

# On the Use of Niching for Dynamic Landscapes

**Walter Cedeño**

Research & Development  
Hewlett-Packard Company  
2850 Centerville Road  
Wilmington, DE 19808  
walter\_cedeno@hp.com

**V. Rao Vemuri**

Department of Applied Science  
University of California, Davis &  
Lawrence Livermore National Lab.  
Livermore, CA 94550  
vemuri1@llnl.gov

## Abstract

*Application of genetic algorithms to problems where the fitness landscape changes dynamically is a challenging problem. Genetic algorithms for such environments must maintain a diverse population that can adapt to the changing landscape and locate better solutions dynamically. A niching genetic algorithm suitable for locating multiple solutions in a multimodal landscape is applied. The results show the suitability of such approach to locate and maintain solutions in a dynamic landscape.*

## 1. Dynamic Landscapes

In this paper we examine the behavior of a niching algorithm on a multimodal dynamic landscape. A multimodal dynamic landscape is a search space where the locations and heights of peaks change with time. A superior solution (i.e., a peak) at time  $t_0$  could become an inferior solution at time  $t_1$ . An example of such a landscape occurs in stock markets where parameters influencing the price of stocks are constantly changing. In order to obtain the best return for the money invested, a stock broker must continually monitor these parameters and move the money so that the best payoff at that time is realized. Investment solutions that were not optimal at one time could potentially become optimal at some other time. In such a landscape, keeping a set of possible investment strategies is smart and necessary.

Multimodal dynamic landscapes, such as the one described above, present a challenging problem to any search technique. First, the heights and widths are constantly changing. Second, new peaks that are constantly emerging must be located quickly to avoid missing solutions that could potentially be useful. Third, old peaks that were found useful have since been flattened. Finally, solutions that have been found

previously must be re-evaluated after some time to maintain an accurate value of its fitness.

A technique for multimodal dynamic landscapes must be able to meet these challenges by locating multiple optima, existing ones and new ones, and maintaining them. Additionally, it must minimize the number of function re-evaluations for solutions found thus far without affecting the convergence properties of the technique. Later, we will see how the Multi-Niche Crowding (MNC) Genetic Algorithm (GA) (Cedeño, 1995) accomplishes these tasks.

## 2. Background

There have been many attempts to apply GAs to dynamic landscapes. None of them considered applying a niching technique to such problems. In one of the approaches part of the population is re-initialized (Eshelman, 1991; Grefenstette, 1992) after it has converged. In this approach it is very hard for the newly introduced solutions to establish themselves when the population contains highly fit individuals. Maresky et. al. (1995) introduced an operator called *selectively destructive re-start* that improves the previous approach by reinitializing the chromosome in a solution with certain probability. The probability is determined by a combination of factors such as population size, improvement of the best-in-generation individual, and the number of function evaluations. The main difficulty with this approach is to find the appropriate re-initialization probability for the problem at hand.

In another work, Cobb and Grefenstette (1993) compared a partial re-start of the population with two approaches that manipulate the mutation rate of the GA. In the first approach the mutation probability is set at a higher value than in a standard GA. In the second approach, called *triggered hypermutation*, the mutation rate is dynamically changed to high values when the time-averaged performance of the GA deteriorates. It was shown that mutation based approaches worked

better than population re-initialization for environments without abrupt changes. Some of the drawbacks are the reduction in the performance of the GA, in terms of the improvement to the average fitness of the population. Additionally, the mutation rate selected affects the performance of the GA greatly when applied to rapidly changing environments.

Some other approaches used sophisticated chromosome encoding schemes, such as including the previous history of the individual (Goldberg and Smith, 1987; Ng and Wong, 1995). In these studies the chromosomes are encoded using genome structures, like diploid or triallelic schemes, that are able to preserve genetic information that will be beneficial if the environment changes. In a separate study Dasgupta and McGregor (1992) used a tree structure representation of the population. In this approach, called the *sGA* (structured GA) nodes at a higher level in the tree regulated the activation and deactivation of genes at lower levels in the tree. These approaches were shown to work well in a landscape where the solution seems to oscillate between two peaks. It is unclear if these approaches will work on problems with many peaks.

