

A Study on the Configuration of Migratory Flows in Island Model Differential Evolution

Rodolfo Ayala Lopes
Department of Electrical
Engineering
Federal University of Minas
Gerais
Belo Horizonte, MG, Brazil
rodolfo.ufop@gmail.com

Rodrigo C. Pedrosa Silva
Department of Electrical and
Computer Engineering
McGill University
Montreal, Canada
rodrigo.silva@mail.mcgill.ca

Alan R. R. Freitas
Department of Computer
Science
Federal University of Ouro
Preto
Ouro Preto, MG, Brazil
alandefreitas@iceb.ufop.br

Felipe Campelo
Department of Electrical
Engineering
Federal University of Minas
Gerais
Belo Horizonte, MG, Brazil
fcampelo@ufmg.br

Frederico G. Guimarães
Department of Electrical
Engineering
Federal University of Minas
Gerais
Belo Horizonte, MG, Brazil
fredericoguimaraes@ufmg.br

ABSTRACT

The Island Model (IM) is a well known multi-population approach for Evolutionary Algorithms (EAs). One of the critical parameters for defining a suitable IM is the *migration topology*. Basically it determines the Migratory Flows (MF) between the islands of the model which are able to improve the rate and pace of convergence observed in the EAs coupled with IMs. Although, it is possible to find a wide number of approaches for the configuration of MFs, there still is a lack of knowledge about the real performance of these approaches in the IM. In order to fill this gap, this paper presents a thorough experimental analysis of the approaches coupled with the state-of-the-art EA Differential Evolution. The experiments on well known benchmark functions show that there is a trade-off between convergence speed and convergence rate among the different approaches. With respect to the computational times, the results indicate that the increase in implementation complexity does not necessarily represent an increase in the overall execution time.

Categories and Subject Descriptors

I.2.6 [Learning]: Parameter learning; G.1.6 [Optimization]: Global Optimization

General Terms

Algorithms, Performance

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.
Copyright 2014 ACM 978-1-4503-2662-9/14/07 ...\$15.00.
<http://dx.doi.org/10.1145/2598394.2605439>.

Keywords

Island Model; Migration Topology; Configuration of Migratory Flows; Differential Evolution

1. INTRODUCTION

The Island Model (IM) is a multi-population approach for Evolutionary Algorithms (EAs) [6, 17] originally inspired by the model of punctuated equilibrium [9]. The first studies about IMs connected them with Genetic Algorithms (GAs) [9, 22] and it has been shown that they are able to largely enrich the GAs balance between exploration and exploitation in the solution space [26]. Due to this characteristic, nowadays, it has also been successfully coupled with a wide variety of EAs such as Differential Evolution (DE) [16, 32], Evolution Strategies (ES) [27] and Genetic Programming [3].

Generally speaking, the IM divides a population of candidate solutions (also called *individuals* or simply *points*) into subpopulations. Each subpopulation is referred to as an *island*, and remains relatively isolated from the others during the evolutionary process of a given EA. The islands evolve independently for some time, and then a *migration* process is performed, by which islands can exchange candidate solutions. This migration process has the potential benefit of promoting or advancing the evolution of the subpopulations, introducing new characteristics through the exchange of individuals between the islands.

The migration creates links between islands by which the individuals can move from one island to another. These migratory flows play an important role in the IM and result in two main effects on the underlying optimization process: the first, which is a positive effect, consists of the speed-up caused by information exchange; whereas the second one, which can be a drawback in certain applications, is the communication and processing overhead of the required information flow. In view of these drawbacks the IM is usually incorporated to EAs when handling more complex problems [26,

[6, 14], nevertheless improvements have also been reported in simple problems [16].

The adequate setting of the migratory flows, which can be static or dynamic, has an important impact on the IM performance [26]. The majority of the approaches presented in literature are static, usually, based on computer networks and architectures [16]. On the other hand, dynamic approaches are able to alter the migratory flow during the search process, and can use information gathered by the algorithm during the optimization process to guide the creation or removal of the inter-island links [16].

Although it is possible to find several different procedures to the configuration of the migratory flows, the majority of studies just compare them with traditional static configurations [4, 30]. Therefore, the goal of this paper is to provide thorough experimental investigation of this procedures and foster the understanding of their impact in the context of Differential Evolution Algorithms. The choice of DE is justified by its easiness of implementation and high performance proven in a series of optimization competitions¹ and reported in a variety of problems, such as, flow shop scheduling [21], digital filter [12] and induction motor design [24].

