Island-Cellular Model Differential Evolution for Large-Scale Global Optimization

Rodolfo A. Lopes
Universidade Federal de Ouro Preto
Ouro Preto, Minas Gerais - Brazil - 35400-000
rodolfo.ufop@gmail.com

Alan R. R. de Freitas
Universidade Federal de Ouro Preto
Ouro Preto, Minas Gerais - Brasil - 35400-000
alandefreitas@iceb.ufop.br

ABSTRACT

The Island-Cellular Model (ICM) is an important population distribution approach for Evolutionary Algorithms (EAs). This hybrid approach combines the Island Model (IM) and Cellular Model (CM) in a two-layer hierarchical model. Although the ICM has been shown to be an efficient way to implement EAs, there is still a lack of knowledge about its parameters and its performance for Large-Scale Global Optimization (LSGO) problems. However, the ICM approach is able to enrich the evolutionary search by keeping the population diversity in EAs. Thus, this paper proposes to implement the ICM approach for LSGO problems using the Differential Evolutionary (DE) algorithm. It also proposes an experimental study of the ICM parameters by investigating their impact on the EA. Experimental results on Large-Scale Global Optimization Benchmark Functions show that the ICM approach for the DE algorithm improves its performance. Furthermore, the results collected from different ICM approaches indicate that there is a trade-off between solution quality and convergence speed.

CCS CONCEPTS

•**Theory of computation → Mathematical optimization;** *Bio-inspired optimization;*

KEYWORDS

Island-Cellular Model, Island Model, Cellular Model, Differential Evolution, Large-Scale Global Optimization

ACM Reference format:
Rodolfo A. Lopes and Alan R. R. de Freitas. 2017. Island-Cellular Model Differential Evolution for Large-Scale Global Optimization. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 8 pages.
DOI: http://dx.doi.org/10.1145/3067695.3084208

1 INTRODUCTION

The Island-Cellular Model (ICM) is a hybrid approach for Evolutionary Algorithms (EAs) [10, 12]. Hybrid distributed approaches are able to improve the scalability and problem-solving capability [10] of EAs. A wide variety of studies have successfully connected hybrid distributed approaches with different EAs [4, 5, 13, 17]. Particularly, related works about the ICM have numerically shown their good performance with Genetic Programming (GP) algorithms [9, 10].

Basically, the ICM combines two distributed models: the *island* and the *cellular* models. Thus, in this approach, different layers hierarchically couple the models. Firstly, the Island Model [6] forms the upper layer of ICM. The main idea of an IM is to divide a population of candidate solutions (also called individuals) into subpopulations, known as islands. The islands remain relatively isolated and evolve independently. Periodically, the migration process occurs and individuals can be exchanged between islands. Additionally, the lower layer implements the Cellular Model (CM) [6] for each island of the model. This approach arranges the individuals into a toroidal grid in which they only can reproduce with other neighbors. Both approaches allow EAs to keep the population diversity and improve their performance [12].

Although EAs have been shown to be powerful to problems solvers, as the size and complexity of problems increase, new challenges have been posed to these algorithms. In response to these challenges, hybrid models to distributed EAs have been proposed. Their success is due to their characteristics, such as parallelism and exploitation of the solution space. Therefore, the goals of this paper are: (i) to propose an Island-Cellular Model Differential Evolution (DE) algorithm for Large-Scale Global Optimization (LSGO) problems; and (ii) to provide an experimental investigation of the ICM and their parameters to foster the understanding of their impact in the context of the DE. The reasons for the DE choice can be clarified by its high performance, reliability, simplicity and easy implementation [29].

In order to achieve the goals defined, this paper presents a thorough experimental analysis of the ICM performance. For this, ICM versions of the Differential Evolution algorithm were implemented. The experiments were conducted in the Large-Scale Global Optimization Benchmark Functions [16]. The results show that there is a compromise between achieving good solutions and fast convergence rates. Furthermore, in general, the ICM significantly improves the EAs performance.

