Eugenic Evolution Utilizing a Domain Model

Matthew Alden
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712
malden@cs.utexas.edu

Aard-Jan van Kesteren
Department of Computer Science
University of Twente
Enschede, The Netherlands
ajvankesteren@hotmail.com

Risto Miikkulainen
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712
risto@cs.utexas.edu

Abstract

In this paper we introduce The Eugenic Algorithm with Modeling (TEAM), an evolutionary search algorithm that employs statistical analysis to promote construction of high-fitness chromosomes. A model of gene/fitness correlations is automatically generated to direct the construction process. When applied to the combinatorial optimization problems of finding a maximally weighted cut in a graph and minimizing the two-dimensional Rosenbrock function, TEAM performs well compared to other evolutionary algorithms at evolving high-fitness solutions.

1 INTRODUCTION

The field of combinatorial optimization focuses on problems with a finite number of possible solutions. For many such problems, an optimal solution cannot be found analytically, or the problem is too large for exhaustive search. This class of interesting problems includes the traveling salesman problem, maximally weighted cut in a graph, integer programs, the subset sum problem, and maximal clique in a graph. Near-optimal solutions can often be found reasonably fast using techniques such as hill-climbing and simulated annealing (SA; Kirkpatrick and Sherrington, 1988). These standard techniques operate by incrementally improving suboptimal solutions. Evolutionary algorithms, which utilize optimization strategies modeled after biological evolution, implement a more global search and have been shown to be particularly powerful on combinatorial optimization problems.

Numerous evolutionary algorithms have been designed to operate on a population of binary chromosomes, a convenient structure for encoding solutions to combinatorial problems. The traditional approach, exemplified by genetic algorithms (GAs; Holland, 1975), produces new chromosomes via recombination of existing chromosomes, with a component of mutation. The information inherited by a single new chromosome is derived from only a small percentage of the total information present in the population.

Recently, evolutionary algorithm research has progressed towards increasingly constructive techniques for generating new chromosomes. Among these algorithms are Binary Simulated Crossover (BSC; Syswerda, 1993), Population Based Iterative Learning (PBIL; Baluja, 1994), and the Eugenic Algorithm (EuA; Prior, 1998). These algorithms construct chromosomes based on information in the entire chromosome population. A probability indicating allele preference is calculated for every gene, and these probabilities are then used to bias the allele selection process towards an estimated ideal chromosome.

This paper presents a second-generation constructive algorithm, The Eugenic Algorithm with Modeling (TEAM), that applies statistical methods to the identification of desirable alleles. A model of gene/fitness correlations is built and used during chromosome construction to aid allele selection.

We next describe the TEAM algorithm. In Section 3 we present and discuss the results of applying TEAM to two combinatorial optimization problems, finding a maximally weighted cut in a graph and minimizing the two-dimensional Rosenbrock function (De Jong, 1975), showing that TEAM performs better than stochastic hillclimber, standard GA, and EuA. Finally, we speculate how the algorithm might be improved by using different statistical tests and generalizing the model.

2 TEAM

2.1 OVERVIEW

At a high level, TEAM operates similar to other GAs. A population of chromosomes is evolved based on feedback from a chosen chromosome evaluation function, $f(c)$. Augmenting the standard chromosome population, TEAM maintains a gene/fitness correlation model, hereafter referred to as the *model*, and several sets of fitness values used in updating the model, hereafter referred to as the *model statistics*. For every gene, these additional structures are used to estimate the allele with the greatest likelihood of producing a high-fitness chromosome.

The algorithm begins by initializing three data sets as follows: (1) build the initial population, (2) initialize the model statistics based on the initial population, (3) initialize the model based on the model statistics. Population evolution proceeds via repetition of the following steps: (1) create a new chromosome, (2) select and replace an existing chromosome in the population with the new chromosome, (3) update the model statistics using the new chromosome, (4) reconstruct the model if necessary. Evolution terminates after a specified number of generations has elapsed.

Chromosome creation involves assigning an allele to every gene in the genome. The first genes assigned are those with the strongest observed influence on fitness. The allele with the strongest observed correlation to high fitness is assigned to each gene. Allele/fitness correlation is partially based on previous allele assignments, therefore a partially constructed chromosome contributes to its own construction.

After the entire chromosome has been constructed, a chromosome in the existing population is selected for extinction by one of several heuristics. The selected chromosome is removed from the population, and the newly constructed chromosome added. The model statistics are updated with information about the new chromosome. The model is updated periodically, after a specified number of elapsed generations since the previous update.

