ICGA 93

The ECOlogical Framework II: Improving GA Performance At Virtually Zero Cost

Yuval Davidor
Department of Computer Science,
The Weizmann Institute,
Rehovot 76100, Israel.

Takeshi Yamada and Ryohei Nakano
NTT Communication Science Laboratories,
2 Hikaridai, Seika-cho, Soraku-gun,
Kyoto, 619-02, Japan.

Abstract

The basic concepts of the ECO framework for implicit speciation were introduced in an earlier note. In this paper the ECO framework is further investigated in the NP-complete domain of Job Shop Scheduling (JSS). The performance of a GA running under the ECO framework shows significant improvement compared to the same GA running without the ECO framework on several JSS benchmark problems (10x10, 20x5, 15x20, 10x15, 10x20, and 15x15). Since the ECO framework virtually does not increase the computation overhead, it is suggested that until full control over convergence is achieved for genetic algorithms (and evolutionary computation in general), running a GA under the ECO framework is a 'money saver' which can improve performance significantly.

1 THE ECO PARADIGM REVISIT

The most fundamental difference between standard GA models and the ECO framework is the range of genetic interaction among population members. By and large, in standard GAs population members are processed globally and are indistinguishable during the mating and replacement phases. This results in a centralized genetic dynamics, and the implications of global interactions in relation to convergence are well understood since the early GA simulations [Cavicchio, 1970, De Jong, 1975, Goldberg and Richardson, 1987, Mauldin, 1984]. This lack of control is partly the reason for the many attempts to introduce into GAs better control over convergence through specialized operators and parallelism. The original ECO model was presented in an early note together with first results which suggested that local genetic interactions introduce improved robustness into the GA simulation [Davidor, 1991]. In this section the ECO principles for implicit speciation are represented and contrasted against centralized models.

1.1 THE GENETIC OPERATORS TURNED LOCAL

The ECO model requires local genetic interactions. Therefore, reproduction, recombination, and selection have to be changed from global to local mechanisms. This section discusses these necessary changes to turn the genetic operators local.

In the ECO framework population members are held on a 2-dimensional grid and result in an eight immediate grid/population member neighborhood. Genetic interactions are determined locally according to grid position and immediate local population. Each grid node contains one and only one population member. For isomorphic considerations, the opposite edges of the grid are connected together and form the surface of a torus.

1.1.1 Reproduction

The ECO reproduction is a type of steady state reproduction [Whitley, 1988, Syswerda, 1989, Syswerda, 1991]. At each reproduction cycle, a grid element i, j is selected at random. The population member which occupies this i, j'th node is defined as the first parent, and together with its immediate adjacent (in grid terms) 8 population members form the local and temporarily active sub-population [Davidor, 1991, Davidor, in press].

The average fitness of this 9 member sub-population is calculated (first parent + 8 neighbours). The fitness of the first parent in the i, j'th node relative to the average fitness of the 9-member sub-population determines how many offspring are going to be produced in the current reproduction cycle. For example, if the relative fitness of a given first parent is 1.4, then this first parent will reproduce at least one offspring, and another one with probability 0.4, while a relative fitness of 0.5 means a probability 0.5 for one offspring and a probability 0.5 for no offspring at all.

1.1.2 Mating

After determining how many offspring the first parent will help produce, a second parent is selected with replacement probabilistically relative to fitness from the sub-population, excluding the first parent, for each offspring produced.

Please note that the type of crossover mechanism, whether it is a two point, uniform, or other problem specific crossover, is not mentioned as well as the representation format etc. as it is not relevant to the implementation of the ECO environment. By thus defining the choice of parents one is free to use whatever crossover, mutation, and other genetic mechanisms one wishes.

1.1.3 Replacement

After producing the offspring and calculating their fitness, each offspring is introduced into the grid by selecting at random a grid node from the 9 grid member environment. The population member in the selected node and the offspring are in conflict due to limited habitat resources (as only one population member is allowed at each grid node). The conflict is resolved by retaining the best out of the two solutions (such as in ranking or elitism).

2 JOB SHOP SCHEDULING

The $n \times m$ (minimum-makespan) job-shop scheduling problems are defined as follows. n jobs have to be processed on m machines. The processing of a job on a machine is called an operation. The processing order of operations for each job and their processing times are given. The objective is to determine the operating sequences of the machines so as to minimize total processing time called the makespan.

