness of the partial distributions in the probabilistic model built upon the first population. With certain appropriate criteria, we can prune the likely ineffective portions of the model.

In the next section, we provide a reference implementation of the proposed mechanism on ECGA with a judging criterion for deciding whether or not each distribution in a given marginal product model is ineffective.

3 Model Pruning for ECGA

This section starts by reviewing the essential part of the extended compact genetic algorithm (ECGA). Based on the proposed mechanism of detecting the inconsistency of statistics gathered from the two populations, a technique is devised to identify the ineffective parts of the probabilistic model. Finally, an optimization algorithm incorporating the proposed technique is described in detail.

3.1 Extended Compact Genetic Algorithm

ECGA [9] uses a product of marginal distributions on a partition of variables. This type of distributions belongs to a class of probabilistic models known as marginal product models (MPMs). In such a model, subsets of variables are modeled jointly, and each subset is considered independent of others. In this work, the conventional notation is adopted that variable subsets are enclosed in brackets.

In ECGA, both the structure and the parameters of the model are searched and optimized using a greedy approach to fit the statistics of the selected set of promising solutions. The

measure of a good MPM is quantified based on the minimum description length (MDL) principle [19], which assumes that given all things are equal, simpler distributions are better than complex ones. The MDL principle penalizes inaccurate as well as complex models, thereby, leading to a near-optimal distribution. Specifically, the measure is the complexity of the MPM which is quantified as the sum of the model complexity, C_m , and the compressed population complexity, C_p .

The model complexity, C_m , quantifies the model representation in terms of the number of bits required to store all the marginal distributions. Suppose that the given problem is of length ℓ with binary coding, and the variables are partitioned into m subsets with each of size k_i , $i = 1 \dots m$, such that $\ell = \sum_{i=1}^m k_i$. The marginal distribution corresponding to the i th variable subset requires $2^{k_i} - 1$ frequency counts to be completely specified. Taking into account that each frequency count is of length $\log_2(n + 1)$ bits, where n is the population size, the model complexity can be given as

$$C_m = \log_2(n + 1) \sum_{i=1}^m (2^{k_i} - 1) .$$

The compressed population complexity, C_p , quantifies the suitability of the model in terms of the number of bits required to store the entire selected population (the set of promising solutions selected by the selection operator) with an ideal compression scheme applied. The compression scheme is based on the partition of the variables. Each subset of the variables specifies an independent “compression block” on which the corresponding partial solutions are compressed. Theoretically, the optimal compression method encodes a message of probability p_i using $-\log_2 p_i$ bits. Thus, taking into account all possible messages, the expected length of a compressed message is $\sum_i -p_i \log_2 p_i$ bits, which is optimal. In the information theory [20], the quantity $-\log_2 p_i$ is called the *information* of that message and $\sum_i -p_i \log_2 p_i$ is called the *entropy* of the distribution. Based on the definition, the compressed population complexity can be derived as

$$C_p = \log_2 n \sum_{i=1}^m \sum_{j=1}^{2^{k_i}} -p_{ij} \log_2 p_{ij} ,$$

where p_{ij} is the frequency of the j th partial solution to the i th variable subset observed in the selected population.

In the calculation of C_p , it is assumed that the j th possible partial solution to the i th variable subset is encoded by using $-\log_2 p_{ij}$ bits. This assumption is fundamental to our proposed technique to identify the likely ineffective distributions built by ECGA. More precisely, the information of the partial solutions, $-\log_2 p_{ij}$, is a good indicator of inconsistency of the statistics gathered from the two populations.

3.2 Model Pruning

We propose a technique for ECGA to identify the possibly ineffective parts of a marginal product model based on the notion that ECGA uses the compression performance to quantify the suitability of a probabilistic model for the given set of solutions. The degree of compression can be considered as a representative metric for the fitness of modeling, because all good compression methods are based on capturing and utilizing the relationships among sampled data. Thus, if the compression scheme of the MPM built on one set of solutions is incapable of compressing another set of solutions sampled in the same condition, it is likely that the MPM is partially incorrect. With this property, we can systematically check a given MPM for ineffective portions.