2.2 CHROMOSOME CREATION

Chromosome creation involves ordering the genes and then assigning an allele to every gene. The assignment of an allele x to gene g_i is called a *binding*, written $g_i = x$. For binary chromosomes, $x \in \{0, 1\}$. A chromosome c of length l is defined by a set of bindings, $\{c_{g_1}, c_{g_2}, \ldots, c_{g_l}\}$, where c_{g_i} is the allele of gene g_i in chromosome c. A chromosome c is said to satisfy a set of bindings S if for every binding $g_i = x$ in S, $c_{g_i} = x$.

The first step of chromosome creation, gene ordering, is based on *selectivity*. The selectivity of a gene is an estimation of how clearly the observed fitness values in the population suggest a particular allele for that gene. For example, if the set of fitness values of chromosomes satisfying binding $g_i = 0$ is $\{12, 15, 16, 17\}$, and the set of fitness values of chromosomes satisfying binding $g_i = 1$ is $\{18, 19, 20, 23\}$, then the gene has high selectivity; the allele 1 appears to consistently lead to higher fitness chromosomes. Alternatively, if the respective sets of fitness values were $\{12, 15, 19, 20\}$ and $\{16, 17, 18, 23\}$, the gene's selectivity would be low. Genes are ordered by decreasing selectivity. The selectivity of a gene g_i relative to population P is estimated by:

$$Sel(g_i, P) = \begin{cases} 1.0, \text{if } \left|[P]^F_{g_i=0}\right| \leq 1 \vee \left|[P]^F_{g_i=1}\right| \leq 1 \\ t\_test\left([P]^F_{g_i=0}, [P]^F_{g_i=1}\right), \text{otherwise} \end{cases},$$

where $[P]^F_{g_i=x}$ is the set of fitness values of chromosomes from population P that satisfy binding $g_i = x$. Function t_test returns one minus the observed significance level (OSL) used in Student's t-test (Press et al., 1992), a statistical hypothesis test indicating whether two sample sets are believed to come from the same normal distribution. A high value for t_test, near 1, indicates that the sets are probably not from the same distribution, while a low value, near 0, indicates that there is insufficient evidence to make that distinction.

Once the genes have been ordered, they are sequentially bound to alleles in this order. Set B, initially empty, contains the bindings that define the partially constructed chromosome. As each gene is bound, the resulting binding is added to set B.

To select an allele for gene g_i, a set B'_i is first calculated. B'_i is a subset of B containing the bindings deemed most relevant to the pending binding of gene g_i. Construction of B'_i is described in Section 2.3. An estimate of the allele, x_i, for gene g_i most likely to lead to a high-fitness chromosome is calculated from B'_i as:

$$x_i = \begin{cases} 0, \text{if } \overline{[P]^F_{B'_i \cup g_i=0}} > \overline{[P]^F_{B'_i \cup g_i=1}} \\ 1, \text{otherwise} \end{cases},$$

where $[P]_{B'_i \cup g_i=y}$ is the subpopulation of P composed of chromosomes satisfying all bindings in the set $B'_i \cup g_i = y$, and $[P]^F_{B'_i \cup g_i=y}$ is the set of fitness values of chromosomes in that subpopulation. The gene g_i is bound to allele x_i with probability $1 - \frac{1}{2}(1 - Sel(g_i, [P]_{B'_i}))^\alpha$, otherwise gene g_i is bound to allele $1 - x_i$. Parameter α regulates the dependence of allele selection on selectivity. This probabilistic scheme assigns alleles based on confidence in

expected outcome. Statistically influential genes are more likely to be assigned x_i, while genes not appearing to strongly influence fitness are assigned more randomly. In this way, the system performs neighborhood search to fine tune less influential genes. As the last step, to promote diversity, the binding of gene g_i is mutated with probability p_m.

