XXXVI Ibero-Latin American
Congress on Computational
Methods in Engineering
CILAMCE
Rio de Janeiro, 22-25 Nov 2015

MIGRATION POLICIES TO IMPROVE EXPLORATION IN PARALLEL ISLAND MODELS FOR OPTIMIZATION VIA METAHEURISTICS

Thiago T. Magalhães

thiagotm@lncc.br

National Laboratory for Scientific Computing - LNCC/MCTI

Eduardo Krempser

krempser@lncc.br

Oswaldo Cruz Foundation - FIOCRUZ

Helio J.C. Barbosa

hcbm@lncc.br

National Laboratory for Scientific Computing - LNCC/MCTI

Abstract. *Island models are an interesting and powerful alternative in order to achieve at least two objectives in optimization: to increase the speed up of the algorithms (taking advantage of the growing computational power) and, at the same time, to improve the quality of the obtained solutions. Among many others, the "migration policy" is an important aspect that guide the behavior of the model. We suggested a useful topology of communication between the islands that compose the model, and employed 10 different policies, among well known and new suggestions of implementations. Comparisons were performed, applying a hybrid AG+DE model to solve a benchmark including 30 scalable problems with different levels of complexity, for 30 dimensions. The quality of the obtained results by the different techniques was studied, as well the distinct behaviors, in terms of sub-populations diversity.*

Keywords: *Metaheuristics; Hybrid algorithms; Parallel optimization*

1 Introduction and organization of the paper

Metaheuristics are a relevant class of methods in the context of optimization problems. They have been widely used to solve several hard problems, which could not easily be solved by classical methods (Dréo, 2006) (Yang, 2010) (Gendreau and Potvin, 2010). In particular, two features of population-based metaheuristics deserve to be highlighted: (1) high computational demand and (2) natural parallelism. These two aspects motivate the development of many parallel models and implementations in metaheuristics (Alba, 2005), and an important discovery in this area is the set of techniques that have the capacity to, at the same time, improve the speed up of the algorithms as well as the quality of the obtained solutions (Starkweather et al., 1991). This possibility opens the way, for example, to the treatment of increasingly larger and more complex problems (Cung et al., 2002), and to the use of the large computational power currently available and which increases daily. As a result, a large number of works aiming at the study of different proposals for parallel metaheuristics have been published. In general, these works suggest strategies and implementations, analyze sets of parameters, and compare the behavior and the performance of different possible implementations (Crainic and Toulouse, 2003).

In this same way, this work presents a study on a relevant aspect of one of the most common parallel models for optimization via metaheuristics, the island model. This approach constitutes a useful strategy in parallel metaheuristics. However, it presents a relatively large quantity of parameters and possible ways of implementation. As a result, there is a growing interest in the study of these possibilities, aiming at increasing its performance with respect to speed up and/or efficiency of optimization. In this sense, a very important aspect of this model is its migration policy. We expect that some possibilities of implementation for this issue can significantly affect the exploration process of the search space of candidate solutions and, consequently, the quality of the final results obtained by the optimization algorithm. Thus, a relevant research area corresponds to the design, analysis, and testing of the migration policy in island models, and this work is an effort in this area.

This paper is organized as follows: initially, in the Section 2, a general view of the islands model is presented, briefly. We will give a background constituted by a brief history, the theoretical aspects and some considerations about the model functioning, which will be relevant to the understanding of the objectives of this work. Then Section 3 presents a further history about this research area and the related literature, while exposing the central proposals and goals of this research. Details about our implementation, such as the metaheuristics applied, the model parameters and features will be explained in Section 4. Section 5 approaches the methodology of experimentation and lists the main analysis tools. Finally, Sections 6 and 7 list the principal results obtained, promotes a discussion, and a presents a brief set of conclusions.

2 Island models background

As the availability of multiple machines became a reality in the scientific world, and more and more applications and interesting results were obtained from them, the model today known as island model emerged in the context of metaheuristics. Initially, implementations of distributed genetic algorithms (Pettey et al., 1987) (Tanese, 1989) (Starkweather et al., 1991), already presenting the central features of island models, were proposed although not yet identified by this nomenclature. Some time later, the use of the expression "island model" to classify the distributed models in genetic algorithms (Schwehm, 1996) (Cantú-Paz, 1998) (Alba and Troya,

1999) appeared. The ambitious goal of the island model is to exploit the parallel hardware to speed up the algorithms and, at the same time, to increase the quality of the exploration in the search space of solutions [1]. As noted before, one immediate consequence of this is the growth in the range of problems that might be tackled by these algorithms.

Remembering that we applied this model to population-based metaheuristics, firstly a population of candidate solutions is created. However, unlike the canonical implementations, in island models the population is partitioned into several sub-populations, and mating restrictions arise. Each one of these sub-populations is named an island and, in parallel architectures, each one of these islands can be run in one processor or machine. A candidate solution (or individual) of an island can only interact (or mate) with another individual of its own island. However, there exists a fundamental operator named migration. It is responsible for transferring information (usually represented by a set of individuals) between the islands, according a series of parameters. Thus, while an individual of an island cannot directly interact with individuals of other islands, migration – that occurs periodically – ensures a certain level of communication between the sub-populations.

For this reason, we can say that island models are based on semi-isolated evolution (Augusto, 2009). The simplest representation of this iterative process is given in Algorithm 1. In this representation, control parameters for migration and metaheuristics are omitted. The most important consideration is that the common island models algorithm is composed of two parts: the isolated evolution periods, and the instants of information exchange (migration windows). The sub-populations evolve separately in the course of m iterations and, then, the migrations are performed. This process occurs n times until the total number of iterations previously established is reached [2].

Algorithm 1: Simple representation of a generic Island Model algorithm

```
m = migration interval
n = total iterations / migration interval
Initialize sub-populations
for i := 0 to n do
    for j := 0 to m do
        Perform a metaheuristic iteration in each island
    Migrate individuals
```

Concerning the control parameters of the model, we can form two groups: (*i*) general parameters and (*ii*) migration parameters (Skolicki, 2005). While the first group includes parameters from the metaheuristics that operate in each island and from the allowed computational budget (as population size, number of islands, total number of iterations...), the second group includes parameters and strategies concerning the migrations process. These parameters can be listed and described as follows (Skolicki, 2005):

- *Migration size*: quantity of individuals that migrate between the islands;

[1] In addition, it is important to note that this is a logical model. In other words, it is not dependent on the availabilty of multiple machines, and can be implemented in serial machines, but not necessarily with speed up gains, in this case.

[2] Despite this basic algorithm, is important to note that also there are implementations of adaptive approaches for migration intervals.

- *Migration interval*: quantity of metaheuristics iterations performed in isolation by each island between two migration windows – the variable m, in Algorithm 1;
- *Migration topology*: defines the neighbourhood relations between the islands. In other words, which island can send/receive individuals to/from which islands;
- *Migration synchronicity*: describes the way by which migrations occur, with respect to parallelism implementation. If the model is synchronous every island performs migration at the same time, and one island waits for another to perform migrations. If the model is asynchronous the migration windows can occur independently, for each island;
- ***Migration policy***: there should be a selection pressure concerning the migrations. Not every member of a sub-population should be able to migrate, and the criteria that define which individual migrates from one island to another exerts influence on the search process. According to the majority of authors, the migration policy concerns only those criteria.

