We are IntechOpen, the world's leading publisher of Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Diversity-Based Adaptive Evolutionary Algorithms

Maury Meirelles Gouvêa Jr. and Aluizio Fausto Ribeiro Araújo
Pontifical Catholic University of Minas Gerais
Federal University of Pernambuco
Brazil

1. Introduction

In evolutionary algorithms (EAs), preserving the diversity of the population, or minimizing its loss, may benefit the evolutionary process in several ways, such as, by preventing premature convergence, by allocating the population in distinct Pareto optimal solutions in a multi objective problem, and by permitting fast adaptation in dynamic problems. Premature convergence may lead the EA to a non-optimal result, that is, converging to a local optimum. In static problems, standard EAs work well. However, many real world problems are dynamic or other uncertainties have to be taken into account, such as noise and fitness approximation. In dynamic problems, the preservation of diversity is a crucial issue because EAs need to explore the largest number of regions possible. Standard genetic algorithms (SGA) are not suitable for solving dynamic problems because their population quickly converges to a specific region of the solution space.

The loss of diversity is caused by selection pressure and genetic drift, two factors inherent in EAs. The loss of diversity may lead the EA to a non-optimal result, despite the fact that after a period of time, EA tends to find the global optimum. In static problems, loss of diversity might not be a very critical problem. However in dynamic environments lack of diversity may degrade EA performance. Especially in dynamic problems, the preservation of diversity is a crucial issue because an EA needs to explore the search space aggressively.

One option for reacting to a change of the environment is to consider each change as the arrival of a new optimization problem to be solved. This is a viable alternative if there is time available to solve the problem. However, the time available for finding the new optimum may be short and also sometimes the algorithm cannot identify the environmental change. When the new optimum is close to the old one, the search can be restricted to the neighborhood of the previous optimum. Thus, some knowledge about the previous search space can be used. However, reusing information from the past may not be promising depending on the nature of the change. If the change is large or unpredictable, restarting the search may be the only viable option.

The approaches that handle dynamic environments, addressing the issue of convergence, can be divided into the following categories (Jin & Branke, 2005): (i) generating diversity after a change, (ii) preserving diversity throughout the run, (iii) memory-based approaches, and (iv) multi-population approaches. The first two approaches cover the diversity problem.

In (i), an EA runs in standard way, but when a change is detected, some actions are taken to increase diversity. In (ii), convergence is avoided all the time and it is expected that a more dispersive population can adapt to changes. In (iii), EA is supplied with a memory so as to be able to recall useful information from past generations. In (iv), the population is divided into several subpopulations allowing different peaks in the environment to be tracked.

The preservation of diversity has advantages that can be supported by theory, such as those cited above, and from Nature. The loss of diversity because of the extinction of species may produce irreversible ecological disturbance for an ecosystem. A high diversity level produces abilities which allow populations or species to react against adversities, such as diseases, parasites, and predators. An appropriate level of diversity allows populations or species to adapt to environmental changes. On the other hand, a low diversity level tends to limit these abilities (Amos & Harwood, 1998). From the point of view of the evolutionary process, the loss of diversity also represents serious problems, such as population convergence to a specific region of the solutions space; thus, EA losses its main feature, the global search. In order to preserve the diversity of the population it is necessary to create strategies to adjust one or more EA parameters, such as the mutation rate, selection pressure, etc. These strategies are known as diversity-based algorithms.

This chapter presents a survey on diversity-based evolutionary algorithms. Two classes of models are presented: one to minimize the loss of diversity and another to control population diversity based on the desired diversity range or level. Several methods to measure the diversity of the population and the species are presented as a foundation for diversity control methods. The rest of this paper is organized as follows. Section 2 presents parameter setting and control in EAs. Section 3 describes several methods for measuring diversity. Section 4 presents methods to preserve and control population diversity in evolutionary algorithms. Finally, Section 5 concludes this chapter.

2. Parameter tuning and control in evolutionary computation

The EA parameters can affect population diversity directly. For instance, a larger mutation rate causes disturbances in the offspring and, consequently, increases the diversity of the population in the next generation. On the other hand, the greater the selection pressure is, the fittest individuals tend to survive or generate more offspring. Thus, these individuals tend to be genetically similar, thus decreasing the diversity of the population.

We can set parameter values by parameter tuning and parameter control (Angeline, 1995; Eiben et al., 1999; Hinterding et al., 1997). Parameter tuning finds appropriate values for the parameters before the algorithm is used, and these parameters are fixed during the run. For example, Bäck & Schutz (1996) suggest the following mutation probability

$$p_m = \frac{1.75}{N\sqrt{L}}, \tag{1}$$

where N is the population size and L is the individual length.

Parameter control changes parameter values on-line in accordance with three categories (Eiben et al., 1999; Hinterding et al., 1997): deterministic, adaptive, and self-adaptive control methods. The next three subsections present these categories.

2.1 Deterministic control methods

Deterministic techniques in which the control rule is triggered when a number of generations has elapsed since the last time the rule was activated. For example (Hinterding et al., 1997), the mutation rate may be defined as

$$p_m(k) = 0.5 - 0.3 \frac{k}{K}, \qquad (2)$$

where k is the current generation and K is the maximum number of generations. This strategy aims to produce high exploration in the beginning of the evolutionary process as a way to seek out promising regions in the solutions space. During the evolutionary process, the mutation rate decreases in order to benefit the exploration continuously. Thus, the diversity of the population tends to decrease throughout the evolutionary process.

