Efficient Evaluation Relaxation Under Integrated Fitness Functions

Laura A. Albert & David E. Goldberg
Department of General Engineering
University of Illinois at Urbana-Champaign
104 S. Mathews St.
Urbana, IL 61801
{lalbert, deg}@uiuc.edu

In many industrial applications of genetic algorithms (GAs), there is a tradeoff between more and less accurate functions. On one hand, more accurate yet more expensive evaluations can be used, and on the other, noisy, inexpensive evaluations are made. If computation time is expensive, a fast noisy fitness function may be preferred over a slow, accurate fitness function [1]. Some progress has been made in understanding the tradeoffs in the noisy evaluations where the noise is due to randomness or *variance* alone [2],[3]. Less concern has been shown when the error cannot be averaged out by sampling – where a good portion of the error may be due to *bias*.

In this poster, we consider the general situation when evaluation error is the result of a biased function surrogate. In particular, we consider fitness functions whose cost and accuracy vary because of discretization errors of integration. We assume that the building blocks are exponentially scaled and that the GA experiences domino convergence where the convergence time complexity is linear in the number of building blocks [4].

The following time model can be used to describe a GA with a sampling fitness function and can be extended to describe a GA with an integration fitness function:

$$T = (\alpha + \beta n)GN \tag{1}$$

Here, α is the overhead per individual per generation, β is the time to calculate one sample, n is the number of samples or grid points, G is the total number of generations and N is the population size.

The choice of the number of grid points for the integration, n, is usually constant for the entire run of the GA. This choice of n is too precise at the beginning of the run, which wastes computation cycles early on. In order to improve the computational efficiency, the number of grid points can be exponentially increased throughout the duration of the run — it will take 2^n grid points to discriminate between two individuals that are converging to the nth building block. In the first few generations, the first, or most salient, bit is being considered by the selection operator because its contribution to the fitness is the greatest, and the other bits are ignored.

Using a crude discretization in the early generations introduces some noise into the GA which increases the expected time of convergence [3]. Nevertheless, the savings in computing the function evaluations outweighs the cost of making more total evaluations. The expected speedup can thus be written as the quotient of the time model in (1) and the time when using the exponentially increasing grid points as described above:

$$S = \frac{(\alpha + \beta n)G}{\sum_{k=1}^{k} (\alpha + \beta \cdot 2^i) g} \tag{2}$$

where g is the time for each bit to converge when considering the noisy model and k is the number of bits. Our experiments consider a one-dimensional test function, and the observed speedup was 2.2.

References

[1] Fitzpatrick, J. M. & Grefenstette, J. J. (1988). Genetic algorithms in noisy environments. *Machine Learning*, 3, pp. 101-120.

[2] Miller, B. L. & Goldberg, D. E. (1996). Optimal sampling for genetic algorithms. *Proceedings of the Artificial Neural Networks in Engineering*, Vol. 6 (pp. 291-298). New York: ASME Press.

[3] Miller, B. L. (1997). Noise, sampling and efficient genetic algorithms. (Doctoral Dissertation, University of Illinois). IlliGAL Report No. 97001.

[4] Thierens, D, Goldberg, D. E. & Periera, A. G. (1998). Domino convergence, drift and the temporal-salience structure of problems. *The 1998 International Conference on Evolutionary Computation Proceedings* (pp. 535-540). Piscataway, NJ: IEEE Service Center.

MAX-CSP and Fitness Landscapes

Meriema Belaidouni
LERIA, Université d'Angers
2 bd Lavoisier
F-49045 Angers Cedex 01

Jin-Kao Hao
LERIA, Université d'Angers
2 bd Lavoisier
F-49045 Angers Cedex 01

1 ABSTRACT

Our analysis of MAX-CSP landscape leads to an autoregressive model AR(1). The approximation of this model on neighborhood relations shows that the autocorrelation is not always sufficient to predict their difference for local search. In this case the interpretation of mean and standard deviation values can be helpful. The statistical analysis forecasts confirm previous results on Simulated Annealing.

2 METHOD

Given a time series $\{f_t\}_{t>0}$, the Box and Jenkins methodology [1] allows to identify the nature of the phenomenon represented by the sequence of observations. It consists in three stages: I) IDENTIFICATION: An appropriate model within the class of ARMA models is specified. In the case of an AR(1) model, the observations are described by the equation: $f_t - E(f_t) = a_1(f_{t-1} - E(f_t)) + \epsilon_t$, where $E(f_t)$ is the mean of the time series $\{f_t\}_{t>0}$, a_1 is the AR coefficient (and also the autocorrelation), ϵ_t is a white noise with zero mean and finite variance. All ϵ_t are independent from each others. II) ESTIMATION: The model parameters are estimated. In the case of an AR(1) model only a_1 and $E(f_t)$ need to be estimated. The measure R^2 allows to appreciate the quality of fit. It takes its value in the interval [0,1]. The higher R^2, the better the fit. III) MODEL EVALUATION: There should be no serial dependency between residuals ϵ_t.

3 EXPERIMENTS AND RESULTS

This section aims to apply time series analysis to random MAX-CSP landscapes. The families of instances (F.I) are generated by the standard model $< n, d, p_1, p_2 >$, where n is the number of variables, d the number of values per variable, p_1 the density and p_2 the tightness [3]. The families are $A =< 100, 10, 8\%, 25\% >$, $B =< 100, 10, 15\%, 25\% >$, $C =< 100, 10, 50\%, 25\% >$. For each family, two

Table 1: AR(1) parameters for MAX-CSP landscapes obtained with neighborhoods N_1 and N_2.

	Neighborhood Relation N_1			
F.I	$\hat{E}(f_t)$	$\hat{a}_1$	$\hat{\sigma}(\epsilon_t)$	R^2
A	$99.02 \pm .31$	$.979 \pm 10^{-3}$	1.72 ± 10^{-2}	$.960 \pm 10^{-3}$
B	$185.52 \pm .20$	$.979 \pm 10^{-3}$	2.362 ± 10^{-2}	$.959 \pm 10^{-3}$
C	$618.58 \pm .79$	$.979 \pm 10^{-3}$	4.32 ± 10^{-2}	$.960 \pm 10^{-3}$
	Neighborhood Relation N_2			
F.I	$\hat{E}(f_t)$	$\hat{a}_1$	$\hat{\sigma}(\epsilon_t)$	R^2
A	$78.50 \pm .37$	$.979 \pm 10^{-3}$	1.65 ± 10^{-2}	$.968 \pm 10^{-3}$
B	$180.66 \pm .23$	$.979 \pm 10^{-3}$	2.33 ± 10^{-2}	$.960 \pm 10^{-3}$
C	$618.72 \pm .50$	$.979 \pm 10^{-3}$	4.32 ± 10^{-2}	$.960 \pm 10^{-3}$

neighborhood relations N_1 and N_2 are used. N_1: two configurations are neighbors if they differ by one variable value. N_2: two configurations are neighbors if they differ at the value of a single *conflicting* variable. A variable is said conflicting if it is involved in some unsatisfied constraints. Table 1 shows the results of the analysis of costs generated by 100 random walks of length 100.000. The main results are: 1) As confirmed by $R^2 \approx 96\%$, the studied MAX-CSP landscapes are AR(1), 2) The AR coefficient a_1 is close to one indicating that these landscapes are smooth and rather easy to search, 3) Neighborhoods N_1 and N_2 have the same coefficient a_1. N_2 costs are smaller than N_1 costs as shown by the mean $E(f_t)$ and the standard deviation $\sigma(\epsilon_t)$. These remarks, lead us to expect that N_2 is better than N_1 for local search. The forecast is confirmed by previous results on Simulated Annealing [2].

4 CONCLUSIONS

The statistical study shows that random MAX-CSP landscapes follow in good approximation an AR(1) model. If the autocorrelation is not significantly different, the mean and standard deviation may help to compare neighborhoods. In future work, it would be interesting to study structured MAX-CSP landscapes.

References

[1] G. E. P. Box and G. M. Jenkins. *Holden Day.*, 1970.
[2] J. K. Hao and J. Pannier. *AIM*, 1998.
[3] B. M. Smith and M.E. Dyers. *J. Art. Int.*, 81:155–186, 1996.

A Study of the Scalability of a Data Mining Genetic Algorithm on a Distributed Database

Agnès Braud and **Christel Vrain**
LIFO, Université d'Orléans,
rue Léonard de Vinci, BP 6759,
45067 Orléans Cedex 2, FRANCE
{Agnes.Braud,Christel.Vrain}@lifo.univ-orleans.fr

We are interested in speeding up Genetic Algorithms when applied to Concept Learning from examples stored in a database. In this context, the cost for computing the fitness function can be very high and thus considerably slow down the algorithm. We have already studied a parallel algorithm in which the population is distributed among processors, but still behaves like a single panmictic one [Braud & Vrain, 1999]. In that case, the database is duplicated on processors, which is not realistic when it is large. In this poster paper, we study another parallel approach which allows the treatment of databases that cannot be stored on a single processor, and to decrease the cost of the evaluation of individuals. A subset of the learning set is associated to each processor at the beginning of the learning process. The population evolves on a single processor and the algorithm is similar to the sequential one, except for the evaluation step which is done is parallel. Each time the population must be evaluated, the processors evaluate all the individuals on their subsets of examples; then all the results are gathered on a processor to obtain the individuals qualities. An individual represents a disjunction of conjunctive hypotheses, expressed in an attribute-value representation, following the coding proposed in GABIL [DeJong *et al.* , 1993].

The evaluation step is decomposed as described below (we call Processor 1, the processor where all the treatments, except the evaluation step, are computed):

1 - Processor 1 sends all the individuals of the population to the other processors;

2 - each processor computes, for each individual, the number of examples of the local database this individual covers and the number of counter-examples it rejects;

3 - each processor sends its results to Processor 1;

4 - Processor 1 merges the results received and thus obtains the qualities of the individuals on the whole database.

This algorithm has been studied both from a theoretical and an experimental point of view. The theoretical study has been modeled using the BSP [Valiant, 1990] paradigm, that allows developers to predict performance of their program on a given architecture. This has enabled us to predict that under some conditions, the speedup obtained will be interesting and our algorithm will be scalable. Experiments performed on a biprocessor Sun Ultra Sparc 2 with increasing sizes for the database show that the parallel algorithm is quite twice quicker than the sequential one for the largest databases. Tests have also been performed on a CRAY T3E-1200, using up to 32 processors. They show that when the size of the population or the size of the database is large enough, we obtain an increase in speedup that is about $2 \times log(p)$.

This parallelization study is devoted to Data Mining tasks, as the underlying algorithm exploits the distribution of the database to speed up the computation of the fitness function. It is simple, but the cost study and then the experimentations have shown that it really decreases computation costs and its simplicity confers it a large flexibility to be extended and enhanced.

References

[Braud & Vrain, 1999] Braud, A., & Vrain, C. 1999. A Parallel Genetic Algorithm based on the BSP Model. *Pages 160–162 of: Proceedings of the GECCO Workshop Program.*

[DeJong *et al.* , 1993] DeJong, K.A., Spears, W.M., & Gordon, D.F. 1993. Using Genetic Algorithms for Concept Learning. *Machine Learning*, **13**, 161–188.

[Valiant, 1990] Valiant, L. G. 1990. A Bridging Model for Parallel Computation. *Communications of the ACM*, **33**(8), 103–111.

An Exploration in Combatting Premature Convergence

Clare Bates Congdon
Colby College Computer Science Department
5846 Mayflower Hill Drive
Waterville, ME 04901
ccongdon@colby.edu

This abstract describes work with an application of genetic algorithms to a specific task from biology (phylogenetic tree construction) [1]. Immigration and an alternative approach for addressing the premature convergence problem are compared: The new method is comparable to immigration, but its patterns of discovery are different, suggesting that this approach warrants further study.

The alternative approach (A) is similar to immigration (I) in that the population is divided into subpopulations that in general remain distinct from each other (crossovers happen only within a given subpopulation). At specified generations, the following happens: 1. The populations are combined into one large population. 2. The large population is sorted by fitness, and canonical duplicates are removed. 3. For each population, n trees from each of m of the best fitnesses are selected at random to seed a new population. 4. The remainder of each population is filled in with randomly generated trees.

As an initial comparison, several runs were done with the task described in [1]. All runs used population sizes of 5000, split into 10 subpopulations of 500 each, 2000 generations, and immigration or destruction events happening every 500 generations (these events do not happen following the last generation, so this constitutes 3 events per run). Other parameters include an elitism rate of 25%, first mutation rate of 10%, second mutation rate of 100%, and crossover rate of 70%. For A, n and m were varied to save either the 10 or the 20 best fitnesses and to save either one or two random trees of each of these fitnesses. For I, rates of 5%, 10%, and 20% were run. An eighth set of four experiments was run (D runs) with the parameters described above (including 10 distinct populations with subpopulations kept distinct). For each distinct variation, four experiments (with different random seeds) were completed for a total of 16 A runs 12 I runs, and 4 D runs.

In terms of being able to locate trees of the best fitness, A slightly outperforms I, finding trees of the best known fitness in 7/16 runs (43.75%), while I discovers the 279 trees in 5/12 runs (41.7%). The D runs are unable to find trees of fitness 279. If one plots the average generation taken to find each fitness from 284 to the known best of 279, the trajectories of the A runs and the I runs are similar, and the two are comparable in terms of the average generation to reach 279. The progress of the A runs is marked by a "punctuated equilibrium" dynamic: The effects of mixing the populations is more likely to be immediate than with I. When improved fitnesses are discovered in the 284-279 fitness range, they happen within 100 generations of an A migration 50% of the time. In contrast, these improvements happen within 100 generations of an I immigration 38.8% of the time.

The effect of the immigration process is to introduce new genetic material into a converging population, possibly providing variation within the population that might prove useful for crossovers. On the other hand, the effect of the process in A is a mass extinction of solutions, in which only a handful of promising solutions are saved. After a destruction event, each population will in general contain a sampling of the best solutions from across all populations. This differs from immigration, in which each subpopulation tends to retain the bulk of its genetic material. In addition to the melding of information from subpopulations, the A process is also marked by extremely high infusion of random individuals.

While the punctuated equilibrium dynamic of method A is intriguing, clearly more runs and more variations on parameter settings need to be explored.

References

[1] C. B. Congdon and E. F. Greenfest. Gaphyl: A genetic algorithm approach to cladistics. In *GECCO Workshop on Data Mining with Evolutionary Algorithms*, pages 85–88, 2000.

A Constructive Genetic Algorithm for the Linear Gate Assignment Problem

Alexandre César Muniz de Oliveira
Depto. Informática
Universidade Federal do Maranhão
65085-580, São Luís, MA,Brazil
acmo@deif.ufma.br

Luiz Antonio Nogueira Lorena
Lab. Ass. de Computação e Matemática Aplicada
Instituto Nacional de Pesquisas Espaciais
12201-970 São José dos Campos - SP – Brazil
lorena@lac.inpe.br

Abstract

We present in this paper an application of the *Constructive Genetic Algorithm (CGA)* to the *Linear Gate Assignment Problem (LGAP).* The *LGAP* happen in very large scaling integration (VLSI) design, and can be described as a problem of assigning a set of circuit nodes (gates) in an optimal sequence, such that the layout area is minimized, as a consequence of optimizing the number of tracks necessary to cover the gates interconnection. The *CGA* evolves a dynamic population composed of schemata and structures and uses heuristics in fitness function definitions.

1 CGA APPLICATION TO LGAP

The *Constructive Genetic Algorithm (CGA)* was proposed recently as an alternative to a traditional *GA* approach (Lorena, 2001), particularly, for evaluating schemata directly. The population, initially formed only by schemata, evolves controlled by recombination to a population of well adapted structures (schemata instantiation) and schemata.