Suppose that the population of solutions, P , is split into two populations S and T . The model building process is performed on S' , the set of promising solutions selected from S , and

builds the model M . We use the statistics collected from T' , the set of solutions selected from T , to examine the built probabilistic model, M . Since marginal models function independently, they can be inspected separately. Recall that a variable subset, which specifies a marginal model, is viewed as a “compression block” that encodes each possible partial solution according to the distribution. The j th possible partial solution to the i th variable subset is encoded using $-\log_2 p_{ij}$ bits, where p_{ij} is the frequency of the j th possible partial solution to the i th variable subset observed in S' . Assume that the given problem is of length ℓ with binary coding, and there are m variable subsets with each of size k_i , $i = 1 \dots m$, in the built model M . For the i th marginal model, $i = 1 \dots m$, we can check whether or not

$$\sum_{j=1}^{2^{k_i}} (-\log_2 p_{ij}) q_{ij} > k_i, \quad (1)$$

where q_{ij} is the frequency of the j th possible partial solution to the i th variable subset collected from T' . If Equation (1) holds, the compression scheme employed in the i th marginal model is considered inappropriate for compressing the corresponding partial solutions in T' , because k_i -bit partial solutions are encoded with strings of expected lengths more than k_i bits. Such a condition indicates that the model is probably ineffective because T' does not agree on this part of the built model. Otherwise, it should be able to compress the partial solutions in T' .

From the machine learning perspective [21], a good model should generalize well to unseen instances. Otherwise, it captures coincidental regularities among the training data. If the model building is performed on the portion where linkage is not sensible from the given set of solutions, it will “overfit” these partial solutions which are not subject to the proper selection pressure. Consequently, the regularities captured by this part of the model tend to be inconsistent with the actual problem structure. Furthermore, the partial solutions which are not subject to the proper selection pressure appear to be random with a high probability, and it leads to the phenomena of random drifting mentioned in section 1. Drifting is random by nature, and two different populations tend to drift in different directions. Thus, we can use the statistical inconsistency between S' and T' to recognize the drifting portions of the solutions and identify the probably ineffective parts of the model. By removing these ineffective parts, we can forge an effective partial model.

An issue in practice of the Equation (1) calculation is that there might be possible partial solutions which do not exist in the set of selected solutions, and $-\log_2 p_{ij}$ is left undefined because $p_{ij} = 0$. When such a situation occurs, p_{ij} 's of which the values are zero are assigned a constant less than $1/n$, and all p_{ij} 's are normalized such that the sum is 1.

3.3 ECGA with Model Pruning

In this section, the optimization procedure incorporating ECGA and the proposed technique is described. This integration assists ECGA to achieve better performance when disparate scalings exist in the problem.

The procedure is presented in Algorithm 1 and starts at initializing a population of solutions. After initialization, the fitness values of solutions are evaluated, and the entire population is randomly split into two populations. Selection is performed on the two populations separately with the same selection pressure. Model building is performed on the first population. The other population is used to prune the built model by utilizing the proposed technique. Finally, all the solutions in the population are altered by sampling the remaining marginal distributions. These steps are repeated until the stopping criteria are met.

A prominent difference between the procedure in Algorithm 1 and common EDAs is that the sampling may not include all decision variables. The solutions are altered by sampling only on

Algorithm 1 ECGA with Model Pruning

Initialize a population P with n individuals of length ℓ .
while the stopping criteria are not met **do**
 Evaluate individuals in P .
 Split P into S and T at random.
 $S' \leftarrow$ apply t -wise tournament selection on S .
 $T' \leftarrow$ apply t -wise tournament selection on T .
 $M \leftarrow$ build the MPM model on S' .
 $M' \leftarrow$ prune M based on the inconsistency with T' .
 for each marginal distribution D in M' **do**
 for each individual $\mathbf{s} = s_1 s_2 \cdots s_\ell$ in P **do**
 Change the alleles in \mathbf{s} partially by sampling D .
 end for
 end for
end while

the marginal distributions survived pruning. Thus, a solution string may not be modified entirely in an iteration. The technique hence assists the EDAs to avoid random drifting and inaccurate processing of less salient building blocks by postponing the process until the linkage becomes sensible. In this way, better performance can be achieved in terms of function evaluations.

