Evaluation Relaxation Using Substructural Information and Linear Estimation

Kumara Sastry[1], Claudio F. Lima[2], and David E. Goldberg[1]

[1]Illinois Genetic Algorithms Laboratory (IlliGAL), Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801

[2]DEEI-FCT, University of Algarve, Campus de Gambelas, 8000-117 Faro, Portugal

ksastry@uiuc.edu, clima@ualg.pt, deg@uiuc.edu

ABSTRACT

The paper presents an evaluation-relaxation scheme where a fitness surrogate automatically adapts to the problem structure and the partial contributions of subsolutions to the fitness of an individual are estimated efficiently and accurately. In particular, the probabilistic model built by extended compact genetic algorithm is used to infer the structural form of the surrogate and a least squares method is used to estimate the coefficients of the surrogate. Using the surrogate avoids the need for expensive fitness evaluation for some of the solutions, and thereby yields significant efficiency enhancement. Results show that a surrogate, which automatically adapts to problem knowledge mined from probabilistic models, yields substantial speedup (1.75–3.1) on a class of boundedly-difficult additively-decomposable problems with and without additive Gaussian noise. The speedup provided by the surrogate increases with the number of substructures, substructure complexity, and noise-to-signal ratio.

Categories and Subject Descriptors

G.1.2 [**Numerical Analysis**]: Approximation; G.1.6 [**Numerical Analysis**]: Optimization; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms

Keywords

Estimation of distribution algorithms, extended compact genetic algorithms, fitness surrogates, efficiency enhancement, evaluation relaxation, linear regression, speed-up

1. INTRODUCTION

One of the promising areas in genetic and evolutionary algorithms (GEAs) is the design and development of *competent* genetic algorithms—GAs that solve hard problems quickly, reliably, and accurately [9]. One such class of competent genetic algorithms is estimation of distribution algorithms (EDAs) [15, 18]—search methods that replace the traditional variation operators of evolutionary algorithms by building and sampling a probabilistic model of promising solutions. EDAs have been shown to successfully solve boundedly difficult problems—both on a single and hierarchical level—oftentimes requiring polynomial (usually sub-quadratic) number of function evaluations. However, even sub-quadratic number of function evaluations can be daunting especially when the fitness evaluation involves complex simulations, computations, and models. This is usually the case with most real-world optimization problems. Therefore, there is a premium on a variety of *efficiency-enhancement* techniques to speedup EDAs.

One such efficiency-enhancement technique commonly used in GEAs, and to a lesser extent in EDAs, is *evaluation relaxation*. In evaluation relaxation, an accurate, but expensive fitness function is replaced with a less accurate, but computationally inexpensive surrogate function. However, surrogate functions are usually of fixed form and often hand-designed based on the search problem. A key advantage of EDAs is that the problem structure is automatically *learned* in terms of a probabilistic model and we can incorporate this knowledge into the design of the surrogate function. Initial studies [19, 26] on the design of surrogate functions that incorporate knowledge of key variable interactions has yielded substantial speedups not possible from naïve surrogate designs. However, the initial effort on incorporating the probabilistic model into surrogate design used a simple schema-theorem based method for fitting coefficients of the surrogate function. While the previous method is accurate on a class of deterministic additively separable problems, the speedups don't carry over to other class problems, specifically noisy and hierarchically-decomposable problems.

Once the structural form of the surrogate is inferred from the probabilistic models, built by EDAs and other methods that automatically identify key variable interactions, we can use various methods of system identification, estimation, and regression to estimate the coefficients of the resulting surrogate. That is what we do here. As a first step, we use the models built by EDAs—specifically the ex-

tended compact genetic algorithm (eCGA) [11]—to decide on the structural form of the surrogate and use least-squares method [7, 12, 14, 20] to estimate the coefficient values of the surrogate function. In other words, the probabilistic model of eCGA yields the functional form of the surrogate function and the least-squares method yields the partial fitness contributions of different subsolutions of the underlying search problem. Moreover, the use of standard and well-established estimation methods such as least squares and recursive least squares will enable us to tap into a rich and established area of estimation, regression and system identification theory and help in designing accurate surrogates that yield maximal efficiency enhancement.

The paper is structured as follows. The next section provides an outline eCGA, followed by a brief review of past work in evaluation relaxation in Section 3. In Section 4 we provide the details of designing a surrogate whose form is inferred from probabilistic models of EDAs, and whose coefficients are estimated using linear regression. In Section 5, we develop facetwise models to understand the strengths and limitations of the proposed evaluation-relaxation scheme and also discuss key empirical results. Finally, we discuss future work followed by summary and conclusions.

2. EXTENDED COMPACT GENETIC ALGORITHM

Estimation of distribution algorithms [15, 18] replace traditional variation operators of genetic algorithms by building a probabilistic model of promising solutions (that survive selection) and sampling the corresponding probability distribution to generate the offspring population. The extended compact genetic algorithm (eCGA) [11] uses a product of marginal distributions on a disjoint partition of variables of the problem to model highly-fit individuals and sample new ones. Each partition of variables corresponds to a linkage group, so that important substructures can be effectively recombined as in a population-wise building block crossover. In eCGA, new solutions are generated from the following probability distribution

$$p(X) = \prod_{i=1}^{m} p(X_{I_i}),\qquad(1)$$

where $X = (X_1, X_2, \ldots, X_\ell)$ is a vector that contains all the variables of the problem and I_i is the index set that contains the index of the variables that belong to the ith marginal distribution.

