

# Multiobjective optimization for dynamic environments

**Lam T. Bui**

ARC Centre for Complex Systems  
The Artificial Life and Adaptive  
Robotics Lab, School of ITEE, Uni.  
New South Wales, ADFA, Canberra,  
ACT, Australia, 2600,  
l.bui@adfa.edu.au

**Jürgen Branke**

Institute AIFB  
University of Karlsruhe  
76128 Karlsruhe  
Germany

branke@aifb.uni-karlsruhe.de

**Hussein A. Abbass**

ARC Centre for Complex Systems  
The Artificial Life and Adaptive  
Robotics Lab, School of ITEE, Uni.  
New South Wales, ADFA, Canberra,  
ACT, Australia, 2600,  
h.abbass@adfa.edu.au

## Abstract-

**This paper investigates the use of evolutionary multi-objective optimization methods (EMOs) for solving single-objective optimization problems in dynamic environments. A number of authors proposed the use of EMOs for maintaining diversity in a single objective optimization task, where they transform the single objective optimization problem into a multi-objective optimization problem by adding an artificial objective function. We extend this work by looking at the dynamic single objective task and examine a number of different possibilities for the artificial objective function. We adopt the *Non-dominated Sorting Genetic Algorithm version 2* (NSGA2). The results show that the resultant formulations are promising and competitive to other methods for handling dynamic environments.**

## 1 Introduction

Real life problems are hardly static. Optimizing portfolio decisions for example need to be updated every now and then to reflect changes in the stock market. Optimizing a university timetable needs to adapt to constant changes because of staff absence and room unavailability because of special events. Even the optimization of a flight path needs to adapt to changes on the fly; especially under the new free-flight control arrangements.

*Evolutionary algorithms* (EA) are widely used to deal with optimization problems in dynamic environments (DE) [5]. The nature of dynamism can vary from one problem to another. When using EAs to solve DE problems, we are usually interested in the ability of the algorithm to adapt and recover from the changes.

One of the main problems facing an evolutionary method when solving DE problems is the loss of genetic diversity. By maintaining a sample of previously visited areas in the search space while improving still the quality of evolved solutions, the time for adaptation decreases when the environment changes. A variety of methods have been proposed in the literature by maintaining diversity, maintaining a separate memory to store the best solutions found in each generation, or by using multi-populations to track areas in a changing landscape.

In this paper, we investigate the use of evolutionary multi-objective optimization methods (EMOs), which are known to be highly effective in preserving the genetic diversity [2], for DE problems. We use the moving peaks

benchmark (MPB) as a standard testbed for our experiments with different values of change severities. When formulating the problem as a multi-objective problem, we use two objective functions. In all experiments, the first objective is the original single dynamic objective, while the second objective is an artificial objective to promote diversity. We examine six different possibilities for the artificial objective. The first is based on a time-stamp associated with each chromosome. The second works by generating the value of the artificial objective at random. The third works by reversing the optimization of the first objective (*i.e.* maximizing the function if the original problem is minimization and vice-versa). The last three options are derived from the Euclidean distance in the decision space: the distance to the closet neighbor, the average distance to all individuals in the population and the distance to the best individual in the population. All the results will be compared against a traditional GA and two classical variants for dynamic environments: the random immigrants[14] and a variation of the hyper mutation algorithm[6], we call MutateHigh. In MutateHigh, each bit in the population is flipped with a probability 0.5; that is, on the average 50% of the chromosome get mutated. Further, NSGA2 is employed as the evolutionary multi-objective technique.

In the rest of the paper, we will review a few work in the changing environment literature, and the use of EMOs for solving single objective optimization problems. We then explain the setup for the different experiments, results, and discussions.

## 2 Related Work

### 2.1 Changing environments

As has been stated in the introduction, many real-world optimization problems are dynamic. Changes in the environment can take various forms such as changes in the parameters, objective functions or problem constraints. To classify the factors in play for defining the difficulties associated with a changing environment problem, a number of classifications have been proposed in the literature such as frequency, severity, and predictability [5, 25].

The frequency of change determines how often the environment changes. As the frequency increases, the time left for adaptation gets shorter and tracking the optima gets harder. The severity of a change indicates the degree of a change, e.g. the distance the optimum moves, or how

strongly the heights of local optima change. The predictability of change defines the pattern of the change such as linearity, randomness or circularity. In the latter, the cycle length defines the amount of time needed before the changes repeat themselves.

The changing environment problem has been an attractive topic in the EA literature. Some detailed reviews can be found in [5, 18]. Generally speaking, there are three main approaches to date: diversity control, memory-based and multi-population approaches.

Diversity control is a common topic in EA in general. To control diversity in a dynamic problem, one can either increase diversity whenever a change is detected - such as the hyper-mutation method [6], the variable local search technique [27] and other methods in [4] and [20] - or maintain high diversity all over the evolutionary run as in random immigrants [14], Aging [12], and the Thermodynamical Genetic Algorithms [22, 21].