4 Experiments

The experiments are designed for observing the behavior of the proposed technique on problem sets with different scalings. Furthermore, different selection pressures are taken into considerations to make more thorough observations. In this study, three scaling models [22] are considered: exponential, power-law, and uniform, and three sets of test functions, Equations (2), (3), and (4), are constructed with f_{trap_4} as elementary functions. For simplicity, the splitting of population is performed in the way that the two resulting populations are disjoint and equally large. The stopping criterion is set such that a run is terminated when all solutions in the population converge to the same fitness value.

$$\text{Exponential: } \sum_{i=0}^{m-1} 5^i f_{trap_4}(s_{4i+1} s_{4i+2} \cdots s_{4i+4}) \quad (2)$$

$$\text{Power-law: } \sum_{i=0}^{m-1} (i+1)^3 f_{trap_4}(s_{4i+1} s_{4i+2} \cdots s_{4i+4}) \quad (3)$$

$$\text{Uniform: } \sum_{i=0}^{m-1} f_{trap_4}(s_{4i+1} s_{4i+2} \cdots s_{4i+4}) \quad (4)$$

4.1 Impact on Population Requirements

This section describes the experimental settings and results of the proposed method, ECGA with model pruning, compared to that of the original ECGA. The problem size ranges from 40 to 80 bits ($m = 10 \dots 20$) with different scaling difficulties. For each problem instance, the required minimum population size is determined by a bisection method such that $m - 1$ building blocks converge to the correct values on average in 50 runs. Two selection pressures are adopted by setting tournament size t as 8 and 16.

The empirical results on exponentially scaled problems are shown in Figure 1. The minimum population sizes required by the proposed method are lower than that needed by the original ECGA. Furthermore, with a higher selection pressure, the population sizes needed by the proposed method grows very slowly. The same situation is observed in the function evaluations that the proposed method works remarkably well when $t = 16$.

Figure 2 shows the results on power-law scaled problems. The results on the required population size are similar to that of the previous experiment set. The proposed method uses fewer function evaluations, but the difference reduces.

The empirical results on uniformly scaled problems are presented in Figure 3. As expected, the proposed method requires larger population sizes than that needed by the original ECGA. The function evaluations used by the proposed method are about twice as many as that spent by the original ECGA under the same selection pressure.

A common phenomena appears in all the three experiment sets that the proposed method needs more generations than the original ECGA does under the same selection pressure. In the next section, we will further explore this phenomena using sets of experiments that augment the population size.

4.2 Insensitivity of Population Sizes

This section describes the experiment sets that reveal the behavior of the proposed method when the population size is increased and presents the results to illustrate the interaction between population sizes and generations for the proposed method. In these experiments, the 60-bit problems ($m = 15$) are adopted, and the population sizes are augmented proportionally to the minimum population sizes.

As presented in Figure 4, only slight decreases in generations are achieved by increasing population sizes on the exponentially scaled problems. The proposed method with $t = 16$ gives the most reduction. With no prominent reductions in generations, the function evaluations grow up as expected in all four settings.

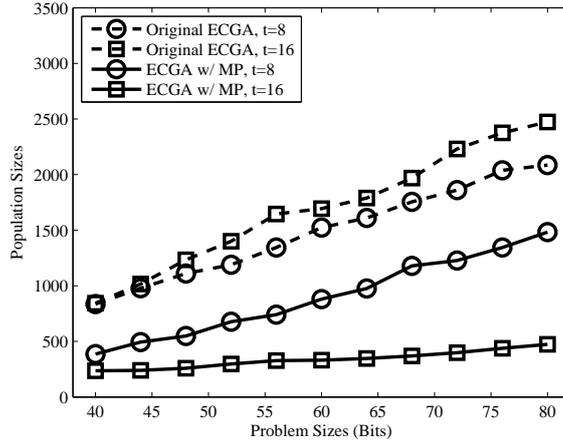
Figure 5 shows the results on the power-law scaled problems. In this case, obvious reductions in generations are observed in the proposed method. However, despite the presence of generation reduction, the function evaluations still grow up with the increasing population size.

