On the Effectiveness of Distributions Estimated by Probabilistic Model Building

Chung-Yao Chuang
Department of Computer Science
National Chiao Tung University
HsinChu City 300, Taiwan
cychuang@nclab.tw

Ying-ping Chen
Department of Computer Science
National Chiao Tung University
HsinChu City 300, Taiwan
ypchen@nclab.tw

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 regrading some parts of the problem and renders EDAs incapable of processing those parts accurately. In this work, we firstly propose a general concept that the estimated probabilistic models should be inspected to reveal the effective search directions. Based on that concept, we design a practical approach which utilizes a reserved set of solutions to examine 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 experimental results indicate that the proposed method can significantly assist ECGA to handle problems comprising building blocks of disparate scalings.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*; I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Design, Verification

Keywords

Sensible linkage, effective distribution, model pruning, estimation of distribution algorithms, EDAs, extended compact genetic algorithm, ECGA, evolutionary computation

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.

1. INTRODUCTION

Genetic algorithms (GAs) are search techniques 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. GAs mimic this mechanism by introducing artificial selections and operators to discover and recombine partial solutions. By properly growing and mixing promising partial solutions, which are often referred to as building blocks (BBs), GAs are capable of solving many problems efficiently. The ability of implicitly processing a large number of partial solutions has been recognized as an important source of GA's power. According to the Schema theorem [13], short, low-order, and highly fit substrings increase their share to be combined, and also as stated in the building block hypothesis [6], GAs implicitly decompose a problem into subproblems by processing building blocks. This 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. This can lead to the vanishing of crucial building blocks and thus the convergence to local optima. In order to overcome this 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) [17]. These methods construct probabilistic models of promising solutions and utilize the built models to generate new solutions. Early EDAs assume no interaction between variables [1, 12]. Subsequent studies start from capturing pairwise interactions [4, 2, 19] to modeling multivariate interactions [11, 18, 5, 16]. With the reasoning of dependencies among variables by building probabilistic models, these approaches can capture the structure of the problem and thus avoid the disruption of identified partial solutions.

Another topic concerning this study is the impact of disparate scalings among different building blocks to the behavior and performance of GAs. For real-world applications, it is often the case that some parts of the problem are more important and contribute more to the fitness evaluation than other parts. This situation can pose two types of difficulties. Firstly, because the processing in the population is statistical in nature, the disparate scalings can cause inaccurate processing of less salient building blocks [8, 10]. The second difficulty is the processing time delay. The lower salience of a building block generally causes it to be processed at a later time compared to those of higher salience. Such a delay may cause that building block to converge under random pressures, instead of properly selective ones. Some other previous studies on this topic include the explicit role of scalings in a systematic experimental setting [9], a theoretical model on convergence behavior of exponentially scaled problems [20], and an extension of that model to larger BBs [14].

Although the aforementioned scaling difficulties exist in many problems and degrades the performance of evolutionary algorithms, there are few investigations concerning the behavior of EDAs with the presence of scaling difficulties. In this study, we make an attempt to explore how scaling difficulties affect EDAs and propose a countermeasure to assist EDAs on problems of different scalings. Specifically, we propose a notion that the estimated probabilistic models should be inspected to reveal the effective search directions and provide an implementation of the proposed idea on the extended compact genetic algorithm (ECGA) [11].

In next section, we will look at how scaling difficulties shadow EDAs' ability in recognizing building blocks and cause inaccurate processing on parts of the solutions. After that, a general approach will be proposed to prevent such a problem. In section 3, an implementation of the proposed approach on ECGA is detailed. Section 4 presents the empirical results, followed by the discussion on the results in Section 5. Finally, section 6 concludes this paper.

2. EFFECTIVE DISTRIBUTIONS

The ability of EDAs to deal with the building block disruption problem primarily comes from the explicit modeling of promising solutions by using 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 [11], 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 cannot 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, consider a k-bit trap function,

$$\begin{aligned} f_{trap_k}(s_1 s_2 \cdots s_k) &= trap_k(u) \ , \text{ where } u = \sum_{i=1}^{k} s_i \\ &= \begin{cases} k, & \text{if u = k} \ , \\ k-1-u, & \text{otherwise.} \end{cases} , \end{aligned}$$

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,

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

where $s_1 s_2 \cdots s_{16}$ is a solution string and we choose ECGA, 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 obtained and shown in Table 1. In Table 1, variables are denoted by their index numbers. Each group of variables represents a marginal model in which a marginal distribution resides, and the converged variables are crossed out.