The remainder of this paper is organized as follows: Section 2 describes the Large-Scale Global Optimization Problems, Differential Evolution algorithm and reviews of Cellular Model and Island Model; Section 3 presents the Island-Cellular Model for Differential Evolution algorithm; Section 4 describes details of the experimental setup and statistical tests used; Section 5 presents the results obtained in the Large-Scale Global Optimization Benchmark Functions; and, finally, Section 6 presents a final discussion and conclusions of the paper.

GECCO '17 Companion, Berlin, Germany

978-1-4503-4939-0/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3067695.3084208

2 OVERVIEW

2.1 Large-Scale Global Optimization Problems

Recently, most real-word optimization problems exhibit a large number of decision variables. This type of optimization problems are known as Large-Scale Global Optimization (LSGO) problems. Generally speaking, the increasing number of decision variables in an optimization problem increase their landscape complexity and the domain search exponentially. Due to these characteristics, this research area has received special attention over the last years by the most important conferences and journals, as mentioned in [20].

With the purpose of tackling these new optimization challenges, standard metaheuristics have been specially designed to attempt to solve them. Valuable attempts to implement variants of Genetic Algorithms (GA) [11], Particle Swarm Optimization (PSO) [14], Differential Evolution (DE) [24] and other metaheuristics [2, 3] have been developed for LSGO. However, according to [20], the performance of these metaheuristics tend to deteriorate due to characteristics of LSGO.

2.2 Differential Evolution Algorithm

Proposed by Storn and Price [25], Differential Evolution (DE) is one of the most important Evolutionary Algorithms (EAs). Initially designed for global optimization over continuous spaces, DE has been also applied to problems with discrete domain [22, 23]. The DE algorithm is an optimization method considered reliable and versatile [29].

Similarly to other EAs, the DE is a population-based optimization method and its basic idea is to iteratively evolve a population of candidate solutions (also called individuals). For this, individuals from the same population are combined to generate new candidate solutions. Then, if these new individuals have a better fitness value, they replace their parents for the next generation. The classic version of DE has three control parameters: (1) *NP*, which defines the population size; (2) *F*, which defines the scale factor of the perturbations generated by the mutation operator; (3) *CR*, which defines the crossover constant. More details are described in Algorithm 1 and they can also be found in [25, 29].

2.3 Models of Distributed Evolutionary Algorithms

Distributed EAs (dEA) are popular and efficient approaches to implement EAs. These approaches are especially recommended when designed to solve problems with extremely high computational cost of fitness evaluation or a huge number of local optima [12]. Furthermore, they can improve its capacity to find satisfactory results within a reasonable time compared to classic versions of the algorithm [12].

According to [12], the models of dEAs can be classified into *dimension-distributed* and *population-distributed*. The first one, dimension-distributed model, proposes to divide partitions of the problem dimensions, as for example: Coevolution Model [26]. The main advantage of dimension-distributed models is the dimension reduction. However, these algorithms are not suggested to solve problems with complex interdependencies. The purpose of the second approach is to distribute individuals of the population, such as in a Cellular Model [6], Island Model [6], and Hybrid Models [9, 10, 12]. Population-distributed models have been successfully applied to EAs and it can be explained by its efficiency and scalability [12].

Algorithm 1: Differential Evolution

```
F, CR, NP ← InitializeParameters();
pop ← InitializePopulation(NP);
while not StopCriteria() do
    pop_aux ← ∅;
    for each individual x from pop do
        Select other 3 random individuals a, b and c from pop
          where $a \neq b \neq c \neq x$;
        Select a random index j between 1 and the number of
          variables;
        for each position i of individual x do
            r ← real value between 0 and 1 with uniform
              distribution;
            if $r \leq CR$ or $i = j$ then
                $\mathbf{y}_i \leftarrow \mathbf{a}_i + \mathbf{F} * (\mathbf{b}_i - \mathbf{c}_i)$;
            else
                $\mathbf{y}_i \leftarrow \mathbf{x}_i$;
        if y is better than x then
            Add y in pop_aux;
        else
            Add x in pop_aux;
    pop ← pop_aux;
Return the best solution found;
```

Additionally, as previously reported, population-distributed models are beneficial to the evolutionary search by improving the population diversity, thereby avoiding local optima and facilitating the search of the whole solution space [12].