Memory-based techniques employ an extra memory that implicitly or explicitly stores the useful information to guide future search. Implicit memory usually uses redundant representations [8, 9, 13, 15] to store information during the evolutionary process. It can also be used to encrypt algorithmic parameters which need to be tuned up. In explicit memories [21, 16], specific information, which may include solutions, get stored and retrieved when needed by the evolutionary mechanism.

The last approach uses sub-populations to chase the optima in different search areas. Sub-populations are maintained and each becomes specialized on a part of the search space. This facilitates the process of tracking the optima as they move. An example in this group is the self-organizing-scouts method [5] and the multi-national GA [26].

A key issue when testing an algorithm in general is to use a suitable benchmark problem, where the different factors that control the difficulty in solving a problem can be controlled appropriately. A number of authors have suggested different benchmarks such as a class of trap functions in [3] or the moving-peaks problem [5, 16], and a close variant thereof [23].

## 2.2 Evolutionary Multiobjective Optimization

Many real-world problems involve several, usually conflicting objectives. In those cases, there is usually no single optimal solution, but a set of equally good alternatives with different trade-offs. Evolutionary algorithms are particularly suitable to solve such problems, since they maintain a population of solutions and can search for a large number of alternatives simultaneously. Most work in EMO is based on the concept of dominance. A solution  $x$  is said to dominate a solution  $y$  if and only if solution  $x$  is at least as good as  $y$  in all objectives, and strictly better in at least one objective. This concept is then used during selection by favoring non-dominated solutions. For comprehensive books on the topic, the reader is referred to [10, 7]. To date, many different EMOs have been developed, the most prominent include include NSGA2 [10] and SPEA2 [29].

There are a number of approaches applying EMOs to

solve single-objective problems. Knowles and Corne [19] proposed a multi-objective optimization approach in which the primary objective is decomposed into a number of objectives and the optimization process takes place on these objectives. Abbass and Deb [2] proposed to add an artificial objective to promote diversity. Three different artificial objectives were discussed in their work. The first is based on a time-stamp for each chromosome using the generation number. The second is by generating the second fitness at random. The third is by reversing the optimization of the first objective (*i.e.* maximizing the function if the original problem is minimization and vice-versa). In terms of promoting diversity, Toffolo and Benini [24] used diversity explicitly as an objective for EMO problems. They used the sum of the Euclidean distances between an individual and all other individuals in the population as a diversity measure. They also proposed to measure the distance to the closest individual only instead of calculating it between every pair of individuals in the population. More recently, Jensen [17] proposed the concept of helper-objectives to introduce new objectives for the optimization process. He still uses the idea of objective decomposition as Knowles and Corne, but he allows the decomposition to change.

In all of the previous work, the focus was on stationary environments. On the one hand, there are difficulties associated with decomposing the objective into sub-objectives such as the objectives must be decomposable in the first place and how to decide on the number of decompositions. On the other hand, Abbass and Deb had a different focus, where they did not use any mutation in order to study the role of EMO in promoting diversity without having a bias when using a mutation operator. Hence, they were more interested in studying the diversity aspect and did not, in fact, propose a method for solving the single objective optimization problem. With the purpose of solving optimization problems in dynamic environments, Yamasaki [28] introduced a technique to employ the concept of time series as an artificial criteria to define dominance relationship within the context of a simple EMO.

In the current paper, we look at the use of EMO for dynamic environments. We adopt ideas from Abbass and Deb [2] and Toffolo and Benini [24] in defining artificial objectives while, in contrast to their use of a simple EMO algorithm in static environments, we use the NSGA2 algorithm [10, 11] in the changing environment.

## 3 Methods

NSGA2 is one of today's most successful EMO algorithms. It is based on two principles: convergence to the Pareto-optimal front is ensured by non-dominated sorting. This method ranks individuals by iteratively determining the non-dominated solutions in the population (non-dominated front), assigning those individuals the next best rank and removing them from the population. Diversity within one rank is maintained by favoring individuals with a large crowding distance, which is defined as the sum of distances between a solution's neighbors on either side in each dimension of the objective space. Furthermore, NSGA2 is an eli-

tist algorithm [10, 11], i.e. it keeps as many non-dominated solutions as possible (up to the size of the population).

One of the main challenges for using EMOs to solve single objective optimization problems is to define at least the second objective. For the sake of simplicity, we will restrict our discussion to bi-objective problems. The first objective is always taken as the objective of the single optimization problem, while the second objective is an artificial one. There are a number of options to define the artificial objective:

- **Time stamp:** The first artificial objective is a time stamp of when an individual is generated. As in [2], we stamp each individual in the initial population with a different time stamp represented by a counter that gets incremented every time a new individual is created. From the second population, all individuals in the population get the same time stamp that is set to the population size plus the generation index. The time stamp then serves as a second objective to be minimized (i.e., old individuals are favored in the selection).
- **Random:** The second artificial objective is simply to assign each individual a random value as the second objective to be minimized. Some bad individuals may be assigned smaller random values and get a chance to survive; hence they may deem to be useful when the environment changes.
- **Inversion:** The third approach inverts the original objective function by minimizing it if it was a maximization problem, and vice-versa. The underlying idea of this is to slow the selection pressure since it could include a large number of Pareto-optimal solutions at each generation. However, the number of these solutions will be controlled by the limitation of the archive size and niching mechanism.