This kind of probability distribution belongs to a class of probabilistic models known as marginal product models (MPMs). For example, the following MPM, [1,3][2][4], for a 4-bit problem represents that the 1st and 3rd variables are linked, and the 2nd and 4th variables are independent.

In eCGA, both the structure and the parameters of the model are searched and optimized to best fit the data (promising solutions). A greedy MPM search, which is performed every generation, starts with the simplest possible model and sequentially searching for more complex ones. That is, starting with a model that treats all variables as independent and then sequentially merging partitions of variables whenever a certain metric is improved. The merging continues until the metric can no longer be improved [24].

The metric used in eCGA to score candidate models is the minimum description length (MDL) metric [21], that penal-

izes both inaccurate and complex models, thereby leading to an (sub)optimal distribution. According to the MDL principle, good models are those under which the representation of the distribution using the current structure, along with the representation of the population compressed under that distribution, is minimal. Numerically, the MPM complexity is given by the sum of model complexity and compressed population complexity. The model complexity quantifies the model representation in terms of the number of bits required to store all the marginal probabilities. The compressed population complexity quantifies the data compression in terms of the entropy of the marginal distribution over all partitions. Same as in other data compression algorithms, the more biased the marginal distributions are, the more the population can be compressed, which means a more effective representation of the non-linearities of the data.

Like traditional GAs, eCGA starts with an initial population (usually randomly generated) that is evaluated and submitted to a selection operator that gives preference to high-quality solutions. The set of selected individuals is then used to build the probabilistic model for the current generation. After the model structure is learned and its parameters are estimated, the offspring population is generated by randomly sampling subsets from the modeled individuals, according to the probabilities of the subsets stored in the MPM. After the offspring is evaluated, the replacement of some (or all) parents by offspring individuals takes place, which ends the eCGA main loop. The next generation proceeds again from the selection phase until some stopping criteria is satisfied.

While eCGA—and other EDAs—are primarily used for efficiently exchanging key sub-structures, they provide additional information about the underlying search problem. The probabilistic model of the population that express (in)dependencies among decision variables of the problem is a good source of information that can be exploited to further enhance EDA performance. Examples of using information from the probabilistic model for another purpose besides mixing are fitness inheritance [26, 19], induction of global neighborhoods for mutation operators [24], hybridization and adaptive time continuation [16], sub-structural niching [23], and on-line [28] population size adaptation, or simply to assist the user in a better interpretation and understanding of the non-linearities of the problem. In this study, we are interesting in exploiting the probabilistic model built by eCGA in inferring the structural form of an inexpensive surrogate function. Before proceeding with the description of the proposed approach for surrogate design, we first provide a brief overview of past work on evaluation relaxation in EDAs.

3. EVALUATION RELAXATION IN EDAS

In *evaluation relaxation*, an accurate, but computationally expensive fitness function is replaced by a less accurate, but inexpensive surrogate function, and thereby the total number of costly fitness evaluations are reduced [2, 10, 13, 19, 22, 26, 27]. The low-cost, less-accurate fitness estimate can either be (1) *exogenous*, as in the case of approximate fitness functions [2, 13, 17], where, external means are used to develop the fitness estimate, or (2) *endogenous*, as in the case of *fitness inheritance* [27] where, some of the offspring fitnesses are estimated based on fitness of parental solutions.

Evaluation relaxation in GAs dates back to early, largely

empirical work of Grefenstette and Fitzpatrick [10] in image registration where significant speedups were obtained by reduced random sampling of the pixels of an image. Approximate models have since been used extensively to solve complex optimization problems in many engineering applications such as aerospace and structural engineering [2, 6]. Following early empirical work design theories have since been developed to understand the effect of approximate surrogate functions on population sizing and convergence time and to optimize speedups. See [22] and the references therein for further details. While surrogates used in evolutionary algorithms can be readily used with EDAs, the probabilistic models of EDAs can be especially helpful in developing accurate surrogates and thereby provide significant speedup [19, 26] when compared to simple *endogenous* surrogates [27]. Since our proposed method is closely related to the *endogenous* probabilistic fitness estimation models of Sastry, Pelikan, and Goldberg [26], we briefly describe their methodology in the following paragraphs.

Sastry, Pelikan, and Goldberg [26] and Pelikan and Sastry [19] proposed a fitness inheritance method for EDAs, specifically for eCGA and the Bayesian optimization algorithm (BOA) [18]. Similar to earlier fitness inheritance study [27], all the individuals in the initial population were evaluated using the expensive fitness function. Thereafter, an offspring was evaluated either using a surrogate with a user-specified inheritance probability p_i, or using the expensive fitness function with a probability $1 - p_i$. However, unlike fitness inheritance method of Smith *et al* [27], Sastry *et al* used the probabilistic models of eCGA to determine the structural form of the surrogate. That is, the MPM used in eCGA, which partitions the variables of the underlying search problem into linkage groups, were used to determine the variable interactions used in the surrogate. Therefore, the process of learning a surrogate model was sub-divided into estimating the fitness contributions of all possible subsolutions in every partition according to the linkage map that is automatically and adaptively identified by the probabilistic model of eCGA. Sastry *et al* used all evaluated parents and offspring in estimating the partial contributions of the subsolutions (or schemata) to the overall fitness of a candidate solution.