Linear gate assignment problems (LGAP) are related to gate matrix layout and programmable logic arrays folding. An example of a gate matrix and the representation used for structures and schemata follows:

1 0 1 1 0 0 1 0 0	1 0 0 0 1 1 0 1 0	1 ? 0 ? ? ? 0 1 ?
0 0 0 1 0 1 0 0 1	0 0 0 1 0 1 1 0 0	0 ? 0 ? ? ? 1 0 ?
1 1 0 0 1 0 0 1 0	0 1 1 0 0 0 0 1 1	0 ? 1 ? ? ? 0 1 ?
1 0 0 1 1 0 1 0 0	1 0 1 0 0 1 0 1 0	1 ? 1 ? ? ? 0 1 ?
0 0 0 1 0 1 0 1 1	0 0 0 1 0 1 1 0 1	0 ? 0 ? ? ? 1 0 ?
0 0 0 0 1 0 1 1 0	1 0 1 0 0 0 0 0 1	1 ? 1 ? ? ? 0 0 ?
0 0 0 0 0 1 0 0 1	0 0 0 1 0 0 1 0 0	0 ? 0 ? ? ? 1 0 ?
1 2 3 4 5 6 7 8 9	s_j= (7 2 5 9 3 4 6 1 8)	s_k=(7 # 5 # # # 6 1 #)
Gate matrix	Permutation (structure)	Permutation (schema)

Two fitness functions are defined on the space of all schemata and structures that can be obtained using this representation. The evolution process considers the two objectives on an adaptive rejection threshold, which gives ranks to individuals and yields a dynamic population. The first function reflects the total cost of a given permutation of gates, and the other drives the evolutionary process to a population trained by a heuristic. The chosen heuristic is the 2-Opt neighborhood.

The initial population is composed exclusively of schemata. Two structures and/or schemata are selected for recombination. The first is called the *base* (s_{base}) and is randomly selected out of the best ranked individuals. The second structure or schema is called the *guide* (s_{guide}) and is randomly selected out of the total population. The current labels in corresponding positions are merged. A new filling operator is proposed to complement a schema, substituting the # labels for gate numbers. A local search mutation is always applied to structures, no matter how they are created (after recombination or after the filling process). The search at 2-Opt neighborhood of the structure was used.

The *CGA* for *LGAP* was run on Intel Pentium II (266Mhz). All best previous results comes of *Microcanonical Optimization* - MCO approach (Linhares,1999). The *CGA* reached all the best results (number of tracks) for instances taken from the literature, but it appears to be more robust than other approaches.

	MCO		**CGA**		
Problem	**Time (s)**	**Tracks**	**Time (s)**	**Generations**	**wire length**
wli	10	**4**	5	5.00	35
wsn	10	**8**	15	7.00	115
v4050	10	**5**	5	5.00	51
v4000	10	**5**	5	5.00	66
v4470	700	**9**	665	33.00	269
v4090	100	**10**	20	13.50	132
x0	700	**11**	755	92.57	343
w1	10	**4**	10	5.00	57
w2	400	**14**	185	19.50	283
w3	3900	**18**	3062	186.00	761
W4	61700	**27**	52246	225.00	1932

References

L.A.N.Lorena and J.C.Furtado (2001). *Constructive Genetic Algorithm for Clustering Problems*. Evolutionary Computation. To appear. Available from http://www.lac.inpe.br/~lorena/cga/cga_clus.PDF

A.Linhares, H.H.Yanasse, and J.R.A.Torreao (1999). *Linear Gate Assignment: a fast statistical mechanics approach*. IEEE Trans. on Comp. Aided Design of Integrated Circuits and Systems. Vol. 18(12),1750-1758.

A Genetic Algorithm for Expert System Rule Generation

John C. Determan
Idaho National Engineering and Environmental Laboratory
P.O. Box 1625, Idaho Falls, ID 83415-2211
jcd@inel.gov

James A. Foster
Computer Science Department, University of Idaho P.O. Box 1010, Moscow, ID 83844-1010
foster@cs.uidaho.edu

Extended Abstract

We applied genetic algorithms to fuzzy rule generation to compute expert system rules from data. Our work introduces several innovations that improve both the speed of the rule generation process and the accuracy of the generated rules (Determan 2000). Our work also contributes to the application of GAs to combinatorial problems in general.

Our genetic data clustering (GDC) algorithm first locates data points in the training data that belong together, it then calculates exact means and standard deviations over these clusters, and finally, it builds fuzzy rules from the cluster means and standard deviations. We employ subtractive clustering (Chiu, 1994) to initialize our population of solutions, but do away with backpropagation. We designed crossover and mutation operators specifically for data clustering. We used self-adaptive techniques, including variable length chromosomes and a variable probability of mutation. Our objective function combines training set performance and the Xie-Beni cluster validity index (Xie and Beni, 1991). Use of a cluster validity index helps to aviod over-training on the data.

To test our algorithm, we used two data sets from a comparative study of neural network derived classification algorithms, the Enhanced Learning for Evolutive Neural Architecture (ELENA) project (Aviles-Cruz, et al. 1995). These data sets were the Anderson Iris data and a LANDSAT image data set. The testing methodology used in the ELENA project (Blayo et al., 1995) was followed.

Data sets used to test classification algorithms generally consist of data points labeled by some integral set of values. It is a useful bounding calculation to treat these labeled data as already clustered (by classification) and form fuzzy rules by taking the average and standard deviations of the data points in these clusters. These rules represent the results that the GDC algorithm could obtain with "perfect" clustering on a given data set.

The GDC algorithm achieved average performance compared to a suite of classification algorithms found in the literature. Tests have shown that performance of the GDC algorithm is mixed. Specifically:

- Simple bounding calculations show that the GDC algorithm computes rule sets that are as good as the rule representation format will allow.
- Comparised with ELENA results, our method produces classification rules that achieve about average results. The rule representation is the primary limitation to performance.

REFERENCES

Aviles-Cruz, C., Voz, J.L., Van Cappel, D. (1995). Databases, Enhanced Learning for Evolutive Neural Architecture, ESPRIT basic Research Project Number 6891, R3-B1-P.

Blayo, F., Cheneval, Y., Guerin-Degue, A., Chentouf, R., Aviles-Cruz, C., Madrenas, J., Moreno, M., Voz, J.L. (1995). Benchmarks, Enhanced Learning for Evolutive Neural Architecture, ESPRIT basic Research Project Number 6891, R3-B4-P.

Chiu, S. L. (1994). Fuzzy Model Identification based on Cluster Estimation. Journal of Intelligent and Fuzzy Systems, 2, 267–278.

Determan J. (2000). Automatic Expert System Rule Generation On Nondestructive Waste Assay Data, Master's Thesis, University of Idaho.

Xie, X. and Beni, G. (1991). A Validity Measure for Fuzzy Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence, 13: 8, 841-47.

Evolutionary Hill-Climbing, Virtual Constraints, and Recurrent Dynamic CSPs*

Gerry Dozier
Department of Computer Science and Software Engineering
Auburn University, AL 36849-5347
email: gvdozier@eng.auburn.edu

1 Introduction

The recurrent dynamic constraint satisfaction problem (rDCSP) [3] is a special class of CSP where constraints are temporarily added and/or deleted over time. The objective of rDCSP-solvers is to discover one or more solutions that remain consistent (no constraint violations) and are not 'damaged' when constraints are added. Recent research on rDCSPs has shown that by identifying, as virtual, those dynamic constraints that have an above average probability of being added to the set of active constraints can improve the performance of rDCSP-solvers [3]. Although this frequency-based approach works well for rDCSPs where the dynamic constraints have different addition probabilities, it does not work well for rDCSPs where the addition probabilities are the same for all dynamic constraints.

2 A New Approach

We have modified an evolutionary hill-climber (EHC) [1] so that it identifies virtual constraints based on constraint violations rather than frequency count. In this new method, the violation-based approach, the EHC keeps track of the number of times a dynamic constraint is violated while it is active or virtual. This is referred to as the violation count of a dynamic constraint. Those dynamic constraints with an above average, non-zero violation count are identified as virtual constraints.

3 Some Preliminary Results

In this work we solve the recurrent dynamic form of the classical N-Queens CSP [2]. Recurrent dynamic N-Queens CSPs, can be viewed as a triple (N,σ,α), where N represents the number of queens, σ denotes the percentage of static constraints, and where the active rate (addition probability) of the dynamic constraints is represented by is α. Each time a solution is found a new occasion is created by randomly activating dynamic constraints based on their active rate. Static constraints are always active.

In solving a rDNQP, a solution must be discovered for an initial occasion and k additional occasions. The the fraction of the k additional occasions for which an algorithm must perform additional search (when its solutions have been damaged) is known as its population instability rate (ι). The effectiveness of our new approach can be seen in Figure 1.

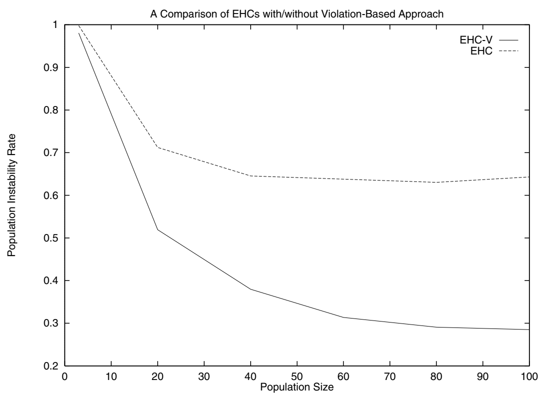

Figure 1: The ι's of EHCs that do not identify virtual constraints are compared with the ι's of those EHCs that utilize the violation based approach (EHC-V) on the 100 instances of the (30,0.1,0.5) rDNQP using population sizes of 3, 20, 40, 60, 80, 100. The violation-based approach has the best (statistically significant) performance for all population sizes.

* This research was supported by the National Science Foundation under grant #IIS-9907377.

References

[1] G. Dozier, J. Bowen, and A. Homaifar, "Solving Constraint Satisfaction Problems Using Hybrid Evolutionary Search," *The IEEE Transactions on Evolutionary Computation*, pp. 23-33, vol. 2, no. 1, April 1998.

[2] B. A. Nadel, "Constraint Satisfaction Algorithms", *Computational Intelligence*, vol. 5(4), pp. 188-224, 1989.

[3] R. J. Wallace and E. C. Freuder, "Stable Solutions for Dynamic Constraint Satisfaction Problems," *Proceedings of the 1998 Conference on the Principles and Practice of Constraint Programming - CP'98*, pp. 447-461, 1998.

A Self Adaptive Hybrid Genetic Algorithm

Felipe P. Espinoza
Department of Civil &
Environmental Engineering
University of Illinois
4129 Newmark Lab, MC-250
205 N. Mathews Ave.
Urbana, IL 61801
fespinoz@uiuc.edu
(217)-333-6979

Barbara S. Minsker
Department of Civil &
Environmental Engineering
University of Illinois
3230 Newmark Lab, MC-250
205 N. Mathews Ave.
Urbana, IL 61801
minsker@uiuc.edu
(217)-333-9017

David E. Goldberg
Department of General Engineering
University of Illinois
117 Transportation, MC 238
104 S. Mathews Ave.
Urbana, IL 61801
deg@uiuc.edu
(217)-333-0897

Abstract

This paper presents a self-adaptive hybrid genetic algorithm (SAHGA) and compares its performance to a non-adaptive hybrid genetic algorithm (NAHGA) and the simple genetic algorithm (SGA) on two multi-modal test functions with complex geometry. The SAHGA is shown to be far more robust than the NAHGA, providing fast and reliable convergence across a broad range of parameter settings. For the most complex test function, the SAHGA required over 75% fewer function evaluations than the SGA to identify the optimal solution.

1 INTRODUCTION

A hybrid genetic algorithm (HGA) is the coupling of two processes: the simple GA and a local search algorithm. The purpose of this study is to develop a self-adaptive HGA (SAHGA) that can be used to competently solve different applications without extensive trial-and-error experimentation.

2 HYBRID GENETIC ALGORITHM

We present 2 different HGA approaches. The first one called Non-Adaptive Hybrid Genetic Algorithm (NAHGA) and the second one is called Self-Adaptive Hybrid Genetic Algorithm (SAHGA). Both algorithms combine an SGA with local search. The local search step is defined by three basic parameters: local search gap, probability of local search, and number of local search iterations. The major difference in the approaches is that the SAHGA adapts in response to algorithm performance as the algorithm converges to the solution. This algorithm incorporates Baldwinian and Lamarckian evolution to update the local search information into the population.

3 EXPERIMENTS

To evaluate the performance of the SAHGA method, we worked with two different multi-modal functions with multiple basins of attraction randomly distributed. Function 1 (F1) has conical basins of attraction and Function 2 (F2) has elliptical basins of attraction. F1 represents the best case for local search, in which only one local search is required to find the local minimum, and F2 represents a more realistic case in which multiple local searches are required to find the local minimum. In order to evaluate the behavior of the SAHGA with respect to the NAHGA and SGA, we performed several experiments to test the capabilities of the method. The experiment tested the behavior of local search, maximum number of local search iterations, and probability of local search. Finally, we investigated the reliability of the SAHGA method in comparison with the SGA.

4 RESULTS AND CONCLUSIONS

The results clearly indicate that the adaptive capabilities of the SAHGA algorithm enabled robust solution of complex, multi-modal problems for a much greater range of parameter settings than the NAHGA. Compared with the SGA, the SAHGA was able to solve complex problems much faster because of the combined effect of smaller population sizes and increased information from local search. For the same level of reliability, the SAHGA required as much as 95% fewer function evaluations than the SGA for function f1 and as much as 75% fewer function evaluations for function f2. Further research is needed to assess the performance of the algorithm on other types of functions.

Graph Crossover

Al Globus
Computer Sciences Corp.
NASA Ames Research Center
Moffett Field, CA 94035

John Lawton
University of California
Santa Cruz, CA

Todd Wipke
University of California
Santa Cruz, CA

Most genetic algorithms use string or tree representations. However, many systems are best represented by graphs; e.g., molecules and circuits. To apply genetic algorithms to graphs, a good crossover operator is necessary. We developed a general-purpose crossover operator for directed and undirected graphs, and used this operator to evolve molecules and circuits. Graph crossover is non-trivial because, unlike strings or trees, graphs cannot always be divided into two parts at a single point because graphs may contain cycles. A steady-state, tournament selection genetic algorithm (JavaGenes) was developed to test our graph crossover operator.

Crossover is easy to implement for strings and trees because these data structures can be divided into two pieces at any point. Although graph crossover can be accomplished by breaking edges, it is more complex because 1) any edge may be a member of one *or more* cycles, 2) graph fragments produced by division may have more than one crossover point ("broken edge") that requires reattachment during fragment combination, 3) when two fragments are combined they may have different numbers of broken edges to be merged, and 4) for a graph crossover operator to potentially reach any possible graph from an initial random population, the operator must be able to create and destroy individual cycles, fused cycles, cages, and combinations of fused cycles and cages.

The primary contribution of this paper is to introduce a crossover operator that 1) operates on any connected directed or undirected graph, 2) divides graphs at randomly generated cut sets, 3) can evolve arbitrary cyclic structures given at least some cycles in the initial population, and 4) always produces connected undirected graphs and almost always produces connected directed graphs.

JavaGenes divides an undirected graph by 1) choosing an initial edge at random, 2) repeatedly finding the shortest path between the initial edge's vertices and 3) removing a random edge from this path until a cut set is found.

Fragments are combined by repeatedly selecting random broken edges from each fragment and combining them. When the broken edges in one fragment are exhausted, the remaining broken edges are randomly discarded or attached to a random vertex in the fragment with no remaining broken edges.

To evolve circuits, edges must be directed and there are special input and output vertices in each circuit. This requires changes to the algorithm. During division, instead of using a random bond to choose the vertices to split, the input and output vertices are chosen and edges on the shortest remaining path between these vertices are broken until a cut set is found. During recombination, only fragments containing an input vertex are combined with fragments containing an output vertex, and vice versa. Curiously, in very rare circumstances this results in a disconnected graph. See [Globus 2000] for details.