In all of these approaches the main focus was on increasing the population diversity. Increasing the diversity allows the GA to discover new peaks while at the same time preserving the good solutions found thus far. Our MNC GA exhibits both of these properties implicitly. Solutions from multiple peaks are maintained while at the same time allowing a subset of the individuals in the population to explore other regions of the search space. In the sections below we will show how nicely the MNC GA solves problems in a dynamic landscape.

### 3. The Multi-Niche Crowding GA

The ability of organisms to evolve and adapt to their environment by means of natural selection has provided mother nature with a diverse set of species. This foundation, which is part of modern evolutionary thinking, was laid by Charles Darwin after the publication of his work "On the Origin of Species by Means of Natural Selection". Only organisms well adapted to their environment can survive from one generation to the next, transferring on the traits that made them successful to their offspring. Competition for resources between organisms and the ever changing environment drives some species to extinction and at the same time others evolve to maintain the delicate balance in nature. It is through this interaction between nature and organisms, that species possessing favorable traits for a given environment emerge. In this work we apply the same principles in the MNC GA, a genetic algorithm that evolves a population of mathematical solutions

containing different categories of solutions adapted to niches in a multimodal environment.

MNC GA is a computational metaphor to the survival of species in ecological niches in the face of competition. The MNC GA maintains stable subpopulations of solutions in multiple niches in multimodal landscapes. Each mode or peak is considered to be a niche. The algorithm introduces the concept of *crowding selection* to promote mating among members with similar traits while allowing many members of the population to participate in mating. The algorithm uses *worst among most similar replacement* (WAMS) policy to promote competition among members with similar traits while allowing competition among members of different niches as well.

The benefits of an approach that can locate multiple optima and maintain them throughout the search are many. Consider, for example, a dynamic environment where the optima are constantly changing. A technique that can locate and maintain multiple optima can inform the user when the current configuration is no longer the best based on the parameters in the environment. In other cases abnormal situations may require changes in the current configuration. Having viable alternatives at hand can allow for a smoother transition to the new configuration. An approach that can use a set of solutions to locate multiple optima is more practical for these types of environments. Additionally, there exist many problems where the location of the best  $K$  optima are needed in order to compare different answers and point out further experimentation. The benefits of the MNC GA have been already shown in applications to problems in DNA mapping (Cedeño, Vemuri, and Slezak, 1995), distributed databases (Cedeño and Vemuri 1996), and aquifer management (Cedeño and Vemuri, 1996).

Figure 1 shows an overview of the MNC GA. Initially, all the individuals in the population (size  $n$ ) are created at random and evaluated in parallel. Once the initial population is created, the operations of selection, mating and mutation, and replacement are applied for a given number of generations. In each generation individuals in the population are selected sequentially for mating, one at a time, and their mates are chosen using crowding selection. Then each pair participates in mating producing 2 offspring. The 2 offspring undergo mutation and those that are different to their parents are allowed to participate in replacement. The offspring left are then inserted, one at a time, into the population using WAMS replacement. These steps are repeated for the specified number of generations.

- Generate initial population of size  $n$  at random.

- Evaluate initial population.
- For gen = 1 to MAX\_GENERATIONS
  - For individual = 1 to  $n$ 
    - Use crowding selection to find mate for individual
    - Mate and mutate offspring
    - Insert offspring in population using WAMS replacement

Figure 1: Overview of the Multi-Niche Crowding GA.

In the MNC GA both the selection and replacement steps are modified with some type of crowding (De Jong, 1975). The idea is to eliminate the selection pressure caused by fitness proportionate reproduction (FPR) and allow the population to maintain diversity throughout the search. This objective is achieved in part by encouraging mating and replacement within the members of the same niche while allowing some competition for the population slots among the niches. The result is an algorithm that (a) maintains stable subpopulations within different niches, (b) maintains diversity throughout the search, and (c) converges to different local optima. No prior knowledge of the search space is needed and no restrictions are imposed during selection and replacement thus allowing exploration of other areas of the search space while converging to the best solutions in the different niches.

### 3.1 Crowding Selection

In MNC, the FPR selection is replaced by what we call *crowding selection*. In crowding selection most individuals in the population get a chance for mating in every generation. Application of this selection rule is done in two steps. First, an individual from the population is selected as a parent for mating. Second, its mate is selected, not from the entire population, but from a small group of individuals of size  $C_s$  (crowding selection group size), picked uniformly at random (with replacement) from the population. The mate thus chosen must be the one who is most “similar” to the selected individual. The similarity metric used here is not a genotypic metric such as the Hamming distance, but a suitably defined phenotypic distance metric. The Euclidean distance between two points is such a metric when the MNC GA is applied to function optimization.

