

Genetic Algorithms with Self-Organized Criticality for Dynamic Optimization Problems

Renato Tinós

Departamento de Física e Matemática, FFCLRP
Universidade de São Paulo (USP)
14040-901, Ribeirão Preto, SP, Brazil
rtinos@ffclrp.usp.br

Shengxiang Yang

Department of Computer Science
University of Leicester
University Road, Leicester LE1 7RH, United Kingdom
s.yang@mcs.le.ac.uk

Abstract- This paper proposes a genetic algorithm (GA) with random immigrants for dynamic optimization problems where the worst individual and its neighbours are replaced every generation. In this GA, the individuals interact with each other and, when their fitness is close, as in the case where the diversity level is low, one single replacement can affect a large number of individuals. This simple approach can take the system to a kind of self-organization behavior, known as Self-Organized Criticality (SOC), which is useful to maintain the diversity of the population in dynamic environments and hence allows the GA to escape from local optima when the problem changes. The experimental results show that the proposed GA presents the phenomenon of SOC.

1 Introduction

The research in genetic algorithms (GAs) has been mainly focused on stationary optimization problems, in spite of a significant part of optimization problems in real world being dynamic optimization problems (DOPs) [4]. In DOPs, the evaluation function (or fitness function) and the constraints of the problem are not fixed [20]. When changes occur, the solution given by the optimization procedure may be no longer effective, and a new solution should be found.

The optimization problem can change by several factors, like faults, machine degradation, environmental or climatic modifications, and economic factors. In fact, the natural evolution, which is the inspiration for GAs, is always non-stationary. The occurrence of natural cataclysms, geological modifications, competition for natural resources, coevolution between species, and climatic modifications are only some examples of changes related to natural evolution.

The simplest approach to deal with DOPs is to start a new optimization process whenever a change in the problem is noticed. However, the optimization process generally requires time and substantial computational effort. If the new solution after the change in the problem is, in some sense, related to the previous solution, the knowledge obtained during the search for the old solution can be utilized to find the new solution [4]. In this case, the search for new solutions based on the old solutions can save substantial processing time. Evolutionary algorithms are particularly attractive to such problems. Individuals representing solutions of the problem before the changes can be transferred into the new optimization process.

However, in GAs, the population of solutions generally converges in the fitness landscape to points close to the best individual of the population. If the fitness landscape changes, the actual population can be trapped in local optima located close to the old solution. In fact, the premature convergence of the solution to a local optima is not a problem exclusive to DOPs, but it can be a serious problem in stationary optimization problems too [17]. In order to avoid the premature convergence, several approaches where the diversity level is re-introduced or maintained throughout the run have appeared in literature over the past years [4, 5, 12, 20]. Typical examples of this solution are the GAs with random immigrants and the use of hypermutation [6].

The random immigrants approach, which is inspired in the flux of immigrants that wander in and out of a population between two generations in nature, is very interesting and simple [6]. In the GAs with random immigrants, some individuals of the current population are substituted by random individuals in each generation of the run. A replacement strategy, like replacing random or worst individuals of the population, defines which individuals are replaced by new ones [21].

However, in some cases, when the number of genes in the individual (parameters in the solution) is high and the local optimum where the population is found has fitness much higher than the mean fitness of all possible solutions of the search space, the survival probability of the new random individuals is generally very small. This occurs because the selection methods employed in GAs preserve, directly or indirectly, the best individuals of the population, and the probability that the fitness of the new random individuals is higher than (or close to) the fitness of the current individuals is generally small.

In this paper, instead of substituting the worst individuals or the random individuals in each generation like in the standard random immigrants approach, the worst individual and its next neighbours are replaced. In this way, individuals start to interact between themselves and, when the fitness of the individuals are close, as in the case where the diversity level is small, one single replacement of an individual can affect a great number of individuals of the population in a chain reaction. In order to protect the newly introduced immigrants from being replaced by fitter individuals, they are placed in a subpopulation and are not allowed to be replaced by individuals of the main population. The number of individuals in the subpopulation is not defined by the programmer, but is given by the number of individuals created

in the chain reaction. It is important to observe that this simple approach can take the system to a self-organization behavior, which can be useful in DOPs.

The experimental results suggest that the proposed GA presents a kind of self-organizing behavior, known as Self-Organized Criticality (SOC) [1], which is described in this paper in Section 2. The proposed GA is presented in Section 3, and the results of the experiments with DOPs are presented in Section 4. Finally, Section 5 concludes the paper with discussions on relevant future work.

2 Self-Organized Criticality

Bak, Tang, and Wiesenfeld suggested in [2] that systems consisting of several interacting constituents may present an interesting kind of self-organizing behavior [13]. The authors described this interesting behavior as SOC. It was suggested that several diverse phenomena exhibit SOC, like sand piles, earthquakes, forest fires, electric breakdowns, and growing interfaces.