JavaGenes was applied to evolving pharmaceutical drug molecules and simple digital circuits. To test JavaGenes we attempted to evolve existing drug molecules. Morphine, cholesterol, and diazepam were successfully evolved by 30-60% of runs within 10,000 generations using a population of 1000 molecules. Correct delay and parity circuits were also evolved.

Graph evolution is subject to a patent [Weininger 1995], but the patented crossover algorithm cannot produce connected children with material from both parents. The patent holder does report success using their system for pharmaceutical drug design. Molecules have also been evolved using tree structures [Nachbar 1998] but crossover was not allowed to break or form cycles.

Since representation strongly affects genetic algorithm performance, adding graphs to the evolutionary programmer's bag-of-tricks should be beneficial. In this effort, circuit evolution was marginal but success was achieved evolving pharmaceutical drug molecules.

[Globus 2000] "JavaGenes: Evolving Graphs with Crossover," A Globus, S Atsatt, J Lawton, and T Wipke, www.nas.nasa.gov/~globus/papers/JavaGenes/paper.html

[Nachbar 1998] Robert B. Nachbar, "Molecular Evolution: a Hierarchical Representation for Chemical Topology and its Automated Manipulation," *Proceedings of the Third Annual Genetic Programming Conference*, University of Wisconsin, Madison, Wisconsin, 22-25 July 1998, pages 246-253.

[Weininger 1995] David Weininger, "Method and Apparatus for Designing Molecules with Desired Properties by Evolving Successive Populations," U.S. patent 5434796, Daylight Chemical Information Systems, Inc.

Extra-Intracellular Transgenetic Algorithm

Marco César Goldbarg
DIMAp, UFRN
Campus Universitário – Lagoa Nova
Natal, Brazil, 59 072-084

Elizabeth Ferreira Gouvêa

Abstract

This paper introduces a new Computational Evolutionary algorithm, EITA, which uses extra and intracellular flows. EITA was applied to the Quadratic Assignment and to the Graph Coloring Problems.

1 INTRODUCTION

Computational Transgenetics (CT) is a metaheuristic that brings the following ideas to the evolutionary algorithms context: To use exogenous and endogenous information to interfere on the processes of formation and modification of individuals of a given population; to use the intracellular flow (Kargupta, 1997) to manipulate individuals; to explore new processes of population improvement using transgenetic agents and competition between agents and individuals; to guide the evolutionary process allowing the occurrence of evolutionary jumps. Recent researches show that genes and culture are inherently linked. Thus, individuals evolve both by anatomical and behavioral selection. The rules that cause anatomical and behavioral elements to come together are called *epigenetic* (Lynch, 1998). To bring information to the evolutionary process, CT uses transgenetic agents to manipulate individuals. A CT agent is composed by one or more memes and an operative method that come from epigenetic rules. *Memes* are the elements of cultural concepts. A meme, in this work, is any proposal to construct a set or block of genes (building block). A meme can be obtained from a number of sources, such as heuristics, etc. The transgenetic agents manipulate individuals of a certain population that evolves and may reinforce the whole process adding new information to it. CT algorithms (Goldbarg & Gouvêa, 2000) are designed to consider the intracellular and epigenetic contexts.

2 EXTRA-INTRACELLULAR TRANSGENETIC ALGORITHM

The Extra-Intracellular Transgenetic Algorithm, EITA, can be described as shown below. An EITA has an underlying Genetic Algorithm (GA). The arrows (→),check list signs (✓) and asterisks (*) mark the statements where intracellular manipulations, epigenetic and the underlying GA steps occur. The meme base required by EITA can be thought as a library containing information about the problem and the instance.

```
 ✓  Load a Meme Base
 ✓  Design a set of agents according to the Meme Base
(*) Generate and evaluate an initial population
    Repeat
        →   Load a subset of agents
        (*) Select a set of parents to generate offspring
        →   Manipulate sensitive parents according to
              epigenetic rules
        (*) Crossover
        (*) Mutation
        →   Liberate chromosomes that can be set free
              (end of agent lifetime)
        (*) Evaluate population fitness
    Until the stop criterion be satisfied
```

3 COMPUTATIONAL EXPERIMENTS

In order to check the transgenetic potential, the approach was applied to the Quadratic Assignment Problem (QAP) and to the Graph Coloring Problem (GCP). Computational experiments showed that chromosome manipulation by transgenetic agents is a powerful tool to guide the search in the solution space.

References

M. C. Goldbarg, and E. Gouvêa (2000). Computational Transgenetics. *X Congreso Latino-Iberoamericano de Investigación de Operaciones y Sistemas*, Mexico.

H. Kargupta (1997). Gene Expression: The Missing Link in Evolutionary Computation, *Genetic Algorithms in Engineering and Computer Science*. John Wiley & Sons.

A. Lynch (1998). Units, Events and Dynamics in Memetic Evolution. *Journal of Memetics - Evolutionary Models of Information Transmission,* 2.

Non-Linear Bit Arrangements in Genetic Algorithms

William A. Greene
Computer Science Department
University of New Orleans
New Orleans, LA 70148
bill@cs.uno.edu
504-280-6755

Abstract

Our earlier research laid a theoretical basis for the contention that genetic algorithms can succeed even if bits are arranged in configurations different from a linear sequence. In the current research we show this success happens in practice, for a number of examples.

In (Greene, 2000) we extended Holland's classic Schema Theorem to the setting that bits are arranged, not necessarily in a linear sequence, but at the nodes of a connected graph. For such potentially much liberalized ways of arranging bits (and given reasonable assumptions; see our theorem), one can expect schema of above average fitness to flourish.

In the current research, we put theory aside and see what happens in practice. We consider a dozen examples, of several different types, wherein bits are arranged in alternative geometries. Of course, with each problem and its associated bit geometry, we should use a crossover operator that is compatible with the geometry, and also conducive to the accumulation of desirable chromosomal subparts. This, coupled with the other familiar forces of geneticism, does lead, in our experiments, to convergence to individuals of high or even maximum fitness.

In this abstract we outline several of our examples.

(1) Twenty Queens Problem. This is the extension of the familiar Eight Queens Problem, to a chess board whose edge-size is 20 squares. A population individual is a 20×20 grid of bits, with 0, resp., 1, meaning the square is empty, resp., is occupied by a queen. For crossover, a random row or column or diagonal line is chosen, then a child is formed by copying bits from one side of the line in one parent and from the other side of the line in the other parent. The initial population consists of individuals each of whose 20 * 20 = 400 bits are chosen randomly; note such an individual starts with far too many queens. An individual's error is found by summing the number of inappropriate queens situated on each row and column and diagonal; maximum error = 1482. An individual's fitness is defined as maximum error minus own error; maximum fitness = 1482, too. Note that the fitness landscape is complex, with many global maxima and many local maxima. Many constraints must be satisfied by a solution to this problem. Experimental results: On 20 trials, each stopped after 2000 generations (population size = 100), the best individual encountered had an average fitness of 1472.1, or 99.3% of the maximum fitness.

(2) A population individual is a complete binary tree of 511 bits. There are 256 leaves on the tree's bottom level, which is at depth 8. For crossover, at random we choose one of the 510 proper subtrees, then form a child by replacing that subtree in one parent by that of the other parent. We define the fitness of an individual to be the number of sibling nodes which have the same bit value (whether it be 0 or 1). There are 255 sibling pairs, so maximum fitness is 255. Note that the fitness landscape has many global and many local maxima. Experimental results: On each of 20 trials, an individual of maximum fitness was found, upon average generation number 480.45.

(3) Bits are arranged in a 3-dimensional cube. Specifically, the bits are positioned at the points (j, k, l) of Euclidean 3-space for which each of j, k, l is an integer in the range 0..10. There are $11^3 = 1331$ bits altogether. For crossover, we cut a cube with a random plane, then form a child by copying bits from one side of that plane in one parent and from the other side of that plane in the other parent. Define the point *antipodal* to (j, k, l) to be the one located symmetrically across the center of our cube; thus, it has coordinates (10-j, 10-k, 10-l). An individual's fitness we take to be the number of antipodal pairs in it which have the same bit value (whether 0 or 1); maximum fitness = 665. Experimental results: On 20 trials, each running to at most 2000 generations, the best individual found had an average fitness of 660.4, or 99.3% of the maximum fitness.

Reference

Greene, William A. (2000). A non-linear schema theorem for genetic algorithms. In D. Whitley *et al*. (eds.), *Proceedings of the 2000 Genetic and Evolutionary Computation Conference*, pp 189-194. Morgan Kaufmann Publ., San Francisco, CA.

Fusion of Coevolutionary GA and Machine Learning Techniques Through Effective Schema Extraction

Hisashi Handa+ **Tadashi Horiuchi*** **Osamu Katai****
Takayuki Kaneko** **Tadataka Konishi**+ **Mitsuru Baba**+

+Faculty of Engineering, Okayama University, Okayama 700-8530, JAPAN
*Dept. of Information Engineering, Matsue National College of Technology, Matsue 690-8518, JAPAN
**Graduate School of Informatics, Kyoto University, Kyoto 606-8501, JAPAN

Abstract

A new Coevolutionary GA is introduced with an ability of schema extraction by machine learning techniques. Considerable improvement of its search ability is done by extracting more efficiently useful schemata from the GA population and then by utilizing them. We examine and compare two kinds of machine learning techniques (C4.5 and CN2) to extract schema information.

In this paper, we introduce a new Coevolutionary Genetic Algorithm (CGA) with schema extraction by Machine Learning (ML) techniques. Basic idea in this CGA is to explore various effective genetic information in the population, i.e., schemata, and to exploit the genetic information in order to guide the population to better solutions [1]. Our CGA consists of two GA populations which independently evolve and affect with each other; the first GA, called "H-GA", searches for the solutions, and the second GA, called "P-GA", searches for the effective schema for the H-GA. Thus, each individual in the P-GA consists of alleles in the H-GA and "don't care symbols" representing a schema in the H-GA.

We aim to improve the search ability of our CGA by extracting more efficiently useful schemata from the H-GA. We will adopt ML techniques as a method to extract useful schemata from H-GA. The training data for ML consist of a large amount of genetic information together with their fitness values which have been searched by the H-GA. Namely, we apply an *inductive learning* using the past data which the H-GA has been searching, by regarding the highly fit individuals as "positive instances" and the lowly fit individuals as "negative instances". Then, we can derive a set of rules which characterize the positive instances and negative instances. By focusing on the rules whose consequent part is "positive class", their antecedent parts mean the pairs of an allele and its value which are important for characterizing the positive instances Therefore, we can regard those pairs as parts of useful schemata. We adopt C4.5[2] and CN2[3] as ML techniques. C4.5 generates a decision tree and then classification rules based on the criteria of information entropy. On the other hand, CN2 derive a set of rules directly by using a general-to-specific beam search through rule-space.

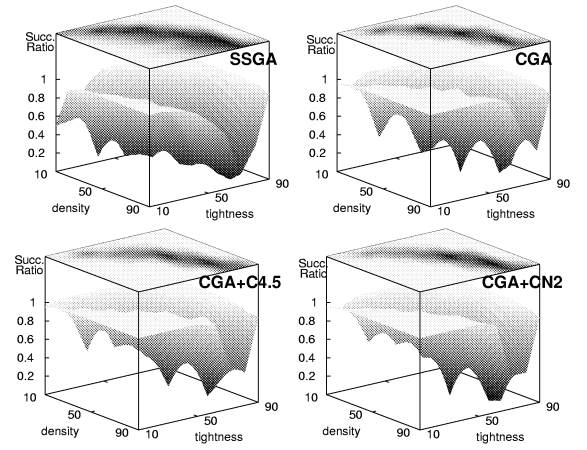

Figure 1: Success ratio for solving general CSPs

We examine Steady-State GA (SSGA), CGA, CGA+C4.5 and CGA+CN2 on general Constraint Satisfaction Problems (CSPs) for a variety of couples of density and tightness. As depicted in Figure 1, experimental results are confirmed us the effectiveness of the proposed method.

References

[1] H. Handa *et. al.*: Coevolutionary Genetic Algorithm for Constraint Satisfaction with a Genetic Repair Operator for Effective Schemata Formation", *Proc. of SMC99*, Vol. III, pp.617-621, 1999

[2] J. R. Quinlan : *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.

[3] P. Clark *et. al.*: The CN2 Induction Algorithm, *Machine Learning*, Vol.3, No.4, pp.261–284, 1989.

Of Heisenberg and Epistasis

Robert B. Heckendorn
Department of Computer Science
University of Idaho
Moscow, ID 83844-1010 USA
heckendo@cs.uidaho.edu

Epistasis is the nonlinear interaction between bits in the domain of a function with respect to computing the value of the function [Jong et al., 1997]. The more bits that simultaneously interact (the higher the epistasis) the greater the degree of freedom to "hide" the optimum anywhere in the subdomain formed by the interacting bits [Heckendorn and Whitley, 1999]. High epistasis, however, is no guarantee of a difficult problem. Nor is low epistasis a guarantee of an easy problem. Still, epistasis represents an important measure of problem structure that can be used to assess a fitness landscape [Heckendorn and Whitley, 1999].

The **Walsh transform** of a function allows us to restate a function over L bit strings as a vector of 2^L **Walsh coefficients**, each representing the "quantity" of epistatic interaction for each of 2^L possible sets of bit interaction. The Walsh transform can be thought to map a function from **function space** to **epistatic space** [Reeves and Wright, 1995] [Heckendorn and Whitley, 1997].

It is well known that the average function value for an L bit function over a hyperplane with k fixed bits can be computed by summing 2^k Walsh coefficients [Goldberg, 1989]. That is, the smaller the hyperplane one sums over the more Walsh coefficients are needed to compute the sum precisely. In this way it is reminiscent of the Heisenberg Uncertainty Principle. I have extend this result by showing a more general theorem called the Epistatic Uncertainty Theorem describing the trade-off between specificity of information about epistasis and function value. I have shown the versatility of the theorem and related corollary; and the practical relationship to the Fast Walsh Transform and in developing techniques for the rapid calculation of epistasis.

The theorem is stated in terms of numbered hyperplanes [Heckendorn and Whitley, 1999]: Given a function $f : \mathcal{B}^L \rightarrow \mathbb{R}$ and a hyperplane partitioning mask $m : m \in \mathcal{B}^L$ then:

$$\frac{1}{|h_{m,n}|} \sum_{x \in h_{m,n}} f(x)\psi_x(unpack(z, \overline{m})) = \sum_{j \,:\, j \in h_{\overline{m},z}} w_j \psi_j(unpack(n, m))$$

Intuitively, this shows the relation between summing rows of a partition matrix of a function and columns of the partition matrix of the Walsh coefficients. Many related theorems have been shown, but this space is too small to contain them.

References

[Goldberg, 1989] Goldberg, D. (1989). Genetic algorithms and walsh functions: Part i, a gentle introduction. *Complex Systems*, 3:129–152.

[Heckendorn and Whitley, 1997] Heckendorn, R. B. and Whitley, D. (1997). A walsh analysis of nk-landscapes. In Bäck, T., editor, *Proc. of the 7th Int'l. Conf. on GAs*, Palo Alto, CA. Morgan Kaufmann Publishers, Inc.

[Heckendorn and Whitley, 1999] Heckendorn, R. B. and Whitley, D. (1999). Walsh functions and predicting problem complexity. *Evolutionary Computation*, 7(1):69–101.

[Jong et al., 1997] Jong, K. A. D., Potter, M. A., and Spears, W. M. (1997). Using problem generators to expore the effects of epistasis. In Bäck, T., editor, *Proc. of the 7th Int'l. Conf. on GAs*, pages 338–339, Palo Alto, CA. Morgan Kaufmann Publishers, Inc.

[Reeves and Wright, 1995] Reeves, C. and Wright, C. (1995). Epistasis in genetic algorithms: An experimental design perspective. In Eschelman, L. J., editor, *Proc. of the 6th Int'l. Conf. on GAs*, pages 217–224. Morgan Kaufmann Publishers, Inc.