Specifically, they used schema theory basis for determining the relative and partial contribution of a schema to the overall fitness. That is, Sastry *et al* defined fitness of a schema h as the difference between the average fitness of individuals that contain the schema and the average fitness of the population:

$$\hat{f}_s(h) = \frac{1}{n_h} \sum_{\{i|x^{(i)} \supset h\}} f(x^{(i)}) - \frac{1}{M} \sum_{i=1}^{M} f(x^{(i)}), \qquad (2)$$

where n_h is the total number of individuals that contain the schema h, $x^{(i)}$ is the ith evaluated individual and $f(x^{(i)})$ its fitness, and M is the total number of individuals that were evaluated. If a particular schema in not present in the evaluated population, its fitness is arbitrarily set to zero.

Pelikan and Sastry [19] used a similar approach for developing a fitness inheritance method in BOA. Given the differences of the probabilistic model evolved by BOA, the methodology for modeling fitness is slightly different from eCGA, but the key idea remains the same: Use the probabilistic model to construct the form of the surrogate, and use

schema-theory basis for determining the partial fitness contribution of subsolutions to the overall fitness. Both studies reported substantial speedup on a class of boundedly-difficult additively-separable problems.

While the automatic and adaptive incorporation of problem knowledge in terms of the probabilistic models built by EDAs and other linkage learners is very powerful, the estimation of the partial fitness contributions of solutions is somewhat ad hoc. Moreover the schema-theory basis for estimating these partial fitness contributions of schemata works well only on certain class of search problems. For example, the fitness inheritance method of Sastry *et al* [26] fails to provide significant speedup on noisy problems, and the method of Pelikan and Sastry [19] fails on hierarchically-decomposable problems.

Therefore, we propose a method to enhance the robustness and accuracy of the surrogate such that substantial speedups are obtained even on problems where the earlier fitness inheritance methods fail. That is, we infer the structure of the surrogate from the probabilistic models and then use standard and robust techniques from system identification—specifically, the least square method—to estimate value of the coefficients (or the partial fitness contributions of subsolutions) of the resulting surrogate function. Details of the surrogate design method is provided in the following section.

4. FITNESS INHERITANCE IN ECGA USING LEAST SQUARES FITTING

Similar to earlier fitness inheritance studies [19, 25, 26, 27], in the proposed method the fitness of all the candidate solutions in the initial population are evaluated using the expensive/accurate fitness function. Thereafter, in the subsequent generations, the fitness of an individual is estimated using the surrogate with probability p_i, or is evaluated using the fitness function with probability $1 - p_i$. Similar to the surrogates used in [26], in the proposed method the surrogate is a polynomial whose order and terms are determined by the probabilistic model and whose coefficients are determined using the evaluated individuals. However, unlike in [26], we use a least squares fitting approach [4, 7, 12, 14, 20] to estimate the coefficient values.

In order to estimate the coefficients of the surrogate, we use both parental and offspring solutions that are evaluated using the fitness function. That is, in the first generation, we use all n evaluated parents and on an average $n \cdot (1 - p_i)$ evaluated offspring to estimate the coefficients, where n is the population size. In the subsequent generations, we use on an average $2n \cdot (1 - p_i)$ evaluated solutions. We limited ourselves to using only the parental and offspring solutions only to keep the proposed method as close to that of Sastry *et al* [26] and therefore obtain a fair comparison between the two methods. However, as mention in Section 6, one of our future goals is to use all or majority of the evaluated solutions to design the surrogate.

Similar to [26], individuals with exact fitness are used to estimate the sub-structural fitnesses of the remaining individuals. These sub-structures that are defined by the probabilistic model can be viewed and directly mapped into schemata. The fitness associated with the different schemas that match an individual is then combined to estimate this fitness. In this study, schema or building-block fitness is defined as the relative (to the average fitness of the population)

fitness contribution to the overall fitness of an individual. The proposed fitness inheritance method differs from [26] in the way the schema fitness is estimated.

After the model is built the linkage groups are treated as building-blocks partitions, thus all possible schemata under this structure are considered. Considering the previous MPM example (Section 2) for a 4-bit problem, whose model is `[1,3][2][4]`, the schemata for which the fitness is predicted are {0*0*, 0*1*, 1*0*, 1*1*, *0**, *1**, ***0, ***1}. The total number of schemas is given by

$$N = \sum_{i=1}^{m} 2^{k_i}, \tag{3}$$

where m is the number of BBs and k_i is the size of the ith BB (number of variables belonging to the BB).

The fitness values of the schemata are estimated as follows. Each individual used for learning is mapped into a binary vector of size N, where each variable of the vector uniquely identifies a given schema. That is, the vector is instantiated by the following delta function

$$\delta(x, h_j) = \begin{cases} 1, & \text{if } x \supset h_j \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where x is the individual to be converted and h_j is the jth schema. Basically, the vector will have value "1" for the schemas that contain individual x and "0" otherwise. After mapping M evaluated individuals using the above function, the following matrix with dimension $(M \times N)$ is obtained:

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,N} \\ a_{2,1} & a_{2,2} & \dots & a_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M,1} & a_{M,2} & \dots & a_{M,N} \end{pmatrix}, \tag{5}$$

where $a_{i,j} = \delta(x^{(i)}, h_j)$, and $x^{(i)}$ denotes the ith individual used for learning the surrogate fitness model. We note that the rank of matrix $\mathbf{A}$ is $N - m + 1$.