The seductive characteristic of systems that exhibit SOC is that they self-organize into a particular critical state without the need of any significant tuning action from outside. The critical state is described by the response of a system to external perturbation. In a system exhibiting noncritical behavior, the distribution of responses to perturbation at different positions and at different times is narrow and well described by an averaged value. In a system exhibiting critical behavior, no single characteristic response exists, i.e. the system exhibits scale invariance. A small perturbation in one given location of the system may generate a small effect on its neighbourhood or a chain reaction that affects all the constituents of the system.

The statistical distributions describing the response of the system exhibiting SOC are given by power laws in the form

$$P(s) \sim s^{-\tau} \quad (1)$$

and

$$P(d) \sim d^{-\alpha} \quad (2)$$

where s is the number of constituents of the system affected by the perturbation, d is the duration of the chain reaction (lifetime), and τ and α are constants. As an example, consider the sand pile model described in [2], where a single grain is added at a random position in every interval of time Δt . In order to characterize the response of the system, one can measure the number of sand grains (s) involved in each avalanche induced by the addition of a single grain and the duration (d) of each avalanche. In the critical state, the statistical distributions describing the response of the sand pile model to the addition of a single grain are given by Eqs. 1 and 2, and the addition of a single grain can affect only a grain in its neighbourhood or can affect the whole sand pile.

Researchers have suggested that SOC occurs in natural evolution too [1]. An evidence of SOC in evolution would be the fact that it does take place through bursts of activity intercalated by calm periods, instead of gradually at a

slow and constant pace. There are many more small extinction events than large events, such as the Cretaceous extinction of dinosaurs and many other species, and extinction events occur on a large variety of length scales [19]. These facts suggested that extinctions propagate through ecosystems, such as avalanches in a sand pile, and perturbations of the same size can unleash extinction events of a large variety of sizes. This would occur because species co-evolve to a critical state [14].

Bak and Sneppen [1] proposed a very simple simulation model to study the connection between evolution and SOC. In the one-dimensional version of the model, the individuals (or species in the authors' terminology) are disposed in a circle, and a random value of fitness is assigned to each one of them. In each generation of the simulation, the values of fitness of the individual with the smallest fitness in the current population, one individual located in its right position, and one located in its left position are replaced by new random values. An analogy of the connection between neighbours in the model is the interaction between species in nature. If, as an example, a prey is extinct, the fitness of its predators will change. The Bak-Sneppen Model can be summarized by Algorithm 1.

Algorithm 1 Bak-Sneppen Model

- 1: Find the index j of the individual with the smallest fitness
 - 2: Replace the fitness of the individuals with index j , $j - 1$, and $j + 1$ by random values drawn with uniform density
-

This simple model can lead to an interesting behavior. In the beginning of the simulation, the mean fitness of the population is small, but, as the number of generation increases, the mean fitness increases too. Eventually, the mean fitness ceases to increase, and the critical state is reached. In the Bak-Sneppen Model, a substitution of the fitness of the worst individual causes the substitution of its two next neighbours. In the critical state, the values of fitness of the neighbours are very often replaced by random numbers with smaller values. The new worst individual can be then one of these two neighbours, which are substituted with its two next neighbours, originating a chain reaction, called extinction event in this paper, that can affect all the individuals of the population. The extinction events exhibit scale invariance and their statistical distributions are given by power laws in the form of Eqs. 1 and 2. Large extinction events generally occur when almost all individuals of the population have similar high values of fitness.

It is important to observe that SOC avoids the situation where the species get trapped in local optima in the fitness landscape in this evolution model. The idea is powerful and relatively simple, and soon researchers proposed the use of SOC in optimization processes. Boettcher and Percus [3] proposed the optimization with extremal dynamics, a local-search heuristic for finding solutions in problems where constituents of the system are connected, e.g. the spin glass optimization problem. Løvbjerg and Krink [16] extended Particle Swarm Optimization with SOC in order to help the control of the optimization process and to maintain the diversity level.

In GAs, Krink and Thomsen [15] proposed the use of the sand pile model previously discussed to generate power laws utilized to control the size of spatial extinction zones in a diffusion model. When an individual is extinct, a mutated version of the best individual of the population is created in its place. It is important to observe that, in the algorithm proposed in [15], SOC appears in the sand pile model utilized to control the size of the extinctions, and not as a result of the self-organization of the constituents of the system (individuals of the GA).

3 Proposed Algorithm

In this paper, we propose the substitution of the worst individual and its two next neighbours for new random individuals in the random immigrants approach. The indexes of the individuals are used to determine the neighbourhood relations. In each generation of the algorithm, the individual with the smallest fitness in the current population (index j), one individual located in its right position (index $j + 1$), and one located in its left position (index $j - 1$) are replaced by new random individuals. We hope that, with this replacement strategy, the system can exhibit SOC in order to increase the diversity level of the population in a self-organized way and, then, to avoid the situation where the individuals get trapped in local optima in the fitness landscape when the problem changes.