2.3 THE MODEL

The model's primary purpose is provide a basis for estimating gene epistatis (Davidor, 1991). The model records the observed relative influence of genes on chromosome fitness in a set of gene rankings. Each ranking describes the influence of a single gene on fitness when used in conjunction with other genes. For example, a potential model for a population of chromosomes with four genes might look like the following:

$$\begin{array}{llll} g_0: & g_3 & g_1 & g_2 \\ g_1: & g_2 & g_0 & g_3 \\ g_2: & g_0 & g_1 & g_3 \\ g_3: & g_2 & g_0 & g_1 \,. \end{array}$$

For each gene, all other genes are sorted based on the quantity $U_r(g_i, g_j)$, an estimation of the amount of influence the combination of genes g_i and g_j has on fitness. An ordered pair of genes, (g_i, g_j), is considered highly influential if for $g_j = x$, the set of estimated fitness values of chromosomes satisfying the bindings $\{g_i = 0, g_j = x\}$ is statistically different from the set of estimated fitness values of chromosomes satisfying $\{g_i = 1, g_j = x\}$. A high U_r value, near 1, indicates high influence, and a low value, near 0, indicates low influence. If $U_r(g_i, g_x) > U_r(g_i, g_y)$ then gene g_x appears before gene g_y in the entry for gene g_i. $U_r(g_i, g_j)$ is calculated as:

$$U_r(g_i, g_j) = \frac{1}{4} \begin{pmatrix} t\_test(F_{g_i=0}, F_{g_i=0,g_j=0}) & + \\ t\_test(F_{g_i=0}, F_{g_i=0,g_j=1}) & + \\ t\_test(F_{g_i=1}, F_{g_i=1,g_j=0}) & + \\ t\_test(F_{g_i=1}, F_{g_i=1,g_j=1}) & \end{pmatrix},$$

The sets F comprise the model statistics. $F_{g_i=x}$ is the set of fitness values of all previously evaluated chromosomes that satisfy binding $g_i = x$. For a chromosome of length l, there are $2l$ possible single bindings, and therefore $2l$ $F_{g_i=x}$ sets. $F_{g_i=x,g_j=y}$ is the set of fitness values of all evaluated chromosomes which satisfy bindings $\{g_i = x, g_j = y\}$. There are $4l^2$ such $F_{g_i=x,g_j=y}$ sets, though all sets such that $i = j$ are unused. These sets F act as a global history; as evolution proceeds the fitness value of every evaluated chromosome is a member of l of the $F_{g_i=x}$ sets and $l(l-1)$ of the $F_{g_i=x,g_j=y}$ sets. $\overline{F_{g_i=x}}$ and $\overline{F_{g_i=x,g_j=y}}$ are estimations of the fitness value of any chromosome satisfying a specified single or double binding. Although these sets are incomplete, we assert that they eventually contain sufficient information to support $U_r(g_i, g_j)$.

During allele selection, a set B'_i is calculated from set B, the set of bindings comprising a partial chromosome. Let $h_{g_i,j}$ be the j^{th} gene in the model entry for g_i. B'_i is initially empty, and for each gene $h_{g_i,j}$ in the model entry for g_i, proceeding via increasing j, we perform the following test: if $h_{g_i,j}$ is part of a binding $b \in B$ and $\left|[P]_{B'_i \cup b}\right| \geq n_{min}$ then add b to B'_i. Parameter n_{min} specifies the minimum number of chromosomes that must be present in $[P]_{B'_i}$.

2.4 REPLACEMENT POLICY

After a new chromosome has been created, an existing chromosome is selected for removal from the population. A typical heuristic for extinction is poor fitness, i.e. the least fit chromosome is removed. However, based on the information in the model we can make a more intelligent selection. We can remove the chromosome that contributed the least to the construction of the new chromosome.

In the course of creating a new chromosome, the sets B'_i are calculated l times. The more times a chromosome appears in instances of $[P]_{B'_i}$, the more that chromosome has contributed to the construction of the new chromosome. Therefore, a reasonable extinction heuristic is to remove the chromosome that appeared least often in instances of $[P]_{B'_i}$. This heuristic regards chromosomes as informational units rather than simple patterns for high fitness, promoting the retention of information in the population.

These policies are called *replace_worst_fitness* and *replace_worst_contributor*, and they will be tested experimentally below.

3 EXPERIMENTS

TEAM was evaluated on two combinatorial optimization problems and compared to two standard algorithms as well as EuA, the predecessor of TEAM.

3.1 ALGORITHMS

3.1.1 Stochastic Hillclimber

A simple Stochastic Hillclimber was used as a strawman in these experiments. The hillclimber maintains one chromosome. Each generation, a new chromosome

is created solely by mutating the existing chromosome. The probability of each bit being mutated is specified by parameter p_m. If the fitness of the new chromosome is higher than that of the original chromosome, the original chromosome is replaced by the new chromosome. Evolution continues until a specified number of generations has elapsed.