The most significant generation reduction is observed on the uniformly scaled problems as presented in Figure 6. With tournament size 16, the proposed method needs only about 50% of generations when using a double sized population. It can be observed that nearly constant function evaluations are needed in this case as long as the population size is sufficiently large. This phenomenon might be further investigated to relieve the burden of users for setting appropriate population sizes.

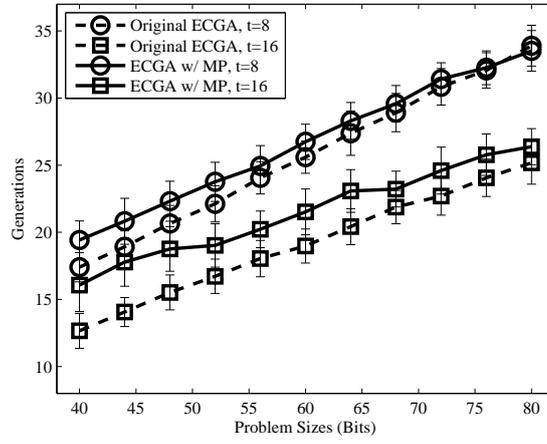
5 Discussion

The proposed method improves ECGA on problems with disparate scalings among building blocks. As illustrated in Figures 1(c) and 2(c), prominent reductions in function evaluations are achieved. Moreover, for the uniformly scaled problems where the linkage sets are completely sensible, it seems that the proposed method uses just nearly twice as many function evaluations as the original ECGA.

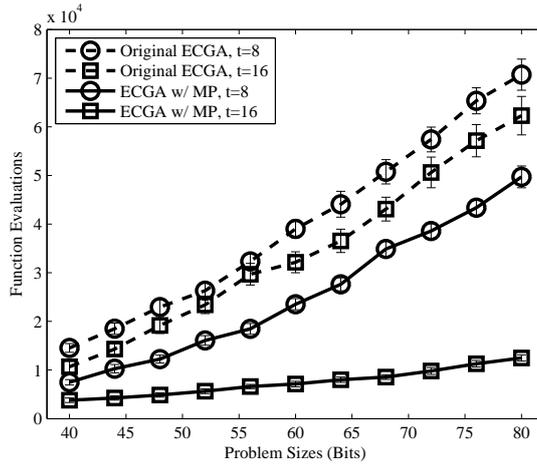
An extraordinary behavior of the proposed method can be observed that when a confined population size is given, it tends to perform a time-space trading in which more generations are spent on overcoming the problem. The most noticeable case is the uniformly scaled problem shown in Figure 6. In this case, the proposed method with appropriate selection pressure reduces