The last three options are determined based on the Euclidean distance and are all to be maximized:

- **Distance to the closet neighbor (DCN),**
- **Average distance to all individuals (ADI), and the**
- **Distance to the best individual of the population (DBI)**

Note that among these options, DCN and ADI take more time to calculate than the others, because they need to look at  $O(n^2)$  distances. However, by comparing the approaches solely on the basis of function evaluations, this aspect is deliberately ignored here, as the time to calculate distances is assumed to be negligible compared to the time to evaluate an individual.

## 4 Experimental settings

The MPB is a dynamic benchmark problem with a number of peaks changing over time in location, width, and height.

The problem was introduced by Branke [16]. Formally, the benchmark function can be formulated as follows

$$F(\vec{x}, t) = \max(B(\vec{x}), \max P(\vec{x}, h_i(t), w_i(t), \vec{p}_i(t))) \quad (1)$$

in which  $B$  is a time-invariant basis landscape,  $P$  is the peak-shaping function with  $m$  peaks, peak-height  $h$ , peak-width  $w$ , and peak-location  $p$ ,  $i$  ranges from 1 to  $m$ . The details of the problem and its settings can be found in [1]. We selected Branke's second scenario with the basic settings mentioned in Table 1.

Table 1: Parameters for the benchmark problem

Parameters	Values
Number of peaks	50
Number of dimensions	5
Change frequency	25 generations
Peak function	cone

The proposed approaches are tested against the MPB with different values of change severities (low/high). Further, the changes are implemented in both height and width of the peaks to observe their behaviors in different cases of change severities. We fix the distance that the peaks move to 1.

Table 2: Change severities

Height	Width
7	1
7	3
15	1
15	3

NSGA2 uses binary tournament selection, single point crossover and bit-flip mutation. The random immigrants algorithm replaces 20% of the population at each generation with new individuals. The mutation rate of the MutateHigh algorithm is set to 0.5. In order to have a fair comparison, we also employ elitism for all three algorithms.

The behavior of evolutionary methods would normally depend on the crossover and mutation rates used. Therefore, there is a need to examine the different methods with a wide range of parameter values to identify a good setting. The crossover rate  $pc$  is varied between 0.5 and 1 with a step of 0.05 and the mutation rate  $pm$  is varied between 0 and 0.2 with a step of 0.01. For each pair of  $pc$  and  $pm$ , thirty runs are performed with different random seeds, a population size of 100, the number of generations is 1000, and chromosomes are binary encoded.

We record the best individual in each generation as measured on the original single objective function. The difference between the objective value of this individual and the current global optima is known as the *generation error* (GEr). We recorded the generation error and derived the average generation error of each last generation just before a new change occurs (AGEr). Further, the diversity of the population is also recorded overtime. It is calculated as the

average distance between all pairs of individuals in the population.

## 5 Results and discussion

At the first consideration, we try to find the best performance each approach achieved over a wide range of crossover and mutation rates, and change severities. These results are summarized in Tables 3,4,5, and 6. Each table is for one pair of change severities and the result associated with each approach in the table corresponds to the minimal value of the AGER and its associated pair of  $pc$  and  $pm$ . Despite the MutateHigh has a better performance in all cases of change severities than traditional GA and random immigrant, the best overall performance belongs to DCN and ADI, representatives of EMO approaches.

Table 3: The best  $pc$  and  $pm$  for each approach and the AGE  $\pm$  the standard error with change severities of 7.0 and 1.0

The artificial objective function	pc (pm)	AGER
Traditional GA	0.55 (0.04)	11.48 $\pm$ 0.60
Random Immigrants	1.00 (0.01)	11.47 $\pm$ 0.56
MutateHigh	0.75 (0.01)	11.95 $\pm$ 0.59
Time-based objective	0.60 (0.11)	12.06 $\pm$ 0.64
Random objective	0.60 (0.10)	11.29 $\pm$ 0.55
Inverse objective	0.55 (0.06)	12.37 $\pm$ 0.87
DCN	0.75 (0.05)	9.52 $\pm$ 0.45
ADI	0.70 (0.06)	9.74 $\pm$ 0.35
DBI	0.50 (0.09)	12.24 $\pm$ 0.55

Table 4: The best  $pc$  and  $pm$  for each approach and the AGE  $\pm$  the standard error with change severities of 7.0 and 3.0