Also, the relative (to the average) fitness of each evaluated individual is kept in a vector with dimension $(M \times 1)$ as

$$\mathbf{f} = \begin{pmatrix} f(x^{(1)}) - \bar{f} \\ f(x^{(2)}) - \bar{f} \\ \vdots \\ f(x^{(M)}) - \bar{f} \end{pmatrix}, \tag{6}$$

where $f(x^{(i)})$ is the evaluated fitness of the ith individual used for learning and $\bar{f}$ is the average fitness of all M evaluated individuals (both from parent and offspring population). The average fitness is then given by

$$\bar{f} = \frac{1}{M} \sum_{i=1}^{M} f(x^{(i)}). \tag{7}$$

Given that there are N different schema fitnesses to estimate, the fitness coefficients associated with the N binary variables can be displayed as vector of dimension $(N \times 1)$

$$\hat{\mathbf{f}}_{\mathbf{s}} = \begin{pmatrix} \hat{f}_s(h_1) \\ \hat{f}_s(h_2) \\ \vdots \\ \hat{f}_s(h_N) \end{pmatrix}, \tag{8}$$

where $\hat{f}_s(h_j)$ is the fitness of schema h_j.

The task of estimating the relative fitness of each schema can be stated as finding a vector $\hat{\mathbf{f}}_{\mathbf{s}}$ that satisfies the equality:

$$\mathbf{A}\hat{\mathbf{f}}_{\mathbf{s}} = \mathbf{f}. \tag{9}$$

In practice, this equality might not be entirely satisfied and one must instead seek for minimizing the difference between left and right terms of Equation 9. For that, it is used a multi-dimensional least squares fitting approach. Thus, under the least squares fitting principle the problem of estimating the fitness of schemata can now be reformulated as finding the appropriate values for vector $\hat{\mathbf{f}}_{\mathbf{s}}$ such that the following squared error function χ^2 is minimized:

$$\chi^2 = \left(\mathbf{A}\hat{\mathbf{f}}_{\mathbf{s}} - \mathbf{f}\right)^T \left(\mathbf{A}\hat{\mathbf{f}}_{\mathbf{s}} - \mathbf{f}\right). \tag{10}$$

The solution to the above problem is a well-known result from literature, therefore details on the resolution are not provided and the interested reader should refer elsewhere [4, 7, 12, 14, 20]. The method used in this paper to perform multi-dimensional least squares fitting was provided by GNU Scientific Library[1] (GSL).

After obtaining the estimates for schema fitnesses, the estimation of an individual's fitness is a straightforward process that consists in summing the average fitness of the population to the fitness of each schema that contains the individual being considered. The estimated fitness of an individual x is then given by

$$f_{inh}(x) = \bar{f} + \sum_{j=1}^{N} \delta(x, h_j)\hat{f}_s(h_j), \tag{11}$$

where $\hat{f}_s(h_j)$ is given by the jth element of vector $\hat{\mathbf{f}}_{\mathbf{s}}$.

It can be shown that the surrogate obtained by using a structure inferred from a perfect model and the coefficients via least squares is identical to that obtained using a Walsh basis [8] of the accurate fitness function. This clearly suggests that given an accurate probabilistic model, we can obtain a surrogate that accurately estimates the fitness of untested solutions. We note that while eCGA can only discover non-overlapping substructures, the proposed surrogate design method can be readily used with other model-building GAs including BOA. The surrogates inferred from both BOA and DSMGA are also in the form of polynomials—usually with more terms than those obtained in eCGA—which are linear in terms of unknown coefficients. We have conducted limited tests on problems with overlapping substructures and the surrogates developed using representative models accurately estimate the fitnesses of untested solutions.

In the following section we discuss the results obtained by the proposed evaluation-relaxation scheme on a class of boundedly-difficult additively-decomposable problems with and without additive Gaussian noise.

5. RESULTS AND DISCUSSIONS

We begin with a description of the test problems used in this study, followed by details of the experimental procedure, population-sizing requirements, and limits on p_i. We then present the speedups provided by a surrogate that automatically adapts to the structure of the search problem.

---

[1]http://www.gnu.org/software/gsl/

5.1 Test Problems

Two different problems are used to test the proposed method: OneMax and $m - k$ Trap functions. In OneMax the fitness is given by the sum of ones in a binary string:

$$f_{OneMax}(X) = \sum_{i=1}^{\ell} X_i. \tag{12}$$

This is a simple linear function with the optimum in the solution with all ones. Therefore, there is no need of linkage learning to be able to solve this problem. While the optimization of the OneMax problem is easy, the probabilistic models built by EDAs such as eCGA and BOA, however, are known to be only partially correct and include spurious linkages. Therefore, the results of the evaluation-relaxation scheme on the OneMax problem will indicate if the effect of using partially correct linkage mapping on the accuracy of the surrogate is significant. In this paper a OneMax function with size $\ell = 50$ is used.

