Multivariate Multi-Model Approach for Globally Multimodal Problems

Chung-Yao Chuang
Institute of Information Science
Academia Sinica
Taipei 11529, Taiwan
cychuang@iis.sinica.edu.tw

Wen-Lian Hsu
Institute of Information Science
Academia Sinica
Taipei 11529, Taiwan
hsu@iis.sinica.edu.tw

ABSTRACT

This paper proposes an estimation of distribution algorithm (EDA) aiming at addressing globally multimodal problems, i.e., problems that present several global optima. It can be recognized that many real-world problems are of this nature, and this property generally degrades the efficiency and effectiveness of evolutionary algorithms. To overcome this source of difficulty, we designed an EDA that builds and samples multiple probabilistic models at each generation. Different from previous studies of globally multimodal problems that also use multiple models, we adopt multivariate probabilistic models. Furthermore, we have also devised a mechanism to automatically estimate the number of models that should be employed. The empirical results demonstrate that our approach obtains more global optima per run compared to the well-known EDA that employs the same class of probabilistic models but builds a single model at each generation. Moreover, the experiments also suggest that using multiple models reduces the generations spent to reach convergence.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*Parameter learning*;
I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*;
I.5.3 [**Pattern Recognition**]: Clustering;
G.1.6 [**Numerical Analysis**]: Optimization

General Terms

Algorithms

Keywords

Estimation of distribution algorithms, EDAs, globally multimodal problems, global multimodality, marginal product models, multi-model approach, extended compact genetic algorithm, ECGA, evolutionary algorithms, genetic algorithm, evolutionary computation.

GECCO'10, July 7–11, 2010, Portland, Oregon, USA.

1. INTRODUCTION

Estimation of distribution algorithms (EDAs) [19, 16, 23] are a class of evolutionary algorithms that replace the traditional variation operators, such as mutation and crossover, by building a probabilistic model on promising solutions and sampling the built model to generate new candidate solutions. Using probabilistic models to summarize the information enables these methods to automatically infer the likely structure of promising solutions and exploit the identified problem regularities to facilitate further search. However, it has been identified [20] that most EDAs typically bypass the issue of global multimodality, i.e., more than one global optimum for the considered problem, and converge to only one of the global optima. Consider that many optimization problems are globally multimodal, it is often preferable or even necessary to obtain as many global optima as possible.

To provide some examples of such demand, [14] stated the benefits of obtaining multiple solutions for a robot path-planning algorithm: In such a situation, the complete information may not be available at the time of planning, thus it is desirable to have several routes planned in advance in case some of them become infeasible, especially when considering dynamic environment. Another instance is in the computational approach to peptide design, which uses estimated docking energy to measure the quality of candidate solutions. Because the docking energy is just an estimate, the solution found may be extremely difficult to synthesize chemically or may even be a false docking positive as indicated in [3]. Therefore, obtaining multiple and diverse solutions is desirable in this task.

In this paper, we propose an approach that builds and samples multiple models at each generation. By iteratively estimating the membership of solutions to each model and readjusting the model structure and parameters, it allows simultaneous modeling of the basins to different global optima. Furthermore, we have also devised a method to compare the suitability of different sets of models, which allows us to automatically estimate the number of models that should be employed. The experiments show that our approach identifies more global optima per run compared to the well-known EDA that employs the same class of probabilistic models. Moreover, the empirical results suggest that using multiple models reduces the generations spent to reach convergence.

In the next section, we briefly review previous studies relevant to this work. After that, section 3 describe the class of probabilistic models employed in this study. In section 4, a process to estimate multiple models are outlined. Section 5

proposes a complexity measure for a given model set, which enables us to formally compare the suitability of two sets of models for a given set of solutions. Section 6 integrates the proposed ideas into an optimization algorithm. The experiments and empirical results are presented in section 7. Finally, section 8 concludes this paper.

2. BACKGROUND

Evolutionary algorithms (EAs) are stochastic search techniques loosely 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. EAs mimic this mechanism by introducing artificial selections and variation operators to discover and recombine partial solutions. From the inception, pioneer evolutionary algorithmists have recognized that detecting interacting variables would be beneficial or even necessary to address hard optimization problems [15, 11]. The detection and utilization of this inter-variable relationship is generally referred to as *linkage problem*, and has attracted a lot of research efforts to design mechanisms that automatically discover such information. In recent years, the most advocated approaches toward this notion are estimation of distribution algorithms (EDAs), which are able to encode the relationship between problem variables by building probabilistic models on promising solutions. In this paper, we focus on discussions of EDAs and their applications to solve globally multimodal problems.