However, this simple idea does not guarantee that the system exhibits SOC as the new random individuals added to the current population, which generally have small values of fitness, are very often substituted by individuals with high values of fitness present in the population. As a consequence, the statistical distribution describing the response of the system to a single extinction will not be a power law, but a narrow one characterized by an averaged value.

In order to protect the newly introduced individuals from being replaced by individuals with high values of fitness, a second strategy should be adopted. In this strategy, the new individuals created during an extinction event are preserved in a subpopulation, which is not defined by the programmer, but is given by the number of individuals created in the current extinction event. The individuals in the current population that do not belong to the subpopulation are not allowed to replace individuals present in the subpopulation. The individuals that belong to the subpopulation are allowed to evolve, i.e. they are submitted to mutation, crossover, and selection. It is important to observe that selection and crossover are allowed only between individuals that belong to the subpopulation.

In the proposed algorithm, there are two major modifications in the standard GA. In the first modification, which is presented in Algorithm 2, the current size of each extinction event, denoted by ext , is recorded, and the minimum and maximum index values of the replaced individuals ($i_{min} - 1$ and $i_{max} + 1$) are utilized to compute the number of individuals affected by the current extinction, i.e. the individuals that belong to the subpopulation. When the chain reaction ceases, i.e. the individual with the smallest fitness do not belong to the subpopulation or is not neighbour of the min-

imum and maximum index values of the replaced individuals, the size of the extinction is set to 1.

Algorithm 2 Extinctions

```

1: Find the index  $j$  of the individual with the smallest fitness
2: Replace the individuals with index  $j$ ,  $j - 1$ , and  $j + 1$  by
   random individuals
3: if ( $j \geq i_{min} - 1$ ) and ( $j \leq i_{max} + 1$ ) then
4:    $ext \leftarrow ext + 1$ 
5:   if ( $j = i_{min} - 1$ ) then
6:      $i_{min} \leftarrow j$ 
7:   end if
8:   if ( $j = i_{max} + 1$ ) then
9:      $i_{max} \leftarrow j$ 
10:  end if
11: else
12:   $ext \leftarrow 1$ 
13:   $i_{min} \leftarrow j$ 
14:   $i_{max} \leftarrow j$ 
15: end if

```

The second modification, which is presented in Algorithm 3, lies in the selection approach for each individual in the new population. Two cases can occur. In the first case, when the index of the new individual was not affected by the current extinction event, the new individual is selected in the standard approach, i.e. the individual is chosen from the whole population according to a given criteria. Otherwise, i.e. if the index of the new individual was affected by the current extinction, the new individual is selected from the subpopulation formed by the individuals replaced in the current extinction (individuals with index values from $i_{min} - 1$ to $i_{max} + 1$).

Algorithm 3 Selection

```

Require: individual's index  $i$ , where  $1 \leq i \leq N$  and  $N$  is the
population size
1: if ( $i < i_{min} - 1$ ) or ( $i > i_{max} + 1$ ) then
2:   Select individual from the whole population
3: else
4:   Select individual from the subpopulation composed by the
   individuals with index from  $i_{min} - 1$  to  $i_{max} + 1$ 
5: end if

```

4 Experimental Studies

In order to evaluate the performance of proposed algorithm, two sets of experiments are carried out. In the first set of experiments, the dynamic test environment for GAs proposed by Yang [22] is employed. In the second set of experiments, evolutionary robots are simulated in dynamic environments. In the experiments, the proposed algorithm is compared to the standard GA, and to two versions of the GA with random immigrants. In the first version, three individuals randomly chosen are replaced by new random individuals. In the second version, the three worst individuals, i.e. the individuals with the smallest fitness, are replaced by new random individuals.

4.1 Dynamic Test Environments

In order to evaluate the performance of different GAs in DOPs, Yang [22] proposed an environment generator based on unitation and trap functions. The unitation function $u(\mathbf{x})$ of a binary vector \mathbf{x} of length l is given by the number of ones in this vector. A trap function is defined as follows

$$f(\mathbf{x}) = \begin{cases} \frac{a}{z}(z - u(\mathbf{x})) & \text{if } u(\mathbf{x}) \leq z \\ \frac{b}{l-z}(u(\mathbf{x}) - z) & \text{otherwise} \end{cases} \quad (3)$$

where a is the local and possibly deceptive optimum, b is the global optimum, and z is the slope-change location which separates the attraction basin sizes of the two optima. A trap function can be a deceptive function for GAs, i.e. a function where there exist low-order schemata that, instead of combining to form high-order schemata, forms schemata resulting in a deceptive solution that is sub-optimal [10]. A trap function is deceptive on average if the ratio of the fitness of the local optimum to that of the global optimum is constrained by the following relation [8].

$$r = \frac{a}{b} \geq \frac{2 - 1/(l - z)}{2 - 1/z} \quad (4)$$

Deception is not the only element that can generate difficulty to a GA. The problem difficulty can also be caused by exogenous noise and scaling, which arises in functions that consist of several schemata with different worth to the solution [11]. A scaling problem can be simulated using additively decomposable functions as follows

$$f(\mathbf{x}) = \sum_{i=1}^m c_i f_i(\mathbf{x}_{I_i}) \quad (5)$$

where m is the number of schemata that are juxtaposed and summed together, I_i is the set of the fixed bit positions that form schema i , and c_i is the scaling factor for each sub-function f_i .