The second function considered is a concatenated $m - k$ deceptive trap problem [1, 5], where the accurate identification and exchange of BBs are critical to EDA success:

$$f_{mk}(X) = \sum_{i=0}^{m-1} f_{Trap}(x_{ki}, x_{ki+1}, \ldots, x_{ki+k-1}), \tag{13}$$

where m is the number of concatenated k-bit deceptive trap functions. The k-bit trap function is defined as follows:

$$f_{trap}(u) = \begin{cases} 1 & \text{if } u = k \\ (1 - d) * \left(1 - \frac{u}{k-1}\right) & \text{otherwise} \end{cases} \tag{14}$$

where u is the number of ones in the string, k is the size of the trap function, and d is the fitness signal between the global optimum and the deceptive optimum ($d = 1/k$ is used). In our experiments we use Trap functions with size $k = \{4, 5\}$, and concatenate 10 copies of each function ($m = 10$).

To test the proposed evaluation-relaxation method on noisy problems, we use $10-4$ trap function with the additive Gaussian noise with zero mean and two different noise-to-signal ratios: $\frac{\sigma_N^2}{(\sigma_f/d)^2} = \{0.01, 0.1\}$, where σ_N^2 is the noise variance. The empirical results obtained in this study are averaged over 50 independent runs. The stopping criteria used in each run is to obtain a solution with at least $m - 1$ building-blocks solved, that is, the optimal solution with an error of $\alpha = 1/m$. Tournament selection without replacement was used with size $s = 8$.

5.2 Population Sizing and Limits on p_i

In the following paragraphs, we describe the population-sizing requirements of the surrogate model, and the optimal value of p_i that yields maximum speedup. Sastry *et al* [26] modeled the error in the fitness estimation by the surrogate as an additive Gaussian noise and predicted that the minimum population size required by eCGA to successfully solve additively-decomposable problems. The minimum population size varies with p_i as

$$n = n_0(1 + p_i), \tag{15}$$

where n_0 is the minimum population size required to correctly identify the linkage groups used when the surrogate is not used ($p_i = 0$). The empirical values of n_o for the test problems used in this study are shown in Table 1 [24]. In this

Table 1: Population size, n_0, used for different problems with size ℓ ($m \times k$), when fitness inheritance is not used ($p_i = 0$).

Problem	ℓ	n_0
OneMax	50	135
Trap, $k = 4$	40	1550
Trap, $k = 5$	50	4150
Trap, $k = 4$, $\frac{\sigma_N^2}{(\sigma_f/d)^2} = 0.01$	40	1914
Trap, $k = 4$, $\frac{\sigma_N^2}{(\sigma_f/d)^2} = 0.1$	40	4430

study, we use the population size according to Equation 15.

Now the question remains as to what the optimal value of p_i—or the probability of using the surrogate—would be that yields maximum speedup? From Section 4 we know that the rank of matrix $\mathbf{A}$ is $N - m + 1$. For additively separable problems with m building blocks of size k each, the rank of $\mathbf{A}$ is $(2^k - 1) \cdot m + 1$. Therefore the minimum number of equations required to use the least-squares method should be greater than $rank(\mathbf{A})$. That is, the minimum number of individuals required for estimating the coefficients is

$$M \geq (2^k - 1)m + 1. \tag{16}$$

Note that since we evaluate on an average $n(1 - p_i)$ parents and $n(1 - p_i)$ offspring, $\bar{M} = 2n(1 - p_i)$. Using this relation and rearranging for n, the minimal population size required to perform least squares fitting is given by

$$n \geq \frac{(2^k - 1) \cdot m + 1}{2(1 - p_i)}. \tag{17}$$

To obtain an upper bound for the proportion of inheritance p_i we use the relation from Equation 15 and solving Equation 17 for p_i, we obtain

$$p_i \leq \sqrt{1 - \frac{(2^k - 1)m + 1}{2n_0}}. \tag{18}$$

Using the approximation $(2^k - 1)m + 1 \approx 2^k m$ and the following population-sizing model for n_0 [24]:

$$n_0 = c_0 2^k m^{1.5} \frac{\sigma_{BB}^2}{d^2}, \tag{19}$$

where c_0 is a problem-dependent constant, σ_{BB}^2 is the fitness variance of a BB partition (or substructure), and d is the signal difference between the two most competing BBs, Equation 18 can now be rewritten as

$$p_i \leq \sqrt{1 - \frac{1}{c_1\sqrt{m}}}, \tag{20}$$

where $c_1 = 2c_0 \frac{\sigma_{BB}^2}{d^2}$. Equation 20—which is depicted in Figure 1 for $m - k$ deceptive trap problems with $k = 4$, and 5—suggests that as the number of building blocks of the search problem increases, the maximum probability of estimating the fitness of an individual using the surrogate asymptotically reaches 1. The maximum p_i for $k = 5$ is slightly higher than that for $k = 4$ because the term c_1 is loosely dependent on k and increases linearly with k. Calculating the upper bounds for $10-4$ and $10-5$ deceptive trap using Equation 20, we obtain $p_i = 0.972$ and $p_i = 0.979$, respectively. Empirically, we obtain the optimal p_i that yields
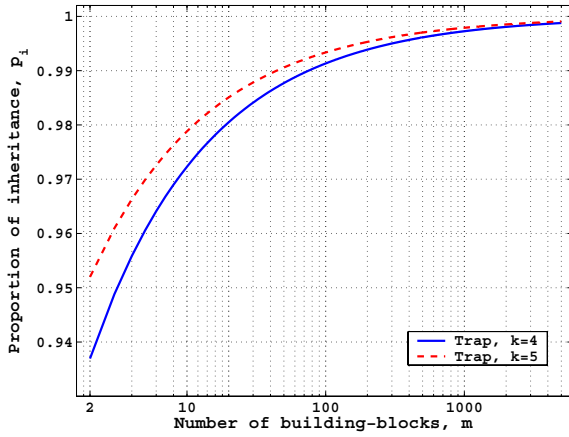
Figure 1: Theoretical curves for the upper bound of proportion of inheritance (p_i) that can be used for the trap function with $k = \{4, 5\}$. The upper bounds for $10-4$ and $10-5$ deceptive trap functions as predicted by theory is $p_i = 0.972$ and $p_i = 0.979$, respectively. Empirically, we obtain $p_i = 0.972$, and $p_i = 0.978$, which are in close agreement with the predicted values.