Job-shop problems are known to be NP-hard whose solution space topology is usually very rugged and multi-modal. We have selected a good GA model for solving job-shop problems, the GA/GT model [Yamada and Nakano, 1992], and optimized its parameters so as to achieve the best performance [Nakano *et al*]. After obtaining optimal performance this algorithm was run under the ECO framework to investigate whether the ECO's improved control over the convergence will improve the overall performance. In the following section we briefly describe the original GA that was used to solve JSS problems.

2.1 GA/GT Algorithm

GT crossover is a genetic recombination operator specialized for the job-shop problem. It can be viewed as a simple scheduling algorithm which produces a new schedule by using two parent schedules p_0, p_1 based on the idea of Giffler and Thompson's active schedule generation [Giffler and Thompson, 1969].

Though a detailed description of the GA/GT algorithm is given in [Yamada and Nakano, 1992], a brief outline of the algorithm is given below:

1. Let o^* be an operation with the smallest completion time among all unscheduled operations.
2. Let G be a conflict set obtained by Giffler and Thompson method [Giffler and Thompson, 1969]. G is a set of operations which overlap their processing with o^* on the machine on which o^* is processed. Let M_i denote the machine.
3. The next operation to be scheduled on M_i is selected from G. Let's assume it to be the j'th operation on M_i. Choose one of the schedules $\{p_0, p_1\}$ according to the value of H_{ij}. Here H is a random bit matrix of size $m \times n$. If p_0 is chosen, then the operation from G which is selected is that that has the earliest starting time on p_0 among G.
4. Repeat these steps until all operations are scheduled.

In Step 3, mutation can be defined by modifying the arbitration criterion so that not the earliest but the k'th earliest operation is selected with small probability. The larger k is, the smaller the probability.

By applying GT crossover twice to the same pair of the schedules using the same bit matrix H, two new schedules are obtained, but the second schedule is constructed by switching the roles of p_0, p_1.

3 RESULTS

The effect of the ECO framework simulation on performance is studied with several JSS benchmarks. We begin with two moderately difficult problems, the 10x10 [Muth and Thompson, 1963] and 20x5 [Muth and Thompson, 1963] problems for which the global optimum is known. For these two problems we present a comparison study between running the GA/GT algorithm with and without the ECO framework. In all experiments the GA running under the ECO framework superseded the non-ECO simulation.

We also present results of a hybrid algorithm combining the GA/GT with a local search running under the ECO framework. This algorithm is applied to seven very difficult JSS problems for which only the latest best solution is known. The seven problems are the abz7, abz8, and abz9 15x20, la21 10x15, la27 and la29 10x20, and la38 15x15 [Applegate and Cook, 1991].

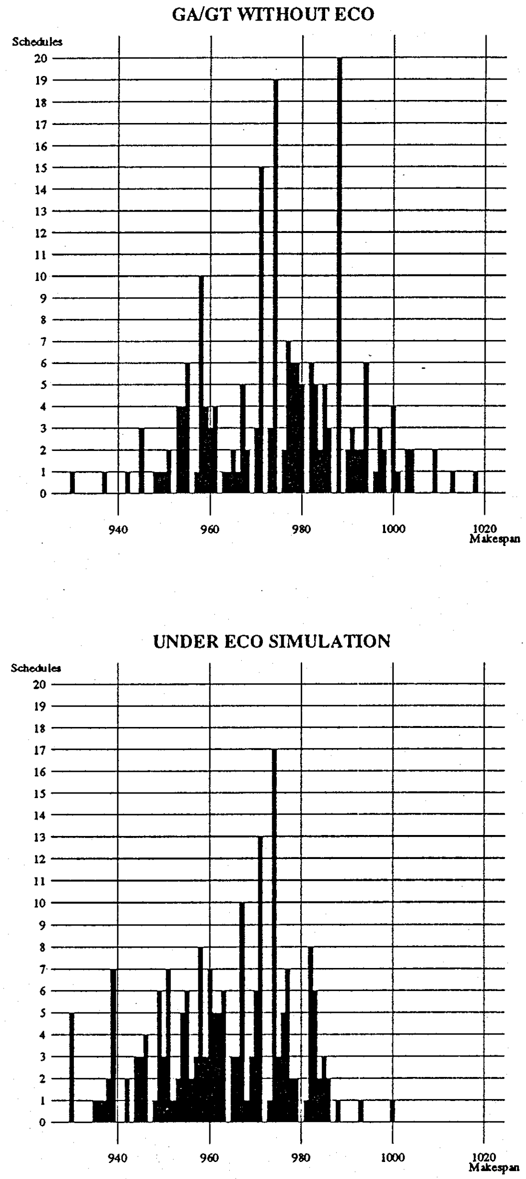


Figure 1: 200 experiments summary of a 10x10 JSS problem with population size 2025. The GA/GT algorithm (top), and the same algorithm running under the ECO simulation framework (bottom) (45x45 grid size for the ECO simulation).