An Analysis of Mate Selection in Genetic Algorithms

Chien-Feng Huang
Center for the Study of Complex Systems, 4477 Randall Lab.
University of Michigan, Ann Arbor, MI 48109
cfhuang@engin.umich.edu

Given a problem to be solved, if a simple GA adopts the fitness proportionate selection scheme for reproduction (Mitchell, 1996), then a pair of parent chromosomes are selected from the population, the probability of selection being an increasing function of fitness. In this implementation, the first individual is passively assigned the mating partner by the mechanism of natural selection. Consequently, each individual is forced to mate with some partners that may not carry promising genetic material. To further elucidate this point, we can consider the following example:

Suppose the population is composed of bit-strings of length 8 and the relevant building blocks are 1111**** and ****1111 (* can be either 1 or 0), and each contributes fitness of 4 to the strings in the population. Then, for example, a string X, 11110000, is of fitness 4, and the optimal string is 11111111, whose fitness is 8. Now given string X, and two candidate mating partners, Y_1 (11110000) and Y_2 (00001111), under the mechanics of the simple GA above, Y_1 and Y_2 have the same probability to be chosen for mating since these two strings are of the same fitness. However, if we are concerned with finding the optimal string, apparently, string Y_2 is better than Y_1 because the mating between Y_2 and X is likely to generate the optimum, yet it is not the case if X mates with Y_1. This implies that natural selection may not have enough capability to distinguish individuals of the same fitness, yet of quite different string structures. To compensate for this deficiency of natural selection, we introduce and investigate several mate selection schemes that allow individuals to be able to recognize different string structures of the same fitness so as to search for more appropriate mating partners, hoping to improve the GAs' performance. Our setup is to first adopt the tournament selection scheme (Mitchell, 1996) as the role of natural selection. Then during each mating event, a binary tournament selection—with probability 1.0 the fitter of the two randomly sampled individuals is chosen—is run to pick out the first individual, then choosing the mate according to the following five different schemes:

A. Run the binary tournament selection again to choose the partner.

B. Randomly choose two candidate partners; then the one more dissimilar to the first individual is selected for mating.

C. Randomly choose two candidate partners; then the one more similar to the first individual is selected for mating.

D. Run another two times of the binary tournament selection to choose two highly-fit candidate partners; then the one more dissimilar to the first individual is selected for mating.

E. Run another two times of the binary tournament selection to choose two highly-fit candidate partners; then the one more similar to the first individual is selected for mating.

With the five mate selection schemes, we present the comparison among the performances of these GAs on one test function: the Royal Road function R_1 (Mitchell, 1996).

For each of the five mate selection schemes, the experiments performed were based on one-point crossover rate 0.7, mutation rate 0.005, population size 128, and the number of maximum function evaluations allowed is 200000. Our objective is to find the global optimum and measure how many function evaluations are needed before the number of maximum evaluations is reached. The results are shown in Table 1, in which the mean function evaluations and the standard deviation were calculated over 30 runs if the optimum was reached in all runs. Otherwise, the number of runs with the optimum being reached is shown.

Table 1: Summary of Experimental Results on R_1

	Mean Function Evaluations to Optimum	
Scheme	Mean	Standard Deviation
A	41551	25916
B	79284	40235
C	0 runs reached optimum	–
D	21601	11965
E	0 runs reached optimum	–

The results show that the speed of scheme D is about two times that of scheme A, and four times that of scheme B. These results indicate that, on Royal Road R_1, it is most beneficial to allow individuals to actively choose dissimilar mates that are filtered by natural selection. On the other hand, schemes C and E were designed to avoid producing lethal offspring via mating being allowed to take place only between relatively similar individuals. The poor performance of these two schemes shows that this idea does not bring forth advantage for finding the global optimum on R_1.

Reference

Mitchell M.(1996). *An Introduction to Genetic Algorithms.* Cambridge, MA: MIT Press.

Mixed Size Tournament Selection

Reinhold Huber
Aero Sensing Radarsysteme GmbH.
c/o DLR Oberpfaffenhofen
D-82234 Wessling

Thomas Schell
Dept. of Scientific Computing
University of Salzburg
A-5020 Salzburg

Abstract

In genetic algorithms, tournament schemes are often applied as selection operators. The advantage are simplicity and efficiency. On the other hand, major deficiencies related to tournament selection are the coarse scaling of the selection pressure and the poor sampling accuracy. We introduce a new variant of tournament selection which provides an adjustable probability distribution, a fine-tuning facility for the selection pressure and an improved sampling accuracy at the cost of a minimal increase of the complexity and with almost no loss of efficiency.

1 THE METHOD

To fit mixed size tournament selection (msTS) to an arbitrary selection scheme X we approximate the distribution of the selection probabilities p_i^X, where i is an index to an individual in the ranked population

The selection probabilities p of tournament selection (TS) with replacement is defined by

$$p_{i,t}^A = \frac{i^t - (i-1)^t}{N^t}$$

The definition above includes the population size $N \in \mathbb{N}$, an index to the sorted individuals i $(1 \leq i \leq N)$ and the tournament size $t \geq 1$.

The selection probabilities of the whole population are represented by a vector $\vec{p}_t^A = \left(p_{1,t}^A \dots p_{N,t}^A\right)$.

We approximate an arbitrary selection scheme X with a weighted sum of TS of varying size t. The parameters of msTS which need to be identified are the maximal tournament size and the weights α_t of the involved tournament schemes.

$$f(\alpha_1, \dots, \alpha_n) := \|\vec{p}^X - \sum_{t=1}^{n} \alpha_t \vec{p}_t^A\|_\infty, \quad n \leq N$$

Obviously, we need to minimize the function $f(\alpha_1, \dots, \alpha_n)$ under the constraints given below.

$$\alpha_t \geq 0, \quad t \in \{1, \dots n\}; \quad \sum_{t=1}^{n} \alpha_t = 1; \quad \alpha_t N \in \mathbb{N}_0$$

1.1 Fitting msTS to Exponential Ranking

The selection probability of the i–th ranked individual using Exponential Ranking (ERk) is given by

$$p_i^E = \frac{c-1}{c^N - 1} c^{N-i}, \quad i \in \{1 \dots N\} \quad c \in [0,1]$$

Table 1 shows the msTS parameters, e.g. ERk with $c = 0.99$ is approximated by 58 1–tournaments, 26 2–tournaments, and 16 3–tournaments.

Table 1: Weighting parameters $\alpha_t N$ and maximum differences D between ERk and msTS with $N = 100$.

c	n	$\alpha_t N$						D
0.99	3	58	26	16				0.000069
0.98	4	31	34	7	28			0.00008
0.975	4	23	29	0	48			0.00021
0.97	4	20	12	1	67			0.000469
0.96	5	11	15	0	1	73		0.000467
0.95	6	5	14	3	0	0	78	0.00049

2 CONCLUSION

We have introduced a new variant of TS i.e. msTS which provides a flexible probability distribution and a fine-tuning facility for the selection pressure. We have demonstrated the adaptability of msTS by simulating ERk. We would like to note that the improvements were achieved at a minimal increase of complexity while preserving the efficiency of the original TS.

On Termination Criteria of Evolutionary Algorithms

Brijnesh J. Jain
Department of Computer Science
Technical University Berlin
Franklinstr. 28/29
D-10587 Berlin, Germany

Hartmut Pohlheim
DaimlerChrysler AG
Research and Technology
Alt-Moabit 96a
D-10559 Berlin, Germany

Joachim Wegener
DaimlerChrysler AG
Research and Technology
Alt-Moabit 96a
D-10559 Berlin, Germany

For the practical use a termination criterion should determine the end of a search process as soon as the EA is not sufficiently efficient, i.e. when the EA is degenerated to a random search or no significant improvements of the best objective value can be expected within a forseeable space of time. The significance of an improvement is highly dependent on the desired precision of the results and on the envisaged goals of the problem solving process. Each improvement corresponds to an economical profit and each iteration causes costs. From this point of view an optimization can run without loss up to a fixed precision. Improvements obtained without loss are considered to be significant.

A suitable point in time to terminate an optimization run prevents premature termination as well as further computations to no avail. Hence the efficiency of a numerical algorithm for optimization is not only dependent on its computational performance but also on its behavior to terminate a run. Furthermore in some real-world applications of EA there are attempts to automate the optimization process as in evolutionary testing. In evolutionary testing, temporal correctness of real-time systems is verified with the assistance of EA whereby the objective functions are discrete. Since real-time systems are often safety-relevant, temporal correctness plays an important role. Consequently, reliable and intelligent termination criteria as well as general recommendations how to employ termination criteria for the automated use of EA are required.

Here an overview of partly well-known termination criteria and a newly defined criterion (ClusTerm) based on cluster analysis is given. Conventional criteria either use objective values or the distribution of individuals in the search space to decide the end of a run. To get other, probably more reliable and intelligent termination criteria, it is reasonable to combine and evaluate both, information about objective values and distribution of individuals. This approach is implemented by using cluster analysis on the fittest individuals. The intelligence behavior of ClusTerm is incorporated in the fluctuations of the aggregate sizes of elitist clusters.

All termination criteria were systematically tested by extensive experiments and evaluated with respect to their reliability and performance. Initiated by the necessity of reliable and intelligent termination criteria in evolutionary testing, where the objective functions are discrete, we primarily considered multidimensional step functions in our empirical analysis. This test set can also be used for benchmarking EA strategies on discrete problems. The results of the empirical analysis were verified by real-world applications in evolutionary testing. The criterion ClusTerm proved to be promising for problem domains with discrete objective functions.

Finally, for each termination criterion guidelines for the practical employment and automated application of EA as well as references when certain criteria should not be used are suggested. Obviously, terminiation criteria behave differently for varying EA strategies and objective functions. Therefore it is impossible to formulate a general rule for optimal use of a termination criterion. But in real-world applications like evolutionary testing an automated employment of EA is desired, since software tester often are not familiar with the behavior of EA. For this reason we give some practical guidelines of the application of termination criteria for unskilled practioners and the automated use of EA. Our recommendations are not only restricted to discrete objective functions. They can also be accepted under reserve if one is dealt with continuous optimization problems. Our suggestions for the continuous case are based on first unsystematical experiments with common continuous test functions.

A Remark on Solving Minimax Problems with Coevolution

Mikkel T. Jensen
Department of Computer Science, University of Aarhus, Denmark.
email: `mjensen@daimi.au.dk`, http: `www.daimi.au.dk/~mjensen/`

Abstract

Minimax optimization problems are relevant to research in scheduling, mechanical structure optimization, network design and constrained optimization. Recent papers have demonstrated that coevolutionary algorithms have a potential for solving this kind of problem. In the present paper it is argued that the approaches used so far will fail if the problem does not have a symmetric property. A new approach solves the problem.

1 Summary

A minimax problem is an optimization problem in which the task is to find the solution $x \in X$ with the minimum worst case cost F, where some problem parameter $s \in S$ is chosen by an adversary. The minimax problem can be formulated: Minimize

$$\varphi(x) = \max_{s \in S} F(x, s) \quad \text{subject to } x \in X. \tag{1}$$

A minimax problem can be seen as a game between two players. The first player controls the *solution*, x, and wants to minimize the cost $F(x, s)$. The second player controls the *scenario*, s, and wants to maximize the cost. Recent papers have used coevolution to solve minimax problems from scheduling domains [Herrmann 1999], constrained optimization and mechanical structure optimization.

The previously proposed algorithms have been based on the idea of having two coevolving populations, P_X holding solutions and P_S holding the scenarios. The idea of using coevolution is to save time by estimating $\varphi(x)$ by only evaluation $F(x, s)$ for a low number of s values. During fitness evaluation, $F(x, s)$ is evaluated for every combination of solution $x \in P_X$ and scenario $s \in P_S$. Every solution $x_i \in P_X$ is assigned the objective function $h(x_i) = \max_{s \in P_S} F(x_i, s)$, which is to be minimized. Every scenario $s_j \in P_S$ is assigned the objective function $g(s_j) = \min_{x \in P_X} F(x, s_j)$. According to [Herrmann 1999] this objective is to be maximized.

The minimax problems solved in earlier coevolutionary work all have a symmetric property. They satisfy

$$\min_{x \in X} \max_{s \in S} F(x, s) = \max_{s \in S} \min_{x \in X} F(x, s). \tag{2}$$

If a minimax problem does not satisfy (2), the previously proposed GAs are likely to fail.

The problem is in the fitness evaluation of the scenarios. Assigning the scenarios the objective function $g(s_j) = \min_{x \in P_X} F(x, s_j)$ will reward scenarios which lead to moderately high values of F for all x, but punish a scenario if it leads to a low value of F for some x, even if it leads to high values of F for other x values. Because of this, the scenario which gives rise to the highest value of F in (1) may be assigned a low fitness, and P_S may fail to converge or converge on a wrong part of the search-space. This in turn can lead to P_X failing to converge to the right solution, since the fitness evaluation will not be an estimate of $\varphi(x)$.

A new fitness evaluation for the scenarios has been developed. The idea is to give all scenarios which lead to a relatively high F value for some x a high fitness. Experiments have demonstrated that a GA similar to the GA in [Herrmann 1999] shows very poor performance on a number of simple numerical minimax problems not satisfying (2), while a GA based on the new fitness evaluation performs much better.

References

[Herrmann 1999] J. W. Herrmann. A Genetic Algorithm for Minimax Optimization Problems. *Proc. 1999 Congress on Evolutionary Computation*, pages 1099-1103, 1999.

Genetic-guided Model-based Clustering Analysis

Hui-Dong Jin **Kwong-Sak Leung**
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, Hong Kong
{hdjin,ksleung}@cse.cuhk.edu.hk
Tel: +852 2609 8412.

Man-Leung Wong
Department of Information Systems
Lingnan College
Tuen Mun, Hong Kong
mlwong@ln.edu.hk
Tel: +852 2616 8093.

Clustering, or the unsupervised classification of data items into clusters, can reveal some intrinsic structures like the optimal number of clusters among data sets. Model-based clustering approaches, describing a data set with a mixture model, are able to determine the optimal number of clusters in theory. The common approach (EnumEM) to select the number of clusters is based on the enumeration strategy by using the Expectation Maximization (EM) algorithm [1]. However, it often fails to find the globally optimal model, especially for ill-separated data sets.

We propose two new genetic-guided model-based clustering algorithms, GAXEM and GAEM, to enhance EnumEM. GAs can be used both to explore the natural clusters and to determine the optimal number of clusters among data sets. This distinguishes the proposed algorithms from other genetic-guided clustering algorithms.

In our algorithms, each mixture model is directly coded as a chromosome to represent a clustering solution. The *Bayesian Information Criterion* (*BIC*) [1] serves for the fitness value of the mixture model. To enhance the performance, we develop several specific genetic operators. The EM algorithm is embedded to calculate the *BIC* value to enable the algorithms to search around these local minima. Besides several intuitive genetic operators, we also develop a specific HAC crossover operator, which is motivated by the model-based Hierarchical Agglomerative Clustering (HAC) algorithm [2]. To get a new child, the closest pair of clusters in two parents merges iteratively, and the mergence ends with the lowest *BIC* value. The main difference between GAXEM and GAEM is that only the former uses the HAC crossover operator.

As shown in Table 1, both GAXEM and GAEM can determine the optimal number of clusters more frequently than EnumEM on 9 synthetic data sets. For example, EnumEM fails to detect 12 clusters among

DataSet		EnumEM			GAXEM			GAEM		
	K	Accu	Suc	Time	Accu	Suc	Time	Accu	Suc	Time
A	4	56.8	3	1799	63.5	6	1374	63.5	6	1079
B	5	51.4	1	2581	53.7	6	4250	53.2	5	2096
C	6	42.6	2	2184	48.3	6	5685	48.3	6	3115
D	7	42.8	1	3135	63.7	5	4885	51.5	3	4803
E	8	62.0	2	3224	64.7	6	9780	63.4	4	4262
F	9	53.9	2	3318	60.8	6	6891	60.8	6	6399
G	10	55.0	1	4369	59.2	4	17921	55.2	2	6906
H	11	43.2	0	8570	52.9	4	28732	42.7	3	15871
I	12	38.5	0	9149	51.4	5	35296	42.3	2	24487
Average		49.6	1.3	4259	57.6	5.3	12757	53.4	4.1	7669

Table 1: The simulation results on 9 synthetic data sets of the EnumEM, GAXEM and GAEM clustering algorithms. 'K' indicates the optimal number of clusters in the data set, 'Accu' indicates the average clustering accuracy value(%) and 'Suc' indicates the successful trials on finding the optimal number of clusters within 6 runs. The unit for the average execution time 'Time' is second.

dataSetI within 6 runs. However, GAEM succeeds twice and GAXEM succeeds 5 times. The significant different performance between GAXEM and GAEM also substantiates the significance of the proposed HAC crossover operator.