3.1.2 Genetic Algorithm

The GENEsYs-1.0 (Bäck, 1992) GA implementation was used in these experiments. Fitness-proportionate selection, elitism, and 2-point crossover were used. The GA is further parameterized by the population size, n, the per-chromosome probability of recombination, p_r, and the per-gene probability of mutation, p_m.

3.1.3 EuA

The predecessor of TEAM, EuA uses statistical predictors of chromosome fitness without maintaining a model or model statistics. A detailed description of this algorithm can be found in (Prior, 1998) and (Polani and Miikkulainen, 2000). This algorithm is parameterized by the population size, n, the probability of allele selection noise (similar to mutation), p_n, and the per-gene probability that an extinct allele will be reintroduced (only applicable when all chromosomes have the same allele for a particular gene), p_i.

3.1.4 TEAM

As described in Section 2, TEAM is parameterized by the population size, n, the minimum number of chromosomes to consider during allele selection, n_{min}, the per-gene probability of mutation, p_m, the selectivity factor, α, the model update frequency, T_{model}, and the chromosome replacement policy, ρ.

3.2 RESULTS

Our two combinatorial optimization test problems differ in how amenable they are to neighborhood search algorithms. Finding a maximally weighted cut in a graph, though an NP-complete problem, has a high degree of hillclimbability, making neighborhood searches effective. The Rosenbrock Function, on the other hand, has many local optima in binary space (Prior, 1998). This problem is very deceptive, making neighborhood searches less effective. The algorithms in this paper direct and focus neighborhood search differently, therefore these problems will illuminate the algorithms' strengths and weaknesses.

3.2.1 Maximally Weighted Cut in a Graph

In this problem we wish to partition the vertices of an undirected, weighted graph into two sets, V_0 and V_1, such that the sum of weights of all edges having one endpoint in V_0 and the other endpoint in V_1 is maximal. A feasible solution to this problem is a partition of the vertices such that every vertex is a member of V_0 or V_1, but not both. Given a graph of n vertices and m edges, we encode a partition as a binary chromosome of length n. If $g_i = 0$, then vertex i is a member of V_0, otherwise $g_i = 1$ and vertex i is a member of V_1.

The sum of weights of all edges crossing the partition is used as the chromosome evaluation function $f(c)$. It is calculated as

$$f(c) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{ij} \left(c_{g_i} \left(1 - c_{g_j}\right) + c_{g_j} \left(1 - c_{g_i}\right) \right),$$

where w_{ij} is the weight of the edge incident on vertices i and j, or valued 0 if no such edge exists. The GENEsYs GA package implicitly minimizes fitness value, therefore an altered chromosome evaluation function $-f(c)$ was used with that package. The graph was constructed randomly, with edge weights uniformly distributed on $[0, 1]$.

Each algorithm was executed 100 times, each evolved for 100,000 generations. The average best fitness value of each generation is shown in Figure 1. The parameters used for each algorithm are: Stochastic Hillclimber: p_m=0.04; GA: n=50, p_r=0.6, p_m=0.001; EuA: n=100, p_n=0.05, p_i=0.01; and TEAM: n=100, n_{min}=20, p_m=0.01, α=0.6, T_{model}=100, ρ=*replace_worst_contributor*. These parameter values were determined experimentally; small variations produce roughly equivalent results.

All algorithms tested eventually generated chromosomes of approximately the same fitness. In this domain of few local optima, only the curves prior to the plateau are noticeably different. The hillclimber makes good progress in early generations, but EuA and TEAM soon catch up and exceed it in performance. The differences between the final average best fitness values of the four algorithms, though relatively small, are statistically significant.

Both of TEAM's chromosome replacement policies, *replace_worst_fitness* and *replace_worst_contributor*, were tested on this problem. The *replace_worst_contributor* policy (shown) generated chromosomes with fitness values approximately 10% better than those under the *replace_worst_fitness* policy. This result suggests that low-fitness individuals can contain valuable information, and are not always the best choice for extinction.