Crowding selection promotes mating among members having similar traits and allows many of the members of the population to participate in mating. This allows members of the same niche to participate in mating more often and preserve those traits that define their species. At the same time mating between different species may occur giving rise to new species. Unlike mating restriction (Deb and Goldberg, 1989) that only

allows individuals from the same niche to mate, crowding selection allows some amount of exploration to occur while at the same time looking for the best individual in each niche.

### 3.2 Worst Among Most Similar Replacement

During the replacement step, MNC uses a replacement policy called *worst among most similar* (WAMS). The goal of this step is to pick an individual from the population for replacement by an offspring. Implementation of this policy follows these steps. First,  $C_f$  “crowding factor groups” are created by picking uniformly at random (with replacement)  $s$  (crowding group size) individuals per group from the population. Second, one individual from each group that is most similar to the offspring is identified. This gives  $C_f$  individuals that are candidates for replacement by virtue of their similarity to the offspring. The offspring will replace one of them. From this group of most similar candidates, we pick the one with the lowest fitness to die and be replaced by the offspring.

After the offspring becomes part of the population it competes for survival with other individuals when the next offspring is inserted in the population. In WAMS replacement offspring are likely to replace low fitness individuals from the same niche. It can also happen that it replaces a high fitness individual from the same niche or an individual from another niche. This allows a more diverse population to exist throughout the search. At the same time it promotes competition between members of the same niche and between members belonging to different niches. A similar technique was used by Goldberg (1989) in classifier systems, but in that work the most similar individual out of a group of low fitness candidates was replaced.

Worst among most similar replacement promotes competition among members with similar traits belonging to the same niche while allowing competition among members of different niches. This replacement technique accomplishes two things. First, by promoting competition among members of the same species in a niche it applies the *survival of the fittest* rule that is so prevalent in nature. Only those that are fit to their environment can survive for many generations, thus allowing the species to evolve to their best potential within their niche. Second, by allowing competition between different species, those species that are a better fit for their environment tend to occupy more slots in the overall population.

Both the selection and replacement steps in the MNC are primarily based on a similarity metric. Fitness is also considered during replacement to promote competition among members of the same niche. Competition among members of different niches occurs naturally.

#### 4. Test Function Used

In this section we describe a multimodal test function with dynamically changing peaks that was used to test the performance of the MNC GA. We also describe the parameter settings for the MNC GA. The test function used here is based on

$$F(x, y) = \sum_{i=1}^P H_i / 1 + W_i [(x - X_i)^2 + (y - Y_i)^2]$$

where  $P$  indicates the number of peaks in the function,  $(X_i, Y_i)$  the coordinates of peak  $i$ ,  $H_i$  the height of peak  $i$ , and  $W_i$  determines the width at the base of peak  $i$ . Note that the lower the value of  $W$ , the wider the base of the peak. We want to test three cases with this function. The first case is based on a landscape where the number of peaks (or optima) at the beginning of the run is the same as when the run ends, but the location and properties of the peak (width and height) are different. For this case we generate 10 peaks (with different width and height) at random. We start the run with the first set of 5 peaks completely manifested in the landscape. Then we dynamically decrease the height of these peaks while at the same time increase the height of the second set of 5 peaks. Toward the completion of the run all the peaks in the first set have disappeared and the peaks in the second set are fully manifested.

The second case is similar to the first case, but here we start with 2 peaks in the first set and dynamically change to the second set containing 8 peaks. This case is more challenging since the MNC GA must be able to dynamically locate and spread the individuals in the population to other peaks in the landscape.

The third and final case is a mirror image of the second case. We start with an initial set of 8 peaks and dynamically change them to a second set containing 2 peaks. This case was done for completeness only, since success in the second case, described above, will provide some results indicating the expected behavior in this simpler case.

In all three cases we used the 10 peaks (generated randomly) shown in Table 1. Case 1 contains peaks 1 to 5 in the first set and peaks 6 to 10 in the second set. Case 2 contains peaks 1 and 2 in the first set and peaks 3 to 10 in the second set. Finally, case 3 contains peaks 1 to 8 in the first set and peaks 9 and 10 in the second set.