Early EDAs, such as the population-based incremental learning (PBIL) [1] and the compact genetic algorithm (cGA) [13], assume no interaction between decision variables, i.e., decision variables are assumed independent of each other. Subsequent studies start from capturing pairwise interactions, such as mutual-information-maximizing input clustering (MIMIC) [6], Baluja's dependency tree approach [2], and the bivariate marginal distribution algorithm (BMDA) [24], to modeling multivariate interactions, such as the extended compact genetic algorithm (ECGA) [12], the Bayesian optimization algorithm (BOA) [22], the estimation of Bayesian network algorithm (EBNA) [10], the factorized distribution algorithm (FDA) [18], and the learning version of FDA [17]. It has been demonstrated that multivariate EDAs are able to solve deceptive optimization problems, which require linkage information to be reliably addressed.

Despite the successful results presented in the literature, most research only examines EDAs on problems containing single global optimum. However, many optimization problems are of several global optima, i.e., globally multimodal, and it is often preferable to identify as many global optima as possible. This issue has been addressed several times on the classical EAs, which are deemed ineffective in such situation, as they usually converge to only one global optimum. To briefly describe this phenomenon, consider that when dealing with a globally multimodal problem, a large enough initial population should contain solutions belonging to basins of different global optima. As there is no mechanism to maintain the balance of distribution between each basin, random fluctuations resulting from the selection process will randomly give preference on one global optimum, i.e., more solutions in the associated basin will be generated, which leads to a reproduction advantage on that basin in the next generation. This advantage, when accumulated over generations, will push solutions belonging to other basins to extinction.

Furthermore, the existence of several global optima often makes the convergence taking longer time than it will otherwise require. This inefficiency comes from combining solutions located in different basins. It has been noted [21] that this juxtaposition usually produces poor solutions because disruption of good solutions are very likely to happen when recombining solutions from different basins. This inefficiency of exploration, depending on the selection pressure, usually takes several generations to disappear as the population drifting toward a single basin in the process described above. Therefore, the interests in studying globally multimodality is not limited in finding more than one optimum, but also in improving the efficiency of evolutionary algorithms.

It is conceivable that EDAs will face the same difficulties described above when dealing with globally multimodal problems. However, there is not much research devoted to this topic. Noticed exceptions include [21], [20] and [9]. In [21], clustering is adopted into a univariate EDA (UMDA) as a means to recognize subpopulations belonging to different basins. They experimented the approach on a simple globally multimodal problem with two peaks. The results demonstrated simultaneous discovery of both optima and improvement of convergence speed when clustering is applied. Advancing the idea of incorporating clustering, [20] proposed an approach based on the unsupervised learning of Bayesian networks. An unobserved variable C is included in the model, which represents the unknown cluster label. The Bayesian network, which includes all problem variables and an artificial cluster variable C, represents a joint probability distribution for the selected solutions. The resulting EDA, which is called unsupervised estimation of Bayesian network algorithm (UEBNA), is expected to detect and maintain the presence of solutions from different basins. Experiments confirm the superiority of the algorithm, in terms of the number of optima found, compared to UMDA and EBNA on a set of graph bisection problems. More recently, [9] proposed another EDA employing clustering, which aims at reducing the computational burden of building multivariate models. The core algorithm is a simple clustered EDA that adopts order-2 probabilistic models, which leads to a very parsimonious approach when compared to multivariate EDAs. The experiments were also carried out on a set of graph bisection problems, and demonstrated that the approach is also capable of addressing globally multimodal problems.

In this paper, we describe an approach that differs from the above mentioned studies in several ways. Firstly, compared to [21] and [9], we consider multivariate probabilistic models. Secondly, unlike that of [20], the proposed algorithm employs, instead of a single model, a multi-model approach. Moreover, our method automates the selection of the number of models that should be used. This eliminates the need of manually tuning the number of clusters that should be employed as the above mentioned studies do. Furthermore, in addition to symmetric problems (e.g., graph bisection problems), we also experimented on globally multimodal problems that are not symmetric, and briefly look at problems that have basins with heterogeneous linkage, i.e., the structural decomposition of each global optimum is not the same.

$[s_1]$	$[s_2\ s_4]$	$[s_3]$
$P(s_1 = 0) = 0.4$	$P(s_2 = 0, s_4 = 0) = 0.4$	$P(s_3 = 0) = 0.5$
$P(s_1 = 1) = 0.6$	$P(s_2 = 0, s_4 = 1) = 0.1$	$P(s_3 = 1) = 0.5$
	$P(s_2 = 1, s_4 = 0) = 0.1$	
	$P(s_2 = 1, s_4 = 1) = 0.4$	

Table 1: An example of marginal product model that defines a joint distribution over four variables. The variables enclosed in the same brackets are considered dependent and modeled jointly. Each variable subset is considered independent of other variable subsets.