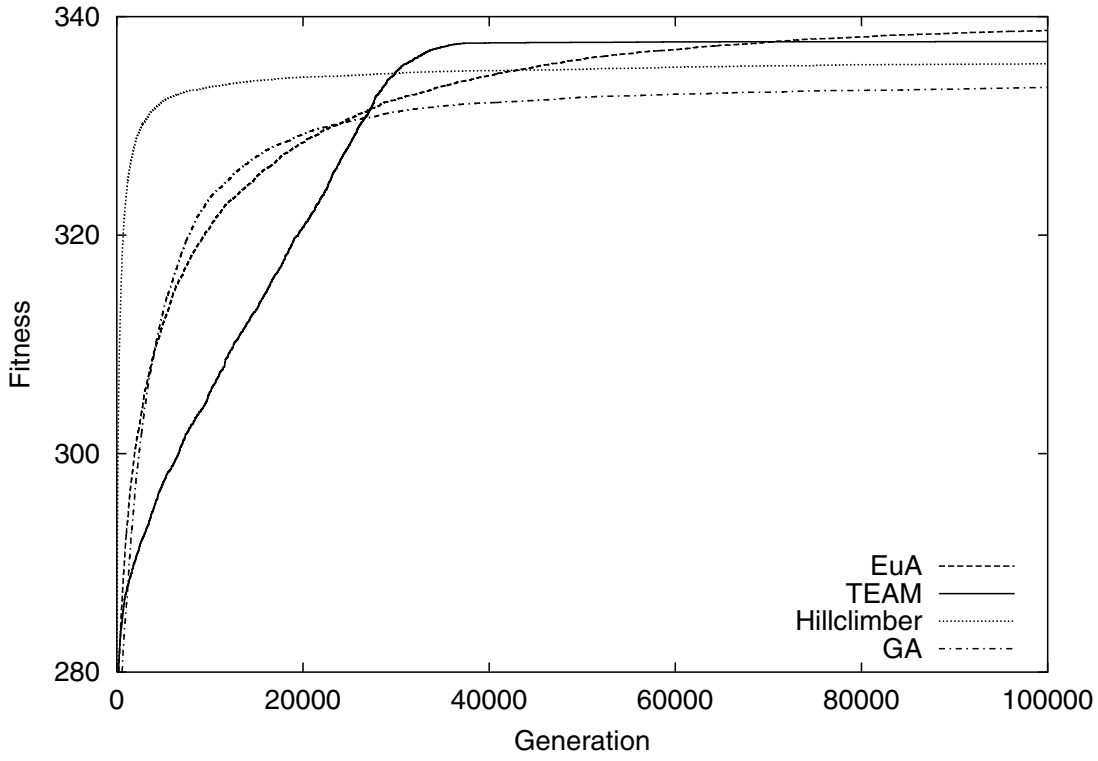

Figure 1: Average Best Fitness of 100 Runs Maximizing Cut Weight Over 100,000 Generations. In this problem with relatively few local maxima, the hillclimber makes good progress early on, but is eventually exceeded by EuA and TEAM. The final fitness differences are statistically significant ($p < 0.05$).

3.2.2 Two-Dimensional Rosenbrock Function

In this problem we wish to minimize the following function:

$$f(x, y) = 100(y - x^2)^2 + (x - 1)^2,$$

where $x, y \in [-5.12, 5.12]$. A feasible solution to this problem is a two-dimensional point, (x, y), which we encode as two floating-point numbers in a chromosome of 64 genes. We calculate the coordinates x and y as follows:

$$x = 5.12\left(1 - 2\sum_{i=1}^{32} 0.5^i c_{g_i}\right),$$

$$y = 5.12\left(1 - 2\sum_{i=33}^{64} 0.5^{i-32} c_{g_i}\right).$$

The coordinate x is encoded in genes 1 through 32, with gene 1 the most significant bit (MSB) in the floating-point encoding of x, and gene 32 the least significant bit (LSB). Similarly, coordinate y is encoded in genes 33 through 64, with gene 33 the MSB and gene 64 the LSB.

All algorithms used a chromosome evaluation function based directly on the Rosenbrock function value, $f(c) = f(x, y)$. 100 runs were performed for each algorithm, each evolved for 50,000 generations. The average best fitness value of each generation is shown in Figure 2. The parameters used for each algorithm are: Stochastic Hillclimber: p_m=0.3; GA: n=50, p_r=0.6, p_m=0.016; EuA: n=100, p_n=0.1, p_i=0.01; and TEAM: n=100, n_{min}=20, p_m=0.01, α=0.2, T_{model}=100, ρ=*replace_worst_fitness*. These parameter values were again determined experimentally, and small variations produce roughly equivalent results.

In this problem, TEAM clearly outperformed all other algorithms: it generated solutions with fitness values several orders of magnitude closer to optimal than its competitors. The differences between the final average best fitness of the four algorithms are statistically significant. The discretization of the problem introduces many local optima, traditionally difficult features for optimization algorithms. Yet during later generations, TEAM's average best fitness continues to improve, suggesting TEAM could yield even better solutions if evolution continued beyond 50,000 generations.

