Reinforcement Learning in Steady-State Cellular Genetic Algorithms

Cin-Young Lee and Erik K. Antonsson

Abstract— **A novel cellular genetic algorithm is developed to address the issues of good mate selection. This is accomplished through reinforcement learning where good mating individuals attract and poor mating individuals repel. Adaptation of good mate choice occurs, thus, leading to more efficient search. Results are presented for various test cases.**

I. Introduction

Cellular genetic algorithms (CGAs) were initially introduced as a parallelized implementation of GAs that provided a more accurate model of evolution. In particular, CGAs introduce the notion of a population topology in which individuals can only interact in a local manner. As implied by its name, CGA employs a cellular, typically toroidal, topology. Mating and competition can only occur between individuals of adjacent cells, which leads to restricted mating and competition. This is in direct contrast to canonical GAs that are randomly breeding with no restrictions on possible mates. These randomly breeding populations are known as *panmictic* populations. It has been shown in both population genetics and evolutionary computation theory that non-panmictic populations (*i.e.*, restricted mating) have increased takeover times when compared to equivalent panmictic populations. Thus, CGAs are often more robust to premature convergence than conventional GAs. Nonetheless, CGAs can still be made to simulate natural evolutionary processes more realistically. Although restricting mating to local interactions roughly approximates the topological structure of real populations, there are additional behavioral aspects of mate selection that can be implemented. In many instances, individuals are more likely to choose similar mates that should produce more fit offspring. One approach to introducing this concept is reinforcement learning of mate selection.

Here, reinforcement learning is implemented in the reproduction/selection phase of CGAs to emulate simple behavioral aspects of individuals in real populations. The basic premise is to provide a mechanism that gravitates individuals with good mating characteristics while simultaneously repelling individuals that produce non-viable offspring. Such a mechanism should ensure good mate selection characteristics, leading to segregation of population members into somewhat isolated niches. Hence, it is believed that the addition of reinforcement learning will allow CGAs to more effectively handle multimodal searches in which niching of the population is often necessary.

Prior to discussing reinforcement learning in CGAs, a brief review of CGAs is provided. Subsequently, reinforcement learning in CGAs is presented. Afterwards, results of the modified CGA on a variety of test problems are shown. The test problems cover a broad range of difficulties and include simple unimodal search, more complex multimodal search, and the one-dimensional Ising model, which has been shown to require population niching [1]. For comparison, a traditional CGA and canonical GA are also applied to each test case. The paper concludes with a summary and discussion of future work.

II. Cellular Genetic Algorithms

Cellular genetic algorithms, also known as fine-grained and diffusion GAs, have an imposed structure on the types of interactions that can occur between individuals. In most cases, individuals occupy cells of a one or two-dimensional toroidal grid. Typically, a radius parameter, ρ, describes the neighborhood of interaction of each individual. For example, $\rho = 1$ indicates that an individual can only mate with an individual at most one cell away (*i.e.*, an adjacent cell). During reproduction, instead of choosing a mate at random as in the canonical, panmictic case, each individual mates with the best performing individual within its neighborhood. This approach has two benefits. Firstly, as mentioned before, the non-panmictic structure leads to increased takeover times that, in some instances, will reduce the probability of premature convergence by maintaining solution diversity. Secondly, the CGA framework lends itself directly to a parallel computing implementation. Much literature is available on CGAs and the interested reader is referred to a subset of early work found in [2]–[5].

III. Reinforcement Learning in CGAs

Restriction of mate selection to a local neighborhood can be beneficial as seen by a variety of successful CGA applications. However, besides limiting mating to occur in a local manner, one could also restrict mating to individuals that will result in good offspring. By preventing matings between incompatible individuals, the search can be made more efficient. In most cases, a population niching approach is taken to achieve this type of mate selection [6]–[9]. No one has yet, to the authors' knowledge, implemented this type of mate selection in a CGA framework. In this section, an approach for maintaining good mate selection in CGAs is developed and presented.