The work was partially supported by RGC Grant CUHK 4161/97E of Hong Kong.

References

[1] Christophe Biernachi, Gilles Celeux, and Gérard Govaert. An improvement of the NEC criterion for assessing the number of clusters in a mixture model. *Pattern Recognition Letters*, 20:267–272, 1999.

[2] Chris Fraley. Algorithms for model-based Gaussian hierarchical clustering. *SIAM Journal on Scientific Computing*, 20(1):270–281, Jan 1999.

Genetic Symbiosis Algorithm for Multiobjective Optimization Problems

Jiangming Mao, Kotaro Hirasawa, Jinglu Hu and Junichi Murata
Department of Electrical and Electronic Systems Engineering, Kyushu University,
6-10-1, Hakozaki, Higashi-ku, Fukuoka 812-8581, Japan

Evolutionary Algorithms(EAs) are often well-suited for optimization problems. Since 1980's, the interest in multiobjective problems has been expanding rapidly. Various evolutionary algorithms for multiobjective problems have been developed which are capable of searching for multiple solutions concurrently in a single run. One of the typical methods for multiobjective problems is VEGA (vector evaluated genetic algorithms) proposed by J.D. Schaffer. VEGA is a natural extension of simple genetic algorithms (SGA) in the sense that the individuals are divided and reproduced independently according to each objective function. The problem of VEGA is difficult to obtain equally distributed solutions in the Pareto space. In order to overcome this problem, Fonseca and Fleming proposed MOGAs (multiple objective genetic algorithm) based on the ranking selection using the concept of dominated and non-dominated solutions. In addition, there have been developed many other multiobjective evolutionary methods such as NPGA(niched Pareto genetic algorithms), NSGA(nondominated sorting genetic algorithm). But, these methods still have the problem, that is, the distribution of the solutions in the Pareto space can not be controlled easily by user's requests.

The reasons are the following two factors, such as selection pressure and genetic drift. We can divide an evolutionary process by using ranking methods into two cases. One is when not all the points are Pareto optimal points and the other is when all points are Pareto optimal points. In the first case, the selection pressure should make the population into the second case because the second case is the target of the optimal selection. Let us pay attention to the second case. In this case, each point has the same chance of producing offspring and we can consider the offspring of them are selected randomly. If we eliminate the effects of mutation and crossover, we can see the effects of genetic drift, making the population converge randomly. Because crossover and mutation should make the population diverge randomly, we can find the distribution of the solutions in the Pareto space can not be controlled easily.

In order to overcome this problem, let us pay attention to ecosystems which hold a very wide diversity. Every species seek their habitants called niche by adapting themselves to the ever changing environments. And in the niche they are interacted with each other by competing, exploiting and benefiting. These relations are generally called symbiosis. In other words, the selection pressure can be produced not only by some environmental conditions such as temperature, water and food, but also some interactions among species. So in this paper, a set of symbiotic parameters which can represent the symbiotic relations similar to ecosystems are introduced to modify the fitnesses for each individual. The modified fitnesses are used for reproduction. The symbiotic parameters are determined by the distance in genome and fitness space between individuals. In the paper, we use Fuzzy Inference to determine the relations between symbiotic parameters and the distance in genome and fitness space. User's requirements can be described as criterion functions. Because the fuzzy rules are so complex like in ecosystems that we can not preset, we train the parameters in fuzzy inference to realize the user's requirement.

According to several simulations on some convex multiobjective functions and some unconvex multiobjective functions, we can find that the fuzzy parameters can be trained to realize the required distribution of individuals in the genome and the fitness space.

A New Parallel Distributed Genetic Algorithm Applied to Traveling Salesman Problems

Mitsunori MIKI
Knowledge Engineering Dept.
Doshisha University
Kyoto, 610-0321 Japan

Tomoyuki HIROYASU
Knowledge Engineering Dept.
Doshisha University
Kyoto, 610-0321 Japan

Takanori MIZUTA
Graduate Student
Doshisha University
Kyoto, 610-0321 Japan

Abstract

This paper proposes a new method of genetic algorithms (GAs) for dicrete optimization problems. For continuous problems, it has been reported that parallel distributed genetic algorithms (PDGAs) show higher performance than conventional GAs. But, for discrete optimization problems, the performance of PDGAs has not been clearly shown. We examine the performance of conventional GAs, distributed GAs, and the proposed method for a typical optimization problem, the Traveling Salesman Problem(TSP). The features of the proposed method are based on multiple crossover operations applied to the entire population (Centralized Multiple Crossover: CMX) and the isolated DGA.

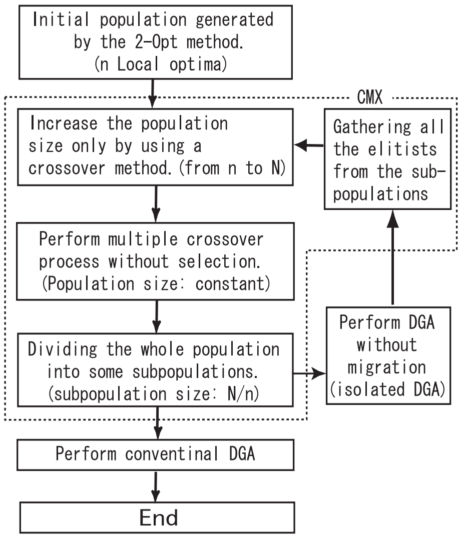
Initial population generated by the 2-Opt method. (n Local optima)
CMX
Increase the population size only by using a crossover method.(from n to N)
Gathering all the elitists from the sub-populations
Perform multiple crossover process without selection. (Population size: constant)
Dividing the whole population into some subpopulations. (subpopulation size: N/n)
Perform DGA without migration (isolated DGA)
Perform conventinal DGA
End

Figure 1: Flowchart of the proposed method.

1 PROPOSED METHOD

The fundamental concept of the proposed method is as follows. For problems with non-separable objective functions such as TSP, the global optimum can be obtained by appropriately combining their minimum elements of the local optima. The local optima is obtained by some heuristic search methods or a DGA without migration. On the other hand, The appropriate combination of the minimum elements of the solutions is performed by using multiple crossovers without selection.

2 EXPERIMENTS ON CMX

Figure 2 shows the effect of the number of repeated CMXs on the histories of the total distance. In this figure, CMX1, 2, and 5 represent one, two, and five times CMX processes, and it is recognized that increase in the number of the CMX processes yields higher performance. Several other experimental results also show the similar tendencies and the proposed method is found to be very effective for discrete optimization problems.

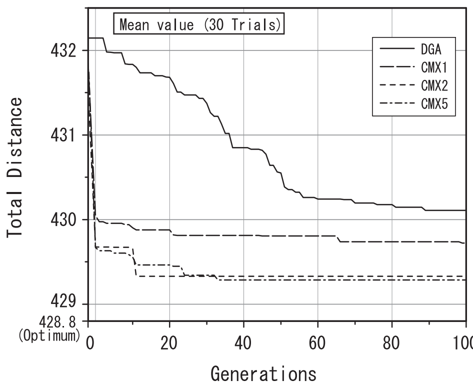

Figure 2: Effect of the repeated CMX processes.

Moments of Schema Fitness Distributions for Multary Alphabets

Bart Naudts[1]
Intelligent Systems Lab
University of Antwerp, RUCA
bart.naudts@ua.ac.be

Luk Schoofs
Intelligent Systems Lab
University of Antwerp, RUCA
luk.schoofs@ua.ac.be

Alain Verschoren
Intelligent Systems Lab
University of Antwerp, RUCA
alain.verschoren@ua.ac.be

This abstract shows the results of (Naudts et al., 2001) in which a number of results recently published by Heckendorn and Whitley, about the moments of schema fitness distributions expressed in terms of Walsh coefficients, are generalized.

String positions run from 0 to $\ell - 1$; the set of positions is denoted by L. Each position takes a value in $\Sigma = \{0, \ldots, n-1\}$. Note that we will often jump between a string in Σ^L and its integer representation as a number between 0 and $n^\ell - 1$. Given a schema $h \in (\Sigma \cup \{*\})^L$, its fitness distribution is defined as the distribution of fitness values of the strings belonging to the schema. The number of strings generated by schema h is denoted with $|h|$. A schema $h = h_0 \ldots h_{\ell-1}$ is a string over the alphabet $\Sigma' = \Sigma \cup \{*\}$, with $*$ the "don't care" symbol. We will use the functions

$$\beta(h_i) = \begin{cases} 0 & \text{if } h_i = * \text{ or } h_i = 0 \\ h_i & \text{otherwise.} \end{cases}$$
$$*(h) = \{i \in L \mid h_i = *\}$$
$$0(s) = \{i \in L \mid s_i = 0\}.$$

The *Balanced Sum Theorem* was proved by (Heckendorn et al., 1999) for binary alphabets. The *Complex Walsh Transform* is described in (Suys, 1998).

Balanced Sum Theorem for Multary Alphabets Using the notations as before, it holds that

$$\sum_{x=0}^{n^\ell-1} \psi_j(x) = \begin{cases} 0 & \text{if } j \neq 0 \\ n^\ell & \text{otherwise.} \end{cases}$$

The Balanced Sum Theorem for Hyperplanes and Multary Alphabets Let $h \in (\Sigma \cup \{*\})^L$ be a schema with $*(h) \neq \emptyset$. It holds that

$$\sum_{x \in h} \psi_j(x) = \begin{cases} 0 & \text{if } *(h) \not\subseteq 0(j) \\ |h|\psi_j(\beta(h)) & \text{otherwise.} \end{cases}$$

Hyperplane Averaging Theorem for Multary Alphabets Let $h \in (\Sigma \cup \{*\})^L$ be a schema with $*(h) \neq \emptyset$. It holds that

$$\frac{1}{|h|}\sum_{x \in h} f(x) = \sum_{j:\, *(h) \subset 0(j)} w_j \psi_j(\beta(h)).$$

The r^{th} moment about the function mean in terms of the Walsh coefficients of the function is given by

$$\mu_r = \sum_{\substack{\forall i:\, 0 \neq a_i \in \Sigma^L \\ a_1 \oplus a_2 \oplus \ldots \oplus a_r = 0}} w_{a_1} w_{a_2} \ldots w_{a_r}$$

where $\oplus$ denotes the addition in $(\mathbb{Z}/n\mathbb{Z})^\ell$.

The r^{th} moment of a schema about the function mean in terms of the Walsh coefficients of the function is given by

$$\mu_r = \sum_{\substack{\forall i:\, 0 \neq a_i \in \Sigma^L \\ *(h) \subseteq 0(a_1 \oplus \ldots \oplus a_r)}} w_{a_1} \ldots w_{a_r} \psi_{a_1 \oplus \ldots \oplus a_r}(\beta(h)).$$

The r^{th} moment of a schema about the mean of the schema in terms of the Walsh coefficients of the function is given by

$$\mu_r = \sum_{\substack{\forall i:\, a_i \in \Sigma^L \\ *(h) \subseteq 0(a_1 \oplus \ldots \oplus a_r)}} w_{a_1} \ldots w_{a_r} \psi_{a_1 \oplus \ldots \oplus a_r}(\beta(h))$$

with the additional restriction of $\forall a_i : *(h) \not\subseteq 0(a_i)$.

References

Heckendorn, R., Rana, S., and Whitley, D. (1999). Polynomial time summary statistics for a generalization of MAXSAT. In *GECCO '99*, pages 281–288. Morgan Kaufmann Publishers.

Naudts, B., Schoofs, L., and Verschoren, A. (2001). Moments of schema fitness distributions for multary alphabets. Technical Report 01–1, University of Antwerp (RUCA).

Suys, D. (1998). *A Mathematical Approach to Epistasis.* PhD thesis, Department of Mathematics and Computer Science, University of Antwerp, Belgium.

[1]Is a post doctoral researcher of the Fund for Scientific Research – Flanders (Belgium) (F.W.O.).

Behavior Analysis of Real–Valued Evolutionary Algorithms for Independent Loci in an Infinite Population

Tatsuya Nomura
Faculty of Management Information, Hannan University
Matsubara, Osaka 580–8502, Japan

Although most of theoretical researches on behavior of Evolutionary Algorithms (EAs) through generations are ones using discrete variables as chromosomes, some researchers have focused their attention on EAs that use real–valued chromosomes. Here, we also provide results on behavior of EAs with real–valued chromosomes for cases that stochastic variables on some loci of chromosomes and them on the others are independent each other, by combining our previous results (Nomura and Shimohara, 2001) with them of Qi and Palmieri (1994).

We formalize EA as follows; a chromosome is a stochastic variable in the m–dimensional Euclidean space, and alternation of one generation consists of roulette selection, recombination, and mutation by addition of independent noise with zero mean, in that order. Furthermore, we assume that the fitness function f has only finitely many global maxima and finitely many discontinuous points, and satisfies $0 < f(x) < \infty$ for $\forall x \in R^m$. Moreover, we describe the probability density function (pdf) of chromosomes in the k–th generation as $p^{(k)}(x)$. Under these assumptions, Qi and Palmieri (1994) proved that (i) the pdf after selection is $\frac{f(x)p^{(k)}(x)}{\int_{R^m} f(x)p^{(k)}(x)dx}$, (ii) the pdf after mutation is the convolution of the pdf before mutation and the pdf of additive noise in mutation, and (iii) selection and mutation maintain Gaussian property in the population.

First, we assume that f is separable for some groups of variables in the sense of $f(x_1, \ldots, x_m) = \prod_{j=1}^{L} f_j(x_{i_{j1}}, \ldots, x_{i_{jd_j}})$ $(\sum_{j=1}^{L} d_j = m)$, and the corresponding groups of loci are independent each other in the k–th generation (that is, $p^{(k)}(x_1, \ldots, x_m) = \prod_{j=1}^{L} p_j^{(k)}(x_{i_{j1}}, \ldots, x_{i_{jd_j}})$ and $p_j^{(k)}$ is the pdf of $i_{j1}, \ldots, i_{jd_j}$–th loci in the k–th generation). Then, it is easily shown that the same groups of loci in the $(k+1)$–th generation are independent each other if recombination maintains this independence.

Second, we consider a special case of the above assumptions on the fitness function and loci. If f is completely separable and all the loci in the k–th generation are independent each other (that is, $L = m$), it can be shown that all the loci in the $(k+1)$–th generation are independent each other if recombination is any of one–point, multi–point, and uniform crossovers. This is shown by the fact that the pdf after these recombinations is described as the sum of the products of the pdf of some loci and that of the other loci before the recombinations (Nomura and Shimohara, 2001).

Finally, as a more special case, we assume that f is completely separate and Gaussian (that is, $f(x) = \prod_{i=1}^{m} \exp(-Q_i(x_i - x_i^*)^2/2)$), and the pdf in the 0–th generation is independent Gaussian with mean values $(\mu_1^{(0)}, \ldots, \mu_m^{(0)})$ and variances $(\Sigma_1^{(0)}, \ldots, \Sigma_m^{(0)})$. Moreover, we assume that mutation is also independent Gaussian with variances $(\sigma_{w1}, \ldots, \sigma_{wm})$ in any k-th generation. If recombination is any of one–point, multi–point, and uniform crossovers, then it is shown that the pdf in any k–th generation is also independent Gaussian with mean values $(\mu_1^{(k)}, \ldots, \mu_m^{(k)})$ and variances $(\Sigma_1^{(k)}, \ldots, \Sigma_m^{(k)})$. Furthermore, we can obtain recursive equations of $\mu_i^{(k)}$ and $\Sigma_i^{(k)}$, and it is shown that $\lim_{k\to\infty} \mu_i^{(k)} = x_i^*$ and $\lim_{k\to\infty} \Sigma_i^{(k)} = \frac{1}{2}\left\{\sigma_{wi} + \sqrt{\sigma_{wi}^2 + \frac{4\sigma_{wi}}{Q_i}}\right\} > \sigma_{wi}$.