It can be observed that the models shown in Table 1 are only partially correct. More specifically, in each generation, only the most salient building block on which the population has not converged is modeled correctly. It is caused by the fact that some part of the problem contributes much more than all others in combine. If one part of the problem is worth essentially more than others, this part of the solution solely determines the chance that one solution will be selected or not. As a consequence, in the population, sufficient information can be provided for only the most salient building block to be modeled correctly, since the model searching is performed based on the selected solutions. The rest parts of the modeling are merely the result of low salience partial solutions "hitchhiking" on the more salient building blocks.

From the above example, we can see that not all BBs can be detected from a given set of selected solutions by probabilistic modeling. Model building algorithms cannot "see" the entire structure of the problem from the selected set of solutions because disparate scalings among different BBs prevent the complete linkage information from being supplied in the selected population. In this work, we will refer this concept as *linkage sensibility* and those problem structures that can be properly identified using the given set of solutions are called *sensible linkage*. Based on these notions, we can re-examine EDAs on the building block disruption problem. It is clear that the disruption problem still exists in the insensible portion of the problem because such parts of the problem cannot be properly modeled. Although the example is an extreme case of scalings that each subproblem is exponentially scaled, in real-world problems, it is oftentimes the case that the constituting subproblems are weighted differently, and the condition implies the linkage might just be partially sensible. Besides the BB disruption problem, the random drifting of the less salient parts of the problem mentioned in the Section 1 even worsen the situation. Those problems are usually handled by increasing population size when EDAs are applied. However, we can deal with this situation in another way if it is possible to distinguish sensible linkage from insensible linkage.

The idea of sensible linkage can be closely mapped to another notion called *effective distributions*. By effective distributions, we mean that sampling these distributions can reliably advance the quality of solutions. Thus, the essential conditions for effective distributions are the consistency with building blocks and provision of good directions for further search. If it is possible to extract effective distributions from the built probabilistic model, we can perform partial sampling using only the effective distributions and leave the rest parts of the solutions unchanged. Thus, the diversity is maintained and we are free from the BB disruption as well as random drifting problems. For instance, let's return to the 16-bit optimization problem. If it is possible to identify those partial models which are really built on the sensible linkage like [1 2 3 4] in the first generation and [5 6 7 8] in the second generation (see the third column of Table 1),

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 marginal model in which a marginal distribution resides. The variables with converged alleles are crossed out.

then we can sample only the corresponding marginal distributions which are, in this case, effective. That is, in the first generation, for each solution string, we re-sample only $s_1s_2s_3s_4$ according to the marginal distribution and keep the alleles of $s_5s_6 \cdots s_{16}$ unchanged. In 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 (note that $s_1s_2s_3s_4$ are already converged). In this way, we do not have to resort to increasing population sizes to deal with the problems that are caused by the disparate BB scalings.

The aforementioned thoughts leave us one complication: the identification of effective distributions. However, direct identification of effective distributions might not be an easy task if not impossible. Hence, it may be wise to adopt a complementary approach—to identify those distributions that are *not* likely to be effective. If there is a way to identify the ineffective distributions, we can bypass them and sample only the rest distributions, thus, to approximate the result of knowing effective distributions. Our basic idea is that if we split the entire population into two sub-populations and use only one sub-population for building probabilistic model, we can utilize the other sub-population to collect the statistics for possible indications of ineffectiveness of partial distributions in the probabilistic model built on the first sub-population. That is, with certain appropriate design, we can prune the likely ineffective portions of the model.

In the next section, our implementation of the above idea on ECGA will be detailed. More specifically, a judging criterion will be proposed to detect the likely ineffective marginal distributions of a given marginal product model.

3. ECGA WITH MODEL PRUNING

This section starts by briefly reviewing ECGA. Based on the idea of detecting the inconsistency of statistics gathered from two sub-populations, a mechanism is devised to identify the possibly ineffective parts of a probabilistic model. Finally, an optimization algorithm incorporating the proposed technique is described in detail.