- **Cellular Model**

The Cellular Model (CM) is an important population-distributed approach for EAs [6, 12]. In this approach, the population is spatially structured in a grid in which each individual can only interact with a set of neighbor individuals to reproduce. Usually, the CM approach arranges the individuals on a toroidal grid of two dimensions. Figure 1 illustrates an example of CM disposing 25 individuals (represented by circles) on a bi-dimensional grid (5x5) in which a parent individual (represented by the gray circle) can only couple with 4 other individuals (represented by the black circle).

Four crucial parameters must be defined in a CM to calibrate its performance:

Grid dimension: defines the number of dimensions in the CM grid;

Grid size: defines the maximum number of coordinates of the CM grid;

Neighborhood pattern: describes the set of neighbors individuals for each individual in the population;

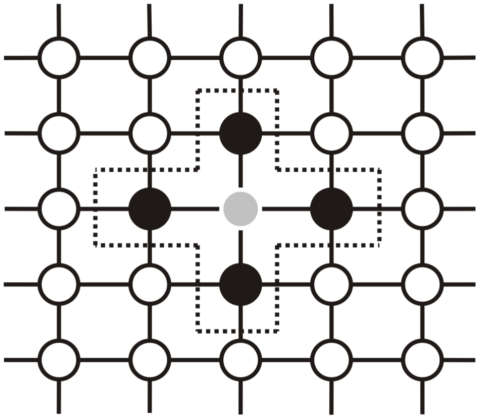


Figure 1: Cellular Model example.

Synchronization: defines which individuals of the current population are replaced in the next iteration.

The classical approach of CM is its parallel implementation. Moreover, the CM has also been proved as an important strategy to keep the population diversity. This approach promotes the slow diffusion of solutions through the grid in EAs and it is able to enrich the exploitation of the solution space. Moreover, studies demonstrated that the CM improves accuracy and convergence [7, 8].

- **Island Model**

The Island Model (IM) is also classified as a population-distributed approach for EAs [6, 12]. Its rationale is to divide a population of solutions into subpopulations. During the evolutionary process, the subpopulations, also known as islands, remain relatively isolated and evolve independently executing its own EAs for some time. Then, solutions can be exchanged between the islands in a process called migration. This cycle repeats until the stop criteria are met.

The independence of islands tends to keep the population diversity, thereby improving the evolutionary process. Nevertheless, the IM collaboration performed by the migration process is another fundamental aspect of this approach. The introduction of new information in the subpopulations by exchanging solutions allows better-performing subpopulations [19].

This model has some parameters to be configured in its design. As reported by [19], these parameters directly impact the IM performance (solution quality and convergence speed) and they are summarized as follows:

Number of islands: defines the number of subpopulations in the model;

Migration topology: describes the communication structure of the IM;

Migratory flow: describes the path of the emigrants (during the migration process) inside the communication structure of the IM;

Migratory frequency: defines the periodicity of migration process;

Migratory rate: defines how many individuals migrate from an island to another;

Migratory policy: describes which individuals will be copied and replaced when the migration process occurs;

Synchronization type: defines the type of synchronization (synchronous or asynchronous) for performing the migration process.

The migratory flow is one of the critical parameters for designing an IM [19]. Many configuration procedures of migratory flow have been presented in the literature, such as Random [27], Ring [6], Roulette Wheel [18], and Tournament [18]. Classified as a static approach, the Ring configuration of migratory flow is defined a priori and does not change over time [19]. The Ring configuration is the probably the most popular approach and it is formed by a closed circuit of islands. The Random, Roulette Wheel, and Tournament approaches are known as dynamic configuration procedures of migratory flow [19]. The basic idea of these approaches is to implement selection methods for choosing immigrant individual [19]. Thus, for each island of the IM, the selection method selects the source island from which individuals will be received and an individual can not be sent to its own island.