2.2 Adaptive control methods

Adaptive techniques consider that the assignment of parameter values is associated with feedback from the evolutionary process. For example, the mutation rate may be defined as in (Srinivas & Patnaik, 1994)

$$p_m(k) = \frac{A}{f^*(k) - \bar{f}(k)}, \qquad (3)$$

where f^* is the fitness of the best individual, $\bar{f}$ is the mean fitness of the population, and A is a constant. This strategy increases the mutation rate as the mean fitness of the population approximates to the best fitness of the population. The objective is to avoid the convergence of the whole population to a specific region of the solutions space. This interaction with the environment by adaptive control methods may be an advantage over deterministic methods because the former may overcome some problems during the evolutionary process, such as local optimum convergence. In dynamic problems, even if the population is located around the global optimum, when the environment changes, it is almost always necessary to spread the population. Using Equation (2), it is not possible to modify the mutation rate based on the environmental change. With adaptive methods, if the algorithm has a mechanism to detect the environment change, the mutation rate can be increased. On the other hand, adaptive methods require more computational effort than deterministic methods.

2.3 Self-adaptive control methods

Self-adaptive techniques encode the parameters in the chromosomes and undergo EA operators. For instance (Eiben et al., 1999), the representation of the i-th individual g_i becomes

$$[g_{i1}, ..., g_{iL}, p_m],$$

in which both the solution vector and mutation probability undergo the evolutionary process. The self-adaptive methods use the evolution principle on the EA parameters, which are modified and undergo the whole evolutionary process – selection, crossover, and mutation. For instance, an individual with L genes in the standard evolutionary algorithm will have L+1 genes in a self-adaptive method, where the extra gene is an evolutionary factor parameter, such as the mutation rate, crossover rate, type of crossover operator, and

so forth. The advantage of this strategy over the other control parameter methods is that the parameters are modified by the effects of evolutions, and tend to persist at all parameter values that produce better individuals. Another benefit of this strategy is its low computational effort because only few genes are added into the individuals, and no extra computation is necessary. The disadvantage of the self-adaptive strategy occurs especially in a dynamic environment where the changes in the environment may not be detected or are detected late. Self-adaptive methods may not be able to avoid premature convergence in the local optimum because, normally, they do not have a direct way to detect it.

3. Diversity measurement

In order to preserve and, especially, to control population diversity it is necessary to measure it. This section presents some of the measurement methods proposed by several authors (Rao, 1982; Weitzman, 1992; Solow et al., 1993; Champely & Chessel, 2002; Ursem et al., 2002; Wineberg & Oppacher, 2003; Simpson, 2004). Rao (1982) created a diversity function based on the probability distribution of a finite set of species. His diversity function uses the distance $d(s_1, s_2)$ between two species s_1 and s_2 defined over a finite set of species, as follows

$$\Gamma = \sum_{i=1}^{n_S} \sum_{j=1}^{n_S} p_i \, p_j \, d(s_i, s_j) \,, \tag{4}$$

where n_S is the number of species and $p_i = P(X = s_i)$.
Weitzman (1992) created a recursive method to compute diversity, as follows

$$\Gamma(S) = \max_{s_i \in S} \Gamma(S - s_i) + d(s_i, S) \; \forall s_i \in S \,, \tag{5}$$

in which there is a unique solution whether if the condition $\Gamma(s_i) = d_0$ is considered, where $d_0 \geq 0$ is a constant.
Solow et al. (1993) proposed a function, named the preservation measure, to calculate the loss of diversity when a species s_i becomes extinct, as follows

$$\Delta\Gamma = -\sum_{s_i \notin S} d(s_i, S) \,. \tag{6}$$

Based on Rao (1982), Champely and Chessel (2002) introduced a function for diversity using the Euclidean distance between species, defined as

$$\Gamma = \frac{1}{2} \sum_{i=1}^{n_S} \sum_{j=1}^{n_S} p_i \, p_j \left[d(s_i, s_j) \right]^2 \,. \tag{7}$$

Simpson (2004) created a heterozygosity-based diversity function, H_e. When H_e is replaced with Γ, Simpson´s diversity function becomes

$$\Gamma = 1 - \sum_{i=1}^{n_a} \left(p_i \right)^2 \,, \tag{8}$$

where p_i is the occurrence rate of the i-th allele, individual, or species from the set S, and n_a is the number of alleles, individuals, or species.

In evolutionary computation, normally, the methods that measure population diversity use two different types of models: one as a function of the distance between individuals (Wineberg & Oppacher, 2003) and another as a function of the distance from the individuals to a reference (Ursem et al., 2002), e.g., the population mean point.

Diversity as a function of the distance between all individuals can be measured as follows

$$\Gamma = \sum_{i=1}^{N} \sqrt{\sum_{j=1}^{i-1} d(g_i, g_j)} \ , \tag{9}$$

where $d(g_i, g_j)$ is the distance (e.g., Euclidean or Hamming) between individuals g_i and g_j. The diversity from Equation (9), with complexity $O(L, N^2)$, has the disadvantage of requiring a large computational effort.