3. MARGINAL PRODUCT MODELS

In this study, we consider a class of probabilistic models known as marginal product models (MPMs). This kind of model forms distribution by using a product of marginal distributions on a partition of the variables. In this kind of distribution, subsets of variables can be modeled jointly, and each subset is considered independent of others subsets. In this work, we adopt a notation that variable subsets are enclosed in brackets. Table 1 presents an example of MPM defined over four variables: s_1, s_2, s_3 and s_4. In this example, s_2 and s_4 are modeled jointly and each of the three variable subsets ($[s_1]$, $[s_2\ s_4]$ and $[s_3]$) is considered independent of other subsets. For instance, the probability that this MPM generates a sample $s_1s_2s_3s_4 = 0101$ is calculated as follows,

$$\begin{aligned} &P(s_1s_2s_3s_4 = 0101) \\ &= P(s_1 = 0) \times P(s_2 = 1, s_4 = 1) \times P(s_3 = 0) \\ &= 0.4 \times 0.4 \times 0.5. \end{aligned}$$

In fact, as its name suggested, a marginal product model represents a distribution that is a "product" over the marginal distributions defined over variable subsets.

The first EDA that employs MPMs is the extended compact genetic algorithm (ECGA) [12]. In ECGA, both the structure and the parameters of the model are searched and optimized with 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 [25], 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 MPM complexity which is quantified as the sum of model complexity, C_m, and compressed population complexity, C_p. The greedy MPM search first considers all variables as independent and each of them forms a separate variable subset. In each iteration, the greedy search merges two variable subsets that yields the most $C_m + C_p$ reduction. The process continues until there is no further merge that can decrease the combined complexity.

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 $\ell = \sum_{i=1}^{m} k_i$. Then 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, the model complexity, 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 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 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 [5], 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. Based on the information theory, the compressed population complexity, 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 of iteratively estimating multiple models. More precisely, we use this notion to recognize a partition of the set of selected solutions and build (then refine) the MPMs based on each subset of solutions.

4. ESTIMATING MULTIPLE MODELS

It can be seen from the previous section that the suitability of a probabilistic model for a given set of solutions can be quantified by the compression performance. The degree of compression is a quite representative metric to the fitness of modeling, because all good compression methods are based on capturing and utilizing the relationship among data. Assuming that we are given a set of solutions and c MPMs, and asked to assign, for each solution, the fittest model among these c MPMs. By the reasoning above, for each solution, we should choose the model which encodes the solution to the shortest description. To be more precise, for each solution x, we should choose the model M_y, $y \in \{1, 2, ..., c\}$,

Algorithm 1 Building Multiple Models

procedure BUILDMODELS(c, S)
  Randomly pick a subset $\{d_y | y \in \{1, 2, ..., c\}\}$ from S.
  Estimate $\{M_y | M_y$ is a univariate model based on $d_y\}$.
  for each x in S **do**
    $y_x \leftarrow y$ such that M_y yields smallest λ for x.
  end for
  repeat
    $y'_x \leftarrow y_x$ for each x in S.
    for each y in $\{1, 2, ..., c\}$ **do**
      $M_y \leftarrow$ greedy MPM search on $\{x | y_x = y\}$.
    end for
    for each x in S **do**
      $y_x \leftarrow y$ such that M_y yields smallest λ for x.
    end for
  until $y'_x = y_x$ for all $x \in S$
  return $\{M_y | y \in \{1, 2, ..., c\}\}$ and $\{y_x | x \in S\}$.
end procedure

with the smallest

$$\lambda = \sum_{i=1}^{m} -\log_2 p_{ix_i}$$

where m is the number of marginal distributions in M_y and x takes the x_ith partial solution in the ith variable subset. In this way, we can utilize these c models to partition the population into c subsets, in which the solutions are best described by the associated MPM.

Using this technique, we can devise a method that iteratively partitions the selected population and rebuilds MPMs on the resulting partition. Our approach starts at building c univariate models estimated from randomly picked c solutions* and smooths the models to prevent zero probabilities. These initial c models are then used to partition the set of selected solutions, S, into c disjoint subsets. Based on each subset, we re-estimate an MPM, thus form a new set of c models. This process iterates until the solutions stop changing their subset membership. The resulting algorithm, as outlined in Algorithm 1, is able to recognize a partition on the selected population and estimate MPMs on the subsets of solutions.

5. QUANTIFYING THE COMPLEXITY OF A SET OF MODELS

In this section, we move on to devise a method for determining the appropriate number of models to use. In order to address this issue, we have to first be able to compare the suitability of two sets of MPMs for modeling a given set of n solutions. The non-triviality of this task resides in that these two sets may contain different number of models and thus, we have to think of a new way to quantify their complexities. It is conceivable that the intuitive $\sum_y (C_m(M_y) + C_p(M_y))$, where $C_m(M_y)$ and $C_p(M_y)$ represent the model complexity and compressed population complexity of M_y, will not work, because in this condition, larger MPM sets will have biased advantage of being able to split the population into smaller subpopulations and build overly-simplified models on the resulting partition.