The landscape is formed from the contribution of each set of peaks to the total value of the function. The contribution of each set of peaks is increased or decreased by 10% every  $g$  generations. In our tests we used values of  $g$  equal to 10 and 20 generations. To be

Table 1: Parameters for the 10 peaks used for all tests.

P	Peak Location	Width	Height
1	(8.37208, 64.8927)	3.81693	63.0384
2	(50.1509, 13.6573)	3.54459	96.1568
3	(51.1377, 28.8592)	1.6538	68.0674
4	(51.712, 9.99087)	4.32051	89.004
5	(9.3263, 19.3406)	1.92525	27.3985
6	(30.6667, 55.8088)	4.41471	73.3882
7	(47.7256, 38.3194)	0.91631	90.1156
8	(28.3306, 12.1417)	4.93385	85.6341
9	(25.8438, 50.7668)	0.20225	13.2312
10	(11.7122, 30.492)	1.02969	82.9828

more exact, let  $h_1(x, y)$  be the function, similar to  $F$ , with peaks defined from set 1. Let  $h_2(x, y)$  be the function, similar to  $F$ , with peaks defined from set 2. The fitness value  $f(x, y)$  of an individual in generation  $k$  is given by the following equation:

$$f(x, y) = (1 - 0.1 \left\lfloor \frac{k}{g} \right\rfloor) h_1(x, y) + 0.1 \left\lfloor \frac{k}{g} \right\rfloor h_2(x, y)$$

Due to the steady state nature of the MNC GA, individuals that survive for many generations (here a generation is defined as the application of  $n$  selection steps) must be re-evaluated to properly adjust their fitness. For these tests we re-evaluated all individuals that have been in the population for more than 10 generations.

#### 5. Results and Discussion

The simulations were done in a 486/33 MHz PC with an application developed using the “ANSI C” language. The genome for each individual is comprised of two chromosomes  $x$  and  $y$  representing the coordinates of the test functions. Each coordinate,  $0 \leq x, y \leq 65.535$ , was encoded using a 32 bit chromosome. The genome for each individual will then consist of a 64 bit string. The individuals in the initial population were generated at random.

The crossover operator used during mating is called *interval crossover*. In interval crossover only one offspring is generated. For each pair of parent chromosomes,  $x_1$  and  $x_2$ , (assume without loss of generality that the cardinal value of binary string  $x_1$  is less than the cardinal value of binary string  $x_2$ ), the offspring’s chromosome is selected at random from the interval  $[x_1 - \epsilon/2, x_2 + \epsilon/2]$ . The value for  $\epsilon$  is usually small, less than 0.001% of the highest cardinal value for the binary string. This allows the offspring to move outside the boundaries delineated by their parents. The usual bit mutation was applied to the offspring after crossover.

The crossover probability ( $\chi$ ) was set at 1.0 with  $\epsilon = 2000$ . The mutation probability ( $\mu$ ) was set at 0.003. Similarity between two individuals was determined by adding the distance (using the chromosome decimal value) between all chromosomes in the genome. For example, for two individuals  $I_1$  and  $I_2$  with genomes  $(x_1, y_1)$  and  $(x_2, y_2)$  respectively, we have the similarity between them given by the sum  $|x_1 - x_2| + |y_1 - y_2|$ . This metric is equivalent to using the phenotype values of the chromosomes. The MNC GA was executed for 120 generations in each run. All other parameters for the MNC GA were set as follows;

Crowding selection size ( $C_s$ ): 20  
 Crowding factor ( $C_f$ ): 4  
 Crowding group size ( $s$ ): 40

These values were selected to allow mating among individuals from different peaks more often and to increase replacement of less fit individuals from the same peak. They may not (with a high probability) be the optimal values for these tests.