In order to accomplish our goal, we present a detailed experimental analysis about a wide variety of configuration procedures presented in the literature. They are, Ring [6], Star [5], Random [30], K-medoids [4], Q-learning [16], Roulette Wheel [15] and Tournament [15]. The IM versions of the Differential Evolution Algorithm [28] were implemented and the experiments were conducted in the Large Scale Global Optimization benchmark. The results show that there is a commitment between achieving good and fast convergence rates. Furthermore, in general, the topologies are very competitive between each other.

The remainder of this paper is organized as follows: Section 2 describes the IM and reviews of the migratory flow; Section 3 presents the different approaches to configure the migratory flow; Section 4 and 5 present the results obtained in the Large Scale Global Optimization benchmark set; and Section 6 presents the final discussion and conclusions of the paper.

2. BACKGROUND

2.1 Differential Evolution Algorithm

As mentioned before, Differential Evolution (DE)[28] is an important EA for global optimization. The reasons for its success can be found in its simplicity and ease of implementation, while at the same time demonstrating reliability and high performance [34].

In DE new candidate solutions are created by combining a parent individual with several other individuals of the same population. Then, this candidate solution replaces the parent if it has a better fitness value. DE has three control parameters: the population size NP , the scale factor of the perturbations generated by mutation F and the crossover constant CR . The basic iterative process of DE can be described as follows:

¹DE has obtained the following prizes: First place in the 2006 CEC contest on constrained optimization with real parameters, third place on the 2007 CEC contest on large-scale optimization problems and first place in the 2009 CEC contest on evolutionary algorithms in uncertain and dynamic environments.

1. Generate a population of solutions \mathbf{x} with random positions in the search space.
2. Set crossover probability CR and differential weight F
3. Until a stopping criterion is not met, repeat the following steps:
 - (a) For each solution \mathbf{x} :
 - i. Select other three different solutions \mathbf{a} , \mathbf{b} and \mathbf{c} from the population which are also different from \mathbf{x} .
 - ii. In order to generate a new solution \mathbf{y} , for each position i of the solution:
 - A. Select a random index j between 1 and the number of variables.
 - B. Generate a real valued number r between 0 and 1 with uniform distribution
 - C. If $(r < CR \vee j == i)$, $\mathbf{y}_i = \mathbf{a}_i + F(\mathbf{b}_i - \mathbf{c}_i)$, else $\mathbf{y}_i = \mathbf{x}_i$
 - iii. If \mathbf{y} is better than \mathbf{x} , \mathbf{x} is replaced by \mathbf{y}
4. Return the best solution found

2.2 Island Model

The Island Model (IM) is a popular and efficient way to implement Evolutionary Algorithms (EAs) on both serial and parallel architectures [6, 16, 27, 32]. It is able to improve the performance of EAs when solving complex problems, providing a better balance between exploration and exploitation.

The IM is easily parallelizable, and in some cases the gain in computational performance may be superlinear [2, 7]. Besides its suitability for parallel implementations, the island model also beneficial to the evolutionary search itself, since some enhancement in solution quality and convergence time can be obtained even in sequential implementations [32, 35].

The idea of IM is to divide the population of solutions into relatively isolated subpopulations called *islands*. Each island executes its own EA and evolves independently. This independence tends to foster the search in distinct regions of the solution space, thereby improving the optimization process. Another fundamental aspect of the IM is the collaboration between islands, which is achieved through a periodical migration process. This process consists of a strategy by which candidate solutions with different characteristics are transferred from one subpopulation to another. Migration allows better-performing subpopulations to aid others by introducing new information in the form of migrating points.

The project of the IM involves taking decisions about quite a few parameters of the model. These parameters are directly related to the IM performance and are summarized as follows:

- **Number of islands:** represents the number of subpopulations in the model;
- **Migration topology:** defines the communication structure of the model;
- **Migratory frequency:** determines how often migration occurs;
- **Migratory rate:** defines how many individuals migrate from a subpopulation to another;

- **Synchronization type:** represents the type of synchronization for the migration process (synchronous or asynchronous);
- **Migratory policy:** defines which individuals will be removed and replaced when migration occurs;
- **Migratory flow:** defines the path of the emigrants inside the communication structure.

2.3 Migratory Flow

The migratory flow (MF) is represented by a set of directed edges. These edges form the neighborhood structure of the IM and represent inter-island links. The movement of individuals between islands follows those links. The migratory flow is a critical parameter to the performance of the IM [16, 26]. It directly impacts the quality of the solution and the pace of convergence [26]. Furthermore, the choice of the most appropriate migratory flow may ensure that an IM is more efficient than others.