3.1 Extended Compact Genetic Algorithm

ECGA [11] uses a product of marginal distributions on a partition of the variables. This kind of probability distribution belongs to a class of probabilistic models known as marginal product models (MPMs). In this kind of model, subsets of variables can be modeled jointly, and each subset is considered independent of other subsets. 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, which assumes that given all things are equal, simpler distributions are better than complex ones. The MDL principle thus penalizes both inaccurate and complex models, thereby, leading to a near-optimal distribution. Specifically, the search measure is the complexity of the MPM which is quantified as the sum of model complexity, C_m, and 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 encoding, 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 ith 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, C_m can be defined as

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

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 picked by selection) 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 optimally 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 [3], 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 corresponding distribution. With the knowledge, C_p can be derived as

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

where p_{ij} is the frequency of the jth possible partial solution to the ith variable subset observed in selected population.

Note that in the calculation of C_p, it is assumed that the jth possible partial solution to the ith variable subset is encoded using $-\log_2 p_{ij}$ bits. This assumption is fundamental to our technique to identify the likely ineffective marginal distributions. More precisely, the information of the partial solutions, $-\log_2 p_{ij}$, is a good indicator of inconsistency of statistics gathered from two sub-populations.

3.2 Model Pruning

The proposed technique to identify the possibly ineffective parts of an MPM is based on the notion that ECGA uses the compression performance to quantify the suitability of a probabilistic model to the given set of solutions. The degree of compression is a representative metric to the fitness of modeling, because all good compression methods are based on capturing and utilizing the relationships among data. Thus, if the compression scheme of the MPM built on one set of solutions is incapable of compressing another set of solutions produced in the same condition, then it is likely that the MPM is, at least, partially incorrect. Using this property, we can perform a systematical checking on the given MPM for the likely ineffective portions.

Suppose that the population of solutions, P, is split into two sub-populations S and T. The model searching is performed on S', the set of promising solutions selected from S. Then we can use the statistics collected from T', the set of solutions selected from T, to examine the built probabilistic model, M. Since each marginal model functions independently, they can be inspected separately. Recalling the description that a variable subset, which specifies a marginal model, is viewed as a "compression block" that encodes each possible partial solution according to the marginal distribution. That is, the jth possible partial solution to the ith variable subset is encoded using $-\log_2 p_{ij}$ bits, where p_{ij} is the frequency of the jth possible partial solution to the ith variable subset observed in S'. Assume that the given problem is of length ℓ with binary encoding, and there are m variable subsets with each of size k_i, $i = 1 \ldots m$, in the built model M. For the ith marginal model, $i = 1 \ldots m$, we can check whether or not

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

where q_{ij} is the frequency of the jth possible partial solution to the ith variable subset collected from T'. If the above inequality holds, then the compression scheme employed in the ith marginal model is not a good one for compressing the corresponding partial solutions in T', because it encodes a k_i-bit partial solution to a bit string of expected length more than k_i bits. Using the earlier reasoning, this condition indicates that the marginal model is likely ineffective because T' does not agree on this part of the model. Otherwise, it should be able to compress the partial solutions from T'.

From a machine learning perspective [15], a good model should generalize well to unseen instances. Otherwise, it captures coincidental regularities among training data. If the model building is performed on the portion where linkage is not sensible from the given set of solutions, then it will "overfit" to those partial solutions that are not subjected to proper selection pressure. Consequently, the regularities captured by this part of model tend to be inconsistent with the actual problem structure. Furthermore, the partial solutions that are not subjected to proper selection pressure appear to be random, and such a situation causes the phenomena of random drifting described previously. By its nature, the drifting is random, and two different sub-populations tend to drift in two different directions. Thus, we can use the statistical inconsistency between S' and T' to locate possible drifting portions, and identify the likely ineffective parts of the model. Hence, we can remove those ineffective parts to forge a partial but more effective model.

An issue in practice concerning the calculation of the above inequality is that sometimes one or several possible partial solutions are absent in the set of selected solutions, and leave $-\log_2 p_{ij}$ undefined because $p_{ij} = 0$. Currently, we handle this problem by assigning a very small value, smaller than $1/n$, to the p_{ij}'s that are zero, and normalizing them such that p_{ij}'s are sum to 1.

3.3 Integration

In this subsection, the optimization process incorporating ECGA and the previously proposed technique is described. This combination helps ECGA to achieve better performance where disparate scalings exist among different parts of the problem. The procedure is presented in Algorithm 1. This process 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 sub-populations. Selections are performed on two sub-populations separately with the same selection pressure. Model building is performed on one sub-population. The other sub-population is used to prune the built model using the technique proposed in the previous section. Finally, all solutions in the population are altered by sampling the remaining marginal distributions in the pruned model. These steps repeat until the stopping criteria are met.