maximum speedup to be $p_i = 0.972$ and $p_i = 0.978$ for $10-4$ and $10-5$ deceptive trap functions, respectively. As a side note, the model proposed by Sastry *et al* [26] did not predict an upper limit on p_i. That is, while the model suggests $p_i = 1$ to be the optimal value, empirical results clearly suggested an upper bound on p_i, for both eCGA and BOA, beyond which the speedup reduced [26, 19]. Interestingly, using Equation 20, we can predict the upper limit of p_i, using the appropriate values of c_1 not only for eCGA but also for BOA, which agrees with empirical results.

5.3 Speedup

To determine the speedup obtained by the proposed evaluation-relaxation scheme, we compute the total number of expensive/accurate fitness evaluations performed to successfully solve at least $m-1$ building blocks for different values of p_i. As mentioned earlier we scale the population size according to Equation 15 and empirically we find that the convergence time is constant with respect to p_i. The speedup of the evaluation-relaxation scheme is given by the ratio of number of function evaluations required when the surrogate is not used to that required when it is used:

$$\eta = \frac{n_{fe}(p_i)}{n_{fe,0}}, \qquad (21)$$

Since all the solutions of the initial population are evaluated and subsequently on an average $n(1-p_i)$ solutions are evaluated. Therefore, the total number of evaluations performed is given by $n + n \cdot t_c \cdot (1-p_i)$. Substituting this relation and Equation 15 in Equation 21, we get

$$\eta = \frac{1}{1 - p_i^2 + \frac{1}{t_c} p_i(1 + p_i)} \leq \frac{1}{1 - p_i^2} \qquad (22)$$

Note that in developing the facetwise model for the speedup, as in previous studies, we neglected the computational cost of the surrogate. We plot the speedup obtained by using
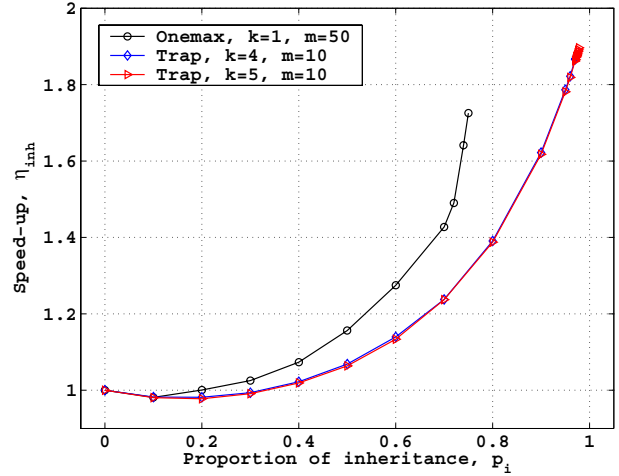

Figure 2: The speedup obtained by using the evaluation-relaxation scheme for OneMax, 10-4 and 10-5 deceptive trap functions. The results are averaged over 50 independent runs. The speedups obtained by the proposed evaluation-relaxation scheme is comparable to those obtained through the endogenous fitness estimation models of Sastry *et al* [26].

the surrogate as a function of p_i for the 50-bit OneMax, 10-4 and 10-5 trap problems in Figure 2. As predicted by the model (Equation 22), the speedup increases with p_i till a certain point which is bounded by Equation 20. As mention in the previous section, beyond a maximum value of p_i, the number of evaluated individuals is not enough to satisfy the minimum requirements of the least-squares method. At this value of p_i, the proposed evaluation-relaxation scheme yields maximum speedup. That is, for OneMax we obtain a maximum speedup of 1.73 at $p_i = 0.75$, for 10-4 trap we obtain a maximum speedup of 1.87 at $p_i = 0.972$, and for 10-5 trap we obtain a maximum speedup of 1.9 at $p_i = 0.978$. Moreover, the speedups obtained are similar to those reported by Sastry *et al*[26]. Using Equations 22 and 20, we can obtain an upper bound on the maximum speedup:

$$\eta^* \leq \frac{c_1}{1 + c_2} \sqrt{m} \qquad (23)$$

where c_2 is related to the convergence time: $c_2 \approx 2c_1/c_t$, and c_t is a constant related to the selection intensity and BB size. The above facetwise model suggests that the maximum speedup obtained by using the surrogate increases with the number of subsolutions which is very promising.