Wineberg & Oppacher (2003) proposed a smaller computational effort than Equation (9), with complexity $O(L, N)$, based on allele frequencies, as follows

$$\Gamma = \frac{N^2}{2L} \sum_{i=1}^{L} \sum_{\forall \alpha \in A} f_i(\alpha)\big[1 - f_i(\alpha)\big] \ , \tag{10}$$

where A is the set of alleles,

$$f_i(\alpha) = \frac{c_i(\alpha)}{N} \ , \tag{11}$$

is the frequency in which the allele α occurs in the i-th gene of the individuals in the population,

$$c_i(\alpha) = \sum_{j=1}^{N} \delta_{ji}(\alpha) \ , \tag{12}$$

is the number of occurrences of α and $\delta_{ji}(\alpha)$ is the delta of Kronecker, which becomes 1 if the gene in *locus i* in chromosome j is equal to α; or 0, otherwise.

Ursem et al. (2002) proposed a model as a function of a reference point in the solutions space, which requires a smaller computational effort and complexity $O(L, N)$. This diversity model shows the population distribution with respect to the population mean point calculated as follows

$$\Gamma = \frac{1}{DN} \sum_{i=1}^{N} \sqrt{\sum_{j=1}^{L} \left(g_{ij} - \overline{g}_j\right)^2} \ , \tag{13}$$

where D is the solutions space diagonal, $D \subset \Re^L$, g_{ij} is the j-th gene of the i-th individual, $\overline{g}$ is the j-th gene of the population mean point, $\overline{g}$, where

$$g_j = \frac{1}{N} \sum_{i=1}^{N} g_{ij} \ . \tag{14}$$

The diversity between species and population diversity have different characteristics. In the former, the species are always different, whereas in the latter, two individuals may be genetically equal. In the diversity of species, a new individual added to a set S increases its diversity. In populations, a new individual may increase or decrease diversity.

4. Diversity preservation and control

The parameter control methods that aim to preserve diversity can be divided into two classes: diversity preservation and diversity control. The diversity preservation methods use strategies that minimize the loss of diversity (Bui et al., 2005; Herrera et al., 2000; Simões & Costa, 2002a; Wong et al., 2003). The diversity control methods have a value or range of desired diversity (Meiyi et al., 2004; Nguyen & Wong, 2003; Ursem et al., 2002). Thus, diversity control strategies aim to minimize the difference between the population and the diversities desired. The next two subsections present some important diversity preservation and control methods in evolutionary computation.

4.1 Diversity preservation

Most methods that deal with population diversity try to avoid loss of diversity without setting a desired value or range. Cobb (1990) created the trigged hypermutation (THM) method that set the mutation probability to a high value (hypermutation) during periods where the time-averaged best performance of the EA worsens; otherwise, the EA maintains a low level of mutation. THM permits the EA to accommodate changes in the environment, while also permitting the EA to perform optimization during periods of environmental stationariness.

Simões & Costa (2001) created a biologically inspired genetic operator called transformation. The computational mechanism is inspired by the biological process and consists of the capacity of the individuals to absorb fragments of DNA (desoxirribonucleic acid) from the environment. These gene segments are then reintegrated in the individual genome. Simões & Costa incorporated transformation into the standard evolutionary algorithm as a new genetic operator that replaces crossover. The pseudo-code of this modified EA is described in Figure 1.

```
k ← 0
Generate P(k)
Evaluate P(k)
Generate initial gene segment pool
while( NOT stop condition )
{
        Select S(k) from P(k)
        Transform S(k)
        Evaluate S(k)
        P(k+1) ← S(k)
        Generate new gene segment pool
        k ← k + 1
}
```

Fig. 1. Transformation-based GA (TGA)

The foreign DNA fragments, consisting of binary strings of different lengths, will form a gene segment pool and will be used to transform the individuals of the population. In each generation k, a sub-population S(k) is selected to be transformed by the pool of gene segments. The segment pool is changed using the old population to create part of the new segments with those remaining being created at random. In the transformation mechanism there is no sexual reproduction among the individuals. To transform an individual the following steps are conducted: select a segment from the segment pool and randomly choose a point of transformation in the selected individual. The segment is incorporated in the genome of the individual. This corresponds to the biological process where the gene segments when integrated in the DNA cell recipient, replace some genes in its chromosome.

Wong et al. (2003) created a method to adjust the crossover, p_c, and mutation probability, pm, in order to promote a trade-off for exploration and exploitation. The evolutionary process is divided into two phases: the first uses pc and pm with values at random; the second adjusts pc and pm according to the fitness enhancements from the first phase. The diversity of the population is maintained by appending a "diversity fitness" into the original individual fitness. Thus, population diversity contributes to survivor selection as a weighted form, that is, there is a weight to balance the original fitness and the diversity fitness.

Shimodaira (2001) designed a method to preserve diversity, called the diversity control-oriented genetic algorithm (DCGA). First, the population is paired, each one is recombined and their offspring are mutated. After that, the offspring and current population are merged. Then, the survivor selection is made from the remaining population in accordance with the following roles:

1. Duplicate structures in M(t) are eliminated and M′(t) is formed. Duplicate structures mean that they have identical entire structures;

2. Structures are selected by using the Cross-generational Probabilistic Survival Selection (CPSS) method, and P(t) is formed from the structure with the best fitness value in M′(t) and the selected structures. In the CPSS method, structures are selected by using uniform random numbers on the basis of a selection probability defined by the following equation:

$$p_s = \left((1-c)\frac{H_i}{L} + c \right)^{\alpha}, \tag{15}$$

where H_i is the Hamming distance between a candidate structure and the structure with the best fitness value, L is the length of the entire string representing the structure, c is the shape coefficient the value of which is in the range of [0.0, 1.0], and α is the exponent. In the selection process, a uniform random number in the range of [0.0, 1.0] is generated for each structure.