Before continuing, it is important to define what is meant by good mate selection. The idea here is either only to allow matings between individuals that produce good offspring (*i.e.*, having a fitness better than its parents') or to choose these matings with a much higher probability than those matings that result in non-viable, or less fit, offspring.

0-7803-7282-4/02/$10.00 ©2002 IEEE

Thus, the problem is to determine which individuals have good mating characteristics and to group these individuals in such a manner that they choose one another as mates with high probability (*aka*, niching).

The problem is solved in the following manner by introducing reinforcement learning into CGAs. A steady-state algorithm is used in which only a single individual is replaced every generation. Reproduction occurs by first selecting a single individual, p, then choosing its mating partners, m_i. Rather than choosing mates deterministically, typical of most CGAs, mates are selected stochastically. The location of each mating partner is determined from a Gaussian random variable with zero mean and σ variance. σ is chosen empricially to be 8% of the population size. Since individuals are not allowed to mate with themselves, the absolute distance is modified to be greater than or equal to 1. This mating scheme ensures that, on average, the most common mating partners will be the adjacent cells. Moreover, individuals closer to p have a higher probability of being chosen as a mate. In order to take advantage of this selection scheme, some mechanism for aggregating individuals with good mating characteristics is required. Conversely, one could disperse individuals with poor mating characteristics to achieve similar niching of the population.

A simple approach is taken in which mates with good offspring are rewarded by moving them closer to p. At the same time, mates with poor offspring are punished by moving them away from p. One option for achieving this type of reinforcement learning is to evaluate the offspring of an individual and move it closer to p if the offspring is better than both parents, further from p if it is worse than both parents, or nowhere if it is better than just one parent. Two potential difficulties of this approach are how to determine the magnitude of the move and how to prevent the displacement of previously well adapted individuals. A tournament like system is developed to circumvent these problems.

Two or more mates are selected using the previously described mechanism. The locations of each mate are placed in an array, D, and sorted according to their Manhattan distance from p. The mates are subsequently relocated along D according to their offsprings' fitness values. So, for example, the mate with the best offspring will be moved to the closest location in D, the worst to the furthest location in D, etc. Hence, it is expected that good mating combinations will tend to gravitate together and poor mating combinations will tend to repel one another.

Figure 1 illustrates the reinforcement learning mechanism for the case of two mates. First, a parent is chosen, as indicated, then two mates are chosen randomly from a Gaussian centered around the parent's location. In this case, Mate 1 is further away from the Parent than Mate 2. After crossover, the offspring are evaluated and it is seen that the offspring of Mate 1, Offspring 1, has the highest fitness. So, since Mate 1 produced a better offspring than Mate 2, it should be moved closer to Parent. This is accomplished by swapping the positions of Mate 1 and Mate 2 as shown. Finally, due to its good fitness value, Offspring 1 replaces Parent in the next generation.

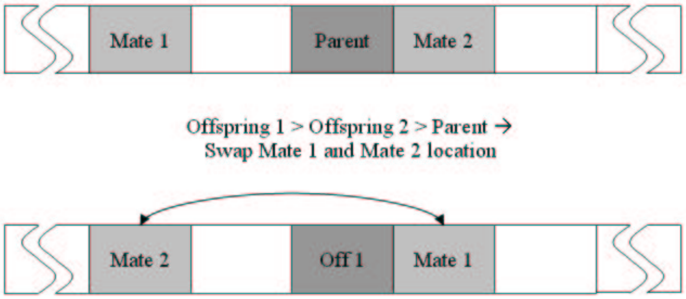
Mate 1
Parent
Mate 2
Offspring 1 > Offspring 2 > Parent →
Swap Mate 1 and Mate 2 location
Mate 2
Off 1
Mate 1

Fig. 1. Diagram of RLCGA in action.