As a different and important interpretation of those parameters, (Martin et al., 1997) refers to other aspects of the migration process. For example, in addition to the traditionally called migration policy, we can cite the *migrants assimilation policy*. The author highlights that, having decided the criteria by which the individuals are selected to be able to migrate, we need to decide how these migrants will be inserted into the new sub-populations. In addition, some authors highlighted the *migration material*. This parameter describes which information will be transfered between the islands. In literature, the most common strategy consists into transfer individuals. However, in this paper we present implementations based in exchange of individuals and other informations. For simplicity, we consider appropriate to visualize this parameter as one more aspect inside the migration policy. Thus, in our work the migration policy will not be considered a simple parameter, but rather a policy, which includes both questions about choice of migrants and strategies of assimilation. The principal aim of this study is to suggest and compare strategies for implementation of this policy, and how these strategies influence population diversity and the quality of the obtained solutions.

3 Related works and proposal

3.1 Related works

This section aims at presenting a brief history about efforts in the literature concerning the migration policies of island models followed by the policies suggested here.

Firstly, it is important to point out two different migrant assimilation strategies. In a first moment, we can find (Schwehm, 1996) describing the migration process in island models, for genetic algorithms. In his work, there is the concept of migration through copies. According to this idea, if an individual migrates from island A to island B, this individual continues to exist in island A. That is, a copy is inserted in population B, overwriting a given individual which will be selected among the individuals that constitute the population of island B. For convenience, we will name this approach as *copy-based* migration. In contrast, (Martin et al., 1997) describes the process of exchange of individuals, as being executed through the real transfer of individuals. In this approach, if an individual migrates from island A to island B, it no longer exists in island A, and occupies a space opened by some individual in B, which migrates from B to another island. For convenience, we will name this approach as *transfer-based* migration. Although the copy-based migrations have become a largely used strategy in island models (Starkweather et al., 1991) (Skolicki and De Jong, 2005) (Ruciński et al., 2010) (Vanneschi et al., 2012) (Apolloni et al., 2014), there are indications that this approach can negatively impact the population diversity. An interesting study, (Skolicki, 2005), reveals that copy-based implementations reduce drastically the global population diversity, whenever a migrations window occurs. However, no studies were found concerning the differences in the quality of the final optimization results between copy-based and transfer-based strategies of migration.

Irrespective of the migrant assimilation strategy adopted, the three simplest and most common strategies to perform the selection of migrants are described in (Martin et al., 1997) as: (1) random, (2) top 50% and (3) best. In the first, migrants are selected randomly from the sub-population. In the second, migrants are selected randomly, among the better half of the sub-population [3]. In the third, the migrant is always the best individual of the sub-population. Clearly, these three strategies present an increasing selection pressure to determine the migration capacity of an individual. The ease of implementation of the random criterion, together with performances which are not substantially worse than other traditional strategies, are probably responsible for the large number of implementations that use this approach (Cohoon et al., 1987) (Martin et al., 1997) (Skolicki and De Jong, 2005) (Apolloni et al., 2014). However, the preference for strategies that implement a certain selection pressure (Ruciński et al., 2010) (Vanneschi et al., 2012) is justified by the need to have migrant individuals that positively impact the evolutionary process. Random strategies can lead to the transfer of weak individuals, who die quickly in new sub-populations and are not able to influence the evolutionary process. On the other hand, highly elitist strategies –like the best strategy, for example– should be used with care and do not seem desirable. This is because such approaches can potentially transfer a select group of dominant individuals among many islands generating premature convergence of the evolutionary process. Sharing this same thought, for example, (Alba and Troya, 2000) defended random-based policies to prevent what has been called the “conqueror effect”. Finally, all these considerations seem to suggest that the balanced fitness-based strategies can be very beneficial to the quality of the obtained solutions (Cantú-Paz, 1999) (Cantú-Paz, 2001).

In addition to the traditional policies, there are studies that suggest the development of techniques based on feedback from the current population of candidate solutions (Pettey et al., 1987). For example, there are efficient adaptive implementations which use the information about the population of candidate solutions to automatically tune some migration parameters, with the passing of generations, and to transform the algorithm features (Noda et al., 2002) (Lardeux and Goëffon, 2010). Typically, this class of techniques presents as a disadvantage the relatively higher complexity of implementation. In contrast, there are techniques which benefit from the information obtained from the population not by changing migration parameters, but by guiding the evolution of the sub-populations. In general, these approaches make similarity calculations to perform an intelligent selection of migrants. In this way, (Magalhães et al., 2014) performed experiments with a strategy derived from average euclidean distances, where individuals more distant from their own sub-population have more chances to undergo migration.

[3] A useful variation of this strategy is the selection based on tournaments (Magalhães et al., 2014). In this approach, tournaments with population members are held, and the best fitness competitors are allowed to migrate. This variation is interesting, specially due to its low computational cost.

The aim of this strategy is not to increase the diversity inside each island, but rather increase the general difference between the islands. However, the employed tests were not sufficient to demonstrate a large advantage of using this approach. Other very interesting strategy in this sense is given by (Araujo et al., 2008). In opposition to the previous strategy, the so-called *multikulti algorithm* aims to maximize the diversity inside each sub-population and not between the islands. To ensure this, the migrant selection criteria is based on the measurement of the distance between each candidate and a representative of the target sub-population, besides the candidate fitness. Thus, the migrant selection is no longer based on information derived from its own sub- population, but on information derived from the target sub-population. In this way, an island can send an individual sufficiently different from the target sub-population, and sufficiently well evaluated, in terms of fitness. This approach derives from the concept that population heterogeneity is favorable to the evolution process, since it increases the possibilities of exploration of the search space and prevents a premature convergence. Additionally, it facilitates the migration of good individuals, which are the ones who have a decisive influence on the evolutionary process.

3.2 Proposal

In this work we implemented a hybrid islands model –which will be detailed in Section 4– aiming at comparing the performances when 10 different migration policies are employed. Due to space constraints, this paper focused on non-adaptive strategies. In the following, the implemented policies are listed and explained:

1. Random copy-based;
2. Random transfer-based;
3. Best-half copy-based;
4. Best-half transfer-based;
5. Higher median euclidean distance;
6. Lower median euclidean distance;
7. Best-half and higher median euclidean distance;
8. Best-half and lower median euclidean distance;
9. Fitness and higher distance to target measurement;
10. Fitness and lower distance to target measurement;

Each one was chosen for a reason and this section aims to clarify this.

Policies 1, 2, 3 and 4 are presented to allow for comparisons between (i) random-based and fitness-based classical strategies and (ii) copy-based and transfer-based strategies. At this point, it is important to note again that direct comparisons between copy-based and transfer-based implementations were not found in the literature. Strategies 3 and 4 are implemented through tournaments, where each tournament includes half of the population members. In copy-based strategies, the criteria used to choose the individual from the target population that will be replaced by the migrant from the source population is related with the criteria used to select migrants. In policy 1, a random individual is discarded and in policy 3 the worst individual from half of the sub-population is discarded.

Policies 5 and 6 calculate the median euclidean distance ϕ between the i-th individual of the population and all others. Then, tournaments are formed, each one including half of the population members, and the selected migrant is the one that maximizes ϕ in policy 5, or the one which minimizes ϕ, in policy 6. These policies are included to complement the tests performed in (Magalhães et al., 2014), by including an opposite approach. Through these tests, we will estimate the real advantages of the use of simple euclidean-distance-based implementations, in terms of quality of the obtained solutions. It will also be possible to observe the differences in results between these two opposite approaches. The first aims to maximize the distances between islands, allocating very different individuals in different islands. In contrast, the second one aims to maximize the distances between individuals inside each sub-population –in other words, maximizing sub-population diversity– allocating very similar individuals in different islands.

Policies 7 and 8 are similar to 5 and 6. However, each member that participates in the tournament is selected through fitness comparisons. Thus, these implementations are useful to analyse the influence of the inclusion of fitness criteria, side by side with similarity measures.