3. If the number of the structures selected in Role 2 is smaller than N; then new structures randomly generated as in the initial population are introduced by the difference of the numbers.

DCGA has an empirical and theoretical justification for avoiding premature convergence. The duplicated offspring decrease the population diversity. Thus, the crossover and large mutation probability tend to produce offspring that are as different as possible from their parents, and that explore regions of the solutions space that have not been explored. The selection pressure and population diversity should be externally controlled independently of the condition of the population, because the algorithm cannot recognize if the population

is in a local or global optimum. If the selection pressure is high, the best individuals near the best one tend to rise and survive in larger numbers, thus causing premature convergence. Shimodaira tried to solve this problem by reducing appropriately the selection pressure in the neighborhood of the best individual to eliminate individuals similar to it. Equation (15) creates a bias between the elimination of individuals with the smallest Hamming distance to the best individual and the selection of individuals with the greatest Hamming distance to the best individual. The greater this bias is, the greater the diversity of the population is.

Grefenstette (1992) proposed an approach based on the flow of population immigrants over generations, called the random immigrants genetic algorithm (RIGA). This approach maintains a population diversity level by replacing some individuals from the current population by random individuals, called random immigrants, in every generation. There are two ways that define how individuals are replaced: replacing individuals at random or replacing the worst ones (Vavak et al., 1996). RIGA inserts random individuals into the population, a strategy that may increase population diversity and benefit the performance of the GA in dynamic environments. Figure 2 shows the pseudo-code of the RIGA.

```
k ← 0
Generate P(k)
Evaluate P(k)
while( NOT stop condition)
{
        Generate a sub-population S_{ri}(k) of n_{ri} random immigrants
        Evaluate S_{ri}(k)
        Replace the worst individuals of P(k) with S_{ri}(k)
        P'(k) ← select( P(k) )
        Q(k) ← recombine( P'(k) )
        Mutate Q(k)
        Evaluate Q(k)
        P(k+1) ← Q(k)
        k ← k + 1
}
```

Fig. 2. Random immigrants GA (RIGA)

4.2 Diversity control

The control aims to keep the process output within prescribed limits (Narendra & Annaswamy, 2005). Figure 3 shows a simple control model. The objective of the control may be stated as the determination of the process input, u, to keep the error between a desired output, Γ_m, and the process output, Γ, within a pre-determined interval. If Γ_m is constant, the problem is called regulation around this value – also known as a set point. If Γ_m is a function of time, the problem is referred as tracking. When the characteristics of the process are known, the aim becomes to determine a control strategy to stabilize the feedback loop in Figure 3 around Γ_m. Otherwise, when the characteristics of the process are unknown, both regulation and tracking can be viewed as an adaptive control problem.

Diversity control methods have a level or range of desired diversity. Thus, it is possible to define a control strategy based on a desired diversity. Ursem et al. (2002) created the diversity-guided evolutionary algorithm (DGEA) with two evolutionary modes:
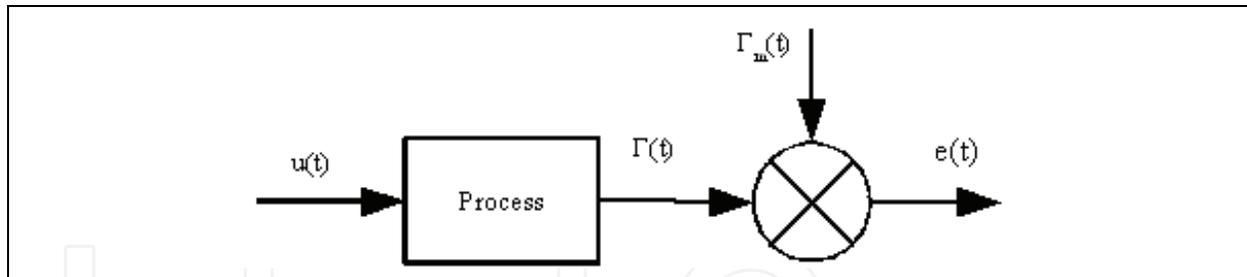
Fig. 3. The simple control scheme

exploitation and exploration. The former applies selection and recombination operators, which tend to decrease population diversity, while the diversity is above a limit d_{low}. When the population diversity drops below d_{low}, DGEA switches to an exploration mode that applies only the mutation operator, which tends to increase the diversity, until a diversity of d_{high} is reached. Ursem et al. used Equation (13) to measure the diversity of the population. Both evolutionary modes, exploitation and exploration, change from one to the other in the evolutionary process as a function of the diversity range. Figure 4 shows the pseudo-code of the DGEA.

```
k ← 0
Generate P(k)
Evaluate P(k)
mode ← 'exploit'
while( NOT stop condiction )
{
        if( diversity( P(k) ) < dlow )
           mode ← 'explore'
        else if( diversity( P(k) ) > dhigh )
           mode ← 'exploit'

        if( mode = 'exploit' )
           P'(k) ← select( P(k) )
           Q(k) ← recombine( P'(k) )
        else
           Q(k) ← mutate( P(k) )

        Evaluate Q(k)
        P(k+1) ← Q(k)
        k ← k + 1
}
```

Fig. 4. Diversity-guided evolutionary algorithm (DGEA)

An important issue is to apply a mutation operator that rather quickly increases the distance to the population mean point. Otherwise, the algorithm will stay in exploration mode for a long time. An advantage for a mutation operator is to use the population mean average point to calculate the direction of each individual mutation. A disadvantage of the DGEA is its non-use of selection, recombination, and mutation together, which is the fundamental principle of EA.