A prominent difference between the above process and the traditional EDAs is that the sampling may not include all variables. As introduced in Section 2, the existing solutions are altered by sampling only the marginal distributions surviving pruning. Thus, a solution string may not be modified entirely in an iteration. This technique hence avoids random drifting and inaccurate processing of low-salience building blocks by postponing the processing until sufficient sensible linkage information is available. In this way, it can achieve better performance in terms of function evaluations if disparate scalings exist in different parts of the problem.

4. EXPERIMENTS

The experiments are designed for observing the behavior of the proposed approach on sets of problems with different scaling difficulties. Furthermore, different selection pressures are also taken into considerations to make a more thorough observation. In this study, three bounding models of scaling [7] are considered: exponential, power-law, and uniform. Based on different scalings, three sets of test functions are constructed as listed by using f_{trap_4} as the elemental function. For simplicity, the splitting of population is performed in the way that two resulting sub-populations are disjoint and of equal size. The stopping criterion is set such that a run is terminated when all solutions in the population converge to the same fitness value.

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

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

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

4.1 Impact on Population Requirements

This section describes the experimental setting and results of the proposed method compared to that of the original ECGA on the three problem sets. The problem sizes range from 40 to 80 bits ($m = 10 \ldots 20$). For each problem instance, the minimum population size required such that, on average, $m-1$ BBs converge to the optimum in 50 runs is determined by bisection. Two selection pressures are adopted by setting the tournament size t to 8 and 16.

Algorithm 1 ECGA with Model Pruning

```
Initialize a population $P$ with $n$ solutions of length $\ell$.
while the stopping criteria are not met do
    Evaluate the solutions in $P$.
    Divide $P$ into $S$ and $T$ at random.
    $S' \leftarrow$ Apply $t$-wise tournament selection on $S$.
    $T' \leftarrow$ Apply $t$-wise tournament selection on $T$.
    $M \leftarrow$ Conduct greedy MPM search on $S'$.
    $M' \leftarrow$ Prune $M$ based on the inconsistency with $T'$.
    for each remaining marginal distribution $D$ in $M'$ do
        for each string $\mathbf{s} = s_1 s_2 \cdots s_\ell$ in $P$ do
            Change the values in $\mathbf{s}$ by sampling $D$.
        end for
    end for
end while
```

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 an appropriate selection pressure, the population size needed by the proposed method grows in a relatively slow rate. 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 required population sizes are similar to the previous set of experiments. The proposed method still uses fewer function evaluations, but the differences are reduced.

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.

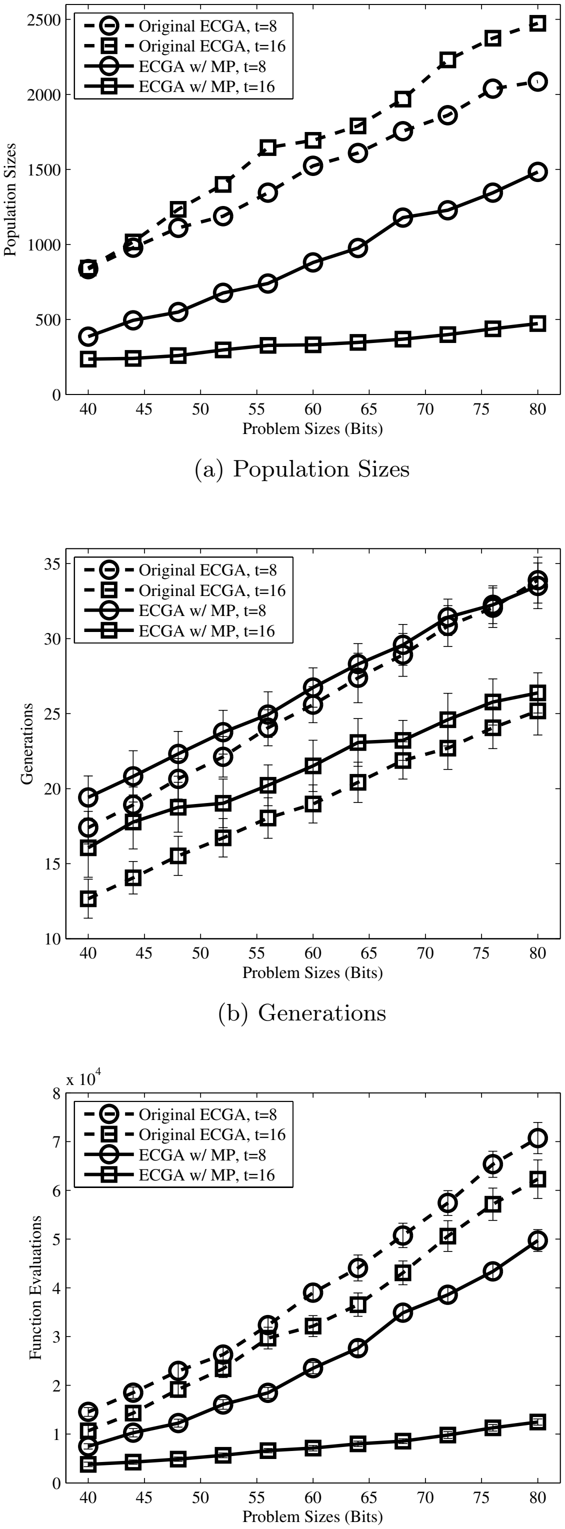
It is noted that a common phenomenon appears in all of the above three sets of experiments that the proposed method needs more generations before convergence than the original ECGA under the same selection pressure. In the next section, we will further explore this phenomenon using sets of experiments that augment the population sizes.