IV. GA Details

Three different GAs are used to determine the utility of reinforcement learning CGAs (RLCGAs in the sequel). These are the just developed RLCGA, a standard CGA, and a canonical, panmictic GA (henceforth referred to as PGA). To ensure commensurate results, all three GAs have the following characteristics. Chromosomes encode solutions as bit strings. Mutation is the standard bit flip operation and crossover is of the two point variety. Mutation probability is fixed at 1/# bits and crossover probability is fixed at 1. Population size is set to 250. For both RLCGA and CGA, the populations are embedded in one-dimensional rings. A steady-state reproduction mechanism is adopted in which a single individual is replaced every generation. The interested reader is referred to [10] for more information on steady-state GAs. A more detailed description of each GA follows.

The reinforcement learning CGA has $N = 2$, *i.e.* two mates are selected each generation. The offspring are evaluated and the mates are re-sorted as described previously. An elitist approach is taken in which the best offspring survives to the next generation only if its fitness is better than p's. If this is the case, the offspring replaces p.

A steady-state CGA is developed for comparison with the RLCGA. The neighborhood radius of the CGA is 1, so individuals can only mate with their nearest neighbors. To keep the number of fitness evaluations per generation commensurate with the RLCGA implementation, both neighbors are mated with p and the best offspring replaces p if its fitness is greater than p's.

The PGA is the canonical GA implementation with unrestricted mating partners that are chosen uniformly at random from the entire population. Again, to keep fitness evaluations per generation commensurate, two mates per parent are chosen and elitist selection of offspring is maintained.

V. Test Problems

Three problems are used to test RLCGA and to determine whether the hypothesis of improved performance in environments requiring niching is true.

0-7803-7282-4/02/$10.00 ©2002 IEEE

The first function is a simple unimodal function as shown in Figure 2. The function can be written as

$$y = \mathcal{G}(50)$$

where $\mathcal{G}(\mu)$ is the Gaussian with μ mean and a variance of 12. For all the GAs, chromosomes are 25 bits in length and are decoded into the range 0–100. This function serves as a quick gauge of RLCGA effectiveness.

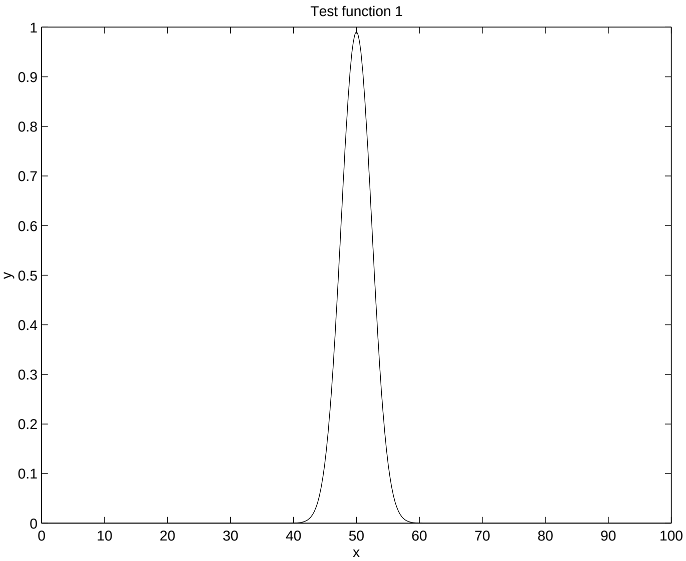

Fig. 2. Test function 1: unimodal function.

The second test function is a multimodal function composed of Gaussians similar to those found in the first test function. This function is shown in Figure 3 and is similar to Beasley's F2 function [7]. Like the first test function, chromosomes are 25 bits long and are decoded into the range 0–100. Due to the multimodal nature of this function, if the RLCGA can take appropriate advantage of its mate selection, it should perform better than both CGA and PGA. The function can be written as

$$y = \mathcal{G}(10) + .9\mathcal{G}(30) + .75\mathcal{G}(50) + .6\mathcal{G}(70) + .5\mathcal{G}(90)$$

where the notation is the same as for the first test function.