3. CONFIGURATION OF MIGRATORY FLOW

The configuration of MF can be divided into static and dynamic, as described below.

3.1 Static Configuration

In static configurations, the migratory flow (all inter-island links) are defined *a priori* and do not change during the evolutionary process [30]. This kind of migratory flow is usually based on computer networks and architectures.

Two traditional static configurations are illustrated in Figure 1. The Ring, consisting of a closed circuit of islands (Fig. 1(a)), is the most studied migratory flow in the literature. The Star is formed by a single central island which is connected with all other islands (see Figure 1(b)).

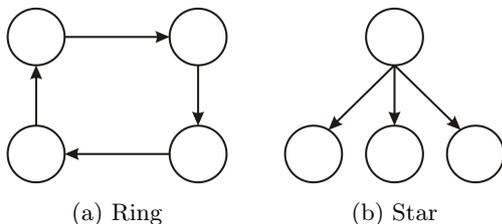


Figure 1: Traditional Configurations.

3.2 Dynamic Configuration

The dynamic configuration is a recent concept to improve the communication structure of the IM. It is able to learn and adjust the MF each stage of evolutionary process [16]. The MF, defined by this mechanism, emerges from heuristic “beliefs”. Thus, the remaining of section presents dynamic configurations found in the literature.

3.2.1 Random Configuration

The Random configuration [30] is arguably the simplest dynamic configuration approach. For each island of the model, the configuration mechanism randomly selects the

source island from which individuals will be received². At the end of the selection process, a complete migratory flow is defined.

The Random configuration provides continuously varying links, and its configuration mechanism is very simple to implement. This dynamic configuration, however, is naïve and makes no use of the knowledge acquired through of the evolutionary process.

3.2.2 Configuration using K-medoids

This configuration mechanism was proposed by Berntsson and Tang [4], and is based on clustering of the best solutions from the islands to find groups of islands that work in similar partitions of the search space. The goal is to reduce the connectivity, and consequently the complexity, of the IM while maintaining good performance [4].

The K-medoids algorithm is an unsupervised learning method based on the automatic division of a set of data into groups based on similarity [29]. This algorithm uses the *medoid* concept, which is the object of a cluster with minimum average dissimilarity to all other objects [25]. The K-medoid algorithm attempts to minimize the squared error, i.e., the distance between points in the cluster and the medoid (designated as the center of the cluster).

In this approach, the configuration of the migration flow is made in two steps:

1. *Definition of groups*, where the best solution from each island is taken and used to define the similar groups using the K-medoids algorithm. The number of clusters and the distance function must be defined by the user³;
2. *Definition of the Migratory Flow*, where the groups defined in the previous step are used to generate the links between islands belonging to the same group. For each group, the link can be created at random or in a ring.

It is worthy noticing that the setting of the number of clusters is a non-trivial problem for this approach. Furthermore, the processing cost of this configuration mechanism increases sharply with the number of islands in the model.

3.2.3 Configuration using Q-learning

A new configuration mechanism, which transforms the IM into a Multi-Agent System (MAS), was proposed by Lopes *et al.* in 2012 [16]. In this MAS, each subpopulation is associated with an agent, and each agent is responsible for learning the links that collaborate the most to the evolution of its subpopulation.

The agents of the model act independently and acquire knowledge from the previous migration processes. The learning process is based on the Q-learning method [33], which is a traditional reinforcement learning procedure. In this method the learning model works through trial-and-error interactions with a dynamic environment [11]. In the early stages of the search process each island can receive immigrants from all others, but as the algorithm progresses the

²It is important to notice that loop arcs are forbidden, i.e., an individual can not be sent to its own island.

³In our study, two clusters and the Euclidean distance are used for this method.

agents learn which migration flows have provided the best individuals, and adapt the flow accordingly.

In this approach, each agent has states and actions. The states describe the islands from which a defined agent can receive individuals during the migration process, with each specific island corresponding to one state. The actions represent the modification or continuation of the island chosen to send solutions.

Studies demonstrate that the Q-learning method converges to an optimal policy of the actions [33]. However, the learning model is very complex and requires the setting of many essential parameters. Furthermore, the parameters may need to be reinitialized due the dynamics of the evolutionary process.