Finally, we suggested two policies which are inspired by concepts from the "multikulti" algorithms (Araujo et al., 2008), and enables the adjustment of the influences of both fitness and similarity measures. Differently from the other policies implemented in this paper, this strategy utilizes an information about target sub-populations in order to choose the migrants, which represents a population in a compact form and may be transfered between the islands before each migration. In our implementation, this information is a vector, where the i-th position is the median of all k values registered in the i-th position of the j-th individual, $0 < j < k$, with k being the total number of individuals in a sub-population. When an island receives the information about its target island, a value for ϕ, which is calculated following the Eq. (1),

$$\phi = \alpha \frac{fitness_j}{max(fitness)} + \beta \frac{dist_j}{max(dist)} \tag{1}$$

is attributed to each individual of the island. In Eq. (1), $dist_j$ represents the euclidean distance between the j-th individual and the information received from the target sub-population. This proposal is very flexible, due to the existence of the α and β coefficients. They may give different priorities to fitness and similarity measures, and can be reset to zero. If desirable, for example, β may be zero and we have a simple normalized fitness-based strategy, similar to policy 4, described above. On the other hand, if α is zero we have a normalized euclidean-distance-based approach. Concerning the setting of priorities, both coefficients may also assume several real values, where, ideally $-1 \leq \alpha \leq 1$ and $-1 \leq \beta \leq 1$. However, we used just two simple combinations for these two constants, as to avoid the need of new parameter adjustments. While $\alpha = -1$, due the employment of the model to solve minimization problems, we have chosen $\beta = 1$ for the policy 9 and $\beta = -1$ for the policy 10. In these configurations, both fitness and distance criteria have the same importance. The signal differences in β encourage the diversity between different islands or the diversity within each island. If $\beta = -1$, individuals more similar to the information have more chances to undergo migration. If $\beta = 1$, individuals less similar to the information have more chances to undergo migration. This configuration change generates a similar effect to that observed in the contrast between policies 7 and 8 .

4 Implementation details

We implemented a hybrid islands model. This approach was chosen based on previous works, which support the hypotheses that hybrid implementations achieve better results than non-hybrid (Zaferani and Teshnehlab, 2011) (Xin et al., 2012). Furthermore, (Magalhães et al., 2014) indicates that hybrid approaches that combine the two very popular metaheuristics Differential Evolution and Genetic Algorithm tend to outperform other parallel implementations, among which non-hybrid and hybrid approaches. This section briefly exposes the general features of these two metaheuristics and of our islands model, citing some related references.

4.1 Optimization Algorithms

Differential Evolution (DE) and Genetic Algorithm (GA) are stochastic and population-based metaheuristics often applied to global optimization. DE was proposed in (Storn and Price, 1997), originally to solve unconstrained optimization problems in continuous spaces. It has attracted growing attention in the literature (Neri and Tirronen, 2010) (Das and Suganthan, 2011), having been extended to solve other problems, including constrained (Mohamed and Sabry, 2012) and discrete problems (Sauer et al., 2011). GA was proposed by (Holland, 1992), and is based on Darwin's natural selection theory (Darwin, 1859) and on biological mutation and reproduction concepts (Bateson and Mendel, 1909) being the most popular evolutionary algorithm. Originally developed to handle discrete problems, their current applications also include real-valued problems (Wright, 1991) as well as a wide range of applications (Goldberg and Samtani, 1986) (Rajasekaran and Pai, 2003).

Since we implemented a hybrid model, where sub-populations that evolve according to different metaheuristics need to communicate among themselves, it was important to choose the same representation to all sub-populations. Thus, we implemented a real-coded representation, and in both DE and GA the i-th candidate solution is a vector $x = \{x_{i,1}, x_{i,2}, x_{i,3}, ..., x_{i,n}\}$, where $x_{i,j} \in \mathbb{R}$, that encodes a possible solution to the problem at hand. Here, n is the number of dimensions (usually the number of variables) from the objective function. In this way, the population of candidate solutions can be visualized as a matrix $X_{m\times n}$, where the element $x_{i,j}$ of this matrix represents the suggested value by the i-th individual to the j-th problem variable. In this context, is important to describe the DE and GA operators that guide the search through the space of possible solutions.

Concerning DE operators, we implemented two well known variants named *target to rand*, as described in (Qin et al., 2009) and *target to best*, as described in (Mezura-Montes et al., 2006). Respectively, these variants are represented in Eq. (2) and Eq. (3), were $\overline{x}_j$ is the j-th position from the target vector, $x_{r_1,j}$, $x_{r_2,j}$ and $x_{r_3,j}$ are the j-th positions from three random vectors being $r_1 \neq r_2 \neq r_3$, and $x_{best,j}$ is the j-th position from the better evaluated vector from the population. Finally, F is usually defined as $0 < F \leq 1$ and represents a weighting factor.

$$\overline{x}_j = x_{i,j} + F(x_{r_3,j} - x_{i,j}) + F(x_{r_1,j} - x_{r_2,j}) \tag{2}$$

$$\overline{x}_j = x_{i,j} + F(x_{best,j} - x_{i,j}) + F(x_{r_1,j} - x_{r_2,j}) \tag{3}$$

In turn, with respect to the GA, due to the the real-valued representation we implemented continuous move operators. Thus, the *BLX-α crossover* operator (Eshelman and Schaffer, 1993), described in Eq. (4), was employed where α is a constant, $x_{i_{(1)},j}$ and $x_{i_{(2)},j}$ are the selected parents and $I = \max(x_{i_{(1)},j}, x_{i_{(2)},j}) - \min(x_{i_{(1)},j}, x_{i_{(2)},j})$. Concerning the mutation operation, we implemented the *non-uniform mutation operator* (Michalewicz, 1992), defined by the Eq. (5). Both operators, as well the values of their parameters were chosen according to a set of experiments performed in (Herrera et al., 1998).

$$\overline{x}_j \in \mathbb{R} \mid \min(x_{i_{(1)},j}, x_{i_{(2)},j}) - I\alpha < \overline{x}_j < \max(x_{i_{(1)},j}, x_{i_{(2)},j}) + I\alpha \tag{4}$$

$$x_{i,\overline{j}} = \begin{cases} x_{i,j} + \Delta(t, MAX - x_{i,j}) & \text{if } \tau = 0 \\ x_{i,j} - \Delta(t, x_{i,j} - MIN) & \text{if } \tau = 1 \end{cases} \tag{5}$$

4.2 Model features and parameters

The hybrid islands model was coded in the C++ programming language, using the Message Passing Interface (MPI) to implement the real parallelism with messages exchange. MPI can administrate the processors and machines and allows the execution of the same implementation in serial machines and in parallel architectures with different quantities of nodes. In this work, we adopted a computational budget according to the one previously established by the CEC 2014 Conference (Liang et al., 2013), for the evaluation of minimization algorithms when applied to the benchmark problems adopted here. We chose 6 nodes (or islands) for our model. For each island, a population of 100 individuals is evolved along 500 iterations of the specific metaheuristic adopted. This configuration results in a total of 300000 executions of the evaluation/fitness function. Concerning the model parameters, while we implemented several migration policies, we have chosen another migration parameters through indication from the literature and, also, after a small set of initial tests.