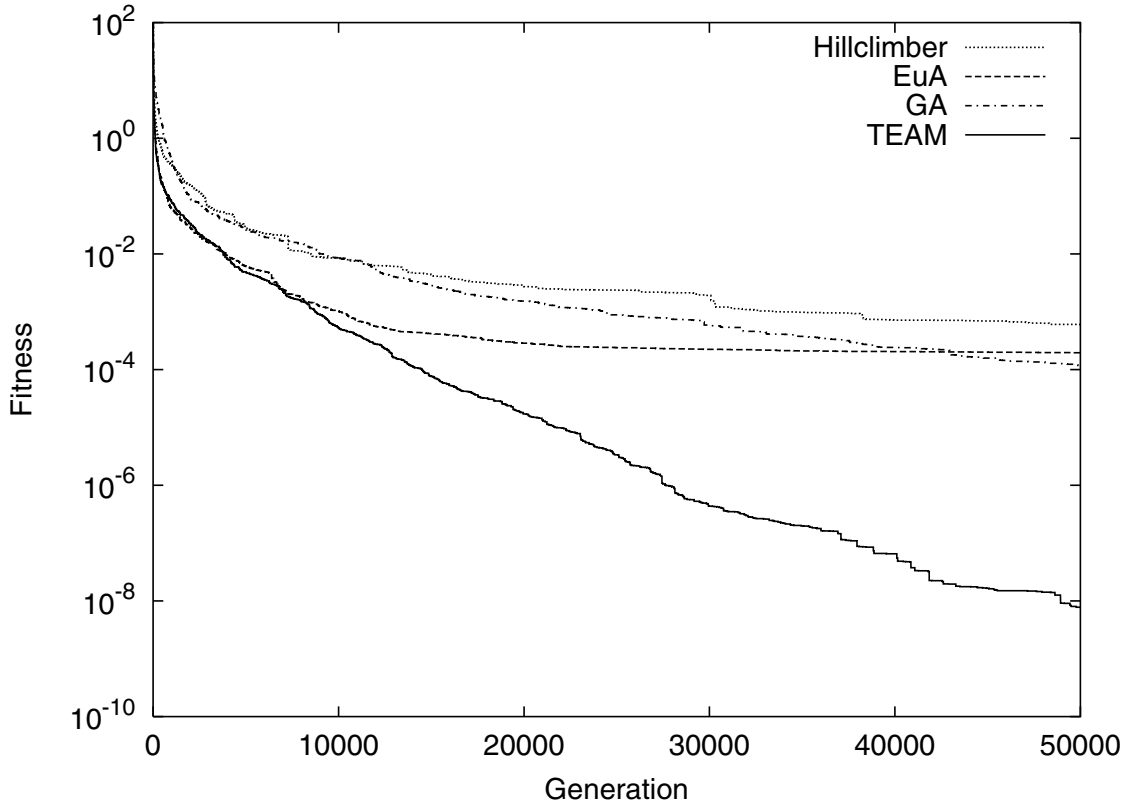

Figure 2: Average Best Fitness of 100 Runs Minimizing the Rosenbrock Function Over 50,000 Generations. In this problem with many local optima and significant deception, TEAM finds solutions several orders of magnitude better than the other algorithms. The final fitness differences are statistically significant ($p < 0.05$).

Both of TEAM's chromosome replacement policies were tried on this problem as well. Interestingly, the alternative *replace_worst_contributor* policy caused TEAM's population to often converge to suboptimal solutions. The more common *replace_worst_fitness* policy (shown) performed significantly better, allowing TEAM to consistently find good solutions.

How does TEAM achieve such a strong performance on this problem? To illustrate, recall that the order in which genes are bound during chromosome creation is determined by selectivity. Gene ordering therefore tells us which genes TEAM identified as the most useful, thereby illustrating its progress toward solution. Figure 3 shows such a histogram of the gene order. The gray scale indicates a gene's average rank during chromosome creation over several generations. Dark coloration indicates that the gene is among the first genes bound, and lighter coloration indicates that the gene is bound later.