Following promising results of the proposed evaluation-relaxation scheme on deterministic problems, we also tested the use of surrogate on additively-decomposable problems with additive Gaussian noise. The speedup obtained for 10-4 trap problem with additive Gaussian noise as a function p_i at two different noise-to-signal ratio are presented in Figure 3. Similar to the deterministic noise, the speedup increases with p_i and reaches a maximum value predicted by Equation 20. We also note that while the endogenous fitness inheritance method of Sastry *et al* [26] fails to provide adequate speedup on noisy problems, the surrogate design using probabilistic models and linear estimation methods yields significant speedup for noisy problems as well. For the
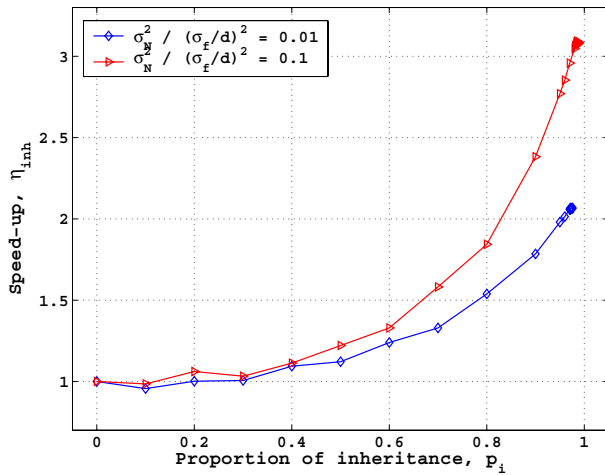
Figure 3: Function evaluation ratio and speedup obtained for 4-bit Trap function with exogenous noise, $\frac{\sigma_N^2}{(\sigma_f/d)^2} = \{0.01, 0.1\}$. The results are averaged over 50 independent runs.

noisy 10-4 trap problem the maximum speedup obtained is 2.1 at $p_i = 0.976$ for noise-to-signal ratio of 0.01 and 3.1 at $p_i = 0.989$ for noise-to-signal ratio of 0.1. Indeed with the proposed method the speedup increases with the noise-to-signal ratio. With increasing noise, higher population sizes are required to successfully solve the problem, and as predicted by the facetwise models (Equations 20, and 22), the fitness for a higher proportion of the population can be estimated using the surrogate and would yield a higher speedup.

6. FUTURE WORK

The results of this study are very promising and warrants additional research in multiple avenues, some of which we have already begun:

Maximize usage of available data for learning and designing the surrogate. In this paper, we only used evaluated parents and offspring—in order to compare the performance of the proposed method with those reported in [26]—to learn the coefficients of the surrogate. However, as the time progresses, we accumulate more evaluations which can be readily used for learning the surrogate. Therefore, we are currently investigating different strategies for using data such as using all evaluated solutions, or using only recently evaluated solutions that yields maximum speedup.

Use recursive least squares (RLS). In this paper, the structural form of the surrogate and the coefficients are learned from scratch every generation. However, a more efficient way for learning and updating the coefficients of the surrogate would be using the recursive least squares (RLS) method [3, 12, 20]. With RLS we can accumulate the knowledge gained in the surrogate over time without explicitly storing *all* evaluated solutions. For non-stationary problems, RLS with some forgetting factor can be used, where more attention is paid to recent data and less relevance is given to old data. However, the structure of

the surrogate, inferred from the probabilistic model built every generation, can change quite significantly over time. This might be a potential stumbling block for using RLS.

Test on problems with overlapping subsolutions and high epistasis. In our experiments we have considered non-overlapping BBs, however, many problems can have different BBs that share common components or have high epistasis. We have performed limited tests, with success, to verify if we can learn an accurate surrogate for problems with overlapping subsolutions and high epistasis. Therefore, we will investigate the speedup obtained through the surrogate in BOA and DSMGA which can both capture overlapping substructures.

Test on hierarchical problems. An important class of nearly decomposable problems is hierarchical problems, in which the variable interactions are present at more than a single level. We are in the process of implementing the evaluation-relaxation scheme with the hierarchical BOA [18] and studying the performance of the surrogate on boundedly-difficult hierarchically-decomposable problems.

Apply to real-world optimization problems. The main motivation to incorporate fitness inheritance in GAs is to reduce the number of fitness evaluations for problems where fitness evaluation is expensive. Therefore, experiments on real-world problems with costly evaluations should be performed to validate the proposed approach as an applied efficiency enhancement technique.

7. SUMMARY & CONCLUSIONS

In this paper, we propose an evaluation-relaxation scheme where a surrogate function is designed to incorporate problem knowledge mined by EDAs and potentially other linkage learning methods to estimate the fitness of some solutions in the population. The structural form of the surrogate is inferred from the probabilistic model built by the extended compact genetic algorithm and a least squares method is used to accurately estimate the coefficient values of the surrogate function. Results show that the proposed methodology yields substantial speedup—1.75–3.1—on a class of deterministic and noisy additively decomposable problems.

We also developed facetwise models to predict an upper bound on the probability of estimating the fitness using the surrogate as opposed to evaluating using the expensive fitness function. The models suggest that the optimal probability asymptotically reaches 1 as the number of building blocks increases. In other words, as the number of BBs increase, we can use the surrogate for estimating fitness of almost *entire* population and evaluate very few individuals. We also developed a simple model for predicting the upper bound on the speedup which suggests that the maximum speedup increases as $\sqrt{m}$, where m is the number of BBs. That is, we find—both from the model and from empirical results—that by using the surrogate, the speedup increases with the number of substructures, substructure complexity, and the noise-to-signal ratio.