With respect to migration size and interval, the literature tends to indicate that these parameters are the two most important ones in an islands model (Skolicki, 2005), and should be chosen together (Starkweather et al., 1991) (Schwehm, 1996). This is probably related to the migration pattern, that represents the degree of connectivity between the nodes of the model (Martin et al., 1997) and which is largely impacted by these two parameters. Despite the existence of both fixed and variable values for both parameters, some papers indicate that there are not large differences between these approaches, in terms of quality of the obtained solutions (Martin et al., 1997). Thus, for simplicity, we adopted fixed values. Some researches show that migrations sizes within the range of 5% and 25% of each sub-population size are good ones (Martin et al., 1997) (Apolloni et al., 2014). Other researchers indicate that good values vary between 10% and 20% of the sub-population size (Schwehm, 1996) (Skolicki and De Jong, 2005) (Ruciński et al., 2010) (Vanneschi et al., 2012). After some initial tests, we set migration size as 5% of each sub-population size (here 5 individuals for each migration). This choice also highlights the distributed character of the evolution process, maximizing the influence of the islands model implementation in results, when compared to traditional models. With respect to the migration interval, there are very different suggestions in the literature. While some works employ large migration intervals (Ruciński et al., 2010) (Apolloni et al., 2014), other researchers prefer intervals of 10 generations between two migrations (Skolicki, 2005) (Vanneschi et al., 2012). There are implementations where migrations can occur for each generation (Schwehm, 1996) and also there are adaptive strategies (Braun, 1991) (Kröger et al., 1991). Here we chose to establish a migration interval of 10 generations. This way we ensure a certain level of communication between islands and, at the same time, avoid premature convergence due to too short intervals (Skolicki and De Jong, 2005).

Concerning the migration topology, there is a large set of possibilities reported in the literature. If our algorithm is to be executed in a heterogeneous hardware, we can choose the topology, according to the architecture (Martin et al., 1997) (Bujok and Tvrdik, 2011), aiming to maximize the efficiency, in terms of speed up and use of the hardware. Also, there are adaptive and dynamic implementations (Starkweather et al., 1991) (Biazzini and Montresor, 2010), among many others (Ruciński et al., 2010). However, apparently the most common topology is the ring topology (Tasoulis et al., 2004) (Izzo et al., 2009) (Weber et al., 2010) (Apolloni et al., 2014), including dynamic versions (Starkweather et al., 1991) (Ruciński et al., 2010). In this same way, we suggested a ring-based dynamic topology, with a stochastic term. In our approach, each island i has two neighbors, as well as the traditional ring topologies. However, they are the islands $i-r$ and $i+r$, cyclically, where $1 \leq r \leq ni$ is a random integer value which is updated at each migration window, and ni is the number of islands. Thus, the two neighbors of an island i are given by $mod(i + r, ni)$ and $mod(i + ni - r, ni)$. Three different configurations for this topology are represented in Fig. 1. Note that if $r = 1$ we have a traditional ring topology. The main advantage of this approach is the fact that as r is sorted for each migration window, all islands have the same distance between themselves and all islands have the same chances of communicating with any other island. This behavior is useful for this work, because allows, for example, that a weak island, geographically enclosed by two strong islands, is able to promote migrations with another weak island, that can benefit from one of their individuals. In this sense, each individual has the same chances to migrate to weak (and low competitive) sub-populations, or to strong (and high competitive) sub-populations.

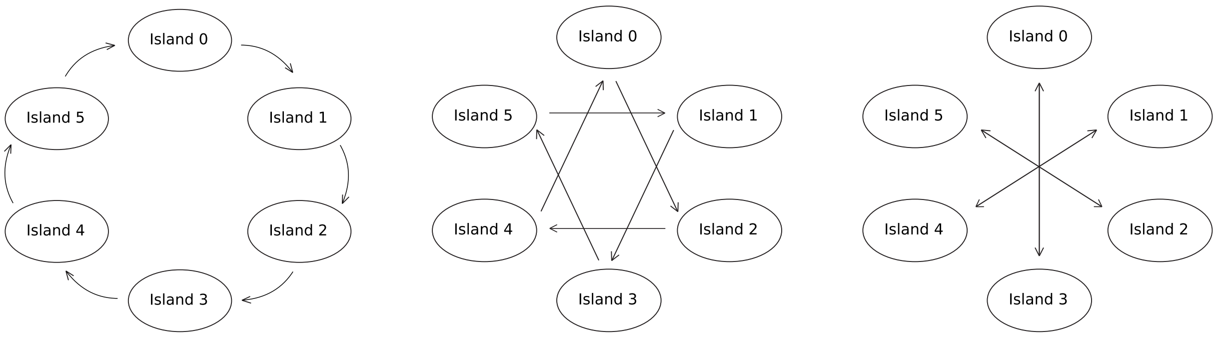

Figure 1: Random ring topology, from left to right: with r=1, r=2 and r=3

Concerning the synchronization of the migrations, asynchronous models tend to lead to more efficient implementations, in terms of speed up. However, they are not ideal for theoretical studies, due the higher difficulty in behavior analysis and due to the larger influence of external factors in the results (Skolicki, 2005) (Ruciński et al., 2010). As this paper is not focused on speed up analysis, we implemented a synchronous model. In this strategy, the migration windows occur at the same time, in all islands.

Finally, with respect to the metaheuristics parameters, we adopted an heterogeneous strategy. Researchers have reported that the use of different parameter sets can be an interesting strategy for optimization using distributed metaheuristcs (Gong and Fukunaga, 2011) (García-Valdez et al., 2014). Thus, for each metaheuristic parameter, we defined a range of possible values. In the beginning of the evolution process each island chose a value within this range, randomly. The expectation is that the existence of different parameters for the islands generates a more diverse search in the space of solutions, together with the heterogeneity in terms of different metaheuristics, increasing the quality of the optimization. The ranges are based in the literature and in previous experiments and are listed in Table 1 where the labels 2 and 3 of the variants from DE are related to (Eq. 2) and (3), respectively.

DE			GA		
Parameter	Lower	Upper	Parameter	Lower	Upper
CR	0.9	1.0	Tournament size	3	7
F	0.6	0.8	Mutation probability	0.03	0.08
Variant	2	3	α	0.12	0.17
			β	2	4

Table 1: Metaheuristics parameters

5 Experiments and methodology

Each one of the 10 configurations of our model was applied to solve 30 scalable problems, in the 30-dimensional configuration, from the benchmark presented in the CEC 2014 competition on single objective real-parameter numerical optimization (Liang et al., 2013). This problem set is composed by 3 unimodal functions ($f1$, $f2$, $f3$), 13 simple multimodal functions ($f4$, $f5$, $f6$, ..., $f16$), 6 hybrid functions ($f17$, $f18$, $f19$, ..., $f22$) and 8 composition functions ($f23$, $f24$, $f25$, ..., $f30$). Furthermore, due the stochastic features of our algorithms, each experiment was performed 30 times. We adopted basic statistical informations such as median and mean and employed the Wilcoxon Statistical Test (Kruskal, 1957).

One relevant feature of a set of candidate solutions is its heterogeneity (Araujo et al., 2008). To analyze this aspect, we determined a measure of the population heterogeneity, given by

$$\varphi = \frac{1}{m-1}\sum_{i=m}^{m}\sqrt{\sum_{j=1}^{n}(x_{best,j} - x_{i,j})^2}. \tag{6}$$

φ is the average euclidean distance between the best individual and each other individual in the population. The larger φ, the larger is the heterogeneity of a given sub-population.