4.2 Time-Space Interactions

This section describes sets of experiments that reveal the behavior of the proposed method when the population size is adjusted and presents the results to illustrate the interactive effect between population sizes and generations for the proposed method. In these experiments, the 60-bit problems ($m = 15$) are adopted as test functions and the population sizes are augmented proportional to the minimum population sizes estimated in the previous sets of experiments.

As presented in Figure 4, only slight decreases in generations are achieved by increasing population sizes on the exponentially scaled 60-bit problem. Among others, the proposed method with tournament size 16 delivered the most reduction. With no prominent reductions in the generations and the increasing population sizes, the function evaluations grow up as expected in all four settings.

Figure 5 shows the results on the power-law scaled 60-bit problem. In this case, prominent reductions in generations are observed in the proposed method. However, despite the presence of these reductions, the function evaluations still grow up with the increasing population size.

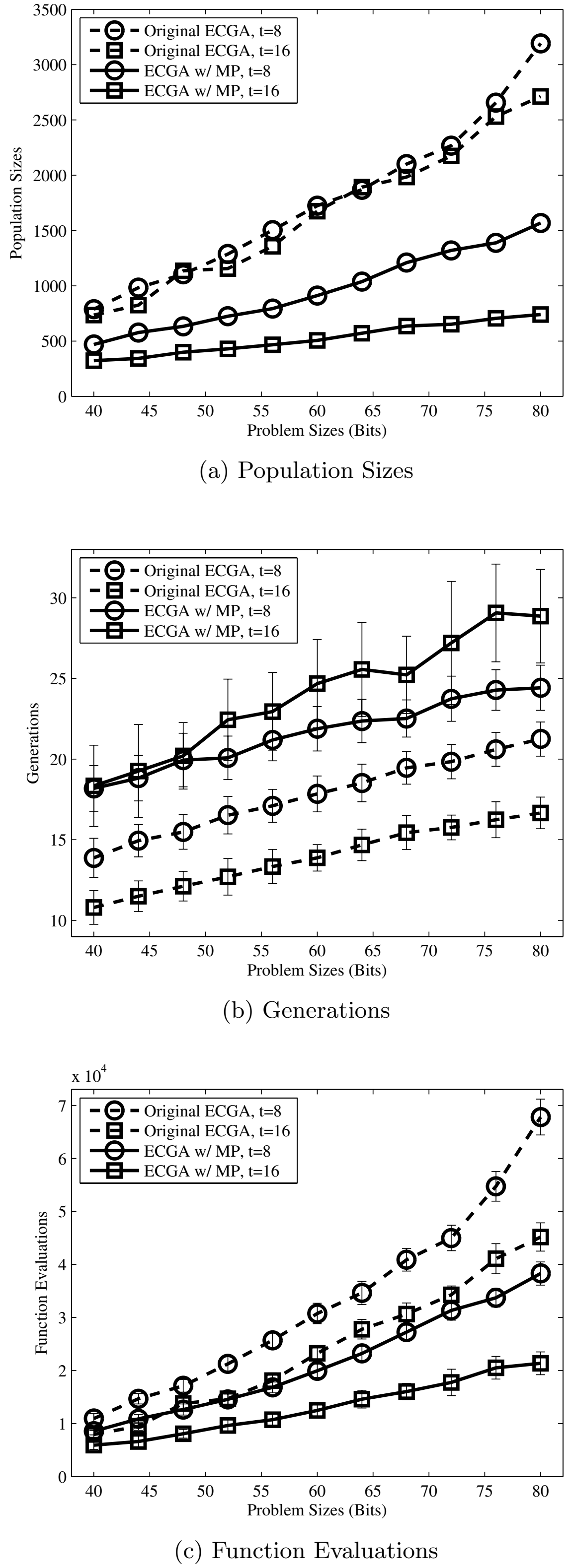


(a) Population Sizes

(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.