Nguyen & Wong (2003) used control theory to adjust the mutation rate in unimodal space. The desired diversity, in generation k, was defined as follows

$$\Gamma_d(k) = \Gamma(0) \exp(-k\tau) , \qquad (16)$$

where $\Gamma(0)$ is the initial diversity and $\tau > 0$ is a constant. Nguyen & Wong model is motivated by the observation that for unimodal search, convergence implies a corresponding reduction in the population diversity and that an exponential convergence rate would need to be accompanied by an exponential reduction of diversity. Nguyen & Wong adopted a diversity measurement based on the radial deviation from the population mean point.

In Nguyen & Wong method, when the current population diversity deviates from the diversity desired, Γ_d, the mutation rate is adjusted as a control problem (Ogata, 1998), as follows

$$p_m(k+1) = p_m(k) \exp\left(\frac{e(k)}{\tilde{e}(k)}\right) , \qquad (17)$$

where $e(k)$ is the deviation or error between the current and desired diversities, $\tilde{e}(k)$ is the square mean error defined as

$$\tilde{e}(k) = \sqrt{\left[e(1)\right]^2 + \left[e(2)\right]^2 + \ldots + \left[e(k)\right]^2} . \qquad (18)$$

From the work of Beyer & Rudolph (1997), Nguyen & Wong hypothesized that EAs can be induced to have linear order convergence for unimodal search if the population diversity can be controlled so as to decrease at a matching exponential rate.

We created an adaptive EA named diversity-reference adaptive control (DRAC) (Gouvêa Jr. & Araújo, 2007). Our approach was based on the model-reference adaptive system (MRAS), an adaptive controller in which the desired performance of a particular process is determined by a model-reference (Astrom & Wittenmark, 1995; Wagg, 2003). The implicit assumption is that the designer is sufficiently familiar with the system under consideration. When a suitable choice is made of the structure and parameters of the model-reference, the desired response can be specified in terms of the model output.

In MRAS, the model and process output are compared and the difference is used to yield the control signal. The system holds two feedback loops: the first loop, an ordinary piece of feedback, comprises the process and controller; and the second changes the controller parameter. Given one process, with an input-output pair $\{u, \Gamma\}$, and one model-reference, with an input-output pair $\{u_c, \Gamma_m\}$, the aim is to determine the control input $u(t)$, for all $t \geq t_0$, so that

$$\lim_{t \to \infty} \left|\Gamma_m(t) - \Gamma(t)\right| = 0 . \qquad (19)$$

Parameter updating is based on feedback from the error. Two widely used methods to yield the control signal using MRAS are (Astrom & Wittenmark, 1995): the MIT rule and the stable adaptation law derived from Lyapunov stability theory.

The MIT rule begins by defining the tracking error, e, which represents the difference between the process output and the model-reference output, as follows

$$e(t) = \Gamma_m(t) - \Gamma(t), \tag{20}$$

where $\Gamma_m(t)$ and $\Gamma(t)$ are the model-reference and process output, respectively, at time t. From this error, a cost function, $J(\theta)$, is formed, where θ is the controller parameter that will be adapted. A typical cost function is

$$J(\theta) = \frac{1}{2}\left[e(t)\right]^2. \tag{21}$$

If the goal is to minimize this cost related to the error, the parameter θ can be changed in accordance with the negative gradient of J, then

$$\frac{d\theta}{dt} = -\eta\frac{\partial J}{\partial \theta} = -\eta\, e\,\frac{\partial e}{\partial \theta}, \tag{22}$$

where $\eta \in [0, 1]$ is the step length of the θ adjustment. The partial derivative $\partial e/\partial\theta$, the sensitivity of the system, establishes how the error is influenced by the adjustable parameter.

Thus, the process output has to track the model-reference output. The block diagram of DRAC, Figure 5, has a block called process comprising the evolutionary process and the diversity evaluator so as to determine the current diversity of the population. The controller sends the control signal, u, to the process as a function of the command signal, u_c, and the parameter, θ. The updating of θ is based on the error between the process and model-reference output. The model-reference, as a whole, represents a behaviour to be tracked by the population diversity.

DRAC computes the population diversity based on Equation (8) as a function of the allele occurrence rate for a given gene. In real-coded EA, the number of alleles is calculated by separating the gene length into defined intervals, i.e., the number of alleles, n_a, is the number of intervals. Thus, the allele that belongs to a given interval j is regarded as allele g_{ij}, i.e., allele j from the gene i.