To visualize the perfomance differences, we employed performance profiles. Performance profiles are an analytical tool for the visualization and interpretation of the results of benchmark experiments (Dolan and More, 2002). They are used here to provide a concise graphical presentation of the relative behaviour of the different combinations of techniques studied. Considering a set of algorithms A and a set of test-problems P, the performance of algorithm $a \in \{1, ..., n_a\}$ in problem $p \in \{1, ..., n_p\}$ will be denoted by $t_{p,a}$. Here, the performance indicator $t_{p,a}$ (larger values are better) is the inverse of the median objective function value found by algorithm a in test-problem p in 30 runs. The performance ratio $r_{p,a}$ is given by

$$r_{p,a} = \frac{t_{p,a}}{\min\{t_{p,a} : a \in A\}}. \tag{7}$$

Performance profiles, are defined by (Dolan and More, 2002) as

$$\rho_a(\tau) = \frac{1}{n_p}\left|\{p \in P : r_{p,a} \leq \tau\}\right|. \tag{8}$$

where $|A|$ denotes the cardinality of a set A. Then $\rho_a(\tau)$ is the probability that the performance ratio $r_{p,a}$ of algorithm $a \in A$ is within a factor $\tau \geq 1$ of the best possible ratio.

Performance profiles have a number of useful properties (Dolan and More, 2002; Barbosa et al., 2010). Here the area under the ρ_a curve (AUC) will be used as an indicator of the overall performance of algorithm a in the problem set P: the larger the AUC, the better the algorithm.

6 Results and discussion

Firstly, the analysis of the results is based on mean and median measurements of the independent runs, for the same problem and algorithm. We performed the Wilcoxon Test, considering 3 significant digits, with the confidence level of 95%. According to these calculations, for 3 problems ($f3$, $f7$ and $f23$) no policy presented results significantly superior to the others. In the $f23$ problem, the Wilcoxon Test indicated that both policies 2 and 10 obtained the best results. Concerning the other problems, our measurements indicate a higher advantage for the policy 3, in general. The policy 3 presented the best mean results in 19 problems, and also the best median result in 19 problems. The policy 4, similar to 3, offered the best mean result in 3 problems, and also the best median result in 3 problems. The policy 5 presented both best mean and best median for problem 27, with police 9 very close. Additionally, the policy 9 was very efficient in solving problems 29 and 30. However, despite this difference between the quantity of best median and best mean obtained by the police 3 and another, the differences in terms of quality of optimization were not so high. This fact can be visualized, in a first moment, in Fig. 2, which represents the performance profile for the all 10 implemented policies.

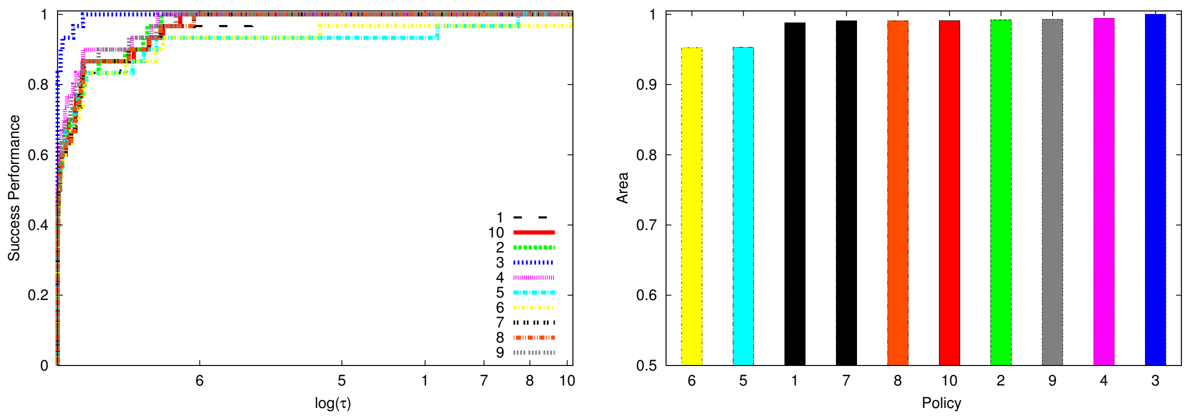

Figure 2: Performance profile for all policies implemented and all functions

Concerning the general behavior of the model the expectation is that, with the course of the generations, the candidate solutions are optimized and the sub-populations become more and more homogeneous, as the global optimum is approached. Different policies have different strategies to guide the diversity of the sub-populations, but our results seem to indicate that the greater proximity to the global optimum really exerts a high influence in the heterogeneity of the sub-populations. This effect can occur due the emergence of a growing number of dominant individuals, which quickly spread their features across the population. The expected behavior when the global optimum is near, is registered in Fig. 3, which exposes the fitness and heterogeneity variations of some sub-population to solve the problem $f1$, for each policy implemented. On the other hand, the expected behavior when the sub-populations cannot find a solution close to the global optimum is presented in Fig. 4, which exposes the fitness and heterogeneity variations of some sub-population when solving problem $f10$ [4].

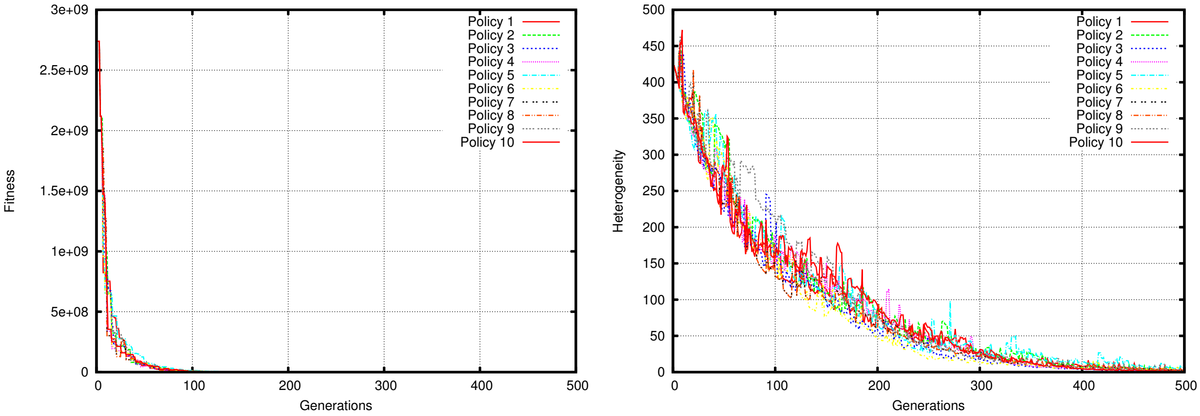

Figure 3: Fitness and heterogeneity variations for all policies, to solve the problem $f1$

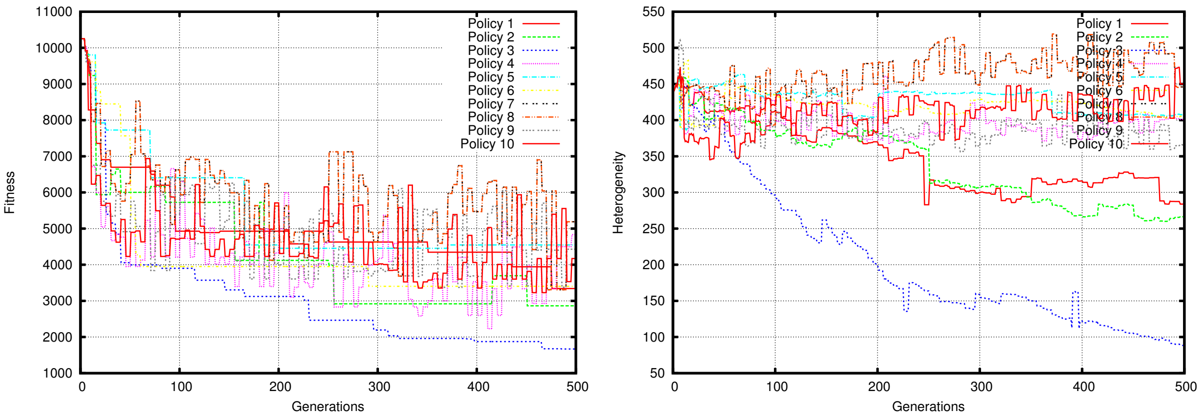

Figure 4: Fitness and heterogeneity variations for all policies, to solve the problem $f10$

Still regarding Fig. 4, we can see a large fitness difference between policy 3 and others. However, before commenting these reports and the performance profile exposed in Fig. 2, it is very important to highlight the differences between the problems – see the Section 5. This is because different strategies have better or worst performances according to problem complexity. In this context, an interesting result is demonstrated in Fig. 5. This figure presents a performance profile for all 10 policies, but considering only the 8 composition functions, which are more complex problems. The areas under the curves show that the policy 9 presents the better results, in this case. This conclusion and the performance differences between the strategies are very distinct from those obtained via Fig. 2.