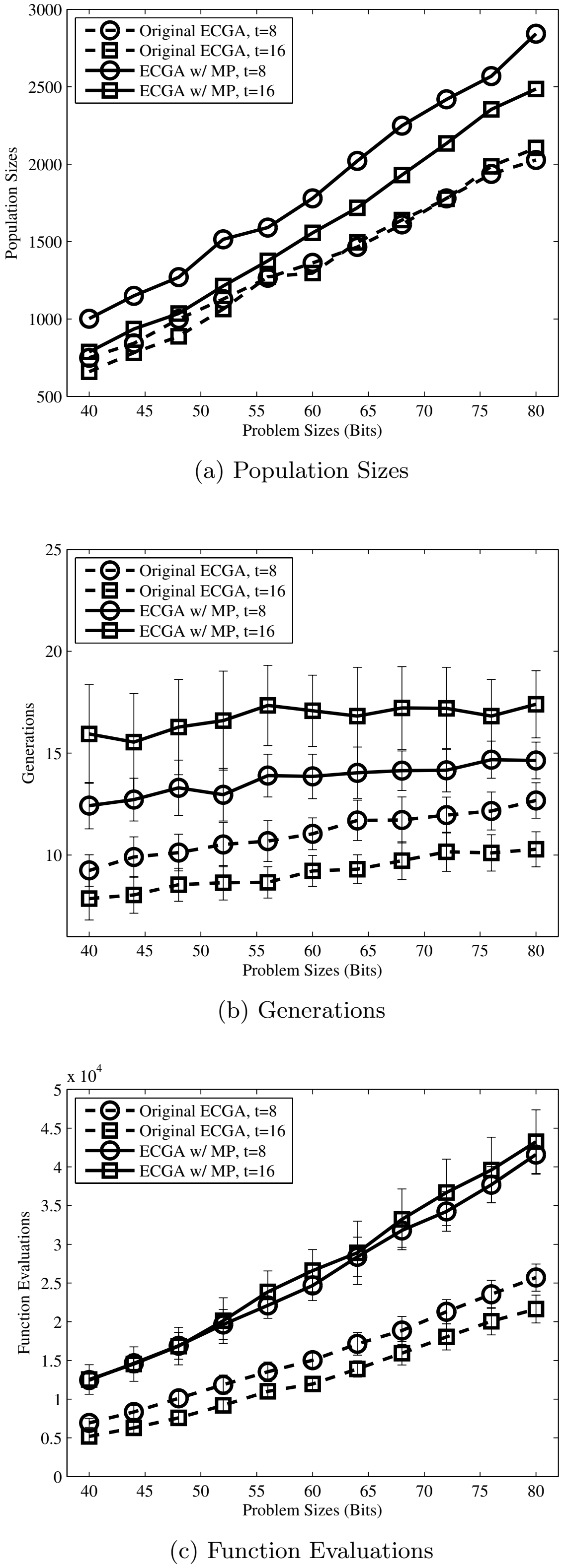


(a) Population Sizes

(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.

The most significant decrease in generations is observed on the uniformly scaled 60-bit problem as shown in Figure 6. With tournament size 16, the proposed method reduced up to 8 generations needed to converge when twice larger population size is used. It somehow keeps the number of function evaluations from climbing up with the population size.

5. DISCUSSION

The proposed method improves the original ECGA on problems where disparate scalings exist among different BBs. As illustrated in Figure 1(c) and Figure 2(c), prominent reductions in fitness evaluations are achieved. Moreover, in the uniformly scaled problems where the linkage 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 using more generations to overcome the problem. The most notable case is on uniformly scaled problems shown in Figure 6 that the proposed method with an appropriate selection pressure reduces the generations aggressively when a larger population is available and thus keeps the function evaluations from rising up. This phenomenon may be worth further investigations in the hope of discovering a way to relieve the burden of setting appropriate population sizes.