3.1 The 10x10 problem

The GA/GT optimal population size for the 10x10 JSS problem is around 2000 (details of the optimal population size for GA/GT can be found in an unpublished manuscript [Nakano *et al*]). Figure 1 summarizes the results of 200 independent experiments running the GA/GT simulation alone and under the ECO framework (summarized in Table 1).

Since the effectiveness of the ECO framework depends on the size of the grid, another experiment is presented which includes a smaller population size of 1024 which results in a 32x32 grid size (Figure 2, and summarized in Table 1).

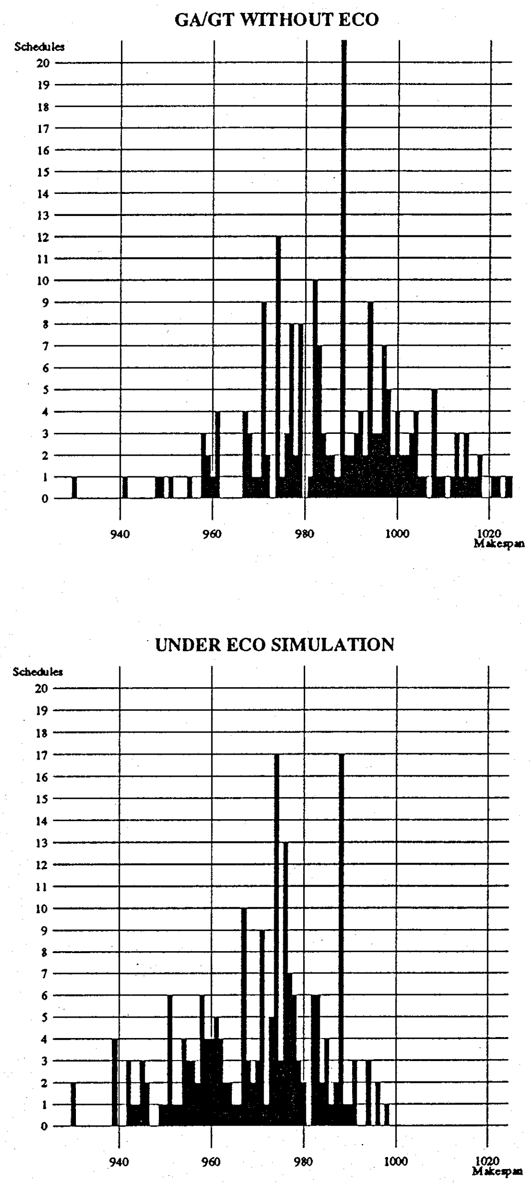
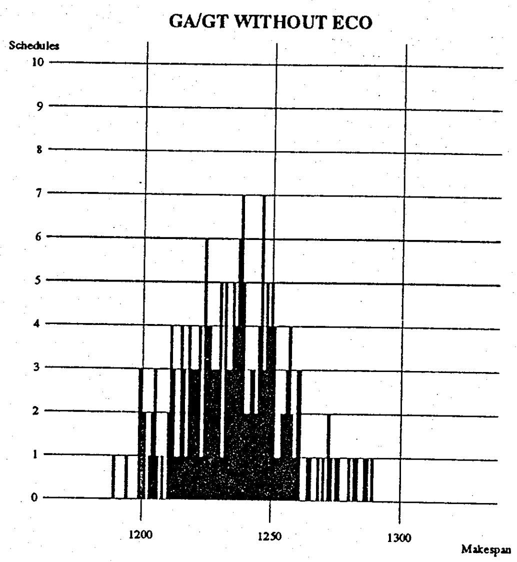


Figure 2: 200 experiments summary of a 10x10 JSS problem with population size 1024. The GA/GT algorithm (top), and the same algorithm running under the ECO simulation framework (bottom) (32x32 grid size for the ECO simulation).

Population Size	2025	1024
GA/GT	975(15)	986(16)
GA/GT with ECO	963(14)	970(14)

Table 1: 200 experiments summary of the off-line performance of the GA/GT and GA/GT under the ECO simulation for the 10x10 JSS problem (variance is given in parentheses).