*In order to pick different enough solutions, our method randomly selects several sets of c solutions, and choose the set of solutions with the largest pairwise Euclidean distance.

Algorithm 2 The Resulting Multi-Model Approach

Initialize a population P with n solutions.
while the stopping criteria are not met **do**
  Evaluate the solutions in P.
  $S \leftarrow$ apply selection on P.
  $c \leftarrow 1$.
  $\mathcal{M}', \mathcal{Y} \leftarrow$ BUILDMODELS(c, S).
  $\mathcal{C}' \leftarrow$ calculate complexity based on $\mathcal{M}'$ and $\mathcal{Y}$.
  repeat
    $\mathcal{M} \leftarrow \mathcal{M}'$.
    $\mathcal{C} \leftarrow \mathcal{C}'$.
    $c \leftarrow c + 1$.
    $\mathcal{M}', \mathcal{Y} \leftarrow$ BUILDMODELS(c, S).
    $\mathcal{C}' \leftarrow$ calculate complexity based on $\mathcal{M}'$ and $\mathcal{Y}$.
  until $\mathcal{C}' \geq \mathcal{C}$
  $O \leftarrow \emptyset$.
  for each model M_y in $\mathcal{M}$ **do**
    $O' \leftarrow$ generate new solutions by sampling M_y.
    $O \leftarrow O \cup O'$.
  end for
  Incorporate O into P.
end while

Reflecting on the advantage of using multiple models, it can be understood that the descriptive power will increase while more models are employed, i.e., we can compress the population better by using more models. By using the MPM estimated from a subset of solutions, we are able to encode that subset to a shorter description compared to that encoded by an MPM estimated from the whole set of solutions. However, we forgot to consider the additional information that associates each solution to its appropriate model. This information, which indicates the assignment of solutions to their most suitable models, should be included in the calculation of the complexity of an MPM set. Again, resorting to information theory, the amount of information spent on tagging all n solutions to their associated models can be quantified as

$$C_t = n \sum_{y=1}^{c} -p_y \log_2 p_y \ ,$$

where p_y is the frequency of assigning a solution to the yth model, assuming there are c models. This quantity represents the number of bits required to store the labels that assign the solutions to their corresponding models. Using this notion, we can now derive a complexity measure for a set of MPMs, $\{M_y | y \in \{1, 2, ..., c\}\}$, on modeling a given set of solutions, S, as

$$C = C_t(\{y_x | x \in S\}) + \sum_{y=1}^{c} (C_m(M_y) + C_p(M_y)) \quad (1)$$

where $y_x \in \{1, 2, ..., c\}$ is the assignment of x to its most suitable model among M_y's and $C_t(\{y_x | x \in S\})$ represents the number of bits needed to store all y_x's as described by the previous formula.

6. INTEGRATION

Combining the ideas presented in section 4 and 5, an optimization process that uses multiple models is proposed in

Algorithm 2. The procedure starts at initializing a population of solutions. After initialization, the solutions are evaluated and selection is performed to obtain the selected set of solutions, S. We then build a singleton MPM set on S and calculate its complexity based on Equation (1). Following that, we gradually increase c and build larger MPM sets. This expansion terminates when the complexity of the newly estimated model set $\mathcal{M}'$ is greater than that of the previous MPM set, $\mathcal{M}$. Finally, we sample the MPMs in $\mathcal{M}$ to create new candidate solutions and incorporate them into the population. These steps repeat until the stopping criteria are met.

In this work, we generate an equal number of solutions for each M_y in the resulting $\mathcal{M}$. The rationale behind this choice has been discussed in [20]. To briefly recap, if the number of solutions to be generated is based on the number of solutions previously assigned to M_y, then we invent a preference to larger basins, no matter the fitness of their associated solutions. Furthermore, in order to study the fundamental characteristics of the proposed approach, we do not use any other means to maintain the population diversity. A plain generational replacement is used, i.e., for each generation, we replace all solutions by the newly generated ones.