Firstly, analyzing Fig. 2, we can make some conclusions for general problems. The position of random-based strategies is expected. When we study a large benchmark containing various types of problems, random-based strategies seem to achieve mediocre results. In general, to migrate random individuals does not seem to be a specially good strategy to solve a specific

[4] In islands models, differently from canonical models, the graph of fitness evolution of a sub-population can present positive as well as negative variations. That is because a sub-population may lose good individuals to another sub-population at every migration window.

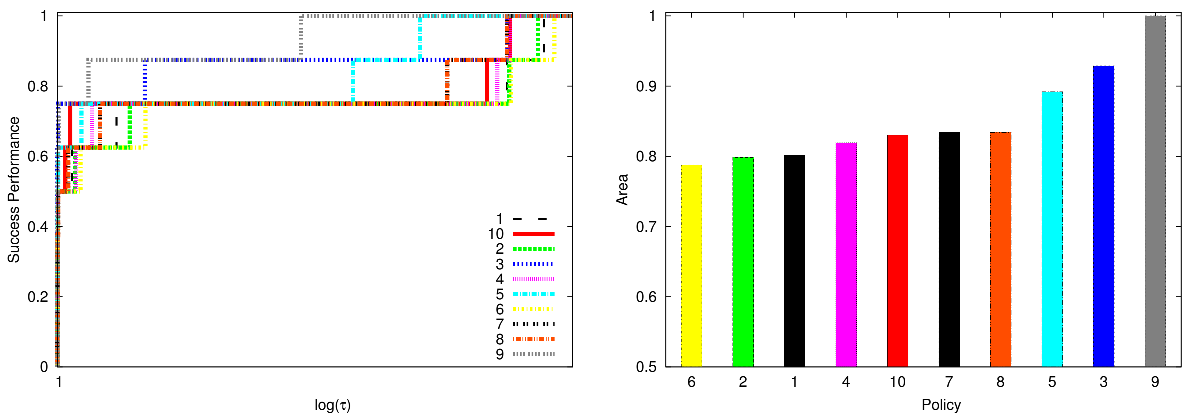

Figure 5: Performance profile for all policies implemented and only composite functions

problem. And, at the same time, it is not a specially bad strategy to solve a specific problem. The relatively poor results obtained by policies 5 and 6 corroborate results from the literature. Both policies are similarity-only based strategies. The results suggest that migration policies require a fitness based criterion, together with other possible criteria. This is useful to avoid the migration of very week individuals, which cannot exert a good influence in the evolution of the sub-populations. If many migrants can not reproduce, the island model evolves the candidate solutions like to an isolated model, due the inefficiency of the migrations.

Observing Fig. 5, the comparisons between the results obtained by the policies 3, 4, 5, 6 and 9 deserve special attention. Firstly, policy 3 presents the best results, when all problems are considered, and the second best results, when only the composite functions are included. This efficiency is obtained with a very simple implementation, and is one of the first strategies suggested in the literature. This is because fitness-only based strategies, as policies 3 and 4, strongly increase the selection pressure of the model, in particular policy 3, which is a copy-based policy. This strategy emphasizes exploitation over exploration, by overwriting promising individuals over weak individuals. This behaviour can be observed in Fig. 6, which focuses on the range of iterations between 300 and 350 for problems 1 and 10. In both problems we can visualize the lower heterogeneity when policy 3 is applied. In case of problem 10, this difference is even larger, as the sub-populations are not close to the global optimum and, as a result, the effect of domination caused by the existence of specially good candidate solutions does not arise. Without such effect, the feature of high exploitation of policy 3 is more highlighted. At first sight it seems a bad strategy, because it reduces the population diversity. However, when the parameter set of the model is not excessively favourable to exploitation, the increase in the selection pressure caused by this approach may be a useful factor to solve many problems, specially the simpler ones.

In case of more complex problems, policy 3 is considerably outperformed by policy 9. In addition, policy 4 –similar to the previous policy, but with lower selection pressure due the transfer-based implementation– is outperformed by a set of other policies. According to Fig. 5, for complex problems it is specially interesting to explore strategies that employ measurements of both fitness and similarity. The Fig. 7 exposes fitness and heterogeneity variations when solving the composition problem $f29$. As expected, the policy 3 maintains a lower heterogeneity within sub-populations and the policy 9 generates a higher heterogeneity inside the sub-populations. Both policies achieve good performances, but policy 9 is superior. We can

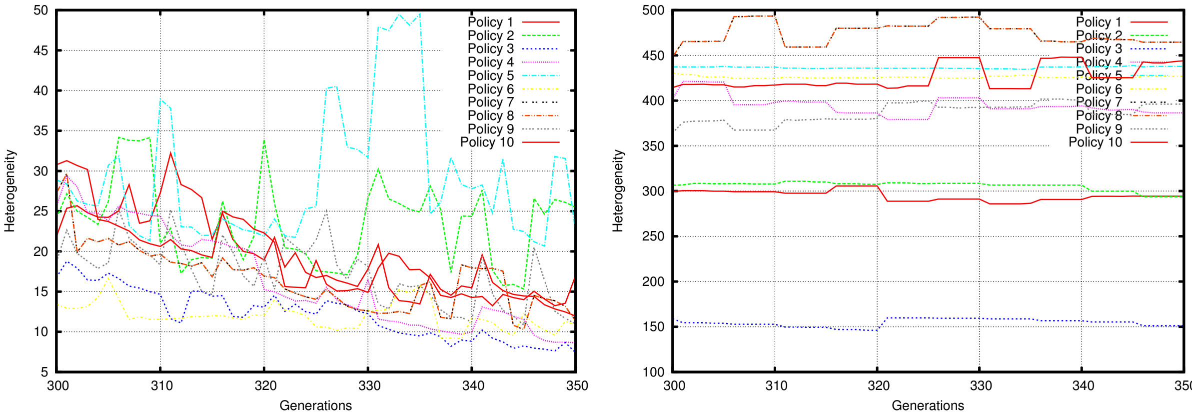

Figure 6: Heterogeneity variations for all policies, to solve problems $f1$ and $f10$

note, in addition, that the difference between the best and the second-best policies is much higher in Figure 5 than in Fig. 2.