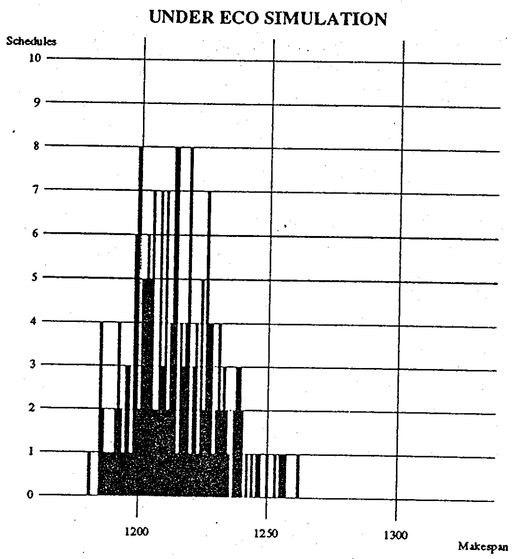
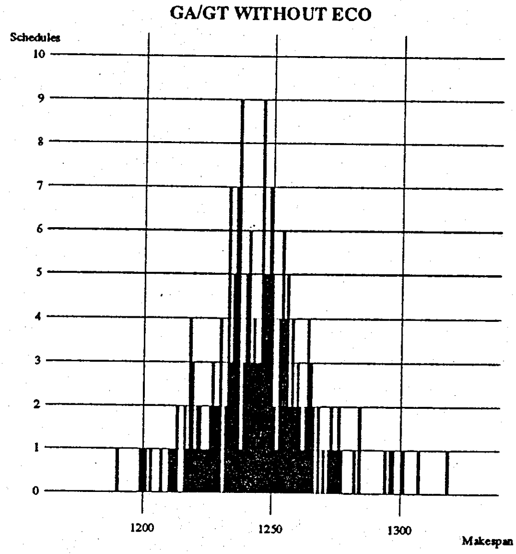


Figure 3: 200 experiments summary of a 20x5 JSS problem with population size 5041. The GA/GT algorithm (top), and the same algorithm running under the ECO simulation framework (bottom) (71x71 grid size for the ECO simulation).



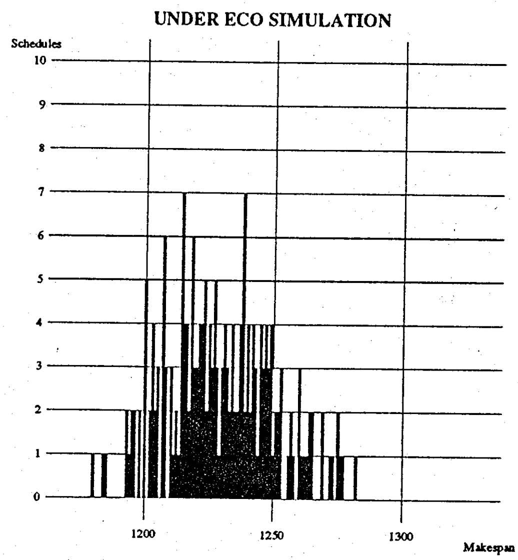


Figure 4: 200 experiments summary of a 20x5 JSS problem with population size 2025. The GA/GT algorithm (top), and the same algorithm running under the ECO simulation framework (bottom) (45x45 grid size for the ECO simulation).

3.2 The 20x5 problem

The 20x5 JSS problem is a more difficult one than the 10x10 problem discussed in the previous section. Its optimal population size for the GA/GT algorithm is around 5000 (details of the optimal population size for GA/GT can be found in an unpublished manuscript [Nakano *et al*]). Figure 3 summarizes the results of 200 independent experiments running the GA/GT algorithm alone, and under the ECO framework.

Figure 4 presents another set of experiments which use a smaller population size 2025. A summary of the two comparison studies is presented in Table 2.

Population Size	5041	2025
GA/GT	1236(19)	1244(19)
GA/GT with ECO	1213(16)	1229(20)

Table 2: 200 experiments summary of the off-line performance of the GA/GT and GA/GT under the ECO simulation for the 20x5 JSS problem (variance is given in parentheses).

3.3 Harder JSS problems

Moderately difficult JSS problems such as 10x10 and 20x5 are not the reason why so much effort is invested in developing special GA models for scheduling because other operation research techniques can handle such complexity well. The real challenge for GAs are really hard problems where OR techniques fail to solve the problem well. That is why we applied the GA technology to harder JSS problems. Unfortunately, GA models thus far, including the GA/GT algorithm, also had substantial difficulties escaping local optimum in harder JSS. With the encouraging results of the ECO simulation framework on moderate problems, we attempted to investigate the strength of the technique on harder problems.