While the fitness function of the test problems used in this study are not at all expensive, the results nevertheless clearly show that the proposed evaluation-relaxation scheme is highly promising and that we can expect considerable speedups with real-world problems—especially those

bounded by the test problems. The accuracy and efficiency of the surrogate design can be enhanced in a number of ways by using concepts from rich areas such as system identification, estimation, and regression. More importantly, the proposed scheme is an effective combine of competence and efficiency-enhancement and is a primary part of principled efficiency-enhancement techniques that promise *supermultiplicative* speedups.

Acknowledgments

8. REFERENCES

[1] D. H. Ackley. *A connectionist machine for genetic hill climbing*. Kluwer Academic Publishers, 1987.

[2] J.-F. M. Barthelemy and R. T. Haftka. Approximation concepts for optimum structural design—a review. *Structural Optimization*, 5:129–144, 1993.

[3] D. P. Bertsekas. Incremental least squares methods and the extended Kalman filter. *SIAM Journal on Optimization*, 6(3):807–822, 1996.

[4] Å. Björk. *Numerical method for least squares problems*. SIAM, Philadelphia, PA, 1996.

[5] K. Deb and D. E. Goldberg. Analyzing deception in trap functions. *Foundations of Genetic Algorithms*, 2:93–108, 1992. (Also IlliGAL Report No. 91009).

[6] J. E. Dennis and V. Torczon. Managing approximate models in optimization. In N. M. Alexandrov and M. Y. Hussaini, editors, *Multidisciplinary Design Optimization: State-of-the-Art*, pages 330–347, Philadelphia, PA, 1997. SIAM.

[7] N. R. Draper and H. Smith. *Applied regression analysis*. John Wiley & Sons, New York, USA, 1966.

[8] D. E. Goldberg. Genetic algorithms and Walsh functions: Part I, a gentle introduction. *Complex Systems*, 3:129–152, 1989.

[9] D. E. Goldberg. *Design of innovation: Lessons from and for competent genetic algorithms*. Kluwer Academic Publishers, Boston, MA, 2002.

[10] J. J. Grefenstette and J. M. Fitzpatrick. Genetic search with approximate function evaluations. *Proceedings of the International Conference on Genetic Algorithms and Their Applications*, pages 112–120, 1985.

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

[12] S. Haykin. *Adaptive filter theory*. Prentice Hall, 1996.

[13] Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing Journal*, 9(1):3–12, 2005.

[14] T. Kailath, A. H. Sayed, and B. Hassibi. *Linear estimation*. Prentice-Hall, Upper Saddle River, NJ, 2000.

[15] P. Larrañaga and J. A. Lozano, editors. *Estimation of distribution algorithms*. Kluwer Academic Publishers, Boston, MA, 2002.

[16] C. F. Lima, K. Sastry, D. E. Goldberg, and F. G. Lobo. Combining competent crossover and mutation operators: A probabilistic model building approach. *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 735–742, 2005.

[17] X. Llorà, K. Sastry, D. E. Goldberg, A. Gupta, and L. Lakshmi. Combating user fatigue in iGAs: Partial ordering, support vector machines, and synthetic fitness. *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1363–1370, 2005.

[18] M. Pelikan. *Hierarchical Bayesian optimization algorithm: Toward a new generation of evolutionary algorithm*. Springer Verlag, Berlin, 2005.

[19] M. Pelikan and K. Sastry. Fitness inheritance in the Bayesian optimization algorithm. *Proceedings of the Genetic and Evolutionary Computation Conference*, 2:48–59, 2004. (Also IlliGAL Report No. 2004009).

[20] C. R. Rao and H. Toutenburg. *Linear models: Least squares and alternatives*. Springer, Berlin, 1999.

[21] J. J. Rissanen. *Stochastic complexity in statistical inquiry*. World Scientific Publishing Co., Singapore, 1989.

[22] K. Sastry. Evaluation-relaxation schemes for genetic and evolutionary algorithms. Master's thesis, University of Illinois at Urbana-Champaign, General Engineering Department, Urbana, IL, 2001.

[23] K. Sastry, H. A. Abbass, D. E. Goldberg, and D. D. Johnson. Sub-structural niching in estimation of distribution algorithms. *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 671–678, 2005. (Also IlliGAL Report No. 2005003).

[24] K. Sastry and D. E. Goldberg. Designing competent mutation operators via probabilistic model building of neighborhoods. *Proceedings of the Genetic and Evolutionary Computation Conference*, 2:114–125, 2004. Also IlliGAL Report No. 2004006.

[25] K. Sastry, D. E. Goldberg, and M. Pelikan. Don't evaluate, inherit. *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 551–558, 2001. (Also IlliGAL Report No. 2001013).

[26] K. Sastry, M. Pelikan, and D. E. Goldberg. Efficiency enhancement of genetic algorithms via building-block-wise fitness estimation. *Proceedings of the IEEE International Conference on Evolutionary Computation*, pages 720–727, 2004.

[27] R. Smith, B. Dike, and S. Stegmann. Fitness inheritance in genetic algorithms. In *Proceedings of the ACM Symposium on Applied Computing*, pages 345–350, New York, NY, USA, 1995. ACM.

[28] T.-L. Yu, K. Sastry, and D. E. Goldberg. Online population size adjusting using noise and substructural measurements. *Proceedings of the IEEE Congress on Evolutionary Computation*, 3:2491–2498, 2005. (Also IlliGAL Report No. 2005017).