(a) Population Sizes

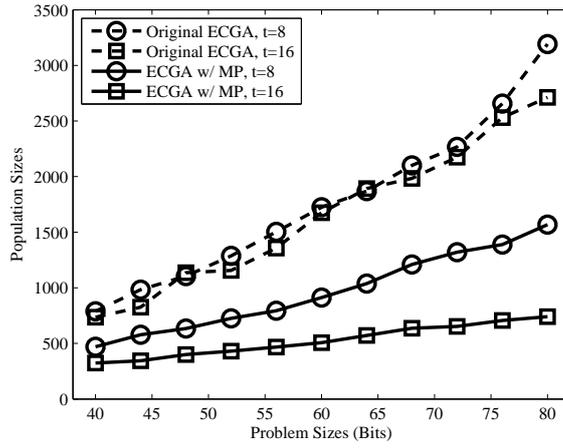


(b) Generations

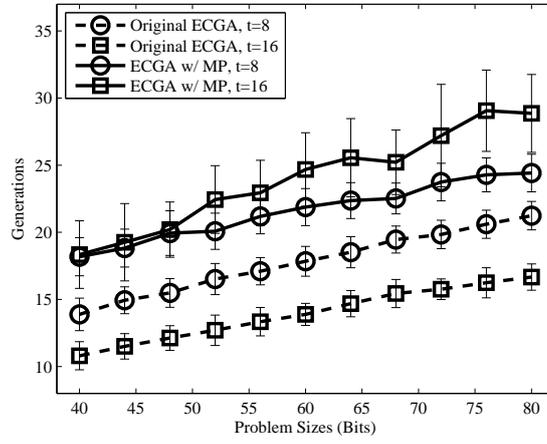


(c) Function Evaluations

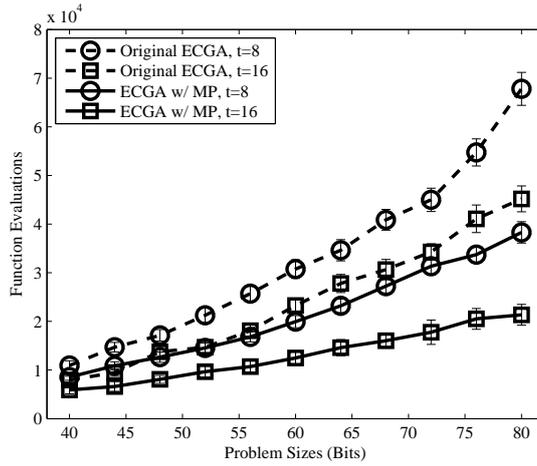
Figure 1: Empirical results of the proposed method compared to the original ECGA on *exponentially scaled problems*. Two tournament sizes $t = 8$ and $t = 16$ are adopted to observe the behavior under different selection pressures.



(a) Population Sizes

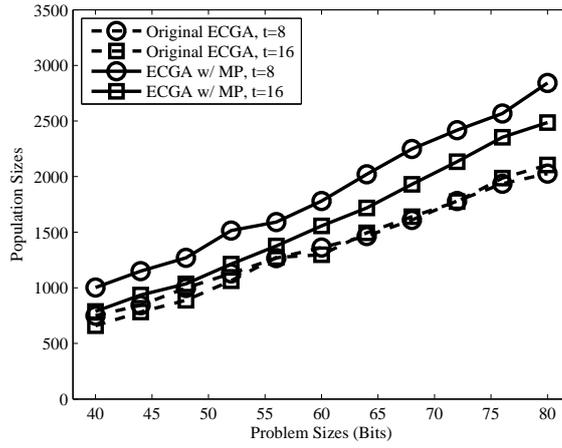


(b) Generations

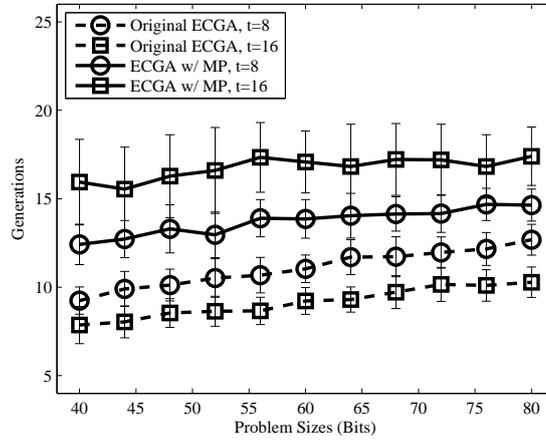


(c) Function Evaluations

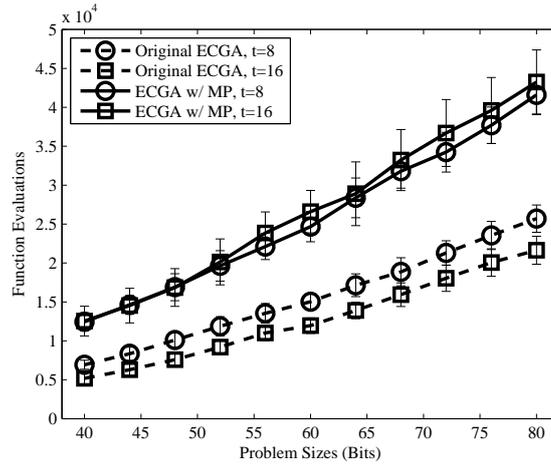
Figure 2: Empirical results of the proposed method compared to the original ECGA on *power-law scaled problems*. Two tournament sizes $t = 8$ and $t = 16$ are adopted to observe the behavior under different selection pressures.