7. EXPERIMENTS AND RESULTS

In this study, our approach is evaluated on the test problems constructed by concatenating several trap functions. A k-bit trap function is a function of unitation[†] which can be expressed as

$$f_t^{(k)}(s_1 s_2 \cdots s_k) = \begin{cases} k, & \text{if } u = k \\ k-1-u, & \text{otherwise} \end{cases} ,$$

where u is the number of ones in the binary string $s_1 s_2 \cdots s_k$. The trap functions were used pervasively in the studies concerning EDAs and other evolutionary algorithms because they provide well-defined structures among variables, and the ability to recognize inter-variable relationships is essential to solve the problems consisting of traps [7, 8]. In order to create globally multimodal problems, we also define a k-bit *inverse* trap function as

$$\bar{f}_t^{(k)}(s_1 s_2 \cdots s_k) = \begin{cases} k, & \text{if } u = 0 \\ u-1, & \text{otherwise} \end{cases} .$$

The plan is to design test problems that assign different region of search space to different combination of trap and inverse trap functions. For this purpose, we define a set of problem variables to be the *switch variables*, to which the values specify the combination of trap and inverse trap functions to be used to evaluate the corresponding solutions. To demonstrate this composition, consider our first test problem, F_1, which is formed by concatenating ten 4-bit trap or inverse trap functions and one switch variable, s_{41},

$$F_1(s_1 s_2 ... s_{41}) = \begin{cases} G_0(s_1 s_2 ... s_{40}), & \text{if } s_{41} = 0 \\ G_1(s_1 s_2 ... s_{40}), & \text{if } s_{41} = 1 \end{cases} ,$$

[†]A function of which the function value depends only on the number of ones in the binary input string.

where G_0 and G_1 are defined as

$$G_0(s_1 s_2 ... s_{40}) = \sum_{i=0}^{9} \bar{f}_t^{(4)}(s_{4i+1} s_{4i+2} s_{4i+3} s_{4i+4}) ,$$

$$G_1(s_1 s_2 ... s_{40}) = \sum_{i=0}^{9} f_t^{(4)}(s_{4i+1} s_{4i+2} s_{4i+3} s_{4i+4}) .$$

It is obvious that F_1 contains two global optima, which are the solution of all ones and the solution of all zeros.

Our second test problem, F_2, is formed by the same technique as the above and it uses two switch variables to create a problem with four global optima,

$$F_2(s_1 s_2 ... s_{42}) = \begin{cases} G_{00}(s_1 s_2 ... s_{40}), & \text{if } s_{41} s_{42} = 00 \\ G_{01}(s_1 s_2 ... s_{40}), & \text{if } s_{41} s_{42} = 01 \\ G_{10}(s_1 s_2 ... s_{40}), & \text{if } s_{41} s_{42} = 10 \\ G_{11}(s_1 s_2 ... s_{40}), & \text{if } s_{41} s_{42} = 11 \end{cases} ,$$

where the definition of G_{00} to G_{11} are

$$G_{00}(s_1 ... s_{40}) = \sum_{i=0}^{4} (\bar{f}_t^{(4)}(s_{8i+1} ... s_{8i+4}) + \bar{f}_t^{(4)}(s_{8i+5} ... s_{8i+8})) ,$$

$$G_{01}(s_1 ... s_{40}) = \sum_{i=0}^{4} (\bar{f}_t^{(4)}(s_{8i+1} ... s_{8i+4}) + f_t^{(4)}(s_{8i+5} ... s_{8i+8})) ,$$

$$G_{10}(s_1 ... s_{40}) = \sum_{i=0}^{4} (f_t^{(4)}(s_{8i+1} ... s_{8i+4}) + \bar{f}_t^{(4)}(s_{8i+5} ... s_{8i+8})) ,$$

$$G_{11}(s_1 ... s_{40}) = \sum_{i=0}^{4} (f_t^{(4)}(s_{8i+1} ... s_{8i+4}) + f_t^{(4)}(s_{8i+5} ... s_{8i+8})) .$$

It can be seen that, in both F_1 and F_2, the structural decompositions of all global optima are the same. In other words, the subfunctions forming the problem are aligned in variables. In this work, we also experiment on a globally multimodal problem that has basins with heterogeneous linkage, i.e., the subfunction composition in each basin is not the same. This test problem, defined as F_3, is similar to F_1 except that the respective concatenated traps and inverse traps are not aligned in variables,

$$F_3(s_1 s_2 ... s_{41}) = \begin{cases} H_0(s_1 s_2 ... s_{40}), & \text{if } s_{41} = 0 \\ H_1(s_1 s_2 ... s_{40}), & \text{if } s_{41} = 1 \end{cases} ,$$