The third problem is the one-dimensional Ising model as found in [11]. Van Hoyweghen showed that niching was required to find one of the globally optimal solutions of the Ising model [1]. The one-dimensional Ising model is a nearest-neighbor interaction function and can be written as

$$f : \{0,1\}^l \rightarrow \mathbb{R} : \mathrm{x} \mapsto \sum_{\mathrm{i}=1}^{\mathrm{l}} \delta(\mathrm{x_i}, \mathrm{x_{i+1}})$$

where l is the length of the bit string, $x_{l+1} \equiv x_1$, and

$$\delta(x_i, x_j) = \begin{cases} 1 & \text{if } x_i = x_j; \\ -1 & \text{otherwise.} \end{cases}$$

The two global optima of the one-dimensional Ising model are a string of either all zeroes or all ones. It has spin-flip symmetry (*i.e.*, if each bit is flipped the fitness remains the same) that makes it a difficult problem for conventional GAs without niching. A 100 bit Ising model is used in this paper.

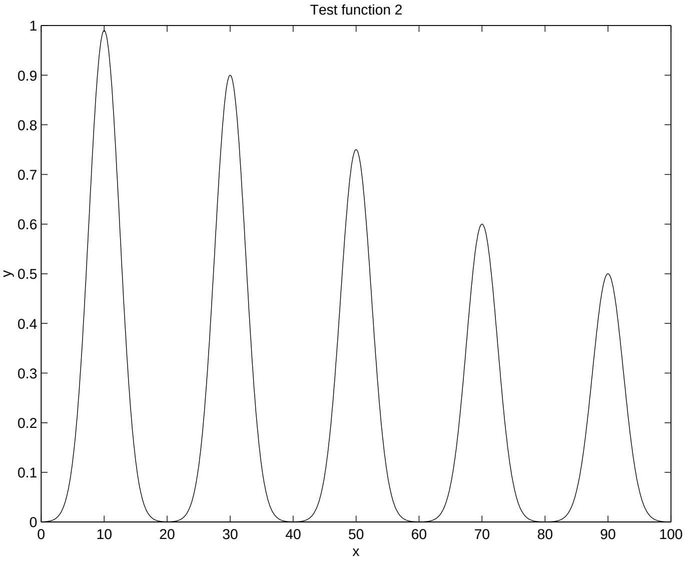

Fig. 3. Test function 2: multimodal.

VI. Results and Discussion

For each test problem, both the steady-state CGA and RLCGA were run 200 times to obtain statistically significant results. Additionally, results for a canonical, panmictic, steady-state GA (PGA) are presented for comparison.

Table I shows the number of generations required to find the optimum for the unimodal test function. It is apparent that RLCGA outperforms the CGA even when taking into account the modest standard deviations. Not surprisingly, because niching is unnecessary for the unimodal test function, PGA outperforms both of the cellular implementations; though the performance gain over RLCGA is marginal.

	Mean	σ	**Min**	**Median**	**Max**
CGA	808.7	363.0	14	809.5	1719
PGA	564.6	242.8	5	592.5	1158
RLCGA	643.3	252.7	4	658	1217

TABLE I

Generations to convergence for test function 1.

Table II shows the number of generations required to find the optimum for the multimodal test function. The mean time to convergence for all implementations is greater than that for the unimodal case, providing evidence of the more deceptive behavior of the search space. However, PGA convergence time does not rise as much as those for RLCGA and CGA. Moreover, PGA significantly outperforms both RLCGA and CGA. These two observations seem to indicate that the need for niching in the multimodal test problem is not a requisite. Nevertheless, the performance of RLCGA compared to CGA indicates that RLCGA is a promising alternative to CGA. The Ising model results should shed more light on the utility of RLCGA in niching environments.

	Mean	σ	**Min**	**Median**	**Max**
CGA	1504.5	925.9	21	1359.5	6040
PGA	622.7	291.7	10	624.5	1388
RLCGA	844.7	453.0	9	789.5	2658

TABLE II

Generations to convergence for test function 2.