3.2.4 Configuration using Roulette Wheel

This approach was proposed by Lopes *et al.* in 2013 [15], and it is similar to the mechanism of the Random configuration. However, in an attempt to explore the dynamics of the evolution of subpopulations, the selection methods proposed are based on the fitness of the individuals. It assumes that the fittest individuals have great probability of improving the IM performance and thus, for each island, the source of the immigrant individual is defined by a selection method.

The Roulette Wheel selection [10] is a commonly used stochastic method for Genetic Algorithms (GAs) both in reproductive selection and generational selection. In this selection method, to each individual is attributed a probability of selection proportional to its relative performance value - better individuals are selected with greater probability - such that the selection probabilities sum to unity. The selection process is performed by sampling individuals from the resulting probability mass function.

In this configuration mechanism, the probability of selection attributed to the i -th individual is given by:

$$PS_i = \frac{fn(x_i)}{\sum_{j=1}^{NI} fn(x_j)} \quad (1)$$

where $fn(x_i)$ is the normalized fitness of the i -th individual x_i , and NI is the total number of individuals participating in the selection process.

In the IM each island performs this method to choose an immigrant among the best individuals from all other islands. This process is repeated until all islands have their own immigrant.

3.2.5 Configuration using Tournament

Following the same idea, Lopes *et al.* also proposed in 2013 [15], an identical mechanism using the well known Tournament selection algorithm. This selection method is a useful and robust selection approach commonly used by GAs [19]. In this selection process, individuals are selected through tournaments in which s randomly chosen individuals are compared with regards to performance value.

Comparisons are successively performed on pairs of candidate solutions, with the winners competing with other individuals until the tournament size s is reached. In each comparison, a uniformly distributed random value between 0 and 1 is drawn and compared to a parameter kt defined as $\{kt \in \mathbb{R} \mid 0 \leq kt \leq 1\}$. If the the random value is smaller than kt the best individual wins the comparison, otherwise the worse is selected.

In this configuration mechanism, individuals are selected based on their fitness $f(x_i)$. This mechanism works the same way as the approach previously presented. Each island performs a tournament to choose an immigrant among the best individuals of all other islands. This process is repeated until all of them have their own immigrant.

4. COMPUTATIONAL EXPERIMENTS

For the comparison of the different approaches, IM versions of the Differential Evolution Algorithm (DE) [28] were implemented. The DE parameters were defined as $CR = 0.9$ and F randomly chosen in the interval $[0.5, 1.0]$ as suggested in [23]. Each island was composed by 25 individuals and effect of the migration gap⁴ was evaluated for 5, 15, 25 and 50 generations. The migration policy was set as *best substitutes random*, the best individual of an island substitutes a random one in the receiving island. Only one individual replaced per island on any given migration episode.

Regarding the Q-learning approach its parameters were set as $\gamma = 1.0$, $\lambda = 0.5$, $\epsilon = 1.0$ as defined in [16]. For the Tournament approach, the tournament size was defined as 3 and the parameter kt equal 1.0. The parameters of Tournament approach were defined according [15].

Six functions of the Benchmark Functions for Large Scale Global Optimization [31] with known optima were used. f_1 and f_2 are unimodal and f_3 to f_6 are multimodal. All functions were defined with 30 variables. For each configuration 25 independent runs were made in each function. The search was terminated when either of the following conditions were met:

- The maximum number of generations (12,000) has been reached;
- $\Delta f = f_{target} - f_{best} \leq 10^{-4}$;

where f_{target} , f_{best} represent the know optimal value for the function and the best result found by the algorithm so far, respectively.

Two outputs were selected for the analysis, (i) Number of generations for convergence and (ii) percentage of convergence. Regarding the number of generations, only the runs which successfully found the global optimum were taken into account. In this way the percentage of convergence can be considered as a complementary measure of the methods' global performance.

Given the non-normality of the results, the nonparametric multiple contrast test⁵ described in [13] was employed for the statistical analysis of the number of generations. The statistical model considered a unbalanced design [20], with the *configuration procedures*, as experimental factor. The significance level α was previously defined as 0.05.

For the percentage of convergence, the Cochran's Q test [8] was employed, when significant differences were detected for the methods, McNemar's test [18] was applied for the post-hoc comparisons. The obtained p-values were then adjusted by Bonferroni method [1].

⁴The *Migration Gap* defines the number of elapsed generations from one migration to another. A migration gap equals to 10, means that from 10 to 10 generations there will be a migration process.