Using Equations 3 and 5, it is possible to create different dynamic environments where the problem difficulty can be adjusted. In this paper, dynamic environments where the deception difficulty is modified by changing the peak heights of optima are employed [22]. In these dynamic environments, the fitness of an individual \mathbf{x} is given by additively decomposable trap functions defined as follows

$$f(\mathbf{x}) = \sum_{i=1}^m c_i f_i(\mathbf{x}_{I_i}, t) \quad (6)$$

$$f(\mathbf{x}_{I_i}, t) = \begin{cases} \frac{a_i(t)}{z_i}(z_i - u(\mathbf{x}_{I_i})) & \text{if } u(\mathbf{x}_{I_i}) \leq z_i \\ \frac{b_i}{l_i - z_i}(u(\mathbf{x}_{I_i}) - z_i) & \text{otherwise} \end{cases} \quad (7)$$

where: $i = 1, 2, 3$; $\mathbf{x}^T = [\mathbf{x}_{I_1}^T \ \mathbf{x}_{I_2}^T \ \mathbf{x}_{I_3}^T]$; $\mathbf{x}_{I_1} = [x_1 \cdots x_6]^T$; $\mathbf{x}_{I_2} = [x_7 \cdots x_{12}]^T$; $\mathbf{x}_{I_3} = [x_{13} \cdots x_{18}]^T$; $l_i=6$; $b_i = b=1.0$; $z_i = z$; the scaling is given by $c_i = 2^{i-1}$; a_i switches between $A_{min} > 0$ and $A_{max} > b_i$ in every 2000 generations. The parameter A_{min} is constrained by Eq. 4, i.e. it is chosen in order that the trap functions are deceptive on average. In this way, in every 2000 generations, the

global optimum changes between b and $a_i = A_{max}$, and the problem changes between deceptive and non-deceptive. Two environments are utilized to generate the results for all GAs. In the first (Environment 1), $z = 5$, $A_{min} = 0.6$, and $A_{max} = 1.4$. In the second (Environment 2), $z = 4$, $A_{min} = 0.9$, and $A_{max} = 1.9$.

4.1.1 Experimental Design

For each run of an algorithm in a dynamic environment, the individuals of the initial population of the algorithm are randomly chosen. The individuals are selected in each generation according to elitism and the roulette wheel method. The two-point crossover is utilized. For all algorithms, 20000 generations are executed with the number of individuals in the population equal to 100.

The comparison of the results obtained from different algorithms on DOPs is more complex than the same comparison for stationary problems [20]. For DOPs, it is necessary to evaluate not the final result, but rather the optimization process itself. Here, the measure *Adaptability*, proposed in [20] and based on a measure proposed by De Jong [7], is utilized to evaluate the GAs. Adaptability is computed as the difference, averaged over the entire run, between the fitness of the current best individual of each generation and the corresponding optimum value. The best results for the Adaptability measure are those with the smallest values.

4.1.2 Experimental Results

Table 1 presents the measure Adaptability and the mean fitness of all individuals of the population averaged over 20 trials for environments 1 and 2. The results of three experiments of environment 1 with different values for the crossover rate (p_c) and mutation rate (p_m) are presented. The values of p_c and p_m are indicated in Table 1. The percentage inside the parentheses indicates the sum, over all generations, of the difference between the current value of the maximum fitness and the fitness of the current best individual.

Hypothesis tests, considering the Student's t-distribution, with a level of significance equal to 0.01 indicate that the measure Adaptability is smaller for the proposed GA in these experiments.

4.2 Evolutionary Robotics

Robots in which artificial evolution is used as a fundamental form of adaptation or design are known as evolutionary robots [18]. In the experiments presented in this section, mobile robots are simulated in DOPs using a modified version of the Evorobot simulator developed by S. Nolfi [18]. In the simulator utilized in the experiments presented in this section, the robots are controlled by a recurrent artificial neural network (Elman Network) with synaptic weights adjusted by GAs.

The experiments presented here are inspired in the experiment proposed by Floreano and Mondada [9], where a Khepera robot with 8 infrared distance sensors (six in one side and two in another side of the robot), 2 ambient light