The IM is able to improve the evolutionary search of EAs through a better balance between exploration and exploitation of the solution space [19]. This efficiency can be explained by the maintenance of population diversity, independent evolution, and collaboration between islands of the model. Besides, the IM is a popular way to implement EAs even in a serial architecture and related works have shown the high performance of IMs [6, 28].

3 ISLAND-CELLULAR MODEL

The Island-Cellular Model (ICM) is a hybrid approach based on the hierarchical combination of two distributed models for EAs [10, 12]. The ICM takes advantages of both approaches by providing high scalability and fault-tolerance [10]. For the first time, this approach was successfully connected with Genetic Programming (GP) [10]. According to [9], the ICM provided an efficient implementation of distributed GPs improving the accuracy in the generation of classifiers.

The general idea of the ICM is to create upper and lower layers which are composed by the IM and CM approaches. Firstly, the population of candidate solutions is divided into subpopulations and then the CM is applied for each subpopulation previously defined. This approach accumulates parameters of both the IM and CM approaches. Figure 2 exemplifies an ICM formed by four islands.

This paper proposes to apply the ICM approach to the Differential Evolution (DE) optimization method. For this, the classic version of the DE algorithm is adapted to implement the CM where the population is spatially distributed in a grid. In addition, the CM version of the DE restricts the set of available individuals to generate new solutions by the mutation operator. In the CM version of the DE, only individuals from the set of neighbor individuals can be selected. As mentioned before, the set of neighbor solutions of a parent individual is defined by the CM neighborhood parameter that describes the neighborhood pattern, as illustrated by two examples in Figure 3.

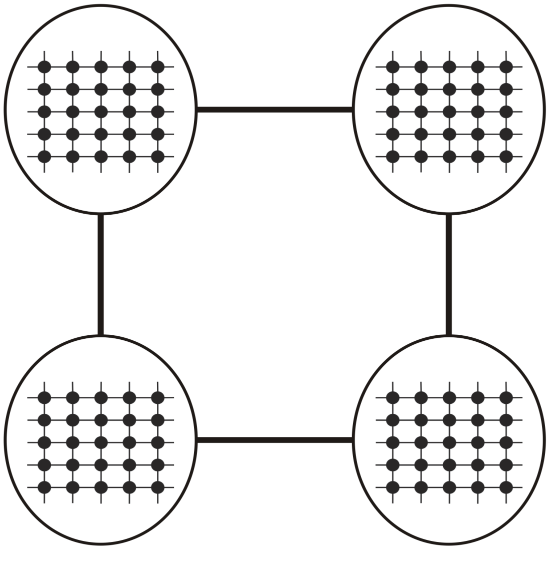
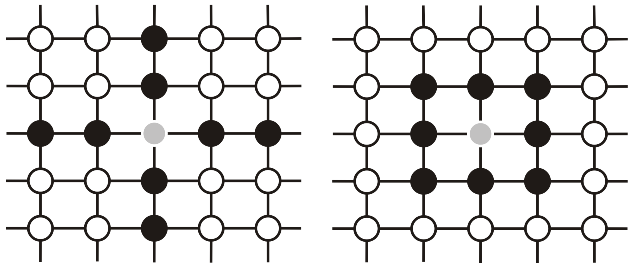


Figure 2: Island-Cellular Model.



(a) Linear 9

(b) Compact 9

Figure 3: Two examples of neighborhood patterns for Cellular Model.

4 EXPERIMENTAL SETTING

In order to accomplish our goals, the ICM, CM and classic versions of the Differential Evolution algorithm were implemented. As suggested in [24], the DE parameters were defined as $CR = 0.9$ and F was randomly chosen in the interval $[0.5, 1.0]$ for each new individual generated. More details on the ICM, CM and classic DE parameters are shown in Table 1.

In order to compare different versions of the ICM and other approaches, the Benchmark Functions for Large-Scale Global Optimization [16] were used. All the fifteen functions were defined with 1,000 continuous variables and their details are described in Table 2. See [16] for more details on the benchmark functions. For each ICM configuration, 10 independent runs were made for each test function from the benchmark. The evolutionary process was terminated when either of the following conditions were met:

- The maximum number of function evaluations ($20*10^6$) has been reached; *or*
- $\Delta f = f_{best} - f_{target} \leq 10^{-3}$

where f_{best} represents the best result found by the algorithm so far and f_{target} represents the optimal value known for the function.