As mentioned previously, the Ising model has been shown to require niching for efficient solving [1]. Since convergence to one of the global optima often required more than 20,000 generations, an arbitrary cutoff point of 7,500 generations was chosen to compare convergence characteristics of the disparate GA implementations. Table III shows the best fitnesses at the 7,500th generation for each GA implementation. RLCGA outperforms the other GAs, but does not seem to significantly outperform PGA. This is somewhat troubling because it implies that RLCGA, even in a search that has provable niching requirements, is only slightly better than a canonical PGA. Figures 4 and 5 show the actual distribution of fitnesses for PGA and RLCGA runs on the Ising model. These figures indicate that RLCGA is clearly superior to PGA. However, in studying the Ising model, Van Hoyweghen showed that if the population size was large enough, then a canonical panmictic GA can find an optimal solution to the Ising model in a similar amount of time as a niching GA [1]. This is a result of the ability of large populations to maintain solution diversity. So, an extra set of experiments was run with a considerably smaller population size of 100. The results of these experiments are shown in Table IV.

The RLCGA with reduced population size exceeds the performance of the other two GA implementations. As expected, PGA performs much worse with the reduced population size as a result of the inability to maintain solution diversity. These results imply that RLCGA is able to effectively manipulate mate selection to form good niching characteristics.

Interestingly, the performance of the CGA drastically improves when the population size is reduced from 250 to 100. This is most likely a manifestation of the exploitation versus exploration trade-off in which, with excessively large populations, the steady-state CGA is unable to converge quickly because of a prediliction for exploration. With the smaller population, a more comfortable balance between exploitation and exploration can be struck, leading to greatly improved performance.

	Mean	σ	**Min**	**Median**	**Max**
CGA	77.2	3.1	68	76	88
PGA	90.8	3.2	80	92	100
RLCGA	93.2	2.5	88	92	100

TABLE III

Fitness after 7500 generations for the Ising model. Population size 250.

	Mean	σ	**Min**	**Median**	**Max**
CGA	90.8	3.0	84	92	96
PGA	85.0	4.2	72	84	96
RLCGA	91.9	3.3	80	92	100

TABLE IV

Fitness after 7500 generations for the Ising model. Population size 100.

VII. Conclusion

A novel cellular genetic algorithm is developed in this paper to more clearly address the issues of mate selection. This is accomplished by using a reinforcement learning algorithm to attract individuals with good mating characteristics while simultaneously repelling individuals with poor mating characteristics. Initial results are promising, as the reinforcement learning cellular genetic algorithm outperforms standard GA implementations on the Ising model, which has been shown to require niching [1]. Furthermore, RLCGA converges more quickly than a standard, albeit steady-state, CGA in all of the test cases. This indicates that RLCGA may be a good alternative to CGA in a variety of applications.

Nonetheless, much work remains to be done, particularly with application of RLCGA to more challenging, real-world problems. Also of interest are population sizing studies to determine the optimal settings for RLCGA, if they exist. Furthermore, the behavior of the RLCGA paradigm in more complex topologies (*i.e.*, two-dimensional toroidal grids) needs to be studied. The effects of increasing the number of mates would also be of great intellectual interest.

References

[1] C. VanHoyweghen, D.E. Goldberg, and B. Naudts, “Building block superiority, multimodality and synchronization problems,” Tech. Rep. 2001020, Illlinois Genetic Algorithms Laboratory, 2001.

[2] R. Collins and D. Jefferson, “Selection in massively parallel genetic algorithms,” in *Proceedings of the 4th International Conference on Genetic Algorithms*. 1991, pp. 249–256, Morgan-Kaufmann.

[3] Y. Davidor, “A naturally occurring niche and species phenomenon: The model and first results,” in *Proceedings of the 4th International Conference on Genetic Algorithms*. 1991, pp. 257–263, Morgan-Kaufmann.

[4] D. Hillis, “Co-evolving parasites improve simulated evolution as an optimization procedure,” *Physics D*, vol. 42, pp. 228–234, 1990.