Table 1: Experimental Results of Algorithms in Dynamic Test Environments

GA		Environment 1 ($p_c = 0.2, p_m = 0.01$)	Environment 1 ($p_c = 0.7, p_m = 0.01$)	Environment 1 ($p_c = 0.2, p_m = 0.001$)	Environment 2 ($p_c = 0.2, p_m = 0.01$)
Adaptability (best individual)	Standard	0.19454 (16.21 %)	0.19575 (16.31 %)	0.20010 (16.67 %)	0.18764 (12.94 %)
	Random immigrants	0.03559 (2.97 %)	0.02808 (2.34 %)	0.04582 (3.82 %)	0.02320 (1.60 %)
	Random immigrants (worst)	0.03946 (3.29 %)	0.02761 (2.30 %)	0.05689 (4.74 %)	0.02579 (1.78 %)
	Proposed algorithm	0.02071 (1.73 %)	0.01168 (0.97 %)	0.02591 (2.16 %)	0.01362 (0.94 %)
Mean Fitness (all individuals)	Standard	0.9265	0.9222	0.9899	1.1659
	Random immigrants	1.0582	1.0514	1.1222	1.2993
	Random immigrants (worst)	1.0719	1.0729	1.1258	1.3207
	Proposed algorithm	1.0339	1.0202	1.0978	1.2585

sensors, and 1 floor brightness sensor should navigate in an arena of 40×45 cm. The robot has a measurable limited energy, which is recharged every time the robot crosses a battery recharge area. The battery recharge area is indicated by a different color of the floor and by a light source mounted in a tower inside the area.

4.2.1 Experimental Design

In the experiments presented in this section, the fitness function is given by the accumulated average rotation speed of the two wheels of the robot during its life time, i.e. while the battery has energy and while it does not crash into a wall or an obstacle, considering a maximum limit of 60 seconds (a fully charged battery allows the robot to move for 20 seconds). The fitness is not computed while the robot remains in the battery recharge area. Although the fitness function does not specify that the robot should return to the battery recharge area, the individuals that develop the ability to find it and to periodically return to the battery recharge area while exploring the arena without hitting the obstacles accumulate more fitness. The neural network utilized to control the robots has 17 inputs (8 infrared sensors, 2 ambient light sensors, 1 floor brightness sensor, 1 sensor for the battery energy, and 5 recurrent units), 5 hidden neurons, and 2 outputs (2 motors).

Experiments with 1200 generations each are presented in this section. In the experiments, the environment where the robot is evolving is changed after each 300 generations. Environment changing frequently occurs in real problems, where some aspects of the environment are frequently modified. Besides, robots are frequently evolved in simulations to avoid damage, and, when a satisfactory behavior is reached, the neural networks utilized to control are transferred to the real robots. In the experiments, four different environments, where the locations of the battery recharge area and internal walls are different, are considered.

In the runs, the individuals of the initial population are randomly chosen. The individuals are represented by a vector of 119 real values. In each generation of the GAs, the 20 best individuals are selected and each one generates 5 children ($N=100$). In both experiments, p_m is equal to 0.01 and crossover is not utilized.

4.2.2 Results

Table 2 presents the adaptability (supposing a maximum fitness equal to 1.0) and the mean fitness of all individuals of the population averaged over 20 trials. Most of the times,

Table 2: Results - Evolutionary Robotics

GA		Experiment 1
Adaptability (best individual)	Standard	0.40391 (40.39 %)
	Random immigrants	0.39226 (39.23 %)
	Random immigrants (worst)	0.41860 (41.86 %)
	Proposed algorithm	0.34351 (34.35 %)
Mean Fitness (all individuals)	Standard	0.22700
	Random immigrants	0.23038
	Random immigrants (worst)	0.22360
	Proposed algorithm	0.26492

a new solution is found, allowing the robot to navigate in the environment and to return to the battery recharge area only when the battery level is low. When the environment changes, the fitness values of the robots become small, and a new solution is searched.

Hypothesis tests, considering the Student's t -distribution, with a level of significance equal to 0.07 indicate that the measure Adaptability is smaller for the proposed GA in the evolutionary robotics experiment.

4.3 Analysis of the Results

In the experiments presented here, the mean values of adaptability for the three GAs with random immigrants (including the proposed GA) are smaller than the mean values for the standard GA, indicating that the average fitness of the best individuals are higher for the GAs with random immigrants. These results can be explained by the fact that the standard GA has difficulties in escaping from the local optima induced by the deceptive problem (e.g., between generations 0 to 1999 of the dynamic test environments experiments) and by changing the global optima (e.g. between 2000 and 3999 in the same experiments). On the other hand, random immigrants inserted in every generation provide diversity to the populations in the last three GAs, which explains their better results.

Let us now analyze the results of the three GAs with random immigrants. First, let us investigate how the proposed algorithm works. In the beginning of the experiments, the individuals of the initial populations generally have small fitness. In the proposed GA, the new individuals that replace the individual with the smallest fitness and its neighbours generally have small values of fitness too. As several individuals in the population have small values of fitness, the probability that one of the neighbours of the current worst individual becomes the new worst is small. As a consequence, a single replacement of an individual generally does not generate large chain reactions of extinctions,