⁵Function *mctp* from *parcomp* package of the statistical software R

5. RESULTS

This section presents the comparisons among the studied approaches. The Figure 2 depicts the interaction of the configuration methods with the number of islands, Figure 3 depicts the interaction of the configuration methods with the migration gap, both in regards to the average number of generations for convergence. Only the configurations and runs which found the global optimal were considered for these graphics.

It can be seen in Figure 2 that, when the methods converge to the global optimum, the approaches which incorporate information about the search process, tend to converge in less generations. However, despite not using any information the Ring configuration has shown competitive performance. Clearly, the Star configuration presented the worst results, indicating that a centralized topology is not a good structure for this kind of optimization.

Furthermore, it is interesting to notice the degrading performance that the static configurations presented with a high number of islands. This behavior can be explained by the difficulty that high quality solutions have to spread over these rigid structures as the number of islands increases. In overall the approaches which present a higher selective pressure on the immigrants, Tournament and Roulette, also presented the best results in this criterion.

Regarding the migration gap, Figure 3 shows that, in general, its increment slowed down the convergence. These results were in some way expected because a smaller migration gap increases the selective pressure within the islands, once they are receiving high quality individuals more frequently. Table 1 presents the results and p-values of the statistical tests, regarding the number of generations for convergence (only runs and configurations that found the global optimum). The approaches marked in bold represent the best configurations in each paired test. The applied test considers only the effect of the topology configurations, ignoring the effects of the other factors and interactions. The results show that the Roulette Wheel and Tournament approaches converged in a number of generations significantly smaller than the other methods and corroborate with the previous analysis.

Configuration 1	Configuration 2	P-value
Q-learning	Star	0.0000
Q-learning	Tournament	$3.4432 * 10^{-12}$
Q-learning	Roulette Wheel	$7.8931 * 10^{-11}$
Ring	Star	0.0000
Ring	Tournament	$1.9072 * 10^{-07}$
Ring	Roulette Wheel	$1.5741 * 10^{-06}$
K-medoids	Star	0.0000
K-medoids	Tournament	$5.2142 * 10^{-06}$
K-medoids	Roulette Wheel	$5.0668 * 10^{-05}$
Star	Random	0.0000
Star	Tournament	0.0000
Star	Roulette Wheel	0.0000
Random	Tournament	$4.5362 * 10^{-05}$
Random	Roulette Wheel	$3.4016 * 10^{-04}$

Table 1: P-values of the statistical analysis for the number of generations

Figures 4 and 5 present, respectively, the interactions of the number of island and migration gap, with the configuration approaches in regards to the percentage of convergence. The percentage of convergence represents the proportion in which all the configurations and runs found the global optimal. In relation to number of islands (Figure 4), the percentage of convergence was improved for all approaches as the number of islands was increased. Regarding the migration gap, higher values have shown to be more beneficial, except the star topology. Although higher migration gaps decrease the convergence speed, they increase the convergence rate. In this context, Random, Tournament and Roulette Wheel presented interesting results once their convergence rate was virtually the same for migration gaps of 25 and 50 generations.

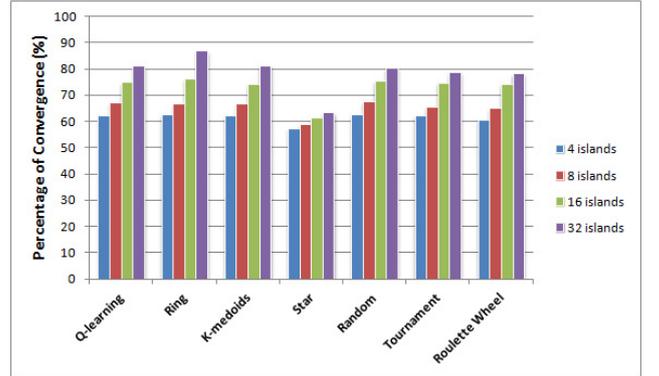


Figure 4: Percentage of Convergence (Number of Islands)

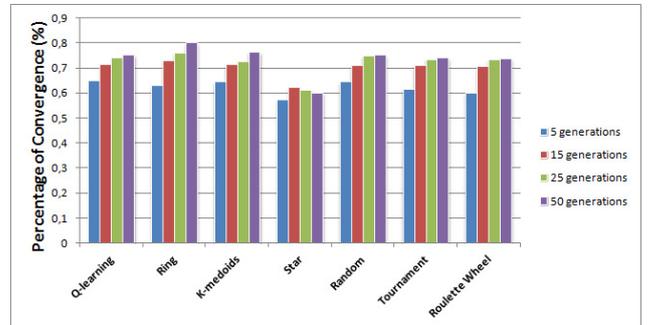


Figure 5: Percentage of Convergence (Migration Gap)

Table 2 presents the results of the statistical tests on the mean of percentage of convergence achieved by the approaches presented and this test considers only the effect of the topologies. In this table, only results that presented significant differences with 95% of confidence were displayed and they show that the Ring and Q-learning converged significantly more times than the other approaches. As well as in the average of number generations, the Star topology was the worse approach between all.