Preliminary results obtained with a hybrid GA/GT + local search algorithm running under the ECO framework suggest that this combination of mechanisms is able to improve the known best of seven very difficult JSS problems (the abz7, abz8, and abz9 15x20, la21 10x15, la27 and la29 10x20, and la38 15x15 [Applegate and Cook, 1991]). For example, this combined algorithm located better solutions than the best known solutions reported in OR literature. In fact, it improved the known best of all the above mentioned problems, a performance that was not possible with the GA/GT algorithm alone. It is premature to report comprehensive analysis of the performance of this algorithm at this stage, but a thorough analysis of this hybrid algorithm running under the ECO framework is under way.

4 CONCLUSIONS

The ECO framework discussed in this paper presents a new synthesis of the conventional genetic operators. In the ECO framework all operators are based on local interaction in a 2D grid topology. This new interaction arrangement results in a rapid local convergence while maintaining both global diversity and speciation as an emergent property of the simulation. Furthermore, this improved control over convergence is achieved with almost virtually no additional computation overhead. In applying the ECO framework to JSS problems significantly improved robustness is obtained. This increased robustness translates into significantly improved on-line and off-line performance.

Sections 3.1 and 3.2 demonstrated the improved convergence of the ECO framework can offer to global GAs in the domain of JSS. Because there is virtually no additional computation cost incurred when applying the ECO framework, the robustness advantage of the ECO framework is clear.

There are only few parameters to set when applying the ECO framework, and their setting is not very sensitive. For example (and assuming a serial architecture), small changes in the grid dimension, type of mutation probabilities and scaling function used, result in minute differences. Nonetheless, too small a grid and the local genetic interaction degenerate into a global one. If a too large a grid is taken, each grid node does not have enough genetic interaction with its environment to make the most of the local genetic pool. However, between these two extremes there are many values which result in good performance.

References

[Applegate and Cook, 1991] Applegate, D., and Cook, W., (1991) A computational study of the job-shop scheduling problem, ORSA Journal on Computing, 3(2), 149-156.

[Cavicchio, 1970] Cavicchio, D. J. (1970) Adaptive search using simulated evolution, (Doctoral dissertation, University of Michigan) Unpublished doctoral dissertation.

[Davidor, 1991] Davidor, Y. (1991) A naturally occurring niche and species phenomenon: The model and first results, 4th International Conference on Genetic Algorithms, 257-263.

[Davidor, in press] Davidor, Y. (in press) Ecological model for evolutionary Computing, Journal of the Institute of Systems, Control and Information Engineers.

[De Jong, 1975] De Jong, K. (1975) An analysis of the behavior of a class of genetic adaptive systems, (Doctoral dissertation, University of Michigan) Dissertation Abstracts International 36(10), 5140B.

[Giffler and Thompson, 1969] Giffler, B., and Thompson, G.L., (1969) Algorithms for solving scheduling problems, journal of operations Research, Vol. 8, pp. 487-503.

[Goldberg and Richardson, 1987] Goldberg, D. E. and Richardson, J. (1987) Genetic algorithms with sharing for multimodal function optimization, 2nd International Conference on Genetic Algorithms, 41-49.

[Mauldin, 1984] Mauldin, M. L. (1984) Maintaining diversity in genetic search, National Conference on Artificial Intelligence, 247-250.

[Muth and Thompson, 1963]
Muth, J. F., and Thompson, G.L. (1963) Industrial scheduling, Prentice-Hall.

[Syswerda, 1989] Syswerda, G. (1989) Uniform crossover in genetic algorithms, 3rd International Conference on Genetic Algorithms, 2-9.

[Nakano *et al*] Nakano, R., Davidor, Y., and Yamada, T. (unpublished manuscript) Optimal population size under the constant computation cost criterion. NTT Communication Science Labs., 2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02, Japan.

[Syswerda, 1991] Syswerda, G. (1991) A study of reproduction in generational and steady state genetic algorithms, In G. J. E. Rawlins (Ed.), Foundations of Genetic Algorithms, Morgan Kaufmann.

[Whitley, 1988] Whitley, D. (1988) GENITOR: a different genetic algorithm, Rocky Mountain Conference on Artificial Intelligence.

[Yamada and Nakano, 1992] Yamada, T. and Nakano, R. (1992) A Genetic Algorithm Applicable to Large-Scale Job-Shop Problems. Parallel Problem Solving from Nature (Brussels,Belgium).