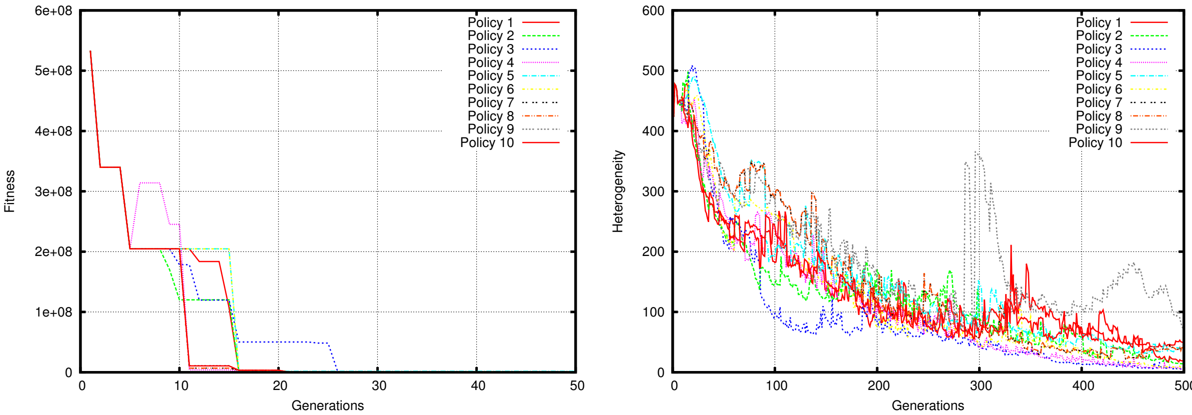

Figure 7: Fitness and heterogeneity variations for all policies, to solve the problem $f29$

Finally, an important consideration is related to the comparison of the performances obtained by the pairs of policies that consider similarity measurements, with respect to the composite functions. In this sense, the better evaluated pair of policies are 5, 6, and 9, 10. Policies 5 and 9 outperform their pairs 6 and 10, respectively. This result join new and previous findings in global optimization with island models. While the literature suggests policies that prize the diversity inside each sub-population as a strategy to ensure an efficient exploration of the search space, the policy 5, which exploits the opposite strategy, outperformed its pair, the policy 6. On the other hand, the better results obtained by the policy 9 indicate that the appreciation of the heterogeneity inside each sub-population may be the best strategy, outperforming all others – including the policy 5. This result corroborates other studies (Araujo et al., 2008). The idea behind the policies that increase the distance between different islands, and not between different individuals from the same island, is to encourage each island to explore a different region of the search space. A surprise is the relative difficulty to identify a pattern in the behavior of this strategy. While Fig. 7 clearly shows the policy 9 stimulating the heterogeneity inside the islands, Fig. 8 seems to demonstrate the opposite behavior, in comparison with its opposite pair, the policy 10. A possible reason for this is related to the proximity to the higher global optimums. According this figure, the behavior is in compliance with the expected in the first iterations of the evolution process. However, as the policy 9 approaches more quickly the best regions of the search space, the diversity also tends to decrease more quickly, with the course of the generations.

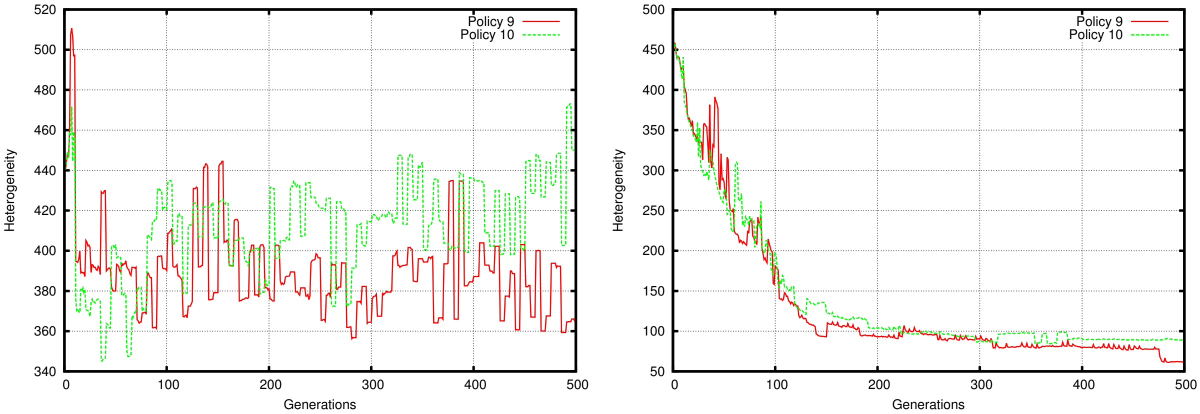

Figure 8: Heterogeneity variation for policies 9 and 10, to solve problems $f10$ **and** $f20$

7 Conclusions

Different strategies present different behaviors concerning the evolution of diversity inside the sub-populations and our results demonstrated that problems from different classes should be solved by different strategies. Policies based only in fitness comparisons, which increase the selection pressure, achieve good results both for copy-based and transfer-based implementations. To solve the basic problems, the best-half copy-based police obtained the best results and this indicates that elitist policies may be good strategies. On the other hand, policies based only in similarity measurements tend to obtain poor results, probably due the high probability of the transfer of weak individuals. In general, one of the policies suggested in this paper, which employs a normalized combination of fitness and similarity measurements also presented specially good results. But its performance was specially satisfactory when we compare results for more complex problems, thus indicating that this strategy should be further studied. Additionally, our results allows interesting conclusions about the relation between diversity inside the islands and differentiation of sub-populations. Following the literature, policies that aims to maintain the heterogeneity inside the sub-populations provided better results. However, is important to note that a policy based on the appreciation of the distances between the islands and not of the heterogeneity inside the sub-populations – the policy 5 obtained results considerably better than the policy 6 in solving the composite functions – also obtained interesting results and should be more studied. Finally, among all this, we can conclude that the exploration of elitist strategies may be efficient to solve more simple problems, while implementations based on similarity and fitness comparisons may be important tools to optimize more complex functions.

8 Acknowledgments

The authors would like to thank the support from CNPq (grants 132789/2014-0 and 310778/2013-1).

Bibliography

Alba, E. (2005). *Parallel metaheuristics: a new class of algorithms*, volume 47. John Wiley & Sons.

Alba, E. and Troya, J. M. (1999). A survey of parallel distributed genetic algorithms. *Complexity*, 4(4):31–52.

Alba, E. and Troya, J. M. (2000). Influence of the migration policy in parallel distributed gas with structured and panmictic populations. *Applied Intelligence*, 12(3):163–181.

Apolloni, J., García-Nieto, J., Alba, E., and Leguizamón, G. (2014). Empirical evaluation of distributed differential evolution on standard benchmarks. *Applied Mathematics and Computation*, 236:351–366.

Araujo, L., Guervós, J. J. M., Cotta, C., and de Vega, F. F. (2008). Multikulti algorithm: Migrating the most different genotypes in an island model. *arXiv preprint arXiv:0806.2843*.

Augusto, D. A. (2009). *Programao Gentica multi-populacional e co-evolucionria para classificao de dados*. PhD thesis, Universidade Federal do Rio de Janeiro.

Barbosa, H. J. C., Bernardino, H. S., and da Motta Salles Barreto, A. (2010). Using performance profiles to analyze the results of the 2006 CEC constrained optimization competition. In *IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE.

Bateson, W. and Mendel, G. (1909). *Mendel's principles of heredity*. Putnam's.

Biazzini, M. and Montresor, A. (2010). Gossiping differential evolution: a decentralized heuristic for function optimization in p2p networks. In *Parallel and Distributed Systems (ICPADS), 2010 IEEE 16th International Conference on*, pages 468–475. IEEE.

Braun, H. (1991). On solving travelling salesman problems by genetic algorithms. In *Parallel Problem Solving from Nature*, pages 129–133. Springer.

Bujok, P. and Tvrdik, J. (2011). Parallel migration models applied to competitive differential evolution. In *Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2011 13th International Symposium on*, pages 306–312. IEEE.

Cantú-Paz, E. (1998). A survey of parallel genetic algorithms. *Calculateurs paralleles, reseaux et systems repartis*, 10(2):141–171.

Cantú-Paz, E. (1999). Migration policies and takeover times in genetic algorithms. In *Proc. Genetic and Evolutionary Computation Conference, Morgan Kaufmann (ISBN 1558606114)*.