Table 3 presents the mean and standard deviation of the running times, in seconds, of all configurations and runs. The majority of the approaches presented similar running

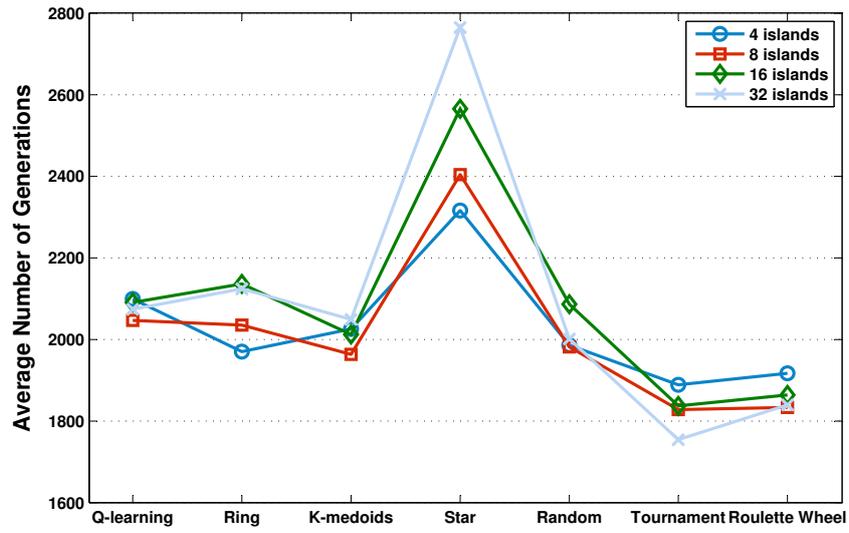


Figure 2: Topology Configuration x Number of Islands

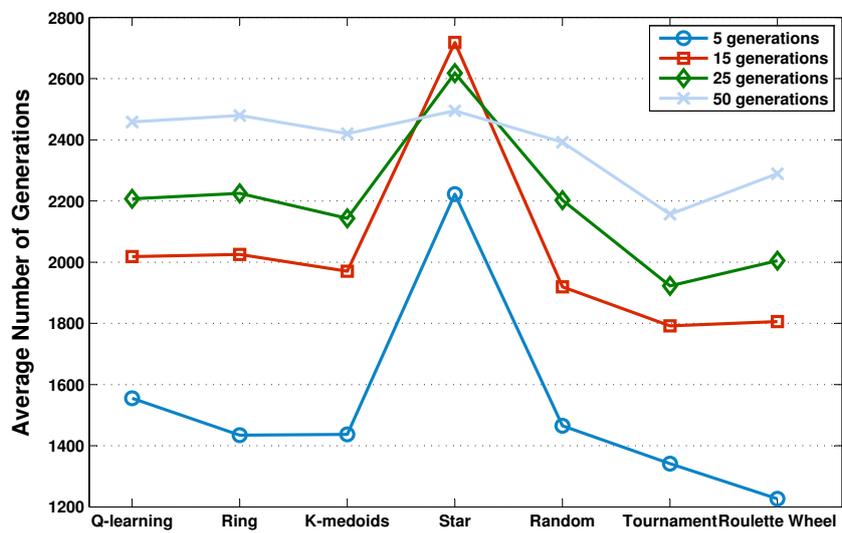


Figure 3: Topology Configuration x Migration Gap

Configuration 1	Configuration 2	P-value
Q-learning	Star	$4.6200 * 10^{-15}$
Q-learning	Roulette Wheel	$6.6087 * 10^{-03}$
Ring	K-medoids	$1.2618 * 10^{-02}$
Ring	Star	$4.6200 * 10^{-15}$
Ring	Random	$4.9833 * 10^{-02}$
Ring	Tournament	$3.8178 * 10^{-05}$
Ring	Roulette Wheel	$1.0258 * 10^{-08}$
K-medoids	Star	$4.6200 * 10^{-15}$
K-medoids	Roulette Wheel	$4.1286 * 10^{-02}$
Star	Random	$4.6200 * 10^{-15}$
Star	Tournament	$4.6200 * 10^{-15}$
Star	Roulette Wheel	$4.6200 * 10^{-15}$
Random	Roulette Wheel	$1.6088 * 10^{-02}$