In DRAC, the model-reference represents a crucial feature of behaviour to be tracked by the evolutionary process. Note that while the evolutionary process aims to determine the optimal solution, the control system regulates the population diversity to track the model-reference. The model-reference is expressed by

$$\Gamma_m(k+1) = \Psi(\Gamma_m(k),\, u_c(k),\, k). \tag{23}$$

where $\Psi(.)$ is a non-linear function, Γ_m is the model-reference output, and u_c is the command signal (i.e., the model input).

From the Hardy-Wimberg model (Hardy, 1908; Weinberg, 1908), it is possible to assume that there is no evolution without loss of diversity. Regarding this premise, a general model-reference should consider two hypotheses: (i) during the evolutionary process, diversity decreases, and (ii) there is a minimum diversity level to maintain a balance between exploitation and exploration. Thus, after each change in the environment, Γ_m goes from its current value to $\Gamma(0)$, to increase exploration.

DRAC proposes a model-reference that decreases its diversity limited by a determined minimum value. This model also forces a strong growth in diversity after changes in the environment. The modifications to the environment are detected by the decrease of the best
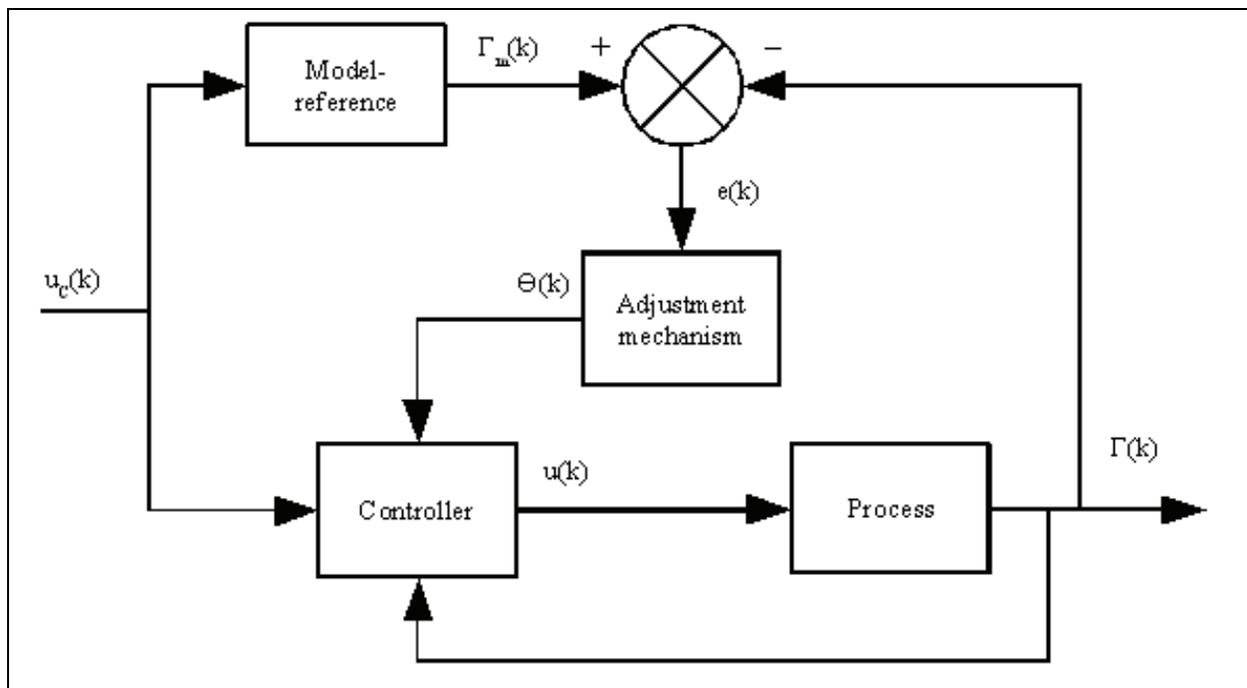
Fig. 5. Block diagram of the DRAC method for EA parameter control

individual. Our model reference is based on heterozygosity dynamics, from an ideal Wright-Fisher population (Wright, 1931; Fisher, 1930), as follows

$$\Gamma_m(k+1) = \Gamma_m(k)\left(1 - \frac{1}{2N_e}\right), \qquad (24)$$

where N_e is the effective population size, i.e., the size of the ideal Wright-Fisher population. The command signal, u_c, is the effective population size, N_e, $\Gamma_m(0)$ is the initial population diversity, $\Gamma_m(0) = \Gamma(0)$, and a minimum diversity value must be defined to avoid zero diversity.

DRAC modifies the selection mechanism, which is conducted in three stages as follows:

1. Selection of the individual with best fitness to assure that the best solution survives to the next generation.
2. Selection of αN individuals based on a standard selection (e.g., roulette wheel or tournament).
3. Selection of $(1 - \alpha)N - 1$ individuals based on their distances from an individual to the population mean point, $\overline{g}$, so as to preserve diversity. This individual fitness is based on the distance, d_i, weighted by the selection pressure, becoming

$$f'_i(d_i) = 1 - \exp(-\beta d_i), \qquad (25)$$

where $\beta > 0$ is the selection pressure. The lower d_i is, the lower is the fitness, $f'_i(.,.)$. Thus, an individual far from $\overline{g}$ is more likely to be selected, thus, preserving the diversity. The selection pressure, β, regulates the influence of the distance d_i upon the selection mechanism, i.e., the larger β is,, the higher the influence of the individual distance d_i is upon the selection, and the higher the diversity in the next generation is.