The artificial objective function	pc(pm)	AGER
Traditional GA	0.55 (0.04)	13.69 $\pm$ 0.75
Random Immigrants	0.65 (0.01)	13.51 $\pm$ 1.06
MutateHigh	0.85 (0.01)	13.78 $\pm$ 0.63
Time-based objective	0.50 (0.11)	12.96 $\pm$ 0.81
Random objective	0.50 (0.08)	12.30 $\pm$ 0.96
Inverse objective	0.50 (0.09)	13.96 $\pm$ 0.87
DCN	0.65 (0.04)	10.42 $\pm$ 0.71
ADI	0.55 (0.04)	9.31 $\pm$ 0.51
DBI	0.80 (0.10)	11.79 $\pm$ 0.71

As an example, in order to have a clearer view, we visualize the values of AGER in Figures 1 and 2 for the averaged generation error AGER with different values of crossover and mutation rates for the best two representatives of EMO and traditional GA approaches: ADI and MutateHigh. Obviously, the absence of mutation deteriorates the quality of solutions. Overall, in all cases of change severities, a good performance was achieved for EMO approaches within an area centered on (0.6,0.07). In this area of the crossover and mutation space, the DCN and ADI approaches produced the best results although they required significantly more computational time. Although the traditional GA and random

Table 5: The best  $pc$  and  $pm$  for each approach and the AGE  $\pm$  the standard error with change severities of 15.0 and 1.0

The artificial objective function	pc(pm)	AGER
Traditional GA	0.55 (0.07)	16.14 $\pm$ 0.71
Random Immigrants	0.55 (0.03)	15.38 $\pm$ 0.81
MutateHigh	0.70 (0.02)	11.96 $\pm$ 0.80
Time-based objective	1.00 (0.10)	12.06 $\pm$ 0.80
Random objective	0.55 (0.10)	14.79 $\pm$ 0.66
Inverse objective	0.50 (0.07)	15.98 $\pm$ 0.89
DCN	0.50 (0.07)	12.68 $\pm$ 0.60
ADI	0.65 (0.06)	13.18 $\pm$ 0.52
DBI	0.60 (0.06)	14.05 $\pm$ 0.61

Table 6: The best  $pc$  and  $pm$  for each approach and the AGE  $\pm$  the standard error with change severities of 15.0 and 3.0

The artificial objective function	pc(pm)	AGER
Traditional GA	0.50 (0.04)	14.79 $\pm$ 0.85
Random Immigrants	0.80 (0.02)	14.67 $\pm$ 0.70
MutateHigh	0.80 (0.02)	12.70 $\pm$ 0.66
Time-based objective	0.55 (0.09)	15.06 $\pm$ 1.00
Random objective	0.65 (0.09)	14.20 $\pm$ 0.83
Inverse objective	0.60 (0.07)	15.28 $\pm$ 0.88
DCN	0.50 (0.06)	12.56 $\pm$ 0.62
ADI	0.50 (0.05)	13.00 $\pm$ 0.63
DBI	0.60 (0.07)	13.96 $\pm$ 0.74

immigrants have quite competitive results in comparison with some of the EMO approaches, they are very sensitive to the mutation rate.

From Tables 3,4,5, and 6, it is interesting to note that the traditional GA approach achieved very similar results as the random immigrants approach. In addition, the small mutation rate for the random immigrants and MutateHigh approaches appears to contribute to the fact that diversity is already maintained through the introduction of new children at random for the random immigrants and the mutation rate for the MutateHigh algorithm. Therefore, the mutation probability did not have much impact on the quality of obtained solutions.

To firm the comparison, we also compare all approaches with the accumulated off-line error in different cases of change severities (See Tables 7,8,9, and 10). The outcome is very consistent with that of the AGER. The EMO approaches, more specifically DCN and ADI, continue to show better performance.

Taking a step further, we investigated the underlying reason for the superior performance of DCN and ADI. In this paper, we hypothesized that diversity plays a key role in solving the dynamic optimization problem. We implemented both explicit and implicit diversity as the second objective function to make the single dynamic problem a bi-objective problem. Further, NSGA2 with crowding distance also provides a good mechanism to preserve diversity in the population. All of these factors are expected to main-