6. SUMMARY AND CONCLUSIONS

This paper started at reviewing previous studies on EDAs and scaling difficulties. It illustrated how scaling difficulties shadows EDAs' ability in recognizing BBs. A notion called *linkage sensibility* was described, and the term *sensible linkage* was proposed to refer to those problem structures that can be extracted by inspecting only the set of selected solutions. Based on the concept, we defined the effectiveness of distributions estimated by probabilistic model building and proposed a general approach to achieve a more effective modeling. Finally, an implementation of the proposed approach on ECGA was described and examined on several test functions with different scaling difficulties.

In this study, we focused on scaling difficulties and their influences on EDAs' ability in recognizing BBs. However, at a higher level, our attempt was trying to resolve an important issue which was 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 adopted and carried over to other EDAs such that more flexible, friendly, and robust EDAs may be developed.

Acknowledgments

The work was partially sponsored by the National Science Council of Taiwan under grants NSC-96-2221-E-009-196 and NSC-96-2627-B-009-001 as well as by the Aiming for the Top University and Elite Research Center Development Plan (ATU Plan), 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.

7. REFERENCES

[1] S. Baluja. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical report, Pittsburgh, PA, USA, 1994.

[2] S. Baluja and S. Davies. Using optimal dependency-trees for combinational optimization. In *Proceedings of the 4th ICML*, pages 30–38, 1997.

[3] T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley-Interscience, 1991.

[4] J. de Bonet, C. Isbell, and P. Viola. MIMIC: Finding optima by estimating probability densities. *Advances in Neural Information Processing Systems*, 9:424–430.

[5] R. Etxeberria and P. Larrañaga. Global optimization using bayesian networks. In *Proceedings of the 2nd Symp. on Artificial Intelligence*, pages 332–339, 1999.

[6] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Boston, MA, USA, 1989.

[7] D. E. Goldberg. *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[8] D. E. Goldberg, K. Deb, and J. H. Clark. Genetic algorithms, noise, and the sizing of populations. *Complex Systems*, 6(4):333–362, 1992.

[9] D. E. Goldberg, K. Deb, and B. Korb. Messy genetic algorithms revisited: Studies in mixed size and scale. *Complex Systems*, 4(4):415–444, 1990.

[10] D. E. Goldberg and M. Rudnick. Genetic algorithms and the variance of fitness. *Complex Systems*, 5(3):265–278, 1991.

[11] G. Harik. Linkage learning via probabilistic modeling in the ECGA. Technical Report 99010, Illinois Genetic Algorithms Laboratory, UIUC, IL, USA, 1999.

[12] G. R. Harik, F. G. Lobo, and D. E. Goldberg. The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, 3(4):287, 1999.

[13] J. H. Holland. *Adaptation in natural and artificial systems*. MIT Press, Cambridge, MA, USA, 1992.

[14] F. G. Lobo, D. E. Goldberg, and M. Pelikan. Time complexity of genetic algorithms on exponentially scaled problems. In *Proceedings of GECCO-2000*, pages 151–158, 2000.

[15] T. M. Mitchell. *Machine Learning*. McGraw-Hill Higher Education, 1997.

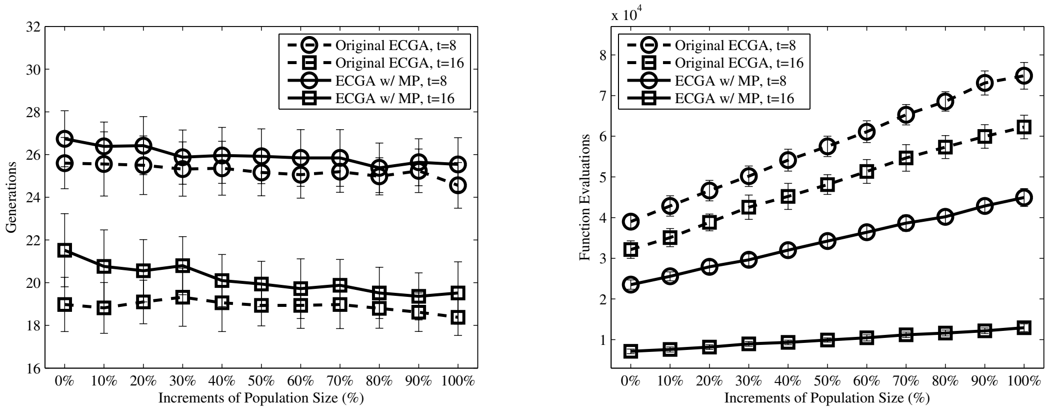
[16] H. Mühlenbein and R. Höns. The estimation of distributions and the minimum relative entropy principle. *Evolutionary Computation*, 13(1):1–27, 2005.