The results are very promising. In all test cases the MNC GA was able to locate the new peaks emerging at different locations of the landscape. Moreover, the niche count adjusted dynamically among the peaks as their height and width changed. The maximum fitness of the

Population size ( $n$ ): 200

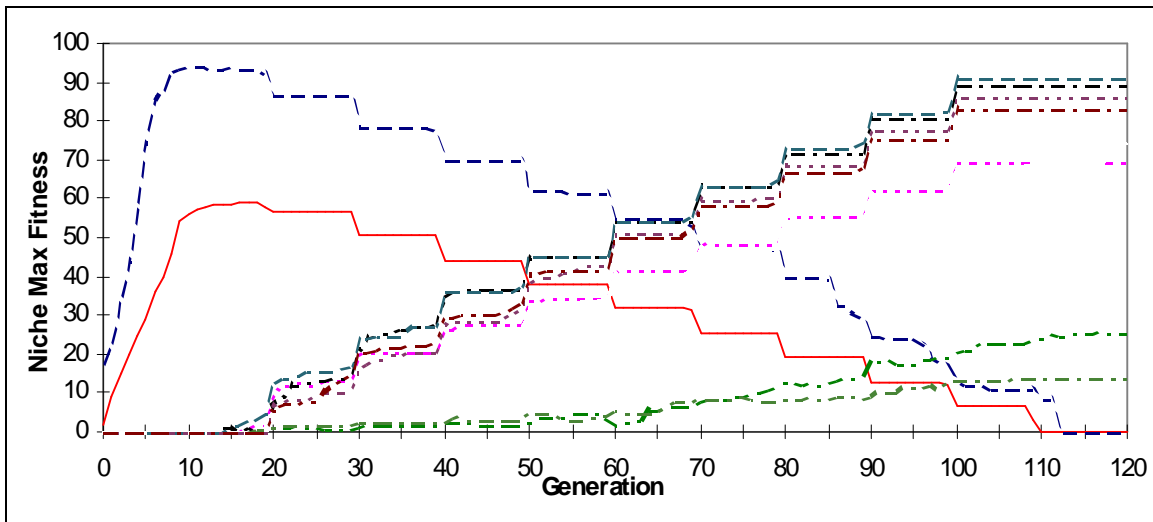
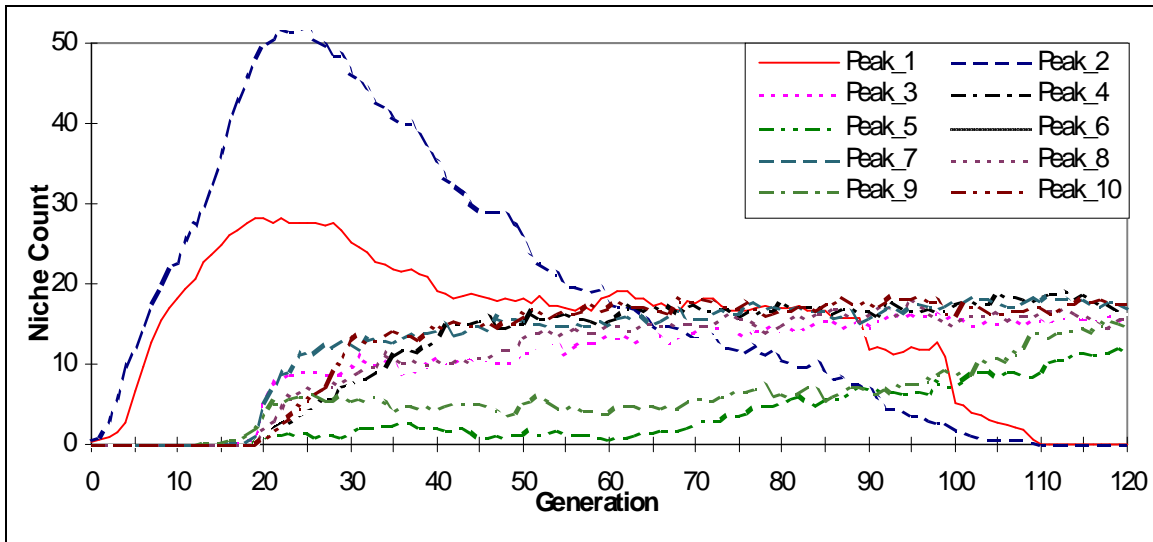


Figure 2: Niche count (top) and niche maximum fitness (bottom) for test case 2 using a value of  $g = 10$ .

highest peaks was found and maintained dynamically during the run.