(a) Population Sizes

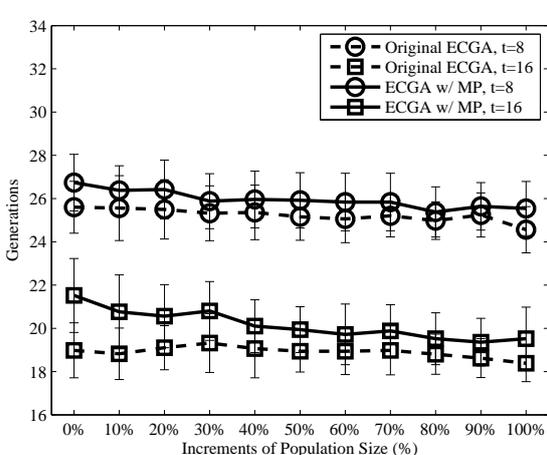


(b) Generations

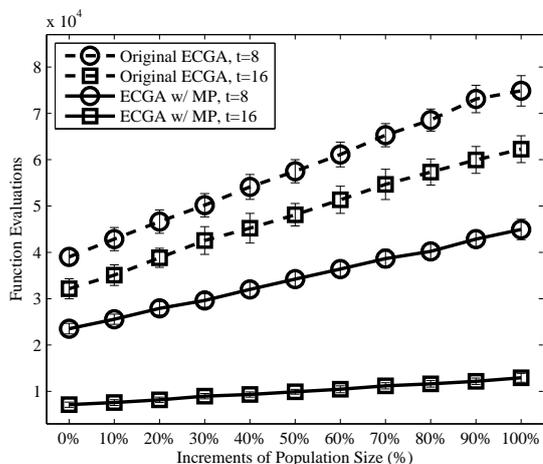


(c) Function Evaluations

Figure 3: Empirical results of the proposed method compared to the original ECGA on *uniformly scaled problems*. Two tournament sizes $t = 8$ and $t = 16$ are adopted to observe the behavior under different selection pressures.

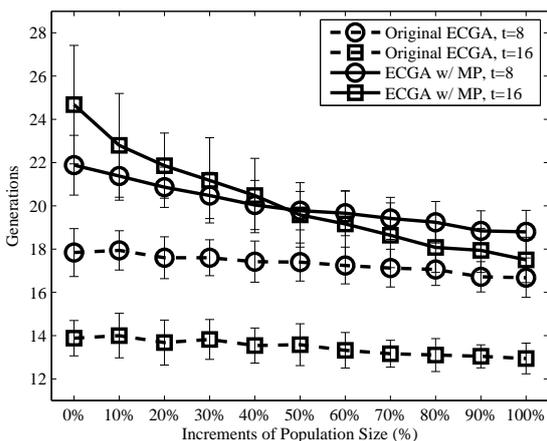


(a) Generations

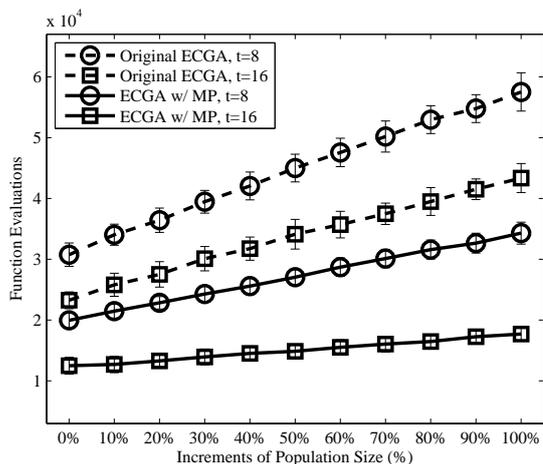


(b) Function Evaluations

Figure 4: Results of increasing population size in solving the *exponentially scaled 60-bit problem*. The population sizes are increased proportionally to the minimum required population sizes.