The experiments collected two outputs for the analysis: (i) number of function evaluations for convergence; (ii) the Δf value achieved. The number of function evaluations measures the convergence speed of EAs evaluated, while the Δf value represents the quality of solutions found.

Table 1: Set of parameters defined for the computational experiments.

Parameter	Values
ICM Parameters	
Number of Islands	*4, 8, 16*
Population Size	*25 individuals per island*
Migratory Flow	*Random[27],* *Ring[6],* *Roulette Wheel[18],* *Tournament*[1][18]
Migration Frequency	*5, 15, 25 generations*[2]
Migration Rate	*1 individual*
Migration Policy	*best substitutes random*[3]
IM Synchronization	*synchronous*
Grid Dimension	*bi-dimensional*
Grid Size	*5x5*
Neighborhood	*Linear 9,* *Compact 9*
CM Synchronization	*synchronous* [4]
CM Parameters	
Population Size	*25*
Grid Dimension	*bi-dimensional*
Grid Size	*5x5*
Neighborhood	*Linear 9,* *Compact 9*
CM Synchronization	*synchronous*
Classical DE Parameter	
Population Size	*25*

Given the non-normality of the results, the nonparametric multiple contrast test[5] was employed for the statistical analysis of the number of function evaluations and quality of solutions (Δf). Details on the nonparametric multiple contrast test are available in [15]. Furthermore, the statistical model considered a balanced design [21], with the ICM configuration as an experimental factor, and the significance level α was previously defined as 0.05.

[1] The tournament size was defined as 3 and the parameter kt equal 1.0 according to [18].

[2] A migration frequency equals to 5, means that from 5 to 5 generations there will be a migration process.

[3] It means that the best individual of an island substitutes a random one in the receiving island.

[4] As suggested in [1], the synchronous update policy is better to any of the asynchronous policies in relation to percentage of hits.

[5] Function mctp from *nparcomp* package of the statistical software R.

Table 2: Properties of optimization functions.

Properties	Functions
Unimodal and Separable	f_1
Unimodal and Partially Separable	f_4, f_8, f_{11}
Unimodal and Non-Separable	f_{13}, f_{14}
Unimodal and Full Non-Separable	f_{15}
Multimodal and Separable	f_2, f_3, f_{12}
Multimodal and Partially Separable	f_5, f_6, f_7, f_9, f_{10}

5 RESULTS

This section presents the comparisons among different configurations of ICM, CM and DE classic. Firstly, the subsection 5.1 presents a study of the ICM parameters. The second part, subsection 5.2, presents a general comparison of studied approaches.

5.1 Island-Cellular Model Parameters Study

Figures 4, 5 and 6 describe the interaction of the ICM parameters studied (neighborhood, migratory flow, and migratory frequency) with the optimization problem in regards to the average number of function evaluations until one of the stop criteria was reached. The Figure 4 demonstrates that different neighborhood patterns presented at least similar performance. However, it can be observed in this figure, the a slightly better performance of the *Compact 9* approach for unimodal and multimodal functions.

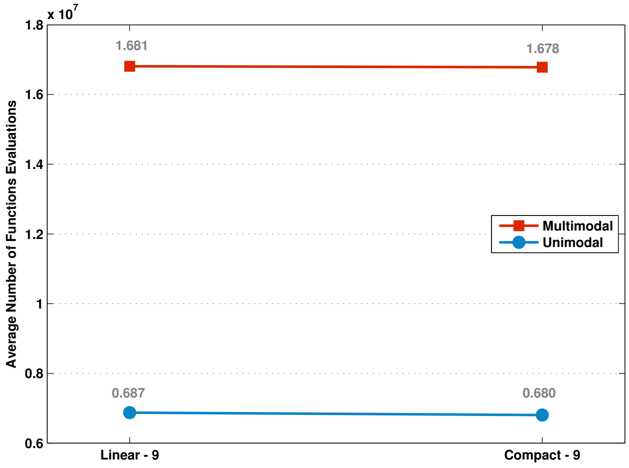


Figure 4: Cellular Neighborhood x Property Function.