Table 2: P-values of the statistical analysis for the convergence percentage

Configuration Method	Average Running Time(s)	Standard Deviation(s)
Q-learning	3.8163	5.5127
Ring	3.4150	4.5910
K-medoids	4.1679	6.2383
Star	5.1738	6.6213
Random	3.7746	5.5178
Tournament	3.8262	5.8423
Roulette Wheel	3.8533	5.7371

Table 3: Running Times (s)

times except the Star which again presented the worst results. The large standard deviations are due to the consideration of the non-successful runs in the measurement.

6. CONCLUSIONS

In this paper, a thorough experimental analysis of a variety of procedures for the configuration of migration flows has been presented. To this end, an IM version of the Differential Evolution Algorithm was implemented and the different procedures were coupled to it. The main goal of the paper was to elucidate the influence these procedures in the IM and their interaction with other IM parameters. In addition to that, we aimed to cover the lack of comparison between these different configuration approaches.

The results on well known benchmark functions demonstrate that there is a difference in the DE performance due to the different migration topologies applied to the IM. In synthesis, they indicate that when the methods converge to the optimal solution, the Tournament and Roulette Wheel approaches converge faster. The reason for that is likely to be the rise in the selective pressure caused by those types of migration topologies. Furthermore, it has been shown that small migration gaps and a greater number of islands may also contribute to improve the convergence pace of the algorithm for these approaches. Nevertheless, this increase in selective pressure seems to decrease the population diversity leading to smaller convergence rates. In this respect the approaches which add less selective pressure, e. g., Ring and the Q-learning, presented better results. Regarding the computational times, the results among the different approaches

were very similar. The addition of complexity due to information gathering in the dynamic approaches did not result in a significant time increase.

Concluding, the study has accomplished its objective, producing more knowledge about the migratory flows in IM which will assist future studies of multi-population approaches.

7. ACKNOWLEDGMENTS

The authors would like to thank Universidade Federal de Ouro Preto and the following Brazilian agencies for the financial support: CNPq, FAPEMIG and CAPES.

8. REFERENCES

- [1] H. Abdi. *Bonferroni and Sidak corrections for multiple comparisons*. Sage, Thousand Oaks, CA, 2007.
- [2] E. Alba. Parallel evolutionary algorithms can achieve super-linear performance. In *Information Processing Letters*, volume 82, pages 7–13. Elsevier Science Inc., 2002.
- [3] D. Andre and J. R. Koza. A parallel implementation of genetic programming that achieves super-linear performance. In *Information Processing Letters*, volume 106, pages 201–218. Elsevier Science Inc., New York, NY, USA, 1998.
- [4] J. Berntsson and M. Tang. Dynamic optimization of migration topology in internet-based distributed genetic algorithms. In *Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 1579–1580, ACM, New York, NY, USA, 2005.
- [5] P. Bujok. Parallel models of adaptive differential evolution based on migration process. In *Aplimat, 10th International Conference on Applied Mathematics*, pages 357–364, Bratislava: Slovak Technical University, 2011.
- [6] E. Cantú-Paz. A survey of parallel genetic algorithms. *Calculateurs Paralleles, Reseaux Et Systems Repartis*, 10:1–30, 1998.
- [7] E. Cantú-Paz. Migration Policies, Selection Pressure, and Parallel Evolutionary Algorithms. In *Journal Of Heuristics*, volume 7, pages 311–334. Kluwer Academic Publishers, 2001.
- [8] W. Cochran and G. Cox. *Experimental Designs, 2nd edition*. John Wiley & Sons, 1957.
- [9] J. P. Cohoon, S. U. Hegde, W. N. Martin, and D. Richards. Punctuated equilibria: a parallel genetic algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, pages 148–154. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, USA, 1987.
- [10] O. A. Jaddan, L. Rajamani, and C. R. Rao. Improved selection operator for ga. pages 269–277. Asian Research Publishing Network, Pakistan, 2005.
- [11] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: a survey. In *Journal of Artificial Intelligence Research*, volume 4, pages 237–285. Access Foundation and Morgan Kaufmann Publishers, 1996.
- [12] N. Karaboga. Digital iir filter design using differential evolution algorithm. *European Association for Signal*