(a) Generations



(b) Function Evaluations

Figure 5: Results of increasing population size in solving the *power-law scaled 60-bit problem*. The population sizes are increased proportionally to the minimum required population sizes.

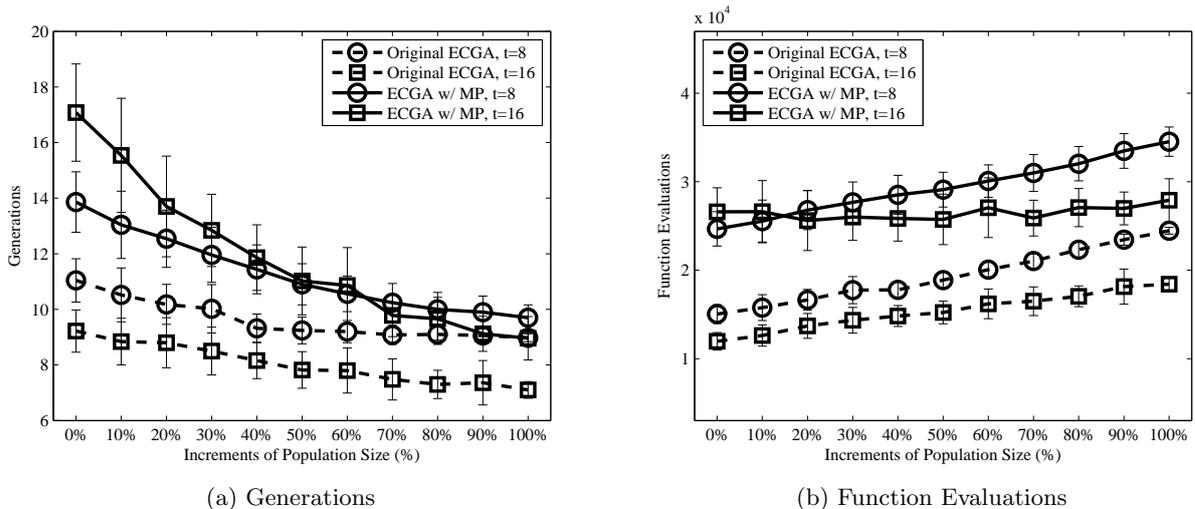


Figure 6: Results of increasing population size in solving the *uniformly scaled 60-bit problem*. The population sizes are increased proportionally to the minimum required population sizes.

the generations aggressively when a larger population size is available and keeps the growth of function evaluations from increasing.

6 Summary and Conclusions

This paper started at reviewing previous studies on EDAs and scaling difficulties and then illustrated how scaling difficulties shadows the ability of EDAs in recognizing building blocks. Based on the proposed concepts of *linkage sensibility* and *sensible linkage*, the effectiveness of distributions estimated by probabilistic model building was defined and a general idea to achieve a more effective modeling was proposed. Finally, an implementation of the proposed technique on ECGA was provided and experimented on the test functions with different scaling difficulties.

EDAs are undoubtedly promising optimization techniques in evolutionary computation. Whereas most studies focus on adopting different probabilistic models, tuning algorithmic parameters, building accurate models on the given population, and applying EDAs on optimization problems, in this study, we attempt to revisit the way EDAs operate and to resolve an important issue which is rarely addressed: what if the information contained in the given population is inevitably insufficient? The approach to solve this problem was proposed and successfully implemented for ECGA. It may be adapted and carried over to other EDAs such that more flexible and robust EDAs can be developed.

Acknowledgments

The work was supported in part by the National Science Council of Taiwan under Grant NSC-96-2221-E-009-196 and Grant NSC-96-2627-B-009-001 as well as by the ATU Plan (Aiming for the Top University and Elite Research Center Development Plan) of the National Chiao Tung University and Ministry of Education, Taiwan. The authors are grateful to the National Center for High-performance Computing for computer time and facilities.