Cantú-Paz, E. (2001). Migration policies, selection pressure, and parallel evolutionary algorithms. *Journal of heuristics*, 7(4):311–334.

Cohoon, J. P., Hegde, S. U., Martin, W. N., and Richards, D. (1987). Punctuated equilibria: a parallel genetic algorithm. In *Genetic algorithms and their applications: Proc of the 2nd Intl. Conf. on Genetic Algorithms, MIT, Cambridge, MA*.

Crainic, T. G. and Toulouse, M. (2003). *Parallel strategies for meta-heuristics*. Springer.

Cung, V., Martins, S. L., Ribeiro, C. C., and Roucairol, C. (2002). Strategies for the parallel implementation of metaheuristics. In Ribeiro, C. C. and Hansen, P., editors, *Essays and Surveys in Metaheuristics*, pages 263–308. Springer.

Darwin, C. (1859). On the origins of species by means of natural selection. *London: Murray*.

Das, S. and Suganthan, P. N. (2011). Differential evolution: a survey of the state-of-the-art. *Evolutionary Computation, IEEE Transactions on*, 15(1):4–31.

Dolan, E. D. and More, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213.

Dréo, J. (2006). *Metaheuristics for hard optimization: methods and case studies*. Springer Science & Business Media.

Eshelman, L. J. and Schaffer, J. D. (1993). Real-coded genetic algorithms and interval-schemata. *Foundation of Genetic Algorithms 2*, pages 182–202.

García-Valdez, M., Trujillo, L., Merelo-Guérvos, J. J., and Fernández-de Vega, F. (2014). Randomized parameter settings for heterogeneous workers in a pool-based evolutionary algorithm. In *Parallel Problem Solving from Nature–PPSN XIII*, pages 702–710. Springer.

Gendreau, M. and Potvin, J. (2010). *Handbook of metaheuristics*, volume 2. Springer.

Goldberg, D. E. and Samtani, M. P. (1986). Engineering optimization via genetic algorithm. In *Electronic Computation (1986)*, pages 471–482. ASCE.

Gong, Y. and Fukunaga, A. (2011). Distributed island-model genetic algorithms using heterogeneous parameter settings. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 820–827. IEEE.

Herrera, F., Lozano, M., and Verdegay, J. L. (1998). Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. *Artificial Intelligence Review*, 12(4):265–319.

Holland, J. H. (1992). Genetic algorithms. *Scientific American*, 267(1):66–72.

Izzo, D., Rucinski, M., and Ampatzis, C. (2009). Parallel global optimisation meta-heuristics using an asynchronous island-model. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, pages 2301–2308. IEEE.

Kröger, B., Schwenderling, P., and Vornberger, O. (1991). Parallel genetic packing of rectangles. In *Parallel Problem Solving from Nature*, pages 160–164. Springer.

Kruskal, W. H. (1957). Historical notes on the wilcoxon unpaired two-sample test. *Journal of the American Statistical Association*, 52(279):356–360.

Lardeux, F. and Goëffon, A. (2010). A dynamic island-based genetic algorithms framework. In *Simulated Evolution and Learning*, pages 156–165. Springer.

Liang, J. J., Qu, B., and Suganthan, P. N. (2013). Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization. *Computational Intelligence Laboratory*.

Magalhães, T. T., Nascimento, L. H. C., Krempser, E., Augusto, D. A., and Barbosa, H. J. C. (2014). Hybrid metaheuristics for optimization using a parallel islands model.

Martin, W. N., Lienig, J., and Cohoon, J. P. (1997). Island (migration) models: evolutionary algorithms based on punctuated equilibria. *Handbook of evolutionary computation*, 6(3).

Mezura-Montes, E., Velázquez-Reyes, J., and Coello Coello, C. A. (2006). A comparative study of differential evolution variants for global optimization. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 485–492. ACM.

Michalewicz, Z. (1992). Genetic algorithms + data structures = evolution programs. *Springer-Verlag, New York*.

Mohamed, A. W. and Sabry, H. Z. (2012). Constrained optimization based on modified differential evolution algorithm. *Information Sciences*, 194:171–208.

Neri, F. and Tirronen, V. (2010). Recent advances in differential evolution: a survey and experimental analysis. *Artificial Intelligence Review*, 33(1-2):61–106.

Noda, E., Coelho, A. L. V., Ricarte, I. L. M., Yamakami, A., Freitas, A., et al. (2002). Devising adaptive migration policies for cooperative distributed genetic algorithms. In *Systems, Man and Cybernetics, 2002 IEEE International Conference on*, volume 6, pages 6–pp. IEEE.

Pettey, C. B., Leuze, M. R., and Grefenstette, J. J. (1987). Parallel genetic algorithm. In *Genetic algorithms and their applications: Proc of the 2nd Intl. Conf. on Genetic Algorithms, MIT, Cambridge, MA*.

Qin, A. K., Huang, V. L., and Suganthan, P. N. (2009). Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, 13(2):398–417.

Rajasekaran, S. and Pai, G. A. V. (2003). *Neural networks, fuzzy logic and genetic algorithm: synthesis and applications (with cd)*. PHI Learning Pvt. Ltd.

Ruciński, M., Izzo, D., and Biscani, F. (2010). On the impact of the migration topology on the island model. *Parallel Computing*, 36(10):555–571.

Sauer, J. G., dos Santos Coelho, L., Mariani, V. C., de Macedo Mourelle, L., and Nedjah, N. (2011). A discrete differential evolution approach with local search for traveling salesman problems. In *Innovative Computing Methods and Their Applications to Engineering Problems*, pages 1–12. Springer.

Schwehm, M. (1996). Parallel population models for genetic algorithms. *Universität Erlangen-Nürnberg*, pages 2–8.

Skolicki, Z. (2005). An analysis of island models in evolutionary computation. In *Proc. of the 2005 Workshops on Genetic and Evolutionary Computation*, pages 386–389. ACM.

Skolicki, Z. and De Jong, K. (2005). The influence of migration sizes and intervals on island models. In *Proc. of the 2005 Conference on Genetic and Evolutionary Computation*, pages 1295–1302. ACM.

Starkweather, T., Whitley, D., and Mathias, K. (1991). *Optimization using distributed genetic* *algorithms*. Springer.

Storn, R. and Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359.

Tanese, R. (1989). Distributed genetic algorithms. In *Proceedings of the third international conference on Genetic algorithms*, pages 434–439. Morgan Kaufmann Publishers Inc.

Tasoulis, D. K., Pavlidis, N. G., Plagianakos, V. P., and Vrahatis, M. N. (2004). Parallel differential evolution. In *IEEE Congress on Evolutionary Computation*, pages 2023–2029.

Vanneschi, L., Codecasa, D., and Mauri, G. (2012). An empirical study of parallel and distributed particle swarm optimization. In *Parallel Architectures and Bioinspired Algorithms*, pages 125–150. Springer.

Weber, M., Tirronen, V., and Neri, F. (2010). Scale factor inheritance mechanism in distributed differential evolution. *Soft Computing*, 14(11):1187–1207.

Wright, A. H. (1991). Genetic algorithms for real parameter optimization. *Foundations of genetic algorithms*, 1:205–218.

Xin, B., Chen, J., Zhang, J., Fang, H., and Peng, Z. (2012). Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: a review and taxonomy. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(5):744–767.

Yang, X. (2010). *Engineering optimization: an introduction with metaheuristic applications*. John Wiley & Sons.

Zaferani, E. J. and Teshnehlab, M. (2011). The parallel evaluation strategy approach to solving optimization problems. *International Conference on Computer and Software Modeling*.