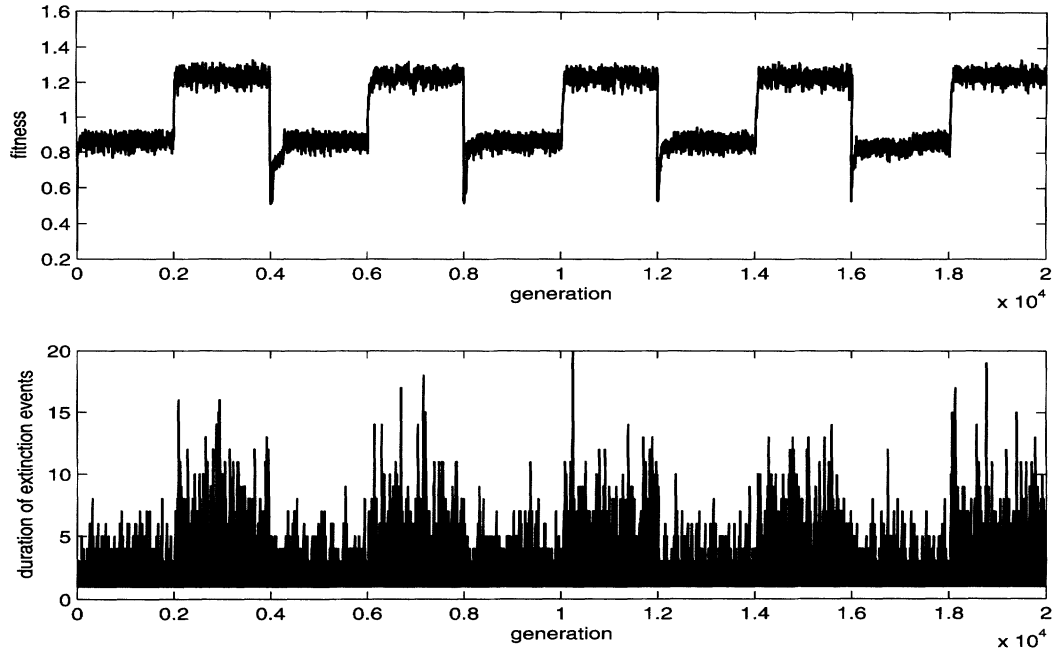


Figure 1: Mean fitness and duration of the extinction events in the fourth trial of the experiment in dynamic test environment 1.

i.e. the distribution of the duration of extinction events is narrow and well described by a small average value. As the number of generations increases, the mean fitness increases too. In this situation, several individuals of the current population have values of fitness higher than the average fitness of the new random individuals. Then, the probability that one of the two neighbours of the old worst individual, which were replaced in the last generation, becomes the new worst individual increases. When this new worst individual is replaced with its two next neighbours, a chain reaction can be developed and the extinction events can have, then, a large variety of sizes. In this case, the extinction events can not be characterized by a narrow distribution.

This situation can be observed in Figure 1, where the mean fitness and the duration of the extinction events in the fourth trial of the experiment in dynamic test environment 1 ($p_c = 0.2$ and $p_m = 0.01$) are plotted. One can observe that, when the global optimum changes from a smaller to a higher value, the mean fitness of the population increased, resulting in higher mean values for the duration of the extinction events and, as a consequence, increasing the diversity of the population. Such interesting behavior is reached by self-organization, and not by a rule imposed by the programmer.

Figure 2 presents the three first steps of an extinction event for a version of the experiment in dynamic test environment 2 with only 10 individuals. The figure respectively shows the fitness of all individuals in the current population in generations 6195, 6196, and 6197. In generation 6195, the individual 3 has the smallest fitness of the population (index j in Algorithm 2). In this way, this individual and its two next neighbours (individuals 2 and 4) are replaced by new random individuals. The individual 4 (index j) has now the smallest fitness in generation 6196, and it

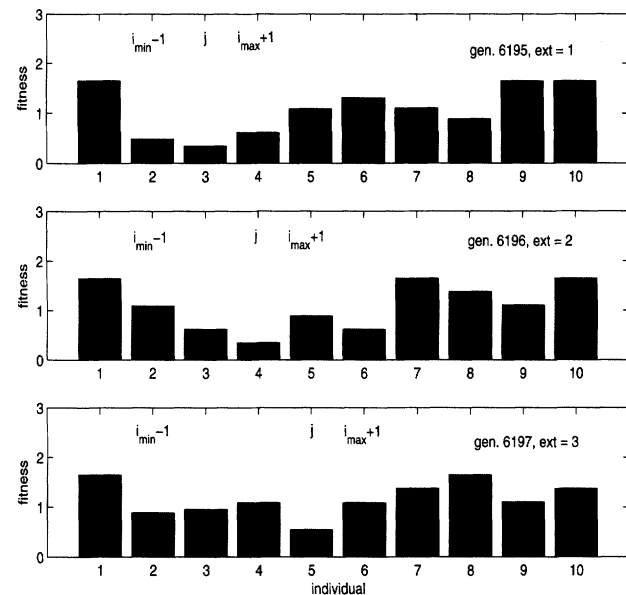


Figure 2: Fitness of the individuals of the current population in generations 6195, 6196, and 6197 for a version of the experiment in dynamic test environment 2 with 10 individuals.

and its two next neighbours (individuals 3 and 5) are then replaced. In generation 6197, the individual 5 (index j) has the smallest fitness. Observe that the chain reaction was propagated because the remaining individuals have values of fitness higher than the individuals in the subpopulation defined by the limits $i_{min} - 1$ and $i_{max} + 1$ (see Algorithms 2 and 3). The individuals that do not belong to this subpopulation are not allowed to replace an individual of this subpopulation.