References

Qi, X. and Palmieri, F. (1994). Theoretical Analysis of Evolutionary Algorithms With an Infinite Population Size in Continuous Space Part I: Basic Properties of Selection and Mutation. *IEEE Transactions on Neural Networks*, 5(1):102–118.

Nomura, T. and Shimohara, K. (2001). An Analysis on Two–Parent Recombinations for Real–Valued Chromosomes in an Infinite Population. *Evolutionary Computation* (in print).

Tackling Multimodal Problems in Hybrid Genetic Algorithms

Prasanna V. Parthasarathy, David E. Goldberg and Scott A. Burns
Department of General Engineering
University of Illinois
Urbana, IL 61801
{parthasa,deg,s-burns}@uiuc.edu

A method is proposed to address the issue of multimodality while using hybrid genetic algorithms (GAs). The issue of handling explicitly multimodal functions using hybrid GAs was addressed in Seront and Bersini (2000). Using a similar idea and also ideas closely related to those of a coevolutionary sharing introduced by Goldberg and Wang (1997) and a variable radius niching technique used in Gan and Warwick (2000), a model building sharer (MBS) has been developed. The proposed method has been applied to a real-world problem, namely sizing of member for fully-stressed design of frame structures.

The motivation for developing the method comes from the fact that traditional niching via sharing does not perform well in hybrid GAs. Some thought shows why sharing, which typically uses phenotypic distance for estimating diversity, fails to perform its job. Figure 1 illustrates this idea. From the figure one can see that, even though the two starting points lead to the same optimum point, conventional sharing will not share their fitnesses as much as Baldwinian sharing would.

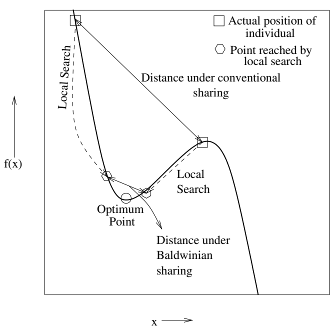
Actual position of individual
Point reached by local search
Local Search
Distance under conventional sharing
f(x)
Local Search
Optimum Point
Distance under Baldwinian sharing
x

Figure 1: Conventional vs. Baldwinian sharing

However, it must be noted that the problem occurs only when using a Baldwinian learning mechanism where the termination point under local search is not back-substituted into the population. Instead, only the fitness of the termination point is taken. In Lamarckian learning where both the fitness and the point itself are taken, conventional sharing should perform just as well as Baldwinian sharing.

The idea that local search takes all points in a given basin of attraction to the same optimum point was exploited in Seront and Bersini (2000). That work used clustering algorithms to identify basins of attraction and evaluated only one point from each cluster to get the optimum point. Selection was however based on the fitness value of points at the starting positions and the worst individual in a cluster was replaced by the optimum point in that cluster.

Table 1: Number of optima using various sharers

Frame	Evals	No. Vars.	Number of optima		
			Baldwinian Sharing	MBS w/ Equal Radius	MBS w/ Unequal Radii
2s2b	10000	6	2	13	16
3s3b	20000	12	4	7	11

In this work, the businessmen-customers model used in Goldberg and Wang (1997) is used to represent the optima and the customers (the regular GA population) are assigned to businessmen based on a variable radius niching technique. The idea of Baldwinian sharing is used by calculating the distance of termination points of the customers from the businessmen. The businessmen evolve by occasionally allowing some of the customers to be evaluated completely by local search.

The ideas were used on a structural member sizing problem and results are shown in Table 1. 2s2b and 3s3b refer to 2-story, 2-bay and 3-story, 3-bay frames respectively. Use of conventional sharing resulted in predominant convergence to a single optimum for both problems.

References

Gan, J., & Warwick, K. (2000). A Variable Radius Niching Technique for Speciation in Genetic Algorithms. See Whitley et al. (2000), pp. 96–103.

Goldberg, D. E., & Wang, L. (1997). Adaptive niching via coevolutionary sharing. In *Genetic Algorithms and Evolution Strategy in Engineering and Computer Science* (Chapter 2, pp. 21–38). West Sussex, England: John Wiley & Sons Ltd.

Seront, G., & Bersini, H. (2000). A new GA-Local Search Hybrid for Continuous Optimization Based on Multi Level Single Linkage Clustering. See Whitley et al. (2000), pp. 90–95.

Whitley, D., et al. (Eds.) (2000). *Proceedings of the Genetic and Evolutionary Computation Conference.* Las Vegas, NV: Morgan Kaufmann Publishers, San Francisco, CA.

Genetic Algorithm for systems with 2D genotype

Andrzej J. Pindor
Information Commons
University of Toronto
130 St. George Street
Toronto, Ont. Canada, M5S 3H1
Email address: andrzej.pindor@utoronto.ca
Phone: (416) 978-5045

Abstract

It is shown that Genetic Algorithm can be used to search for periodic patterns in Game of Life 2D cellular automaton in spite of a very strong *linkage* inherent in the problem.

1 THE PROBLEM IN GA TERMS

The Game of Life cellular automaton (Gardner 1970) has been studied extensively and a large number of interesting structures have been found (see for instance *http//home.interserve.com/~mniemiec/lifepage.htm*). These structures were found by trial and error, aided by intuition. Genetic algorithm has the potential to discover structures which may be impossible to find through human intuition.

The transformation of cells in the Game of Life from one epoch to the next is determined by their local neighbourhoods. In terms of genetic algorithm it means that there a strong *linkage* between corresponding alleles. If the corresponding region of the underlaying lattice is represented as a binary string, the above means that even small *building blocks* (Goldberg 1989) are represented as long schemata and thus are very prone to disruption during crossover. 2D form of the genotype allows one to set up a crossover scheme which minimizes the *linkage disruption.* It also allows exploitation of symmetries of the underlaying square lattice.

Finding period-1 structures (i.e. ones which are stable) turns out to be very easy. However, in case of period-2 structures the linkage is so strong that even with the above choice, most of the offspring have fitness well below the fitness of the parents. In nature, when chances of survival of offspring are low, mating results in a large number of offspring. Consequently, each pair of mating organisms was made to create a large brood of offspring and then the two best organisms from this brood, augmented with the parents, were selected for the new population. This increased the chances that the good *building blocks* were retained and combined into better structures.

In order to avoid convergence to uninteresting structures (very small ones or their independent combinations) the fitness function had to be made dependent on a number of characteristics of the structures, providing measures of various aspects which made a structure 'interesting'. Extensive numerical experiments were required to determine types of functional dependencies which would provide enough time for *building blocks* to assemble into the perfect structures and yet be fast enough to would overcome linkage disruption inherent in crossover.

These experiments were performed finding period-2 structures and the experienced gain was used in the search for period-3 structures.

The problem was programmed in MATLAB.

2 RESULTS

The program finds known period-2 and period-3 structures fitting in the 8x8 region (limitation due to available computer resources). Preliminary results indicate that there are no structures in this region which repeat themselves after two epochs, shifted orthgonally or diagonally.

References

Gardner Martin (1970), *Scientific American*, October 1970

Goldberg D.E. (1989), *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley, Reading, Massachusetts.

GA-Based Multi-Agent Reinforcement Learning for Playing Backgammon

Dehu Qi
CECS Department
University of Missouri-Columbia
Columbia, MO 65211

Ron Sun
CECS Department
University of Missouri-Columbia
Columbia, MO 65211

Abstract

The paper presents a genetic algorithm approach for evolving multi-agent reinforcement learning systems that are made up of a coalition of agents with bidding. The experiment results show the advantage of this approach over the single agent reinforcement learning approach, the pure GA approach and the reinforcement learning with bidding approach.

1 GA-Based Multi-Agent Reinforcement Learning

The general idea of our multi-agent reinforcement learning system is as follows: There are a number of individual agents. Each of them can select actions to be performed at each step, which is done by the Q module in the agent. For each agent, there is also a controller CQ, which determines at each step whether the agent should continue or relinquish control. Once an agent relinquishes its control, to select the next agent, it conducts a bidding process among agents. Based on the bids, it decides which agent should take over next from the current point on, and takes the bid as its own reward.

We use genetic algorithm to evolve this multi-agent reinforcement learning system. Two algorithms were designed in our experiment. Algorithm 1 is a traditional genetic algorithm. We train a set of bidding systems. We then select the best bidding systems by using tournament selection. The new population is composed of the best bidding systems, crossovers of the best bidding systems and mutations of the best bidding systems. After that, we train the new set of bidding system.

In algorithm 1, because the mutation and crossover are done randomly, in some cases, the performance of the newly generated population is worse than that of the old population. In algorithm 2, we restore the old population in case the performance gets worse. We also simplify the algorithm by using only one bidding system to speed up computation. So the crossover and mutation will happen within one bidding system.

2 Experiments

We use our multi-agent reinforcement learning system for playing Backgammon game. Both algorithms' performance against the single agent shows that the system has an advantage over the single agent. The highest winning percentage is 0.92. Between these 2 algorithms, algorithm 1 has a better average winning percentage when compare to algorithm 2. However, the algorithm 2 has a much better winning percentage/time ratio. Further experiment shows that the average performance of algorithm 1 is better than that of the pure GA algorithm, the GA with RL, and RL with Bidding. That means, all 3 components in our system, GA, RL and bidding, are important. They are synergistic. Missing any component leads to worse performance.

3 Conclusion

In sum, we developed a GA bidding approach for performing multi-agent reinforcement learning, to form action sequences to deal with a complex situation: the backgammon game. The experiment result shows the advantage of the system over the single reinforcement learning, the pure GA approach and the bidding system without GA. The result of the experiments suggests that our system may work well in more general complex problems.

The Relevance of Genotypic Learning in the CLGA

Terry P. Riopka and Peter Bock

George Washington University
Dept. of Computer Science, 801 22nd St. NW
Washington, DC 20052
Email: {**riopka**, pbock}@seas.gwu.edu

SUMMARY

A new recombination framework for genetic algorithms referred to as the Collective Learning Genetic Algorithm (CLGA) has been demonstrated which utilizes genotypic learning to guide recombination deterministically in a distributed network of interacting agents [Riopka and Bock, 2000]. In the CLGA, individuals of the population collaborate and exchange knowledge instead of symbols in order to modify their own chromosome strings in a process referred to as *intelligent recombination*. Thus, random crossover is essentially replaced with a consensus of information based on individual experience and observation. In addition, individuals maintain their own strings throughout evolution, preserving a modified string only if it is the same or better than the last.

Although preliminary experiments suggest that the CLGA may be an effective algorithm for searching for solutions to highly epistatic, non-separable combinatorial optimization problems, whether or not the mechanism of genotypic learning is responsible for this apparent success is less clear. It is possible that given the object-oriented framework of the CLGA, *any* recombination method substituted for intelligent recombination might result in reasonable CLGA performance. Before devoting significant effort into testing the CLGA, it is important to establish the degree to which the central concept of the CLGA, specifically genotypic learning, is relevant to its successful operation.

The effect of genotypic learning is shown by comparing the performance of the CLGA with the following recombination mechanisms substituted for intelligent recombination (optimized in the experiments over all levels of epistasis): parameterized uniform crossover (puc), two-point crossover (tpc), mutation-only (m) and random information exchange (r). A non-optimal CLGA was configured heuristically, similar to one used in [Riopka and Bock, 2000] with a population of 572 but without mutation. 50 NK-Landscape problems of size N=20 with random linkage of K=2 and K=10 were tested.

Note that intelligent recombination consistently outperforms random information exchange across all levels of epistasis supporting the hypothesis that genotypic learning is a relevant mechanism for recombination. Intelligent recombination seems to behave like parameterized uniform crossover at low levels of epistasis and like mutation-only at high levels, suggesting an ability to adapt to the level of problem epistasis.

It is not unreasonable to infer that as epistasis increases, the amount of *useful* information available for intelligent recombination decreases, due to decreasing correlation between fitnesses of similar solutions in Hamming space. Consequently, it becomes more and more difficult for the CLGA to learn consistent relationships between bits due to greater variance in solution fitness, causing intelligent recombination to act more intelligently at low levels of epistasis but *like* mutation at high levels of epistasis.

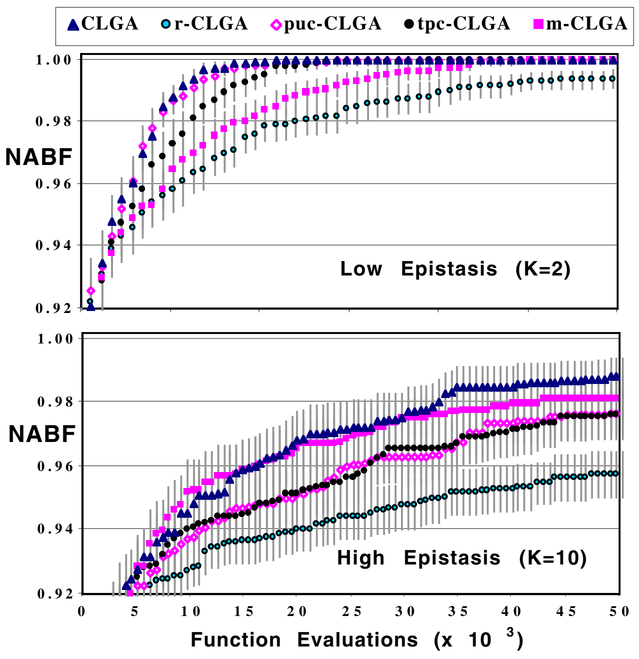

Fig. 1: Normalized average best-so-far fitness (NABF) plots with 95% confidence intervals are shown.

References

Riopka, T.P. and Bock, P. (2000). Intelligent Recombination Using Individual Learning in a CLGA. In *Proc. of the Genetic and Evol. Comp. Conf.,* pp. 104-111.

The Patchy GA and Domination Problems

Robert S. Roos
Dept. of Computer Science
Allegheny College
Meadville, PA 16335

1 INTRODUCTION

Let $G = (V, E)$ be an undirected graph. A *dominating set* is any subset $S \subseteq V$ such that, for all $v \in V - S$, v is adjacent to at least one element of S. The problem of determining, for arbitrary G and integer K, whether or not G has a dominating set of size less than or equal to K, is NP-complete (Garey & Johnson 1979); we denote this decision problem and the corresponding optimization problem by Dom. This paper represents a first attempt to apply genetic algorithms to the dominating set problem. We identify two classes of graphs—random geometric graphs and lattice-partitioned degree-4 graphs—for which GAs significantly outperform the greedy heuristic.

Random geometric graphs have been well-studied in the computer science and mathematics literature. A lattice-partitioned random degree-4 graph is constructed as follows: form a $p \times q$ lattice of cells, each containing three vertices. Within each pair of horizontally and vertically adjacent cells, construct a random matching between the three vertices in each cell (wrap around for cells at the boundaries of the lattice).

2 THE PATCHY GA

The Patchy GA is a steady-state, elitist algorithm. It generates an initial population of dominating sets using a greedy algorithm with random tie-breaking. In each succeeding generation, two chromosomes are chosen, a covering set of subgraphs ("patches") is constructed, and crossover is performed by intersecting the dominating set of one or the other parent with each of the patches and then extending the resulting vertex set to a dominating set for the graph. The worst element of the population is replaced. (Mutation was not investigated in this study.)