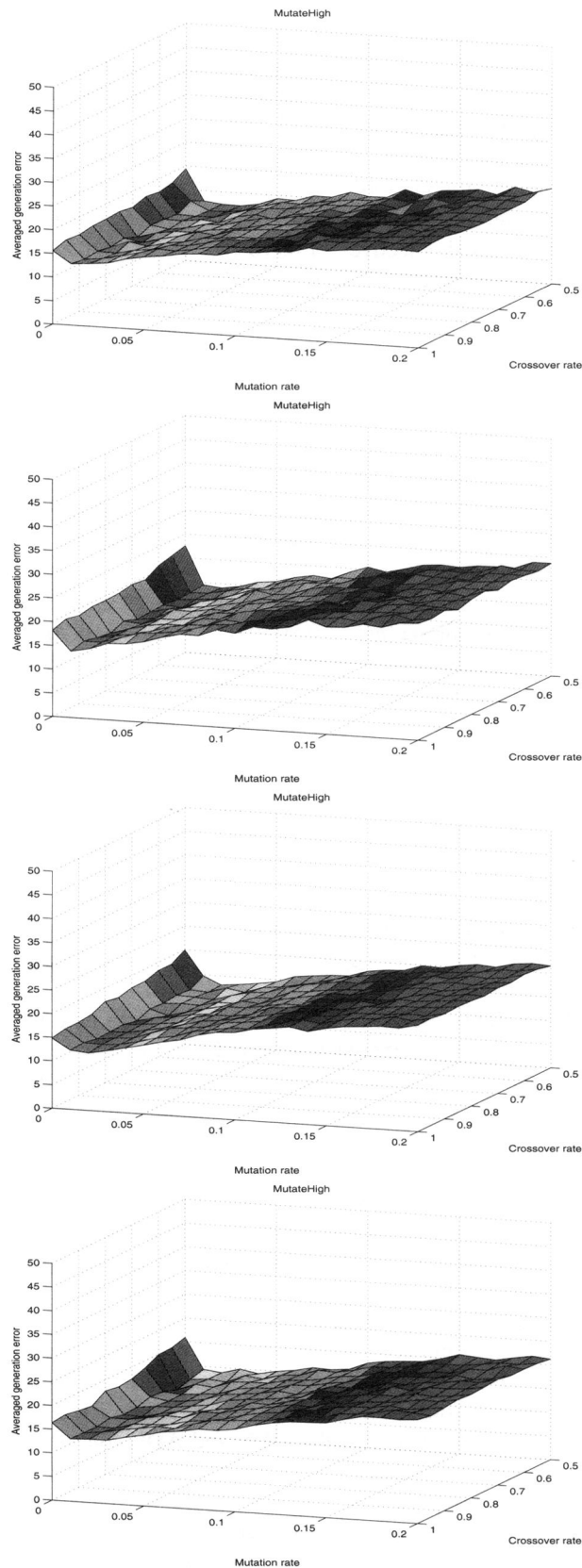


Figure 1: The AGER achieved from the MutateHigh over different ranges of  $pc$  and  $pm$  in four different cases of change severities: (7,1),(7,3),(15,1),(15,3).

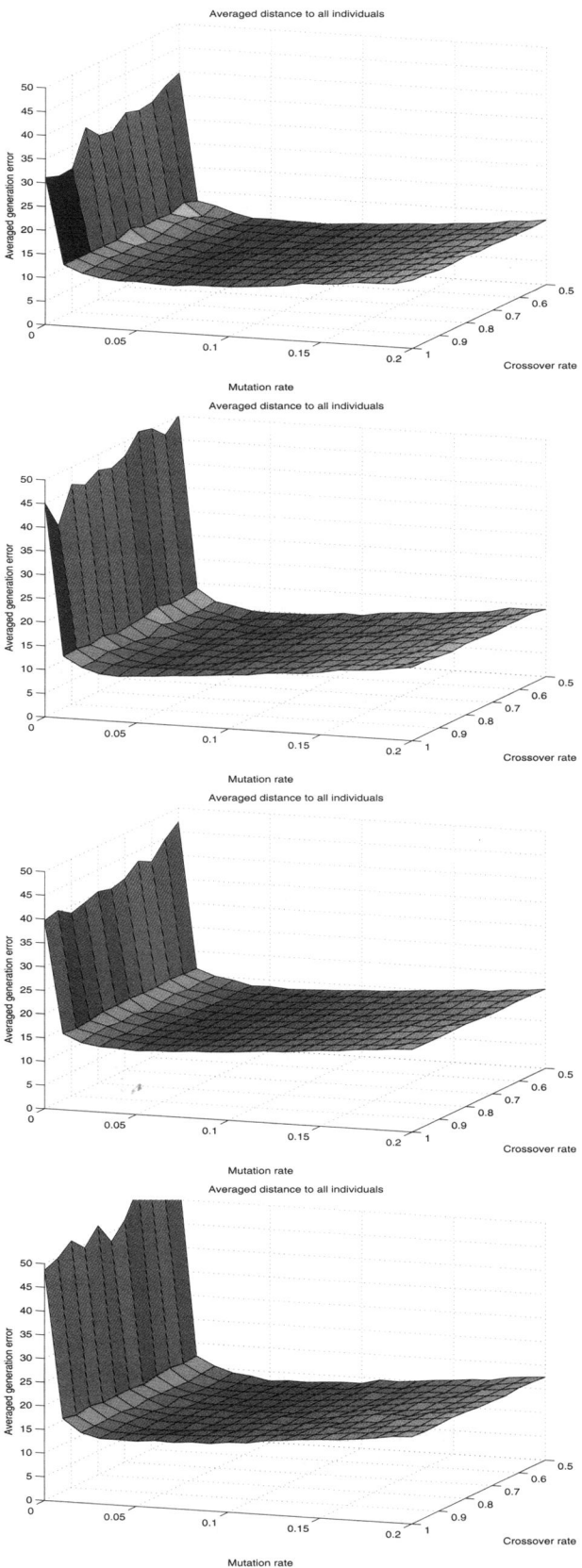


Figure 2: The AGER achieved from ADI over different ranges of  $pc$  and  $pm$  in four different cases of change severities: (7,1),(7,3),(15,1),(15,3).