where H_0 and H_1 are defined as

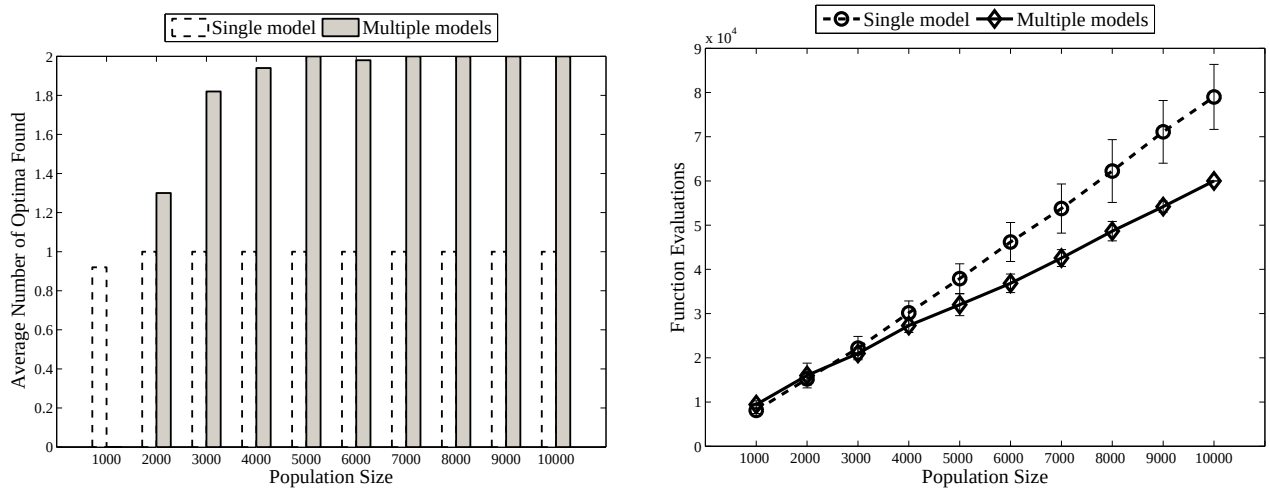
$$H_0(s_1 s_2 ... s_{40}) = \sum_{i=0}^{9} \bar{f}_t^{(4)}(s_{4i+1} ... s_{4i+4}) ,$$

$$H_1(s_1 s_2 ... s_{40}) = \sum_{i=0}^{8} f_t^{(4)}(s_{4i+3} ... s_{4i+6}) + f_t^{(4)}(s_{39} s_{40} s_1 s_2) .$$

It is conjectured that this problem is more difficult as the disruption of good partial solutions is more likely to happen if the respective decompositions are not properly recognized.

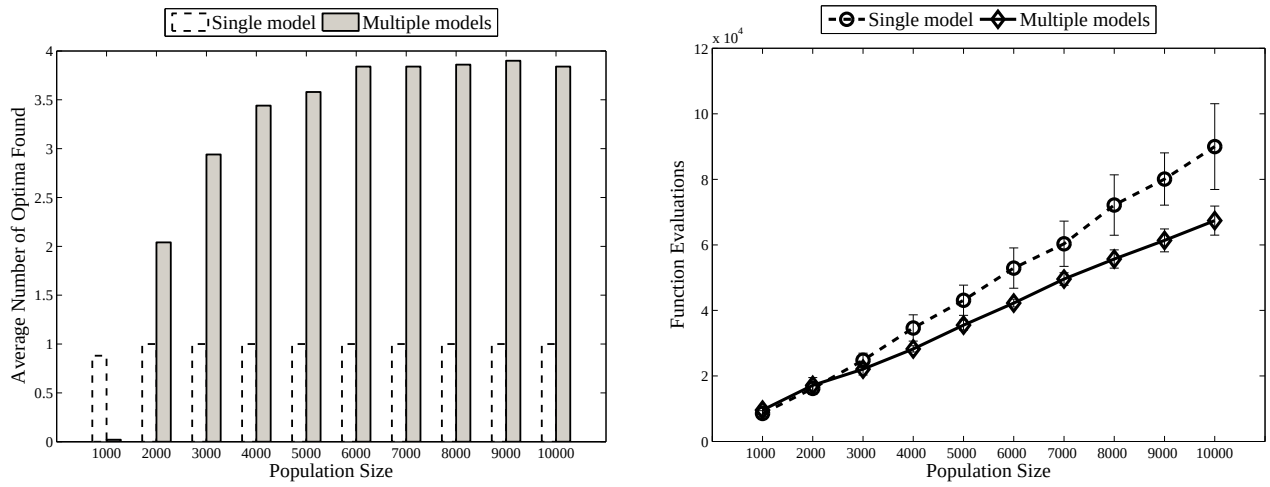
7.1 Experimental Settings

In this study, the proposed approach is compared with the extended compact genetic algorithm (ECGA) [12]. The reason to compare with ECGA is that ECGA uses the same class of probabilistic models, i.e., MPMs, and it is a single model approach which can serve as an illustrative contrast to our multi-model approach. For each of the F_1 to F_3, we run both algorithms using population sizes ranging from



(a) Number of Optima Obtained (b) Function Evaluations

Figure 1: Empirical results of the proposed approach (multiple models) compared to ECGA (single model) on test problem F_1. Different population sizes are experimented to demonstrate the behavior of the algorithms.