The better results of the proposed GA when compared to the other GAs with random immigrants in the experiments presented here can be explained by two major factors. First, the number of different individuals that are replaced in a fixed period of generations is generally higher for the proposed GA. One can observe in Figure 2 that the subpopulation in generation 6197 is formed by 5 individuals (with indexes from $i_{min} - 1$ to $i_{max} + 1$). In the GA with random immigrants where the worst individuals are replaced, it is common that new individuals replace individuals with the same index in next generation, because the new individuals generally have small values of fitness. In this way, the number of different individuals that are replaced in a fixed period of generations is generally smaller in comparison to the proposed GA, and, as a result, the diversity becomes smaller too. This fact can be observed by analyzing the results presented in Table 1, where the mean fitness of the population is smaller in the proposed GA, even though its higher values of fitness of the best individuals (i.e. the adaptability is smaller).

The second major fact that explains the better results for the proposed GA is that the survival probability of a new random individual, which can be evolved to become a solution of the problem, is generally smaller in the standard GAs with random immigrants. This is explained because the values of fitness for the current individuals, whose locations are generally located in (or close to) local maxima after several generations, are generally much higher than the mean fitness of the search space, i.e. the mean fitness of all possible individuals. This occurs because the selection methods employed in GA preserves, directly or indirectly, the best individuals of the population. An immigrant generally survives during the evolution only if its fitness is close to the mean fitness of the population, which is a rare event when the number of parameters in the solution is high or when the local optimum where the population is found has values of fitness much higher than the mean fitness of the search space. On the other hand, the proposed GA preserves a new potential solution in a subpopulation and allows it to evolve while the current extinction event is in progress. When the extinction event ends, evolved versions of possible new solutions given by fair immigrants are generally present in the current population and can be combined with the individuals of the main population to generate new solutions.

In the investigated experiments, like in the fossil recorded data for the extinction events in nature [19], there are more small than large extinction events, and the extinction events occur on a large variety of length scales. In Figure 3, the distribution of the number of extinction events against each size is plotted in a log-log scale for the fourth trial of the experiment on dynamic test environment 1 ($p_c = 0.2$ and $p_m = 0.01$). From Figure 3 it can be observed that the result exhibits power laws (see Section 2), even without any apparent tuning, indicating the presence of SOC. This kind of self-organization behavior arises in systems where many degrees of freedom are interacting and the dynamics of the system is dominated by the interaction between these degrees of freedom, rather than by the intrinsic

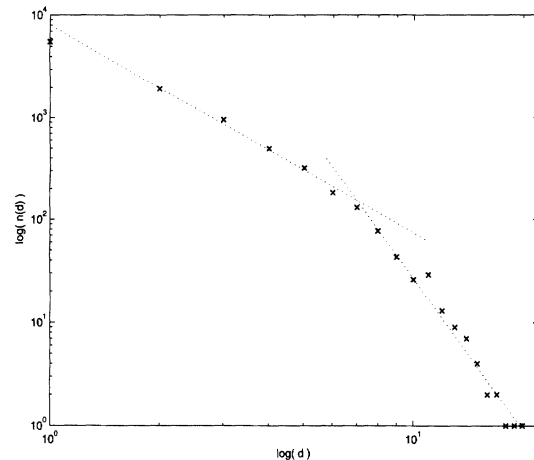


Figure 3: Number of occurrences for each size of the extinction events in the fourth trial of the experiment in dynamic test environment 1.

dynamic dynamics of the individual degrees of freedom [13]. In the proposed GA, the population self-organizes in order to allow the occurrence of extinction events with a large variety of length scales. Large extinction events generally occur when the mean fitness of the population is high, and, as a consequence, the diversity level of the population is small. In this way, the diversity of the population is controlled by self-organization, allowing the GA to escape from local optima when the problem changes.

5 Conclusions and Future Work

In this work, a GA with random immigrants where the worst individual and its next neighbours are replaced in every generation is proposed. In the proposed GA, the individual starts to interact between themselves and, when the fitness of the individuals are high and close, as in the case where the diversity level is small, one single replacement can affect a large number of individuals in an extinction event. In order to avoid that the individuals with the best fitness replaces the newly introduced individuals, these ones are preserved in a subpopulation. The number of individuals in the subpopulation is not defined by the programmer, but is given by the number of individuals created in the extinction event. It is important to observe that this simple approach can take the system to a self-organization behavior, which can be useful in DOPs to maintain the diversity of the solutions and, then, to allow the GA to escape from local optima when the problem changes. In this way, the proposed algorithm is interesting in problems where the new solution is located in a peak that is hardly reached by the location of the old solution by traditional GA operators.