In DRAC, the selection pressure is the control signal, i.e., $u_c = \beta$, and its parameter θ is adjusted as a function of the error between the current, Γ, and model-reference diversity, Γ_m. DRAC model proposes a control signal as a function of the command signal and the controller parameter. The control law is defined as

$$u(k) = \theta(k)u_c(k) \, , \tag{26}$$

The method employed to adjust θ is a particular case of the MIT rule. The parameter θ is updated as a function of the error between the process and model-reference output. The discrete version of the adjustment of θ is defined as an approximation of Equation (22), as follows

$$\theta(k+1) = \theta(k) - \eta' e(k) \, , \tag{27}$$

where $\eta' = \eta \, \partial e/\partial \theta$, $\eta' > 0$, is a constant. This adjustment rule gives no guarantee that the adaptive controller makes the error vanish. Figure 6 shows the DRAC pseudo-code.

```
k ← 0
Generate P(k)
Evaluate P(k)
Γ(k) ← diversity( P(k) )

while (NOT stop condiction )
{
  Select P′(k) from P(k)
  Q(k) ← recombine( P(k) )
  Mutate Q(k)
  Evaluate Q(k)
  Associate P(k) and Q(k)
  Γₘ(k) ← model-reference( k )
  Adjust θ(k+1) as a function of the error, e(k) = Γₘ(k) – Γ(k)
  Select the survivors S(k):
         1. Select the best individual
         2. Select αN individuals by tournament
         3. Select the rest to the next population
  P(k+1) ← S(k)
  Γ(k+1) ← diversity( P(k+1) )
  k ← k+1
}
```

Fig. 6. Diversity-reference adaptive control (DRAC)

5. Conclusion

This paper presented a survey about diversity-based evolutionary algorithms. Two sets of models were presented, one to minimize the diversity loss and another to control the population diversity based on a desired diversity range or level. The problem of the inappropriate level of diversity with respect to the environment and its dynamic can be

avoided or reduced if the population diversity is controlled. For example, DRAC controls population diversity, which tracks a model-reference. The method provides a model-reference of diversity that decreases according to a control law and increases after the environment changes. In DRAC, the evolutionary process is handled as a control problem, and MRAS is used to adjust the control signal.

The model-reference, tracked by the population in DRAC, is based upon principles: (i) from the Hardy-Weinberg theory that, in a population, it is necessary for diversity to decrease in order that there is evolution; and (ii) it is necessary to have a minimum level of diversity in order to benefit exploitation. The diversity control method proposed can accelerate the speed at which the algorithm reaches promising regions in a dynamic environment.

DRAC reveals several possibilities, such as adjusting the model-reference as a function of the environment and its dynamic, especially for small-sized and chaotic dynamics. Another possibility is to use other EA parameters as the control signal, such as mutation and crossover probabilities, the number of individuals selected for crossover, and the number of individuals selected in Stage 3 of the proposed selection mechanism. These parameters have a significant influence on population diversity and the evolutionary process, and they can be investigated and compared with pressure-based selection.

6. Acknowledgment

7. References

Amos, W.; Harwood, J. (1998). Factors affecting levels of genetic diversity in natural populations. *Philosophical Transactions of the Royal Society of London – Series B: Biological Sciences*, Vol. 353, No. 1366, pp. 177—186

Angeline, P. J. (1995). *Adaptive and Self-adaptive Evolutionary Computation,* IEEE Press, Piscataway

Aström, K. J.; Wittenmark, B. (1995). *Adaptive Control,* Addison-Wesley

Bäck, T.; Schutz, M. (1996). Intelligent Mutation Rate Control in Canonical Genetic Algorithms, *International Symposium on Methodologies for Intelligent Systems (ISMIS'96),* pp. 158—167, Springer

Beyer, H. G.; Rudolph, G. (1997). Local performance measures, In: *Handbook of Evolutionary Computation,* Back, T.; Fogel, D. B.; Michalewicz, Z. (Ed.), B2.4:1—B2.4:27, Oxford University Press, Oxford

Champely, S.; Chessel, D. (2002). Measuring biological diversity using euclidean metrics. *Environmental and Ecological Statistics,* Vol. 9, No. 2, pp. 167—177

Cobb, H. G. (1990). *An Investigation into the Use of Hypermutation as an Adaptive Operator in Genetic Algorithms Having Continuous, Time-dependent Nonstationary Environments,* Naval Research Laboratory, Technical Rep. AIC-90-001, Washington

Crow, J. F.; Kimura, M. (1970). *An Introduction to Population Genetic Theory,* Burgess Publishing, Minnesota

Eiben, A. E.; Hinterding, R.; Michalewicz, Z. (1999). Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation,* Vol. 3, No. 2, pp. 124—141

Grefenstette, J. J. (1986). Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics,* Vol. 16, No. 1, pp. 122−128

Fisher, R. A. (1930). *The Genetical Theory of Natural Selection*, Oxford University Press

Gouvêa Jr., M. M.; Araújo, A. F. R. (2007). Diversity-Based Model Reference for Genetic Algorithms in Dynamic Environment, *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*, Singapore, South Korea, September 25-28, IEEE Press