Table 7: The best *pc* and *pm* for each approach and the off-line error with change severities of 7.0 and 1.0

The artificial objective function	pc(pm)	Off-line error
Traditional GA	0.55 (0.04)	11.05
Random Immigrants	1.00 (0.01)	11.21
MutateHigh	0.75 (0.01)	10.53
Time-based objective	0.60 (0.11)	11.19
Random objective	0.60 (0.10)	10.43
Inverse objective	0.55 (0.06)	11.90
DCN	0.75 (0.05)	8.87
ADI	0.70 (0.06)	8.97
DBI	0.50 (0.09)	11.58

Table 8: The best *pc* and *pm* for each approach and the off-line error with change severities of 7.0 and 3.0

The artificial objective function	pc(pm)	Off-line error
Traditional GA	0.55 (0.04)	15.23
Random Immigrants	0.65 (0.01)	14.48
MutateHigh	0.85 (0.01)	13.27
Time-based objective	0.50 (0.11)	12.53
Random objective	0.50 (0.08)	12.02
Inverse objective	0.50 (0.09)	14.44
DCN	0.65 (0.04)	10.83
ADI	0.55 (0.04)	9.13
DBI	0.80 (0.10)	11.18

Table 9: The best *pc* and *pm* for each approach and the off-line error with change severities of 15.0 and 1.0

The artificial objective function	pc(pm)	Off-line error
Traditional GA	0.55 (0.07)	16.68
Random Immigrants	0.55 (0.03)	16.39
MutateHigh	0.70 (0.02)	14.50
Time-based objective	1.00 (0.10)	14.02
Random objective	0.55 (0.10)	14.58
Inverse objective	0.50 (0.07)	16.87
DCN	0.50 (0.07)	13.37
ADI	0.65 (0.06)	12.61
DBI	0.60 (0.06)	13.80

tain the diversity of the population over time. Therefore, it helps EMO approaches to overcome the difficulty of the environmental changes and track the optimum. That probably explains the reason for the superior performance of EMO approaches, especially, DCN and ADI, when compared to the random immigrants and MutateHigh algorithms.

To support this hypothesis, we measure the diversity of the population for each approach over time (See Figures 3, 4). The outcome shows that in all cases of change severities, EMO approaches maintain better diversity when compared with the traditional EA and MutateHigh. The case of MutateHigh indicates the obvious effect of the high mutation rate overtime. The diversity of the population reduces quickly, but when the change happens, it increases

Table 10: The best *pc* and *pm* for each approach and the off-line error with change severities of 15.0 and 3.0

The artificial objective function	pc(pm)	Off-line error
Traditional GA	0.50 (0.04)	17.43
Random Immigrants	0.80 (0.02)	17.64
MutateHigh	0.80 (0.02)	15.44
Time-based objective	0.55 (0.09)	15.38
Random objective	0.65 (0.09)	14.32
Inverse objective	0.60 (0.07)	17.18
DCN	0.50 (0.06)	14.74
ADI	0.50 (0.05)	13.38
DBI	0.60 (0.07)	15.03

drastically, as a result of employing the high mutation rate. Random immigrants also maintains a rather high level of diversity, but as the previous results show, is not really able to use it properly. It seems that the diversity introduced by the EMO approach is more useful than just that diversity generated randomly.

In summary, the above results show that integrating diversity into EMO proposes a promising solution to solve the dynamic optimization problem. The diversity-based objective helps the EMO approach to outperform the random immigrant and MutateHigh algorithms.

## 6 Conclusion

In this paper, we examined the feasibility of applying EMOs in dynamic environments. In order to do that, we used NSGA2 as the evolutionary algorithm and the moving-peaks problem as the testbed. The traditional GA, random immigrants, MutateHigh are chosen to compare their performance against six proposed EMO-based approaches. The criteria to establish the artificial objective are time-based, random, objective reverse, distance to the closet neighbor, the averaged distance to all other individuals, and the distance to the best individual. The experiments demonstrated that the EMO approaches are competitive. Within the EMO approaches, the average distance to all other individuals, and the distance to the best individual options have much better performance in comparison with the others. In future, a work need to be carried out to test the method further in such dynamic conditions as changing with different levels of the shift length and scale it to cope with large number of variables.

## Acknowledgment

This work is supported by the University of New South Wales grant PS04411 and the Australian Research Council (ARC) Centre on Complex Systems grant number CEO0348249.