In the proposed algorithm, the old solutions generated by the standard genetic operators can be combined with the solutions created during an extinction event. In this way, the proposed GA can save considerable computation time when compared to a standard GA where a new optimization process with random individuals is started whenever a change in the problem is noticed.

Studying and combining self-organizing behaviors, such as the self-organized criticality studied in this paper, into GAs have shown to be beneficial for their performance under dynamic environments. Much work can be further done in this area. In this paper a simple neighbourhood scheme is used for indexing individuals for the extinction event. That is, individuals in the population are randomly arranged, where neighbouring individuals may not have certain relationship. This is not always true for natural or evolutionary systems. Developing other neighbouring schemes that assign certain relationship between neighbours instead of the random scheme for extinction events in the proposed GA may further improve its performance for DOPs, which is now under investigation by the authors. Another relevant future work is to compare the self-organizing property with other properties, such as the speciation schemes, for GAs under more comprehensive dynamic environments.

Acknowledgments

The authors would like to thank FAPESP (Proc. 04/04289-6) for the financial support.

Bibliography

- [1] P. Bak. *How nature works: the science of self-organized criticality*. Oxford University Press, 1997.
- [2] P. Bak, C. Tang, and K. Wiesenfeld. Self-organized criticality. an explanation of $1/f$ noise. *Physical Review Letters*, 59(4):381–384, 1987.
- [3] S. Boettcher and A. G. Percus. Optimization with extremal dynamics. *Complexity*, 8(2):57–62, 2003.
- [4] J. Branke. Evolutionary approaches to dynamic optimization problems - introduction and recent trends. In J. Branke, editor, *GECCO Workshop on Evol. Alg. for Dynamic Optimization Problems*, pages 2–4, 2003.
- [5] W. Cedeno and V. R. Vemuri. On the use of niching for dynamic landscapes. In *Proc. of the 1997 IEEE Int. Conf. on Evolutionary Computation*, pages 361–366, 1997.
- [6] H. G. Cobb and J. J. Grefenstette. Genetic algorithms for tracking changing environments. In S. Forrest, editor, *Proc. of the 5th Int. Conf. on Genetic Algorithms*, pages 523–530, 1993.
- [7] K. A. De Jong. *An analysis of the behavior of a class of genetic adaptive systems*. PhD Dissertation, University of Michigan, 1975.
- [8] K. Deb and D. E. Goldberg. Analyzing deception in trap functions. In *Foundation of Genetic Algorithms 2*, pages 93–108, 1993.
- [9] D. Floreano and F. Mondada. Evolution of homing navigation in a real mobile robot. *IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics*, 26(3):396–407, 1996.
- [10] D. A. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company, Inc., 1989.
- [11] D. A. Goldberg. *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Boston, MA: Kluwer Academic Publishers., 2002.
- [12] T. Isokawa, N. Matsui, H. Nishimura, and F. Peper. Coping with nonstationary environments: a genetic algorithm using neutral variation. *IEEE Trans. on Syst., Man, and Cyb. Part A: Syst. and Humans*, 32(4):497–504, 2002.
- [13] H. J. Jensen. *Self-organized criticality: emergent complex behavior in physical and biological systems*. Cambridge University Press, 1998.
- [14] S. A. Kauffman. *The origins of order: self-organization and selection in evolution*. Oxford University Press, 1993.
- [15] T. Krink and R. Thomsen. Self-organized criticality and mass extinction in evolutionary algorithms. In *Proc. of the 2001 Congress on Evolutionary Computation*, volume 2, pages 1155–1161, 2001.
- [16] M. Løvbjerg and T. Krink. Extending particle swarm optimisers with self-organized criticality. In *Proc. of the 2002 Congress on Evolutionary Computation*, volume 2, pages 1588–1593, 2002.
- [17] M. Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, 1996.
- [18] S. Nolfi and D. Floreano. *Evolutionary robotics: the biology, intelligence, and technology of self-organizing machines*. MIT Press/Bradford Books: Cambridge, USA, 2000.
- [19] D. M. Raup. Biological extinction in earth history. *Science*, 231:1528–1533, 1986.
- [20] K. Trojanowski and Z. Michalewicz. Evolutionary algorithms for non-stationary environments. In M. A. Kłopotek and M. Michalewicz, editors, *Intelligent Inf. Systems, Proc. of the 8th Int. Workshop on Intelligent Inf. Syst.*, pages 229–240, 1999.
- [21] F. Vavak and T. C. Fogarty. A comparative study of steady state and generational genetic algorithms for use in nonstationary environments. In T. C. Fogarty, editor, *AISB Workshop on Evolutionary Computing, Lecture Notes in Computer Science*, volume 1143, pages 297–304. Springer, 1996.
- [22] S. Yang. Constructing dynamic test environments for genetic algorithms based on problem difficulty. In *Proc. of the 2004 Congress on Evolutionary Computation*, volume 2, pages 1262–1269, 2004.