The interaction of migratory flows in the ICM approach demonstrates that all configurations have the same behavior for both type of optimization functions, unimodal and multimodal, as shown in Figure 5. Moreover, it is interesting to notice the ICM performance using the Ring configuration. Although the Ring migratory flow configuration does not use any information about the evolutionary search, it has shown competitive performance. Regarding the migratory frequency of ICM, Figure 6 indicates a lower number of generations to perform the migration process tends to degrade the convergence speed. This behavior can be noticed on both unimodal and multimodal optimization functions.

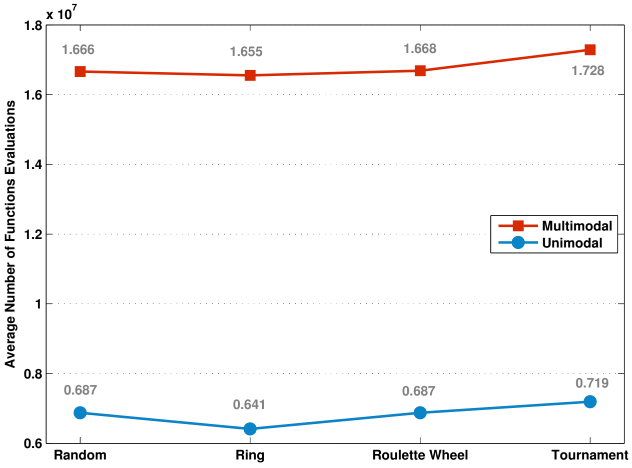


Figure 5: Migratory Flow x Property Function.

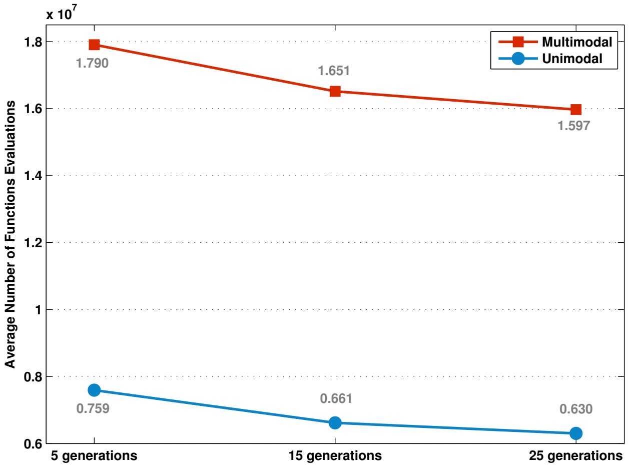


Figure 6: Migratory Frequency x Property Function.

5.2 General Study

Figure 7 presents the comparison performance of different configurations for the ICM according to the parameters previously described in Table 1. For this performance comparison, the quality of solutions and convergence speed are represented and the results

Table 3: Pareto set of Island-Cellular Model configurations.

ICM Configuration	Number of Islands	Migratory Frequency	Migratory Flow	Neighborhood
ICM–01	4 islands	15 generations	Roulette Wheel	Linear - 9
ICM–02	4 islands	25 generations	Random	Compact - 9
ICM–03	4 islands	25 generations	Ring	Compact - 9
ICM–04	4 islands	15 generations	Random	Compact - 9
ICM–05	8 islands	15 generations	Ring	Linear - 9
ICM–06	8 islands	25 generations	Ring	Linear - 9
ICM–07	16 islands	25 generations	Ring	Compact - 9