The most prominent feature of Figure 3 is the concentrated dark bands, i.e. adjacent genes that were bound early in the chromosome creation process. In early generations, the two bands are localized near genes 1 and 2, at the bottom of the histogram, and genes 33 and 34, near the middle of the histogram. Recall that in the floating point encoding of (x, y) in a chromosome, these genes correspond to the most significant bits of the x and y coordinates. Since the two-dimensional Rosenbrock function has a relatively small region of near optimal solutions, this result makes sense: the most important indicators of fitness, and thus the most important genes to initially bind, are those responsible for the largest changes in the point (x, y). It is important to set those genes correctly, otherwise there is no possibility of high fitness.

Gene 33 is identified as selective almost immediately. All points in the domain with fitness relatively near the optimal lay above the line $y = 0$. A chromosome encoding a point above this line must have gene 33 bound to allele 0, therefore this gene is highly selective and should be bound early. As evolution progresses and the alleles of the selective genes become nearly homogeneous in the population, additional genes become se-

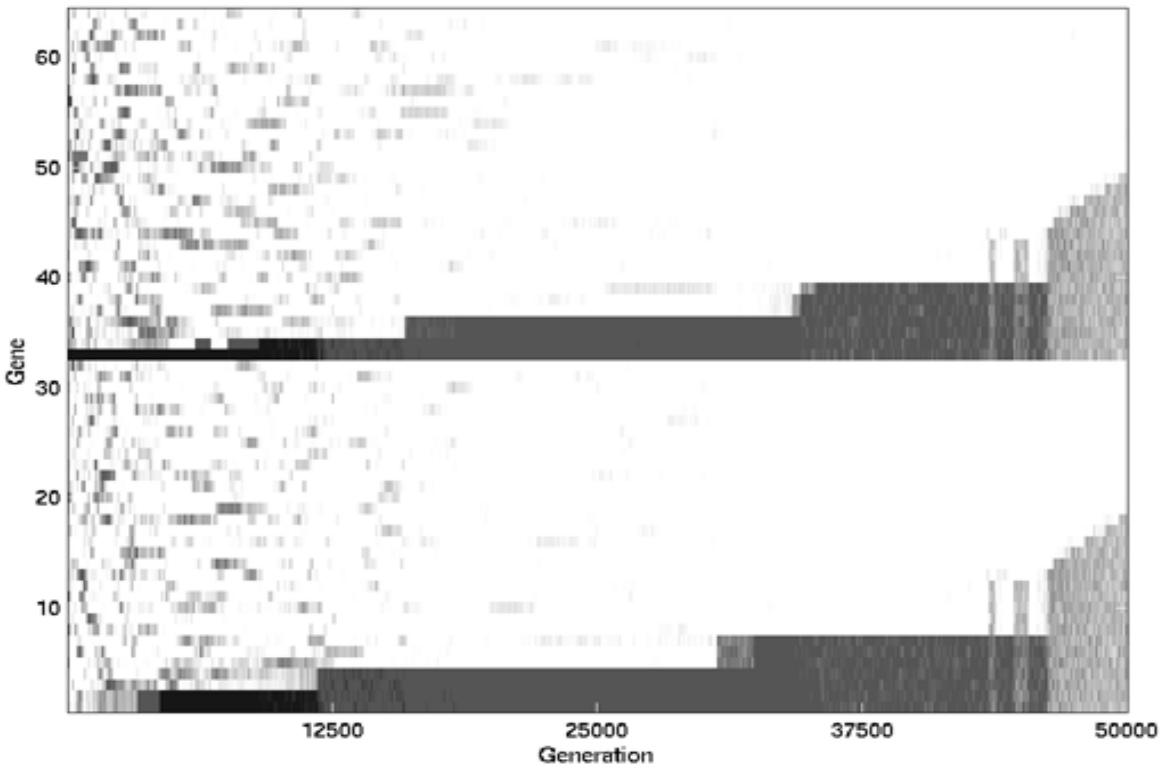

Figure 3: Histogram of Gene Order for Minimization of the Rosenbrock Function Over 50,000 Generations. Dark coloration indicates the gene was bound early in chromosome creation, light means late. TEAM identifies and solves the most selective genes first, thereby making consistent refinement possible.

lective. This process is visible in Figure 3 as a widening of the two dark bands. TEAM's focus on these new selective genes incrementally refines the point (x, y).

This way, Figure 3 illustrates how TEAM solves the problem by identifying the most important genes first. Similar processes take place in other domains where the dependencies may be less obvious, giving us important insight into the domain. To an observer, gene order can be an important clue to identifying underlying inter-gene dependencies in the problem.