Figure 2 shows the niche count and the niche maximum fitness for test case 2. We can observe the algorithm converging to the first two peaks in the initial generations. As the other peaks emerge they were located and all peaks maintained in parallel. After generation 100 the initial two peaks disappeared and so did the individuals in the niche. From the niche maximum fitness graph, in the same figure, we can observe the steps for each peak. It indicates how the MNC GA was able to locate the maxima in the highest peaks, maintain them during the run, and adjust to the changes dynamically. As the peaks in set 1 (peaks 1 and 2) decreased so did their niche count. Between generations 50 and 60, when both sets of peaks contributed equally to the total fitness, we can see all the lines coming together in the charts. After this point the peaks in the second set took over the population.

Other tests can be done to determine with more certainty the applicability of the MNC GA to dynamic environments. For example, we can use a landscape where all peaks are changing independently of each other and new peaks are created at random. Such an environment is more practical and is an area where more research is needed. Nevertheless, the test cases used here have shown the ability of the MNC GA to adapt well to such environments. It locates and maintains the different peaks dynamically without the need of restarting the population, using a higher mutation rate, or using special encoding for the chromosomes.

## 6. Summary

The properties exhibited by the MNC GA are very encouraging. The approach successfully locates multiple optima and maintains stable subpopulations in each peak. The formation of different subpopulations in the niches evolve naturally thus maintaining a very healthy diversity throughout the search. It is this equilibrium between exploration and exploitation of the search space that makes the MNC GA a very good technique for complex problems in static or dynamic environments. A more rigorous analysis must be made before we can claim the success of the algorithm when applied to other problems. The ability to search a complex space in an effective manner and locate multiple optima will be useful in many areas where current optimization techniques do not work well. The same properties make the MNC GA applicable to problems where the search space is changing dynamically.

## References

- Cedeño, W. (1995). The multi-niche crowding genetic algorithm: analysis and applications. *UMI Dissertation Services*, 9617947.
- Cedeño, W. and Vemuri, V. (1996). Genetic algorithms in aquifer management. *Journal of Network and Computer Applications*, 19, 171-187, Academic Press.
- Cedeño, W. and Vemuri, V. (1996). Database design with genetic algorithms. D. Dasgupta and Z. Michalewicz (eds), *Evolutionary Algorithms in Engineering Applications*, Springer Verlag, 3/97.
- Cedeño, W., Vemuri, V., and Slezak, T. (1995). Multi-Niche crowding in genetic algorithms and its application to the assembly of DNA restriction-fragments. *Evolutionary Computation*, 2:4, 321-345.
- Cobb, H. J. and Grefenstette, J. J. (1993). Genetic algorithms for tracking changing environments. In S. Forrest (ed.) *Proceedings of the Fifth International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers San Mateo, California, 523-530.
- Dasgupta, D. & McGregor, D. R. (1992). Non-stationary function optimization using the structured genetic algorithm. In R. Manner and B. Manderick (eds.), *Parallel Problem Solving from Nature*, 2. Amsterdam: North Holland, 145-154.
- Deb, K. and Goldberg, D. E. (1989). An investigation of niche and species formation in genetic function optimization, In J. D. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms*. San Mateo, CA: Morgan Kaufmann, 42-50.
- Eshelman, L. J., ed. (1991). The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In G.J.E. Rawlings (ed.), *Foundations of Genetic Algorithms - 1*. San Mateo, CA: Morgan Kaufmann Publishers.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization & Machine Learning*. Reading MA: Addison-Wesley.
- Goldberg D. E. & Smith R. E. (1987). Non-stationary function optimization using genetic algorithms with dominance and diploidy. In J. J. Grefenstette (Ed.), *Proceedings of the Second International Conference on Genetic Algorithms*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Grefenstette, J.J. (1992). Genetic algorithms for changing environments. In R. Manner and B. Manderick (eds.), *Parallel Problem Solving from Nature*, 2. Amsterdam: North Holland, 137-144.
- Maresky, J., Davidor, Y., Gitler, D., Aharoni, G., and Barak, A. (1995). Selectively destructive restart. In L. J. Eshelman (ed.), *Proceedings of the Sixth International Conference on Genetic Algorithms*. San Mateo, CA: Morgan Kaufmann Publishers.
- Ng, K. P. & Wong, K. C. (1995). A new diploid scheme and dominance change mechanism for non-stationary function optimization. In L. J. Eshelman (ed.), *Proceedings of the Sixth International Conference on Genetic Algorithms*. San Mateo, CA: Morgan Kaufmann Publishers.