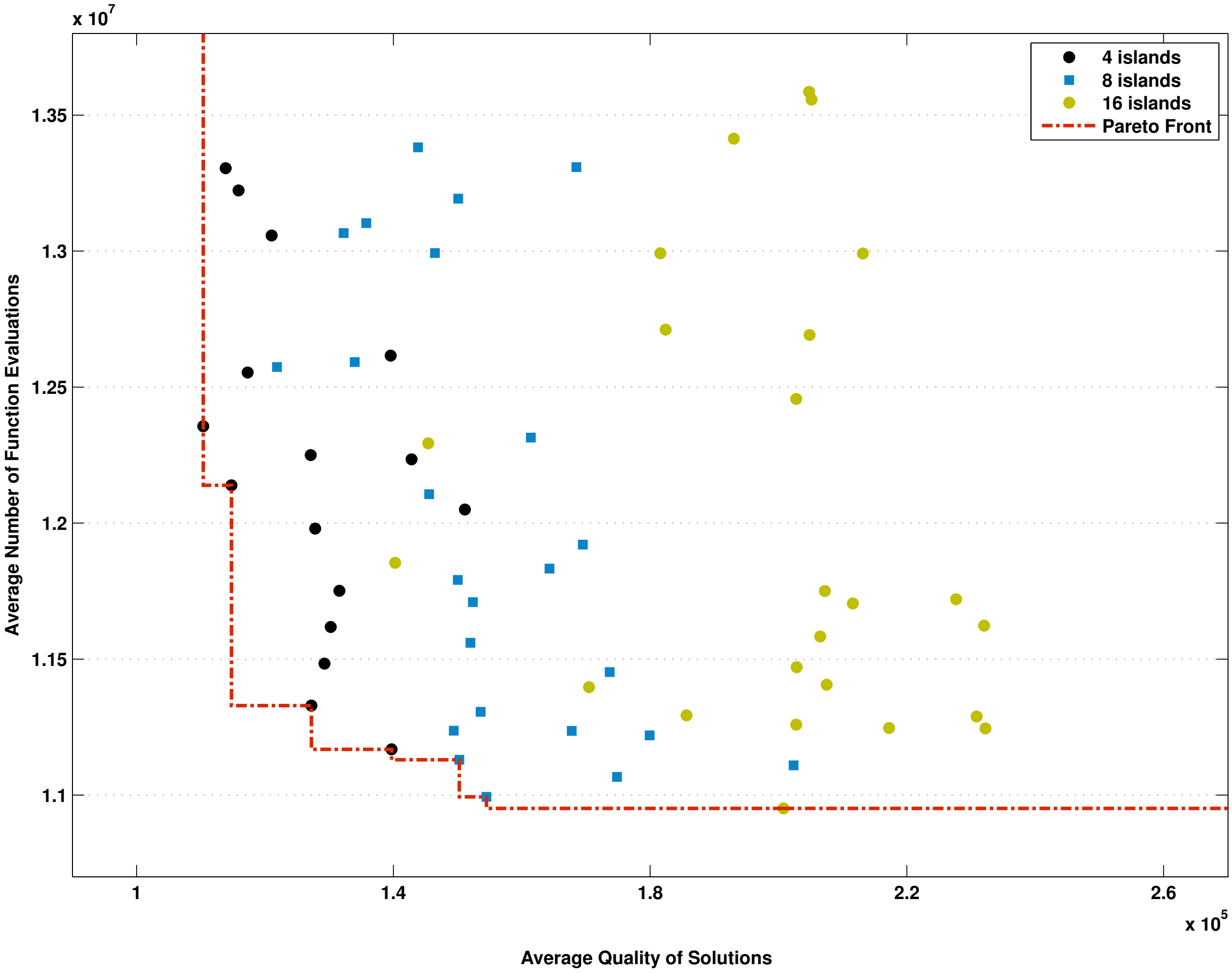


Figure 7: Experimental results of Island-Cellular Model approach.

Table 4: Experimental results of Island-Cellular Model Pareto set.

Approach	Average Number of Function Evaluations	Average Number of Quality Solutions (Δf)
ICM–01	12,138,612.66	114,796.55
ICM–02	11,168,298.0	139,725.12
ICM–03	11,329,323.33	127,228.86
ICM–04	12,356,096.66	**110,395.21**
ICM–05	11,129,529.33	150,296.14
ICM–06	10,993,794.66	154,517.59
ICM–07	**10,950,960.0**	200,796.77

were joined in groups by the number of islands. As observed in Figure 7, the ICM approach with a small number of islands tends to improve the quality of solutions.