(a) Number of Optima Obtained (b) Function Evaluations

Figure 2: Empirical results of the proposed approach (multiple models) compared to ECGA (single model) on test problem F_2. Different population sizes are experimented to demonstrate the behavior of the algorithms.

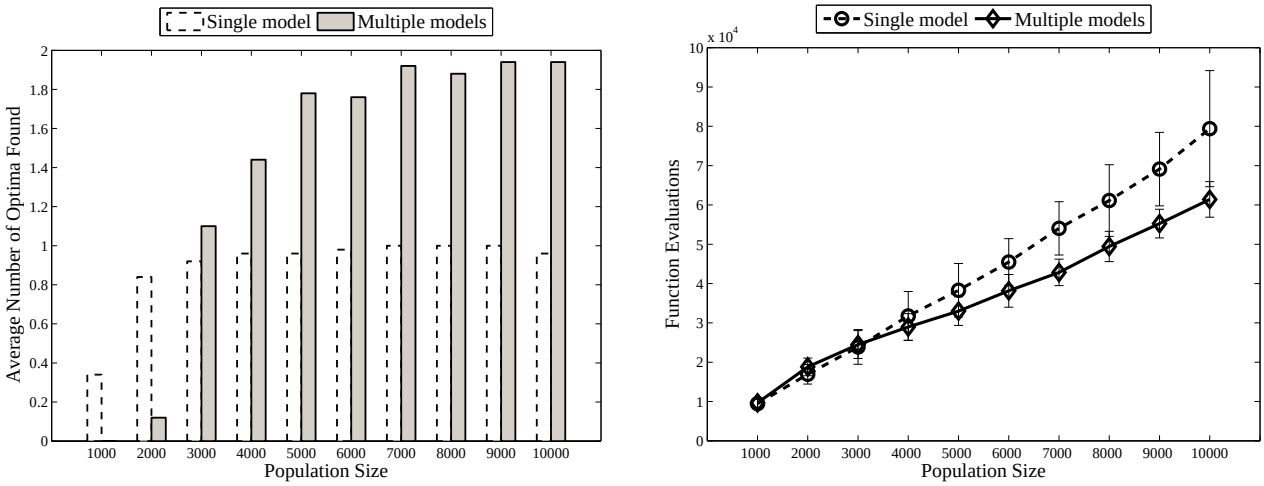
1000 to 10000, and each of these experiments was repeated for 50 times. Tournament selection is adopted as selection method, and we set the tournament size to 16 (which has been reported suitable for ECGA to handle the range of 40- to 80-bit concatenated trap problems [4].) In these experiments, the stopping criterion is set such that a run is terminated when all solutions in the population converge to the same fitness value.

7.2 Empirical Results

The empirical results on test problem F_1 are presented in Figure 1. It can be seen that given sufficient population size, our approach reliably identifies both optima. Comparing to ECGA, it is obvious that no matter how much population size is augmented, ECGA constantly converges to a single optimum. Furthermore, as shown in Figure 1(b), the proposed approach uses less function evaluations when sufficient population size, which makes the algorithm capable of reliably obtaining both global optima, is given. This reduction, as mentioned in section 2, is resulted from utilizing multiple models to prevent the recombination of solutions coming from different basins. In this way, we make the convergence taking shorter time than that needed by ECGA.

Going from two global optima to four global optima, Figure 2 presents the results of experiments on F_2. Similar to the previous experiments on F_1, our approach obtained more global optima than ECGA did, given sufficient population size. The usage of function evaluations shows the same pattern, too: when given large enough population size, the proposed method surpassed ECGA. However, it is also noted that F_2 is harder than F_1 in that sometimes the proposed algorithm didn't obtain all global optima at convergence (four or five times out of 50 runs).

The results on F_3 is the most inspiring. It shows that globally multimodal problem that has basins with heterogeneous linkage is hard to address. As can be seen in Figure 3, even when we use a pretty large population size, the algo-



(a) Number of Optima Obtained

(b) Function Evaluations

Figure 3: Empirical results of the proposed approach (multiple models) compared to ECGA (single model) on test problem F_3. Different population sizes are experimented to demonstrate the behavior of the algorithms.

rithm can not reliably obtain all global optima, comparing to what we have achieved in experiments on F_1. This confirms the previous conjecture that heterogeneous linkage between basins will result in more severe building block disruption. Nevertheless, the proposed approach still performs better than ECGA in both the average number of optima obtained and fitness evaluations spent.

8. SUMMARY

In this paper, we have introduced a multivariate multi-model approach, which builds multiple marginal product models at each generation to guide further search. It iteratively estimates the membership of solutions to each model and readjusts the model structure and parameters. This mechanism is also equipped with a heuristics to choose the number of models to be adopted, which eliminates the need of manual tuning. The empirical evaluations on globally multimodal problems confirm the ability of the proposed approach to obtain more global optima per run compared to extended compact genetic algorithm. Furthermore, the results also suggest that using multiple models can reduce the number of generations spent to reach convergence.