Hardy, G. H. (1908). Mendelian proportions in a mixed population. *Science*, Vol. 78, 49−50

Hinterding, R.; Michalewicz, Z.; Eiben, A. E. (1997). Adaptation in evolutionary computation: a survey, *Proceedings of the 4th IEEE Conference Evolutionary Computation*, pp. 65−69, Indianapolis, USA, Apr 13-16, IEEE Press

Jin, Y.; Branke, J. (2005). Evolutionary optimization in uncertain environments – a survey. *IEEE Transactions on Evolutionary Computation*, Vol. 9, No. 3, pp. 303−317

Magurran, A. E. (2004). *Measuring Biological Diversity,* Blackwell, Oxford

Meiyi, L.; Zixing, C.; Guoyun, S. (2004). An adaptive genetic algorithm with diversity-guided mutation and its global convergence property. *Journal of Central South University of Technology*, Vol. 11, No. 3, pp. 323−327

Morrison, R. W. (2004). *Designing Evolutionary Algorithms for Dynamic Environments*, Springer

Narendra, K. S.; Annaswamy, A. M. (2005). *Stable Adaptive Systems,* Dover Publications, Mineola

Nguyen, D. H. M.; Wong, K. P. (2003). Controlling diversity of evolutionary algorithms, *Proceedings of the Second International Conference on Machine Learning and Cybernetics*, pp. 775−780

Ogata, K. (1990). *Modern Control Engineering,* Prentice-Hall

Rao, C. R. (1982). Diversity and dissimilarity coefficients: a unified approach. *Theoretical Population Biology,* Vol. 21, pp. 24−43

Schwefel, H.- P. (1995). *Evolution and Optimum Seeking*, John Wiley, Chichester

Shimodaira, H. (2001). A diversity control-oriented-genetic algorithm (DCGA): performance in function optimization, *2001 IEEE Congress on Evolutionary Computation (CEC'2001)*, pp. 44–51, Seoul, Korea, IEEE Press

Simões A. ; Costa, E. (2001). On Biologically Inspired Genetic Operators: Transformation in the Standard Genetic Algorithm, *Proceedings of Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pp. 584−591, San Francisco, USA, July 7−11, Morgan Kaufmann Publishers

Smith, R. E.; Goldberg, D. E. (1992). Diploidy and dominance in artificial genetic search. *Complex Systems*, Vol. 6, No. 3, pp. 251−285

Solow, A.; Polasky, S.; Broadus, J. (1993). On the measurement of biological diversity. *Journal of Environmental Economics and Management,* Vol. 24, No. 1, pp. 60−68

Srinivas, M.; Patnaik, L. M. (1994). Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics,* Vol. 24, No. 4, pp. 656−667

Ursem, R. K.; Krink, T.; Jensen, M. T.; Michalewicz, Z. (2002). Analysis and modeling of control tasks in dynamic systems. *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 4, pp. 378–389

Wagg, D. J. (2003). Adaptive control of nonlinear dynamical systems using a model reference approach. *Meccanica,* Vol. 38, No. 2, pp. 227–238

Weinberg, W. (1908). Über den Nachweis der Vererbung beim Menschen. *Jahreshefte Verein f. vaterl. Naturk. in Wurtemberg*, Vol. 64, pp. 368−382

Weitzman, M. L. (1992). On diversity. *Quarterly Journal of Economics,* Vol. 107, No. 2, pp. 363−405

Wineberg, M.; Oppacher, F. (2003). Distances between populations, *The Proceedings of the Fifth Genetic and Evolutionary Computation Conference (GECCO'2003)*, pp. 1481−1492, Chicago, USA, July 12−16, Morgan Kaufmann Publishers

Wong, Y.- Y.; Lee, K.- H.; Leung, K.- S.; Ho, C.-W. (2003). A novel approach in parameter adaptation and diversity maintenance for genetic algorithms. *Soft Computing,* Vol. 7, No. 8, pp. 506–515

Wright, S. (1931). Evolution in Mendelian populations, *Genetics*, Vol. 16, pp. 97−159
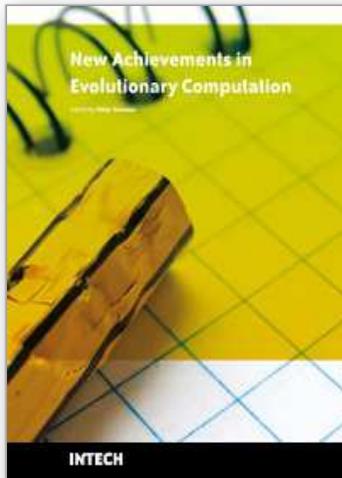
New Achievements in Evolutionary Computation

Edited by Peter Korosec

Evolutionary computation has been widely used in computer science for decades. Even though it started as far back as the 1960s with simulated evolution, the subject is still evolving. During this time, new metaheuristic optimization approaches, like evolutionary algorithms, genetic algorithms, swarm intelligence, etc., were being developed and new fields of usage in artificial intelligence, machine learning, combinatorial and numerical optimization, etc., were being explored. However, even with so much work done, novel research into new techniques and new areas of usage is far from over. This book presents some new theoretical as well as practical aspects of evolutionary computation. This book will be of great value to undergraduates, graduate students, researchers in computer science, and anyone else with an interest in learning about the latest developments in evolutionary computation.

INTECH
open science | open minds