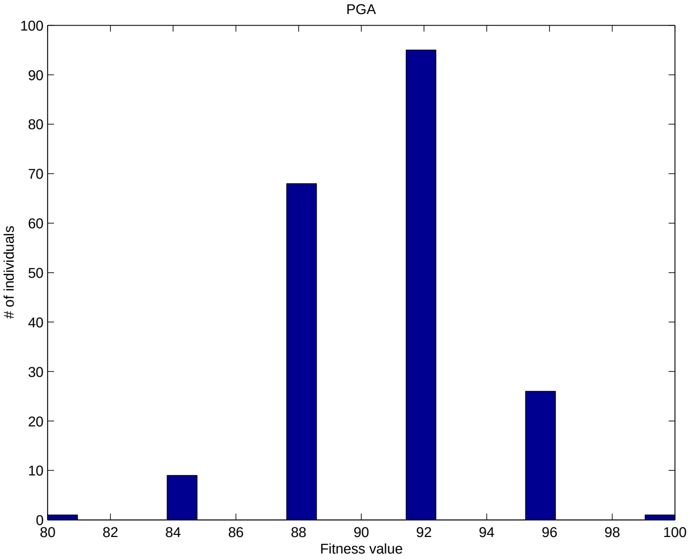

Fig. 4. Histogram of PGA fitnesses for the Ising model. Population size 250

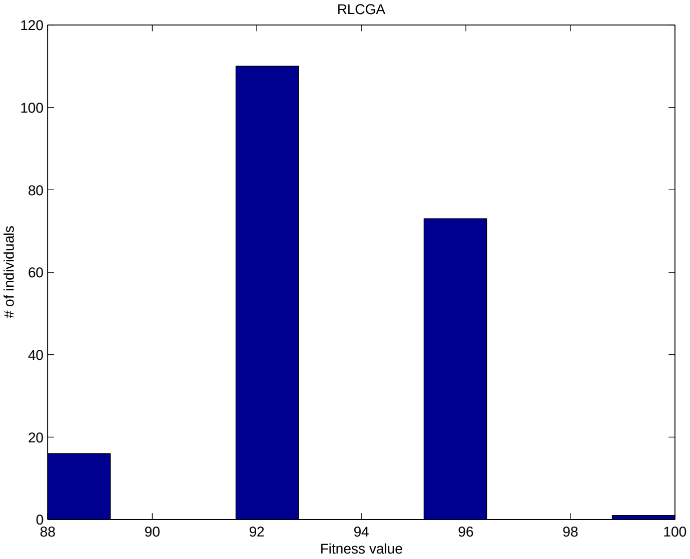

Fig. 5. Histogram of RLCGA fitnesses for the Ising model. Population size 250.

[5] B. Manderick and P. Spiessens, "Fine grained parallel genetic algorithms," in *Proceedings of the 3rd International Conference on Genetic Algorithms*. 1989, pp. 428–433, Morgan-Kaufmann.

[6] W.M. Spears, "Simple subpopulation schemes," in *Proceedings of the Evolutionary Programming Conference*. 1994, World Scientific.

[7] D. Beasley, D.R. Bull, and R. Martin, "A sequential niche technique for multimodal function optimization," *Evolutionary Computation*, vol. 1, no. 2, 1993.

[8] K. Deb and D.E. Goldberg, "An investigation of niche and species formation in genetic function optimization," in *Proceedings of the 3rd International Conference on Genetic Algorithms*. 1989, pp. 42–50, Morgan-Kaufmann.

[9] D.E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Proceedings of the 2nd International Conference on Genetic Algorithms*, 1987, pp. 41–49.

[10] G. Syswerda, "A study of reproduction in generational and steady-state genetic algorithms," in *Foundations of Genetic Algorithms*. 1991, Morgan Kaufmann.

[11] E. Ising, "Beitrag zur theories des ferromagnetismus," *Z. Physik*, vol. 31, no. 235, 1924.