Although we have obtained some promising results on solving globally multimodal problems, it seems that this study arises more issues than it addressed. For example, we are now thinking whether the number of global optima should be included as a factor when measuring the scalability of an optimization algorithm. And, as demonstrated in the results on F_3, the problems that have basins with heterogeneous linkage are obviously more difficult to address, so how can we enhance the approach regarding this aspect of problem difficulty. Furthermore, how can we improve the algorithm to recognize correct partition of population earlier in the run such that the population requirement can be reduced. These questions need further investigations, and we are currently working toward these directions.

Acknowledgments

This research was supported in part by National Science Council of Taiwan under grant NSC98-2631-S-001-001 and and the thematic program of Academia Sinica under grant AS95ASIA02.

9. REFERENCES

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

[2] S. Baluja and S. Davies. Using optimal dependency-trees for combinational optimization. In *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, pages 30–38, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[3] I. Belda, S. Madurga, T. Tarragó, X. Llorà, and E. Giralt. Evolutionary computation and multimodal search: A good combination to tackle molecular diversity in the field of peptide design. *Molecular Diversity*, 11(1):7–21, 2007.

[4] C.-Y. Chuang and Y.-p. Chen. Sensibility of linkage information and effectiveness of estimated distributions. *Evolutionary Computation*, in press.

[5] T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, NY, USA, 1991.

[6] J. de Bonet, C. Isbell, and P. Viola. MIMIC: Finding optima by estimating probability densities. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 424–430. The MIT Press, 1997.

[7] K. Deb and D. E. Goldberg. Analyzing deception in trap functions. In *Foundations of Genetic Algorithms 2*, pages 93–108, 1993.

[8] K. Deb and D. E. Goldberg. Sufficient conditions for deceptive and easy binary functions. *Annals of Mathematics and Artificial Intelligence*, 10(4):385–408, 1994.

[9] L. Emmendorfer and A. Pozo. Effective linkage learning using low-order statistics and clustering.

Evolutionary Computation, IEEE Transactions on, 13(6):1233–1246, Dec. 2009.

[10] R. Etxeberria and P. Larrañaga. Global optimization using bayesian networks. In A. O. Rodriguez, M. S. Ortiz, and R. S. Hermida, editors, *Proceedings of the Second Symposium on Artificial Intelligence (CIMAF-99)*, pages 332–339, Habana, Cuba, 1999.

[11] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[12] G. Harik. Linkage learning via probabilistic modeling in the ECGA. IlliGAL Report No. 99010, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign., 1999.

[13] G. R. Harik, F. G. Lobo, and D. E. Goldberg. The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, 3(4):287–297, November 1999.

[14] C. Hocaoğlu and A. C. Sanderson. Multimodal function optimization using minimal representation size clustering and its application to planning multipaths. *Evolutionary Computation*, 5(1):81–104, 1997.

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

[16] P. Larrañaga and J. A. Lozano. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, volume 2 of *Genetic algorithms and evolutionary computation*. Kluwer Academic Publishers, Boston, MA, October 2001. ISBN: 0-7923-7466-5.

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

[18] H. Mühlenbein and T. Mahnig. FDA: A scalable evolutionary algorithm for the optimization of additively decomposed functions. *Evolutionary Computation*, 7(4):353–376, 1999.

[19] 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*, pages 178–187, London, UK, 1996. Springer-Verlag.

[20] J. M. Peña, J. A. Lozano, and P. Larrañaga. Globally multimodal problem optimization via an estimation of distribution algorithm based on unsupervised learning of bayesian networks. *Evolutionary Computation*, 13(1):43–66, 2005.

[21] M. Pelikan and D. E. Goldberg. Genetic algorithms, clustering, and the breaking of symmetry. In *PPSN VI: Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, pages 385–394, London, UK, 2000. Springer-Verlag.

[22] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz. BOA: The Bayesian optimization algorithm. In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, volume I, pages 525–532, Orlando, FL, 13-17 1999. Morgan Kaufmann Publishers, San Fransisco, CA.

[23] M. Pelikan, D. E. Goldberg, and F. G. Lobo. A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, 21(1):5–20, 2002.

[24] M. Pelikan and H. Mühlenbein. The bivariate marginal distribution algorithm. In R. Roy, T. Furuhashi, and P. K. Chawdhry, editors, *Advances in Soft Computing - Engineering Design and Manufacturing*, pages 521–535, London, 1999. Springer-Verlag.

[25] J. Rissanen. Modelling by shortest data description. *Automatica*, 14:465–471, 1978.