[17] H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions I. binary parameters. In *Proceedings of PPSN IV*, 1996.

[18] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz. BOA: The Bayesian optimization algorithm. In *Proceedings of GECCO-99*, pages 525–532, 1999.

[19] M. Pelikan and H. Mühlenbein. The bivariate marginal distribution algorithm. In *Advances in Soft Computing*, pages 521–535, 1999.

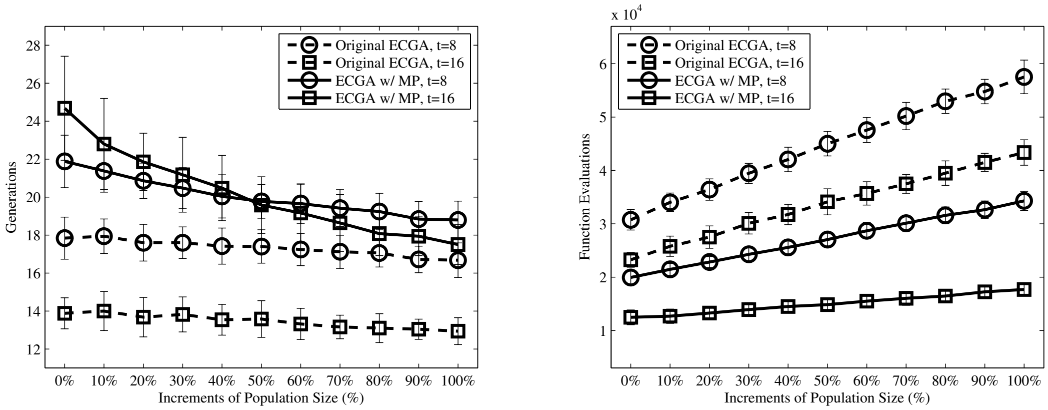
[20] D. Thierens, D. E. Goldberg, and Â. G. Pereira. Domino convergence, drift and the temporal salience structure of problems. In *Proceedings of ICEC '98*, pages 535–540, 1998.



(a) Generations

(b) Function Evaluations

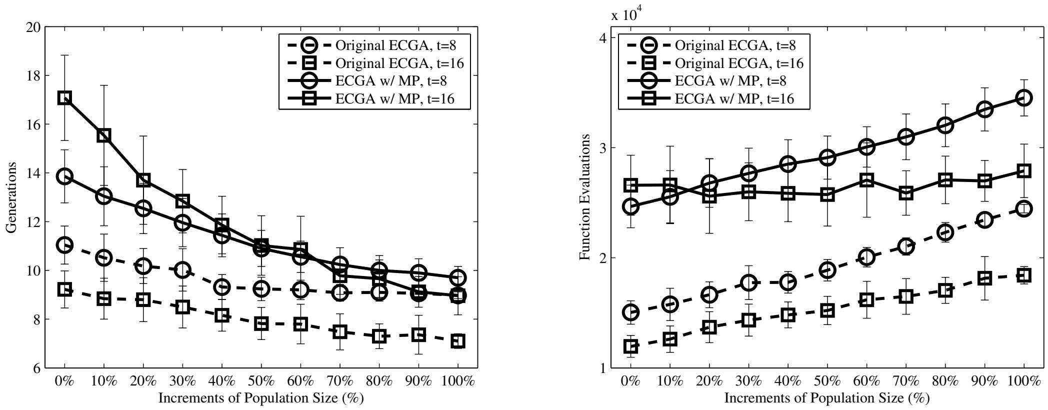
Figure 4: Empirical results of increasing population size in solving the *exponentially scaled 60-bit problem.* The population sizes are increased proportionally to the minimum required population sizes.



(a) Generations

(b) Function Evaluations

Figure 5: Empirical 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.



(a) Generations

(b) Function Evaluations

Figure 6: Empirical 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.