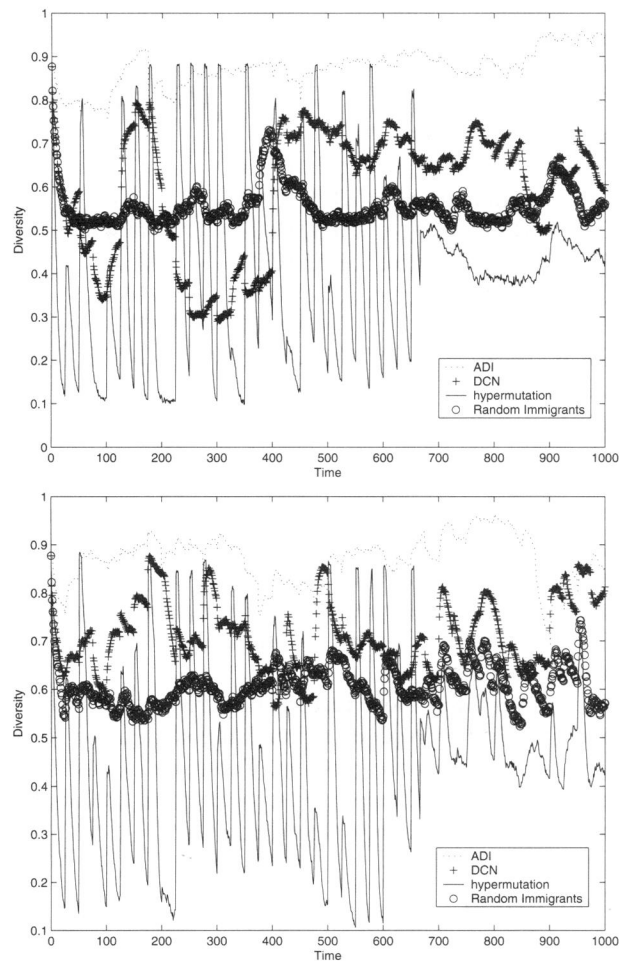


Figure 3: The diversity of the population over time with change severities of (7.0,1.0) and (7.0,3.0)

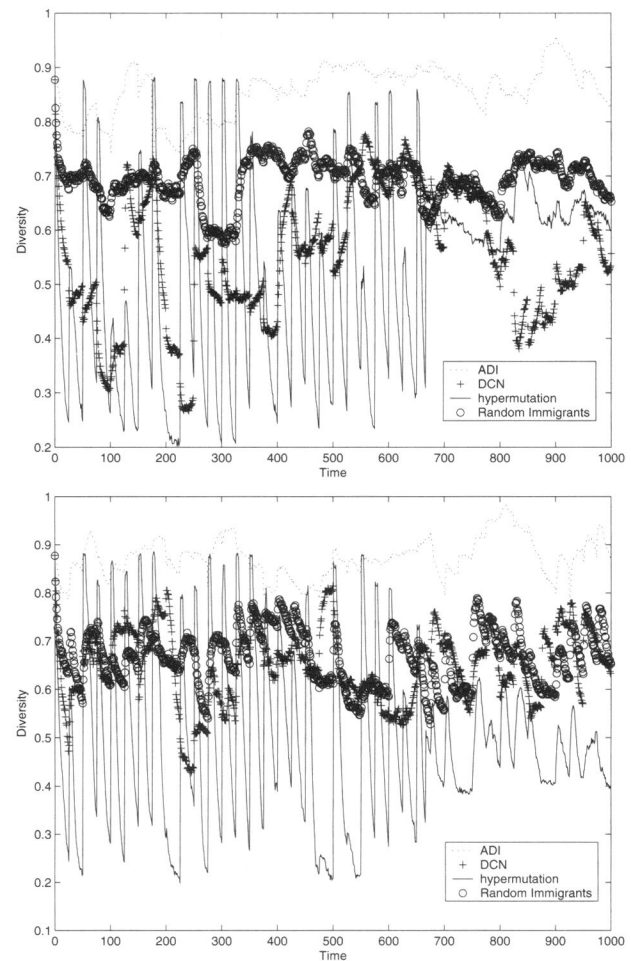


Figure 4: The diversity of the population over time with change severities of (15.0,1.0) and (15.0,3.0)