References

- [1] J. H. Holland, *Adaptation in natural and artificial systems*. Cambridge, MA, USA: MIT Press, 1992.
- [2] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.
- [3] H. Mühlenbein and G. Paaß, “From recombination of genes to the estimation of distributions I. binary parameters,” in *PPSN IV: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*. London, UK: Springer-Verlag, 1996, pp. 178–187.
- [4] S. Baluja, “Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning,” Pittsburgh, PA, USA, Tech. Rep., 1994.
- [5] G. R. Harik, F. G. Lobo, and D. E. Goldberg, “The compact genetic algorithm,” *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, p. 287, November 1999.
- [6] J. de Bonet, C. Isbell, and P. Viola, “MIMIC: Finding optima by estimating probability densities,” in *Advances in Neural Information Processing Systems*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds., vol. 9. The MIT Press, 1997, pp. 424–430.
- [7] S. Baluja and S. Davies, “Using optimal dependency-trees for combinatorial optimization,” in *ICML ’97: Proceedings of the Fourteenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, pp. 30–38.
- [8] 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. London: Springer-Verlag, 1999, pp. 521–535.
- [9] G. Harik, “Linkage learning via probabilistic modeling in the ECGA,” Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign., IlliGAL Report No. 99010, 1999.
- [10] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz, “BOA: The Bayesian optimization algorithm,” in *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, Eds., vol. I. Orlando, FL: Morgan Kaufmann Publishers, San Fransisco, CA, 13-17 1999, pp. 525–532.
- [11] R. Etxeberria and P. Larrañaga, “Global optimization using bayesian networks,” in *Proceedings of the Second Symposium on Artificial Intelligence (CIMA-99)*, A. O. Rodriguez, M. S. Ortiz, and R. S. Hermida, Eds., Habana, Cuba, 1999, pp. 332–339.
- [12] H. Mühlenbein and T. Mahnig, “FDA: A scalable evolutionary algorithm for the optimization of additively decomposed functions,” *Evolutionary Computation*, vol. 7, no. 4, pp. 353–376, 1999.
- [13] H. Mühlenbein and R. Höns, “The estimation of distributions and the minimum relative entropy principle,” *Evolutionary Computation*, vol. 13, no. 1, pp. 1–27, 2005.
- [14] D. E. Goldberg, K. Deb, and J. H. Clark, “Genetic algorithms, noise, and the sizing of populations,” *Complex Systems*, vol. 6, no. 4, pp. 333–362, 1992.

- [15] D. E. Goldberg and M. Rudnick, “Genetic algorithms and the variance of fitness,” *Complex Systems*, vol. 5, no. 3, pp. 265–278, 1991.
- [16] D. E. Goldberg, K. Deb, and B. Korb, “Messy genetic algorithms revisited: Studies in mixed size and scale,” *Complex Systems*, vol. 4, no. 4, pp. 415–444, 1990.
- [17] D. Thierens, D. E. Goldberg, and Â. G. Pereira, “Domino convergence, drift and the temporal salience structure of problems,” in *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation*. IEEE Press, 1998, pp. 535–540.
- [18] F. G. Lobo, D. E. Goldberg, and M. Pelikan, “Time complexity of genetic algorithms on exponentially scaled problems,” in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, D. Whitley, D. Goldberg, E. Cantú-Paz, L. Spector, I. Parmee, and H.-G. Beyer, Eds. Las Vegas, Nevada, USA: Morgan Kaufmann, 10-12 2000, pp. 151–158.
- [19] J. Rissanen, “Modelling by shortest data description,” *Automatica*, vol. 14, pp. 465–471, 1978.
- [20] T. M. Cover and J. A. Thomas, *Elements of information theory*. New York, NY, USA: Wiley-Interscience, 1991.
- [21] T. M. Mitchell, *Machine Learning*. McGraw-Hill Higher Education, 1997.
- [22] D. E. Goldberg, *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Norwell, MA, USA: Kluwer Academic Publishers, 2002.