4 DISCUSSION & FUTURE WORK

The overhead of maintaining the model and constructing the chromosome in TEAM is considerable compared to that of a hillclimber or a GA. However, such more intelligent evolution steps are warranted if they can generate better final solutions than other methods. In the challenging test of minimizing the Rosenbrock function, TEAM indeed produced solutions significantly better than the other algorithms.

TEAM relies on the t-test, which assumes samples are normally distributed. While TEAM is effective in the two selected problems, normality may be a risky assumption in other domains. It might therefore make sense to replace the t-test with a method that works equally well with arbitrary distributions, such as the Mann-Whitney U test (Mendenhall and Beaver, 1994). Instead of scoring chromosomes and genes based on fitness values, scores could be based on rank within chromosome/gene sets. This way it should be possible to apply TEAM reliably to a wide range of domains.

Rank-based calculations have another advantage: they limit the influence high-fitness chromosomes have during chromosome creation. Currently, if there is a small number of individuals with very high fitness, the new chromosome is constructed mostly based on their genes. However, the goal of evolution in TEAM is to produce a population that converges around a few high-fitness individuals. With rank-based calculations a larger number of chromosomes take part in construction, thereby maximizing the amount of information considered during chromosome creation.

The model maintained by TEAM has perhaps the greatest potential for improvement. The current

model is organized around pair-wise relationships between genes. This is an improvement over algorithms that deal with genes in isolation, but combinatorial problems are not limited to only pair-wise dependencies. A method for identifying and exploiting n-gene relationships is needed. Instead of recording gene pairs with an observed relationship to fitness values, we can record gene groups of arbitrary size. We can already identify groups of size two, and larger groups can be formed through expansion and combination of existing groups as additional correlations are found. The identification of gene groups could be run in parallel with population evolution. It is an interesting problem in its own right, and will be studied in detail in future work.

5 CONCLUSIONS

An advanced constructive evolutionary algorithm, TEAM, was introduced in this paper. Using statistical analysis and modeling of gene/fitness correlations, TEAM can evolve chromosomes in domains with complex gene dependencies. Experiments show that TEAM performs better than other problem-independent combinatorial optimization techniques on difficult problems. This result shows that domain information can be extracted and meaningfully applied during population evolution. Future work will focus on improved methods of information extraction and application from limited domain sampling.

Acknowledgments

Thanks to Daniel Polani and John Prior for useful discussions and code for the EuA and EuSane algorithms. This research was supported in part by the NSF under grant IIS-0083776 and by the Texas Higher Education Coordinating Board under grant ARP-003658-476-2001.

References

Bäck, T. (1992). *A User's Guide to GENEsYs 1.0.* Technical report. Department of Computer Science, University of Dortmund.

Baluja, S. (1994). *Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning.* Technical Report CMU-CS-94-163. Carnegie Mellon University, Pittsburgh, PA.

Davidor, Y. (1991). Epistatis Variance: A Viewpoint on GA-Hardness. *Foundations of Genetic Algorithms 1*, 23-35. Morgan Kaufmann, San Mateo, CA.

De Jong, K. A. (1975). An Analysis of the Behavior of a Class of Genetic Adaptive Systems. Doctoral dissertation, University of Michigan, *Dissertation Abstract International*, 36(10), 5140B. (University Microfilms No. 76-9381).

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence.* University of Michigan Press, Ann Arbor, MI.

Kirkpatrick, S., and Sherrington, D. (1988). Infinite range models of spin-glasses. *Physical Review B*, 17:4384-4403.

Mendenhall, W., and Beaver, R. (1994). *Introduction to Probability and Statistics - 9th edition.* Duxbury Press, International Thomson Publishing, Pacific Grove, CA.

Polani, D., and Miikkulainen, R. (2000). Eugenic Neuro-evolution for reinforcement learning. In *Proceedings of the Genetic and Evolutionary Computation Conference* (GECCO-2000, Las Vegas, NV), 1041-1046. Morgan Kaufmann, San Francisco, CA.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical Recipes in C.* Cambridge University Press. Second Edition.

Prior, J. W. (1998). *Eugenic Evolution for Combinatorial Optimization.* Master's thesis, Technical Report AI98-268. Department of Computer Sciences, The University of Texas at Austin.

Syswerda, G. (1993). Simulated Crossover in Genetic Algorithms. *Foundations of Genetic Algorithms 2*, 239-255. Morgan Kaufmann, San Mateo, CA.