## Bibliography

- [1] Online. [http://www.aifb.uni-  
karlsruhe.de/~jbr/MovPeaks](http://www.aifb.uni-<br/>karlsruhe.de/~jbr/MovPeaks).
- [2] H. A. Abbass and K. Deb. Searching under multi-evolutionary pressures. In Zitzler et al., editor, *Proceedings of the Fourth Conference on Evolutionary Multi-Criterion Optimization*, Spain, 2003.
- [3] H. A. Abbass, K. Satry, and D. Goldberg. Oiling the wheels of change: The role of adaptive automatic problem decomposition in non-stationary environments. Technical report, IlliGAL, University of Illinois at Urbana-Champaign, 2004.
- [4] C. Bierwirth and D.C. Mattfeld. Production scheduling and rescheduling with genetic algorithms. *Evolutionary Computation*, 7(1):1–18, 1999.
- [5] J. Branke. *Evolutionary optimization in dynamic environments*. Kluwer Academic Publishers, Massachusetts USA, 2002.
- [6] H.G. Cobb. An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments. Technical Report AIC-90-001, Naval Research Laboratory, 1990.
- [7] C. A. C. Coello, D. A. V. Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York USA, 2002.
- [8] D. Dasgupta. Incorporating redundancy and gene activation mechanisms in genetic search. In L. Chambers, editor, *Practical Handbook of Genetic Algorithms*, pages 303–316. CRC Press, 1995.
- [9] D. Dasgupta and D.R. McGregor. Nonstationary function optimization using the structured genetic algorithm. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature*, pages 145–154. Elsevier Science Publisher, 1992.
- [10] K. Deb. *Multiobjective Optimization using Evolutionary Algorithms*. John Wiley and Son Ltd, New York, 2001.
- [11] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm:



- NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- [12] A. Ghosh, S. Tstutsui, and H. Tanaka. Function optimisation in nonstationary environment using steady state genetic algorithms with aging of individuals. In *IEEE International Conference on Evolutionary Computation*, pages 666–671. IEEE Publishing, 1998.
- [13] D.E. Goldberg and R.E. Smith. Nonstationary function optimisation using genetic algorithms with dominance and diploidy. In J.J. Grefenstette, editor, *Second International Conference on Genetic Algorithms*, pages 59–68. Lawrence Erlbaum Associates, 1987.
- [14] J.J. Grefenstette. Genetic algorithms for changing environments. In R.Männer and B.Manderick, editors, *Parallel Problem Solving from Nature*, pages 137–144. Elsevier Science Publisher, 1992.
- [15] B.S. Hadad and C.F. Eick. Supporting polyploidy in genetic algorithms using dominance vectors. In *International Conference on Evolutionary Programming*, volume 1213 of *Lecture Notes in Computer Science*, pages 223–234, 1997.
- [16] J.Branke. Memory enhanced evolutionary algorithms for changing optimisation problems. In *In Congress on Evolutionary Computation CEC99*, pages 1875–1882. IEEE, 1999.
- [17] M.T. Jensen. Helper-objectives: Using multiobjective evolutionary algorithms for single-objective optimization. *Journal of Mathematical Modelling and Algorithms*, 1(25), 2004.
- [18] Y. Jin and J. Branke. Evolutionary optimization in uncertain environments – a survey. *IEEE Transactions on Evolutionary Computation*, to appear.
- [19] J. Knowles and D. Corne. Reducing local optima in single objective problems by multi-objectivization. In Zitzler et al., editor, *Proceedings of the First Conference on Evolutionary Multi-Criterion Optimization*, Zurich Switzerland, 2001.
- [20] S.C. Lin, E.D. Goodman, and W.F. Punch. A genetic algorithm approach to dynamic job shop scheduling problems. In *Seventh International Conference on Genetic Algorithms*, pages 139–148. Morgan Kaufmann, 1997.
- [21] N. Mori, S. Imanishia, H. Kita, and Y. Nishikawa. Adaptation to changing environments by means of the memory based thermodynamical genetic algorithms. In T.Bäck, editor, *Seventh International Conference on Genetic Algorithms*, pages 299–306. Morgan Kaufmann, 1997.
- [22] N. Mori, H. Kita, and Y. Nishikawa. Adaptation to changing environments by means of the thermodynamical genetic algorithms. In H.-M. Voigt, editor, *Parallel Problem Solving from Nature*, volume 1411 of *Lecture Notes in Computer Science*, pages 513–522, Berlin, 1996. Elsevier Science Publisher.
- [23] R.W. Morrison and K. A. DeJong. A test problem generator for non-stationary environments. In *Proceedings of 1999 Congress on Evolutionary Computation*, 1999.
- [24] A. Toffolo and E. Benini. Genetic diversity as an objective in multi-objective evolutionary algorithms. *Evolutionary Computation*, 11(2):151–168, 2003.
- [25] K. Trojanowski and Z. Michalewicz. Evolutionary algorithms for non-stationary environments. In *Proc. of 8th Workshop: Intelligent Information systems*, pages 229–240, Portland, 1999. ICS PAS Press.
- [26] R.K. Ursem. multinational GAs: multimodal optimization techniques in dynamic environments. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, pages 19–26. San Francisco, CA: Morgan Kaufmann, 2000.
- [27] F. Vavak, K. Jukes, and T.C. Fogarty. Learning the local search range for genetic optimisation in nonstationary environments. In *IEEE International Conference on Evolutionary Computation*, pages 355–360. IEEE Publishing, 1997.
- [28] K. Yamasaki. Dynamic pareto optimum GA against the changing environments. In J. Branke and T. Bäck, editors, *Evolutionary Algorithms for Dynamic Optimization Problems*, pages 47–50, San Francisco, California, USA, 7 2001.
- [29] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm. Technical report, Swiss Federal Institute of Technology, 2001.