To construct patches for geometric graphs, we select a line passing through the unit square and partition the vertices into those to the left and those to the right of the line. For lattice-partitioned graphs we choose as the dividing line a zig-zag path running between the "cells" that were used to partition the vertices into three-element subsets.

3 EXPERIMENTAL RESULTS

We ran Greedy fifty times on each of ten graphs: five random geometric graphs of size 1000 and five lattice-partitioned degree-4 graphs of size 7500. We ran Patchy GA five times on each of the ten graphs, halting it after 50 generations. The best-of-five results for Patchy were consistently better than the best-of-fifty results for Greedy on every one of the graphs tested; the difference in results as a percentage of the smaller of the two results ranged from 0.1% to 3.2%. Moreover, in eight of the ten tests, the worst result from the Patchy GA was better than the best Greedy result, sometimes by as much as 2.2% of the smaller value. The lattice-partitioned graphs were the ones in which the improvement was most noticeable. Differences in dominating set size tended to be close to 1% for geometric graphs, but between 2% and 3% for lattice-partitioned graphs.

References

Garey, M. R., and Johnson, D. S. (1979) *Computers and Intractability.* New York: W. H Freeman and Company.

Hochbaum, D.S. (1985) Easy solutions for the k-center problem or the dominating set problem on random graphs. *Annals of Discrete Math.* **25**, 189–210.

Johnson, D. S. (1974) Approximation algorithms for combinatorial problems. *J. Computer and System Sciences* **9**, 256–278.

A Factorized Distribution Algorithm for problems with integer representation

Roberto Santana, Alberto Ochoa-Rodriguez, Marta R. Soto
Center of Mathematics and Theoretical Physics.
ICIMAF. Calle 15, e/ C y D, Vedado CP 10400. C-Habana. Cuba
{rsantana,ochoa,mrosa}@cidet.icmf.inf.cu

Abstract

In this poster we present a Factorized Distribution Algorithm (FDA) that considers up to second order statistics, and permits to carry out the optimization of integer problems with a high cardinality of the variables.

FDAs are considered as a tractable subclass of Estimation Distribution Algorithms. They apply the selection operator in each generation but do not use the crossover and mutation operators, a factorized probabilistic model of the selected points is constructed instead. There exists a particular subclass of FDAs that considers only up to pairwise dependencies among variables. Trees are the class of dependency graphs used by the FDA we introduce.

The algorithm we present (Fig. 1) reduces the storage requirements of previous FDAs that employ pairwise dependencies by avoiding to store the bivariate probabilities. Our proposal combines classical methods for structural learning of dependencies with a procedure that approximates the bivariate marginals by sampling the data using auxiliary tables.

Tree based FDA for Integers (Int-Tree)

STEP 0: Set $t \Leftarrow 0$. Generate $N \gg 0$ points randomly.

STEP 1: Select $k \leq N$ points according to a selection method.

STEP 2: Create the tree T using an algorithm for finding the mutual information straight from the population, and the MWST method [1].

STEP 3: Create, using T, the auxiliary tables that represent the information stored in the selected set.

STEP 4: Generate N new points using T and the auxiliary tables. Set $t \Leftarrow t + 1$

STEP 5: If the termination criteria are not met, go to STEP 1

Figure 1: General scheme of the FDAs that consider up to pairwise dependencies

The structure of the tree is calculated like in [1]. To generate new vectors a different algorithm is used. Let $X = (X_1, ..., X_n)$ where X_i is a discrete variable with r_i (not necessarily consecutive) possible assignments : $(v_{i,1}; ...; v_{i,r_i})$ and we will denote one of these possible instantiations as x_i, i.e. $x_i = v_{i,j}$ with $j \in \{1, 2, ..., r_i\}$. Two tables are created that allow to identify which are the feasible values $v_j \subset (v_{j,1}; ...; vj_{j,r_j})$ a variable X_j can take given that its parent X_i has been assigned a value $v_i \subset (v_{i,1}; ...; v_{i,r_i})$.

With this work we have tried to partially remedy an unsatisfactory state of affairs in the use of FDAs for the optimization of integer problems. The algorithm shows better results than the Univariate Marginal Distribution Algorithm for the optimization of the functions considered. This algorithm has application to the optimization of constrained problems with integer representation.

Acknowledgments

This work was funded by the Cuban Ministry of Science, Technology and Environment, under the project Low Cost Evolutionary Algorithms.

References

[1] Shumeet Baluja and Scott Davies. Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space. In *Proceedings of the 14th International Conference on Machine Learning*, pages 30–38. Morgan Kaufmann, 1997.

Modeling Tournament Selection With Replacement Using Apparent Added Noise

Kumara Sastry and David E. Goldberg
Illinois Genetic Algorithms Laboratory (IlliGAL)
Department of General Engineering
University of Illinois at Urbana-Champaign
104 S. Mathews Ave, Urbana, IL 61801
{ksastry,deg}@uiuc.edu

This paper analyzes the effects of tournament selection (Goldberg, Korb, & Deb, 1989) with replacement (TWR) on the convergence time and population sizing for selectorecombinative genetic algorithms. In contrast to tournament selection without replacement (TWOR), TWR has not received considerable analytical attention in genetic algorithms literature. TWR is usually considered to be equivalent to TWOR. However, we empirically show that TWR requires more function evaluations for attaining the same accuracy as TWOR.

We observe that though the run duration is same for both TWR and TWOR, the population size required for successful convergence is not. This is because TWR is a noisy scheme when compared to TWOR. In TWR the best individual in the population can have all n copies or none at all, where as in TWOR the best individual has *exactly* s copies. We model this discrepancy as an apparent noise in the population-sizing model (Miller, 1997), similar to that proposed by Goldberg et al. (1992) for roulette-wheel selection.

To quantify the noise term, recognize that the process of selecting an individual is a Bernoulli trial and this process is repeated n/s times. Therefore the process of selecting an individual i in n/s trials is binomially distributed with probability $p_i = s/n$. The variance of the number of tournaments that the individual i participates in is $(n/s)p_i\,(1-p_i)$. Summing over all individuals we get a variance $(s/n)\sum_{i=1}^{n/s}(n/s)p_i\,(1-p_i) = (n-s)/n \approx 1$. Since this process is repeated s times, the variance of tournaments that an individual i participates in is $s(n-s)/n \approx s$. This variance is in the units of squared individuals and has to be converted it into units of squared fitness. We recognize that an individual must change by an amount equal to some proportion of the population fitness variance to increase or decrease its numbers by s. Thus, the appropriate variance due to TWR is $c_o s\sigma_F^2$, where c_o is a constant and is empirically determined to be 0.25. Using this noise due to selection as an external noise term, the approximate population-sizing model for TWR can be written as

$$n = -\frac{-2^{k-1}\log(\psi)}{d}\sqrt{\pi\,(1+c_o s)\,\sigma_F^2}. \qquad (1)$$

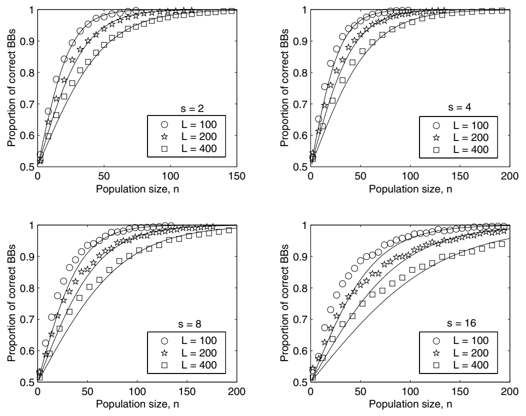

Figure 1: Proportion of correct BBs, given by equation (1) compared to experimental results

The success rate $1-\psi$ computed for different tournament size values and different problems sizes as a function of population size are verified with computational results in figure 1. The experimental results are for OneMax and are averaged over 100 runs. The results demonstrate significant agreement with the analytical relation.

References

Goldberg, D., Deb, K., & Clark, J. (1992). Genetic algorithms, noise, and the sizing of populations. *Complex Systems*, *6*, 333–362.

Goldberg, D. E., Korb, B., & Deb, K. (1989). Messy genetic algorithms: Motivation, analysis, and first results. *Complex Systems*, *3*(5), 493–530.

Miller, B. L. (1997, May). *Noise, sampling, and efficient genetic algorithms.* Doctoral dissertation, University of Illinois at Urbana-Champaign, Urbana, IL.

Toroidal Search Space Conversion for Robust Real-coded Genetic Algorithms

Hiroshi Someya
Tokyo Institute of Technology,
4259, Nagatsuta, Midori-ku,
Yokohama, 226-8502, Japan.
Phone : +81-45-924-5211
hiroshi@es.dis.titech.ac.jp

Masayuki Yamamura
Tokyo Institute of Technology,
4259, Nagatsuta, Midori-ku,
Yokohama, 226-8502, Japan.
Phone : +81-45-924-5212
my@dis.titech.ac.jp

1 INTRODUCTION

In real-coded GAs, most crossover operators like to search the center of search space much more than the other [1]. When a crossover operator has such bias, it will not work on functions whose optima are near the boundary of the search space. To reduce the sampling bias, several methods have been proposed. BEM [1] extends the search space to move the relative position of the optimum toward the center of the search space. BEM allows individuals to be located outside the search space. The functional value of individual i with real vector $\vec{X}^{(i)} = (x_1^{(i)}, \ldots, x_n^{(i)})$ is calculated as the followings:

$$\begin{aligned} f(\vec{X}^{(i)}) &= f(\vec{Y}^{(i)}), \\ \vec{Y}^{(i)} &= (y_1^{(i)}, \ldots, y_n^{(i)}), \\ y_j^{(i)} &= \begin{cases} 2\ \min_j - x_j^{(i)} & : \text{ if } x_j < \min_j \\ 2\ \max_j - x_j^{(i)} & : \text{ if } x_j > \max_j \\ x_j^{(i)} & : \text{ otherwise,} \end{cases} \end{aligned} \tag{1}$$

where, $\min_j$ and $\max_j$ are the lower and upper limits of parameter range respectively. BEM has one parameter, r_e $(0 < r_e < 1)$, that controls how much search space is extended. Although existing methods succeed in reducing the sampling bias, from the viewpoint of robustness, no sampling bias is desirable.

2 TSC

We propose a new method, *Toroidal Search Space Conversion* (TSC), to remove the sampling bias. TSC converts search space with boundary into toroidal one. This conversion is performed as follows:

step1 Extend the search space like BEM with $r_e = 1.0$,

step2 Connect each e-max$_j$ of the extended search space to corresponding e-min$_j$.

When $r_e = 1.0$, the width of the search space doubles. The landscape of the outside of the search space is defined as if mirrors stand on each boundary. An example of the converted search space is shown in Fig.1. A generated individual i is modified as follows:

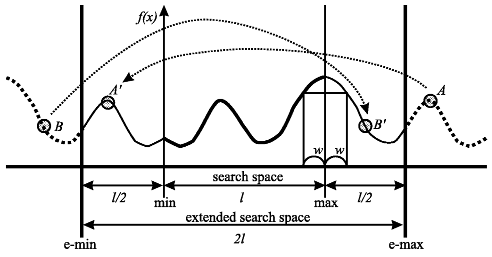

Figure 1: An example of TSC

$$\begin{aligned} \vec{X}^{(i)} &= \vec{Z}^{(i)}, \\ \vec{Z}^{(i)} &= (z_1^{(i)}, \ldots, z_n^{(i)}), \\ z_j^{(i)} &= \begin{cases} x_j^{(i)} + 2l & : \text{ if } x_j < \text{e-min}_j \\ x_j^{(i)} - 2l & : \text{ if } x_j > \text{e-max}_j \\ x_j^{(i)} & : \text{ otherwise.} \end{cases} \end{aligned} \tag{2}$$

For example, A and B, located outside of the extended search space, in Fig.1 are modified to A' and B' respectively. When the distance between A' and B' is farther than that between A' and B (=A and B'), crossover operation using A' and B is performed instead of A' and B'. When the distance between parents is far, crossover does not generate children in the center of the search space, but does them near the boundary close to the parents. In TSC, initial individuals are placed in the extended search space uniformly. Accordingly, by this proposed method, any position on this search space become equivalent to any others. The converted search space has no sampling bias.

References

[1] S. Tsutsui. Multi-parent Recombination in Genetic Algorithms with Search Space Boundary Extension by Mirroring. In *Proc. of PPSN V*, pages 428–437, 1998.

Using Genetic Algorithms to Evolve Cooperative Teams

Terence Soule
Department of Computer Science
University of Idaho
Moscow, ID 83844-1010
email: tsoule@cs.uidaho.edu

Abstract

In this paper we show that genetic algorithms (GA) can be used to evolve teams whose members cooperate and specialize. Our results indicate that with an appropriate cooperation mechanism teams evolved with a GA can perform better than evolved individuals.

1 Introduction

Genetic programming (GP) has had considerable success evolving teams of programs that cooperate to outperform simgle program, see for eample (Hayes et al. 95, Luke and Spector, 96). More recently GP has been used to evolve teams that use explicit cooperation mechanisms to solve problems that do not *require* teams, such as the even-parity problem and function regression problems (Soule, 99, Soule, 00). These problems are well suited to GAs. Thus, it should be possible to use a similar cooperative, team based approach with GAs. This paper uses an explicit cooperation mechanism to solve several function regression problems. This is a first result showing GAs can evolve cooperative teams with specialized members.

2 Experimental Design

We use the symbolic regression problem on the functions $f_1(x) = sin(x)$ and $f_2(x) = sin(x) + 0.5cos(2*x)$. 40 evenly destributed points in the range$(-\pi, pi)$ are used as the test set. The chromosome is 144 bits long. It is divided into 6 sections of 24 bits each. The first 20 bits of each section represents a coefficient. These 20 bits are Grey coded and scaled to give a value in the range -2 to 2. The next 4 bits of each section represent an exponent. These 4 bits are also Grey coded and are scaled to an integer value between 7 and -8. Thus, the solution generated is polynomial with six terms with coefficients between 2 and -2 and exponents between 7 and -8. Teams consist of 3 or 5 chromosomes, the team members. One point crossover between team members is used.

In the first cooperation mechanism a majority vote is applied to each bit in the chromosome, to produce a single chromosome for evaluation (bitwise voting). In the second cooperation mechanism used median voting. Each team member's chromosome is translated into a polynomial and evaluated at each of the 40 test points. At each test point the median of the N member values is used as the 'actual' solution for purposes of determining fitness (median voting). The GA is generational; population size is 300; 50 trials and 50 generations are used; crossover rate is 0.8; mutation rate is 1/chromosome length (0.006944), tournament selection (7), and elitism (5).

3 Results

Median voting performed significantly better than either individuals or bitwise voting on both test functions. Bitwise voting performed no better than normal individuals. (Student's two-tailed t-test $p < 0.05$ was used to test siginifcance.)

Medain voting produced specialized, cooperating team members, but bitwise voting did not. These results show that a GA can be used to evolve cooperative teams similar to those that have been evolved with a GP. However, the choice of cooperation mechanism is clearly important.

4 Bibliography

(Haynes 95) Thomas Haynes, Sandip Sen, Dale Schenfield and Roger Wainwright, "Evolving a Team," *Working Notes of the AAAI-95 Symposium on GP*, pages 23-30 AAAI Press, 1995.

(Luke and Spector 96) Sean Luke and Lee Spector, "Evolveing Teamwork and Coordination woth Genetic Programming," *Genetic Programming 1996*, pages 150-156, Cambridge, MA: MIT Press, 1996.

(Soule 99) Terence Soule, "Voting Teams: A cooperative approach to non-typical problems," *GECCO99*, pages 916-922, Morgan Kaufmann, 1999.

(Soule 00) Terence Soule, "Heterogeneity and Specialization in Evolving Teams," *GECCO2000*, pages 778-785, Morgan Kaufmann, 2000.