- [13] F. Konietschke, L. A. Hothorn, and E. Brunner. Rank-based multiple test procedures and simultaneous confidence intervals. In *Electronic Journal of Statistics*, volume 6, pages 738–759. The Institute of Mathematical Statistics and the Bernoulli Society, 2012.
- [14] D. Levine. A parallel genetic algorithm for the set partitioning problem. In I. Osman and J. Kelly, editors, *Meta-Heuristics*, pages 23–35. Springer US, 1996.
- [15] R. A. Lopes, R. C. P. Silva, F. Campelo, and F. G. Guimarães. Dynamic selection of migration flows in island model differential evolution. In *Proceeding of the Fifteenth Annual Conference Companion on Genetic and Evolutionary Computation Conference Companion*, GECCO '13 Companion, pages 173–174. ACM, 2013.
- [16] R. A. Lopes, R. C. P. Silva, F. Campelo, and F. G. Guimarães. A multi-agent approach to the adaptation of migration topology in island model evolutionary algorithms. In *Brazilian Symposium on Neural Networks, Curitiba, Brazil*, pages 160–165. IEEE, 2012.
- [17] G. Luque and E. Alba. *Parallel Genetic Algorithms: Theory and Real World Applications*. Studies in Computational Intelligence. Springer, 2011.
- [18] Q. McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.
- [19] B. L. Miller and D. E. Goldberg. Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems*, 9:193–212, 1995.
- [20] D. C. Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons, 2006.
- [21] G. Onwubolu and D. Davendra. Scheduling flow shops using differential evolution algorithm. *European Journal of Operational Research*, 171(2):674 – 692, 2006.
- [22] C. C. Pettey, M. R. Leuze, and J. J. Grefenstette. A parallel genetic algorithm. In *Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, pages 155–161. Lawrence Erlbaum Associates, Inc., Cambridge, MA, USA, 1987.
- [23] K. Price, R. Storn, and J. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization*. Natural Computing Series. Springer, 2005.
- [24] C. T. Raj, R. Thangaraj, M. Pant, P. Bouvry, and A. Abraham. Design optimization of induction motors with differential evolution algorithms with an application in textile spinning. *Applied Artificial Intelligence*, 26(9):809–831, 2012.
- [25] A. P. Reynolds, G. Richards, and V. Rayward-Smith. The application of k-medoids and pam to the clustering of rules. In *Intelligent Data Engineering and Automated Learning, IDEAL 2004*, volume 3177 of *Lecture Notes in Computer Science*, pages 173–178. Springer Berlin Heidelberg, 2004.
- [26] M. Ruciński, D. Izzo, and F. Biscani. On the impact of the migration topology on the island model. In *Parallel Computing*, volume 36 of *Parallel Architectures and Bioinspired Algorithms*, pages 555–571. Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, 2010.
- [27] G. Rudolph. Global optimization by means of distributed evolution strategies. In *Proceedings of the 1st International Conference on Parallel Problem Solving from Nature*, Lecture Notes in Computer Science., pages 209–213. Springer, Berlin, 1991.
- [28] R. Storn and K. Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, Dec. 1997.
- [29] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.
- [30] J. Tang, M.-H. Lim, Y.-S. Ong, and M. J. Er. Study of migration topology in island model parallel hybrid-ga for large scale quadratic assignment problems. In *Proceedings of the 8th International Conference on Control, Automation, Robotic and Vision, Kunming, China*, pages 2286–2291. IEEE, 2004.
- [31] K. Tang, X. Yao, P. N. Suganthan, C. MacNish, Y. P. Chen, C. M. Chen, and Z. Yang. Benchmark functions for the congress on evolutionary computation (cec) 2008 special session and competition on large scale global optimization. Technical report, Nature Inspired Computation and Applications Laboratory, USTC, China, 2007.
- [32] D. K. Tasoulis, N. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis. Parallel differential evolution. In *In IEEE Congress on Evolutionary Computation (CEC)*, 2004.
- [33] C. J. C. H. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3-4):279–292, 1992.
- [34] M. Weber, V. Tirronen, and F. Neri. Scale factor inheritance mechanism in distributed differential evolution. *Soft Computing*, 14(11):1187–1207, 2010.
- [35] D. Whitley, S. Rana, and R. B. Heckendorn. The island model genetic algorithm: On separability, population size and convergence. *Journal of Computing and Information Technology*, pages 33–47, 1998.