Schema Evolution Equation Using Quasi-schema Representation

Hideaki Suzuki
ATR International, Inf. Sci. Division
2-2-2 Hikaridai, Seika-cho,
Soraku-gun, Kyoto 619-0288 Japan

Hidefumi Sawai
Communications Research Laboratory
2-2-2 Hikaridai, Seika-cho,
Soraku-gun, Kyoto 619-0289 Japan

Although the rigorous models of genetic algorithms (GAs) [1, 2] are a powerful method to predict the ideal (infinite-population) behavior of a simple GA, they are no use for the analysis of problems with a large string length (or schema order). The state of a population is represented by a $2^{o(H)}$-dimensional frequency vector ($o(H)$ is the order of a target schema H); hence the computational cost for a model calculation is proportional to $2^{3o(H)}$ or $2^{2o(H)} \times T$ (T is the generation number). To reduce the dimension of this vector, here we extend a vector representation method established by Suzuki [3] and formulate a set of evolution equations for *quasi-schemata.*

To analyze the evolution of the target schema H, we separate the whole bit field (binary loci) into E *subfields* $\{F_1, F_2, \cdots, F_e, \cdots, F_E\}$. Then, a quasi-schema $G_{\{i_e\}} \equiv G_{\{i_1,i_2,\cdots,i_e,\cdots,i_E\}}$ is defined as a set of schemata which have the same defining loci as H, i_1 anti-bits in F_1, i_2 anti-bits in F_2, and so on (where an 'anti-bit' is a defining bit different from H's). For example, for $H = $ [000*0**0], if we define the subfields as $F_1 = $ [####....] and $F_2 = $ [....####] ('#'/'.' stand for the inclusion/exclusion of the locus in/from the subfield), quasi-schema $G_{\{2,1\}}$ is

$$\begin{aligned} G_{\{2,1\}} &= [\texttt{110*1**0}] \vee [\texttt{101*1**0}] \vee [\texttt{011*1**0}] \vee \\ &\quad [\texttt{110*0**1}] \vee [\texttt{101*0**1}] \vee [\texttt{011*0**1}]. \end{aligned}$$

With $o_e = o_e(H)$ defined as the number of H's defining loci in the e-th subfield (F_e), the number of different quasi-schemata is $\prod_{e=1}^{E}(o_e + 1)$, and we describe the state of a population with a $(\prod_{e=1}^{E}(o_e+1))$-dimensional frequency vector $\{x_{\{i_e\}}\}$ whose $\{i_e\}$th element is the expected frequency of $G_{\{i_e\}}$. The evolutionary dynamics of an infinite population under GAs are then represented by a set of recursive formulas for selection, mutation, and crossover as

$$x_{\{i_e\}} \overset{\text{sel.}}{\longrightarrow} \frac{f(G_{\{i_e\}})}{\overline{f}} x_{\{i_e\}}, \tag{1a}$$

$$x_{\{i_e\}} \overset{\text{mut.}}{\longrightarrow} \sum_{\{j_e\}} x_{\{j_e\}} M_{\{j_e\}\{i_e\}}, \tag{1b}$$

$$\begin{aligned} x_{\{i_e\}} \overset{\text{cross.}}{\longrightarrow}\ & (1-p_c)x_{\{i_e\}} + p_c \sum_{\{j_e\}} \sum_{\{k_e\}} x_{\{j_e\}} x_{\{k_e\}} \\ & \times C_{\{j_e\}\{k_e\}\{i_e\}}, \end{aligned} \tag{1c}$$

where $f(G_{\{i_e\}})$ is the average fitness of strings belonging to quasi-schema $G_{\{i_e\}}$, $\overline{f}$ is the average population fitness, and p_c is the probability of the crossing-over occurring per string pair. The summation for $\{j_e\}$ and $\{k_e\}$ is taken for all of the $\prod_e(o_e+1)$ quasi-schemata. Rigorous formulas for $M_{\{j_e\}\{i_e\}}$ and $C_{\{j_e\}\{k_e\}\{i_e\}}$ are given under the mutation probability p_m and one-point crossover, respectively, each of which are evaluated with the computational cost proportional to $E \times o(H)$.

From some numerical experiments, it has been proved that the established formulas provide us with a method to predict the schema evolution with sufficient accuracy (for some problems) and a polynomial computational cost ($\propto (\frac{o(H)}{E}+1)^{3E} \cdot E \cdot o(H)$) for fixed E. We infer that the method can be a powerful tool to analyze GAs under a function wherein the evolution of a large-order schema plays a vitally important role.

References

[1] Vose, M.D., Liepins, G.: Punctuated equilibria in genetic search. Complex Systems **5** (1991) 31–44

[2] Whitley, D.: An executable model of a simple genetic algorithm. In: Whitley, D. (ed.): Foundations of Genetic Algorithms 2 (FOGA-2), Morgan Kaufmann Publishers, San Francisco, CA (1993) 45–62

[3] Suzuki, H.: The optimum recombination rate that realizes the fastest evolution of a novel functional combination of many genes. Theoretical Population Biology **51** (1997) 185–200

Simplex Crossover and Linkage Learning in Real-Coded GAs

Shigeyoshi Tsutsui*, David E. Goldberg, and Kumara Sastry****

* Department of Management and Information Science, Hannan University, 5-4-33 Amamihigashi, Matsubara, Osaka 580-5802 Japan, phone: +81-723-32-1224, e-mail: tsutsui@hannan-u.ac.jp
**Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, 117 Transportation Building, 104 S. Mathews, Urbana, IL 61801, U.S.A, phone: +1-217-333-0897, e-mail: {deg, kumara}@illigal.ge.uiuc.edu

Abstract: In this paper, we propose a method of linkage identification in real-coded GAs with *simplex crossover* (SPX) and evaluate it using test functions.

1. Introduction

Previous studies [Tsutsui 99], [Higuchi 00] have proposed *simplex crossover* (SPX) for real-coded GAs. SPX works well on various test functions. However, SPX fails on functions that consist of multiple tightly linked sub-functions. On those functions, SPX should be applied on each tightly linked parameters group, i.e., each simplex should be formed in each subspace in which parameters are tightly linked. Thus, we need a method of identifying those tightly linked parameter groups.

In this paper, we propose a method of linkage identification for real-coded GAs with SPX and evaluate it using test functions.

2. SPX [Tsutsui 99], [Higuchi 00].

The SPX operator uses n+1 parental vectors X_i, $i = 0, 1, ..., n$ for recombination. These (n+1) vectors form a *simplex* in R^n. Then this simplex is expanded in each direction (X_i-O) to some extent, where O is the center of mass of (n+1) parental vectors. Offspring are then generated by *uniformly* picking vector values from this expanded simplex.

3. Linkage Identification

In this study, we consider evaluation functions that can be written as

$$F(X)=F_{\text{tight},1}(X_{\text{tight},1}) +, ..., + F_{\text{tight},S}(X_{\text{tight},S})$$

Here, each sub-function $F_{\text{tight},s}(X_{\text{tight},s})$ (s=1, ..., S) has tight linkage among parameters. For these kinds of functions, we intuitively notice that we it is better to apply SPX in each subspace $X_{\text{tight},s}$ (s=1,...,S) separately. Then the problem is how to identifiy each tight linkage group.

The linkage information among parameters can be obtained by observing the distribution of individuals in a population. If F(X) has a linkage among parameters on loci x_i, x_j, and x_k, then there should be some degree of correlation between x_i-x_j, x_j-x_k, and x_k-x_i, respectively. Thus if we examine the correlation coefficient matrix R =[ρ_{ij}] of parameter values of individuals in a population, linkage among parameters might be detected. However, this normal correlation coefficient can examine only linear correlations among parameters. We must use some non-linear estimation technique such as non-linear regression or higher moment methods. This paper propose a *piecewise interval correlation by iteration* (PICI) algorithm, a more simple and straightforward extension of the linear correlation. It calculates correlation coefficients of piecewise intervals (Fig. 1).

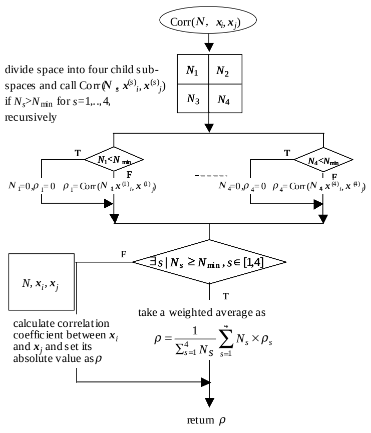
Corr(N, x_i, x_j)
divide space into four child sub-spaces and call Corr(N_s, $x^{(s)}_i$, $x^{(s)}_j$) if N_s>$N_{\min}$ for s=1,..,4, recursively
N_1
N_2
N_3
N_4
T
N_1<$N_{\min}$
F
N_1=0,ρ_1=0
ρ_1=Corr(N_1, $x^{(1)}_i$, $x^{(1)}_j$)
T
N_4<$N_{\min}$
F
N_4=0,ρ_4=0
ρ_4=Corr(N_4, $x^{(4)}_i$, $x^{(4)}_j$)
F
$\exists s \mid N_s \geq N_{\min}, s \in [1,4]$
T
N, x_i, x_j
calculate correlation coefficient between x_i and x_j and set its absolute value as ρ
take a weighted average as
$\rho = \frac{1}{\sum_{s=1}^{4} N_S} \sum_{s=1}^{4} N_s \times \rho_s$
return ρ

Fig. 1 PICI algorithm

4. Experiments

To evaluate the linkage identification method proposed in Section 3, we run a real-coded GA. Here we show only a typical results (see Table 1) on the following function:

$$F_{\text{R2-}n}(X)=F_{\text{R2}}(x_1,x_2)+F_{\text{R2}}(x_3,x_4)+ ... + F_{\text{R2}}(x_{n-1},x_n)$$

where F_{R2} (x_1,x_2) is the Rosenbrock function. We can see the the effect of the proposed method.

Table 1 Results on F2-n(X)

No of parameters	without linkage learning		with linkage learning	
n	#OPT	MNE	#OPT	MNE
4	20	67,721.8	20	54,830.8
8	20	191,046.8	20	135,136.1
12	18	697,515.4	20	225,800.8
16	0	--	20	295,944.5
20	0	--	20	354,135.5
24	0	--	20	402,790.9
28	0	--	20	448,570.9
32	0	--	20	497,641.3
36	0	--	20	540,894.5
40	0	--	20	584,640.5

5. Conclusions

In this paper, we have proposed a method for linkage identification in real-coded GAs with SPX. Without linkage identification, it was difficult for the algorithm with SPX to find the optimal solution on these kinds of test functions. With the proposed linkage identification method, the algorithm found optimum solutions fairly well on the test functions. This work was partialy supported by AFOSR grant No. F49620-00-0163 and NSF grant DMI-9908252.

References

[Higuchi 00] Higuchi, T., Tsutsui, S., and Yamamura, M.:Theoretical analysis of simplex crossover for real-coded Genetic Algorithms, *Proc. of the PPSN VI*, pp. 365-374(2000).

[Tsutsui 99] Tsutsui, S., Yamamura, M., and Higuchi, T: Multi-parent recombination with simplex crossover in real-coded genetic algorithms, *Proc. of the GECCO-99*, pp.657-664(1999).

Fitness Functions for Generated Binary Images

Róbert Ványi
e-mail: vanyi@inf.u-szeged.hu
Department of Informatics,
University of Szeged
Árpád tér 2, H-6720 Szeged, Hungary

now visiting:
Department of Programming Languages
Friedrich-Alexander University
Martensstr. 3, D-91058 Erlangen, Germany

Abstract

There are applications where a binary image is given and a shape has to be reconstructed with some kind of evolutionary algorithms. A solution for this problem usually highly depends on the fitness function. The main goal of this research is to find good fitness functions and fast calculation methods for them.

1 INTRODUCTION

Some experiments have already been done to describe an image using Evolutionary Algorithms, but they used fitness functions not applicable for binary images [1] [2]. This project however uses fitness functions based on the pixel by pixel comparison of two images.

2 FITNESS FUNCTIONS

Some simple functions can easily be given such as the following fitnesses:

simple and modified quadratic error A usual error function is the so-called quadratic error. This function can be slightly modified by distinguishing the different errors.

multi-level fitness When using multi-level fitness, the distance is coded into the destination image before the process, so without the loss of speed, the distance can also be calculated into the fitness.

distance based fitness Using the idea from the previous fitness function a more sophisticated version can also be given, where the fitness is a function of the distance. The tests have shown, that this fitness results in a faster convergence than quadratic error.

2.1 CALCULATING THE FITNESS

One way to calculate fitness is to generate the image and then compare it with the original one. Another way is to do the comparison during drawing. According to the tests, the latter one, called *on-line*, is faster, and it was proven that it can be applied for simple, and sometimes also for complex fitness functions.

2.2 EXTENDED FITNESS FUNCTIONS

As it was mentioned before in some cases the simple fitness functions have to be extended. Some possible extensions are the following:

overwriting pixels This can be handled by marking the already drawn pixels. This can cause a slowdown, but with a special algorithm, this loss of speed can be efficiently reduced.

handling thick lines Since it can be considered as a special case of the previous one, similar ideas can be used to handle thick lines.

3 CONCLUSION

With the help of the theoretical foundations efficient algorithms can be defined for fitness functions, which take more information into consideration. And better fitness functions usually mean faster convergence.

References

[1] C. Jacob. *Principia Evolvica – Simulierte Evolution mit Mathematica.* Dpunkt Verlag, 1997.

[2] J. R. Koza. Discovery of rewrite rules in Lindenmayer systems and state transition rules in cellular automata via genetic programming. In *Symposium on Pattern Formation (SPF-93), Claremont, California, USA*, 1993.

Nested Genetic Algorithms with Problem Division

Dana Vrajitoru
IUSB, Math & Computer Science Department
South Bend, IN 46634-7111, USA
dvrajito@iusb.edu

Abstract

In this paper we propose a new parallel approach to genetic algorithms that has shown interesting performance concerning both the speedup and the quality of the solutions.

1 INTRODUCTION

Most of the parallel GAs divide the genetic population into several independent nests or niches, on which the selection and crossover operations act locally (Cantú-Paz, 1998). The global performance is insured by periodic migration of some individuals between subpopulations. In this paper we introduce a new parallel model for GAs for which the problem to be solved is divided among processes.

2 MODEL DESCRIPTION

Let L be the size of the individual, and np the number of processes. In our model, the process number i is evolving the genes corresponding to the places from $i * L/np$ to $(i+1) * L/np - 1$, where both the process numbers and the gene positions start from 0.

To evaluate the partial individuals generated by each process, we compose complete individuals by periodically exchanging information between processes. We estimate the fitness of each partial individual by averaging the fitness of several complete individuals containing it.

To test the new model we have used the set of standard test functions (minimization problems), the Hamiltonian circuits (HC) (Vrajitoru, 1999), and several deceptive functions. The experiments are based on 40 trials in each case with a population size of 50 and a generation number of 500. We have used a combination of the 1-point, 2-point, uniform and dissociated crossover operators.

Table 1: Average results and speedup

Average Fitness	Standard	Deception	HC
sequential	2.87	2673.73	0.946
parallel	1.69	2812.20	0.948
speedup	27.77	29.10	44.68
nr. of processes	2 - 5	4	4

Table 1 shows the average results of the sequential algorithm on each of these problems, as well as those of the best configuration of our parallel model, for which the processes exchange 2 individuals every 50 generations. The minimum for the set of standard functions is equal to 0. The maximal fitness is equal to 3000 for the deception problems, and to 1 for the HC problems. The third line presents the average speedup for each problem as a percentage of the sequential execution time. The last line contains the number of processes used in each case.

From this table we can conclude that our parallel implementation of GAs is interesting both from the point of view of the quality of the solutions, and of the speedup.

References

[Cantú-Paz, 1998] Cantú-Paz, E. (1998). A survey of parallel genetic algorithms. *Calculateurs Paralleles, Reseaux et Systems Repartis*, 10(2):141–171. Paris: Hermes.

[Vrajitoru, 1999] Vrajitoru, D. (1999). Genetic programming operators applied to genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 686–693, Orlando (FL). Morgan Kaufmann Publishers.