In relation to the general results of different configurations of ICM, the trade-off between the quality of solutions and convergence speed is important to highlight. In order to clarify this point, the non-dominated[6] ICM configurations are identified and the Pareto front[7] is graphically illustrated by a red dashed line in Figure 7. Thus, the ICM approaches overlapping the Parent front described in this Figure are known as the set which is Pareto optimal. Taking notice of the Pareto front, seven different configurations of ICM can be remarked and they are presented in Table 3.

Tables 4 and 5 present the experimental results of the ICM configurations in the Pareto front and the classic versions of CM e DE, respectively. In Table 4, the configurations of the ICM marked in bold represent the best configuration in relation to each objective. Regarding convergence speed, the configuration *ICM–07* presented the minimal averaged number of function evaluations. For the quality of solution, the approach *ICM–04* presented the best performance among all configurations.

The results of the statistical tests regarding the number of function evaluations are described in Table 6. Thus, p-values of the statistical tests help us to determine the significance of the results. The applied test considers all configurations of the Pareto front (see Table 3 and 4) and classical algorithms (CM and DE), but only paired tests of configuration *ICM–07* are presented. In this statistical tests, only the effect of the methods was considered. Given the confidence level previously defined as 95%, the results show that the *ICM–07* configuration significantly spends fewer function evaluations than the DE classic and CM approaches.

Table 7 presents the results of the statistical tests on the mean of the quality of solution (Δf) achieved by all configurations from the Pareto front and classical versions. This test considers only the effect of the methods, ignoring the effects of the functions tests. In this table, only the results for *ICM–04* were displayed and they show that it significantly achieves better solutions than DE classic and CM approaches with 95% of confidence.

Table 5: Experimental results of Cellular Model and Differential Evolution classic versions.

Approach	Average Number of Function Evaluations	Average Number of Quality Solutions (Δf)
DE - Classic	14,584,812.5	56,831,241.48
CM - Linear 9	13,706,617.5	83,503,416.06
CM - Compact 9	13,651,048.5	55,795,725.98

Table 6: p-values of the statistical analysis for the number of function evaluation.

Configuration 1	Configuration 2	p-value
ICM–07	*DE - Classic*	**0.0000**
ICM–07	*CM - Linear 9*	**0.0000**
ICM–07	*CM - Compact 9*	**0.0000**
ICM–07	*ICM–01*	0.1126
ICM–07	*ICM–02*	0.9999
ICM–07	*ICM–03*	0.9959
ICM–07	*ICM–04*	**0.0237**
ICM–07	*ICM–05*	0.9999
ICM–07	*ICM–06*	1.0000

Table 7: p-values of the statistical analysis for the quality of solutions.

Configuration 1	Configuration 2	p-value
ICM–04	*DE - Classic*	**0.0000**
ICM–04	*CM - Linear 9*	**0.0000**
ICM–04	*CM - Compact 9*	**0.0000**
ICM–04	*ICM–01*	0.9999
ICM–04	*ICM–02*	0.9934
ICM–04	*ICM–03*	0.9980
ICM–04	*ICM–05*	0.9466
ICM–04	*ICM–06*	0.9847
ICM–04	*ICM–07*	0.3769

6 CONCLUSIONS

In this article, an Island-Cellular Model has been presented and discussed. The different configurations of the ICM were coupled to a Differential Evolution algorithm and a thorough experimental analysis of the ICM approach was conducted. The main goal of this paper was to study the ICM performance for Large-Scale Global Optimization problems. In addition to that, we aimed to cover the lack of knowledge about the influence of ICM parameters and comparison between different ICM configurations.

The results on well-known large-scale benchmark functions demonstrate that there is a difference in the DE performance due to different configurations of ICM approach. They also show the trade-off between the quality of solutions and convergence speed. In synthesis, the general experimental results indicate that small ICM approaches are able to converge faster and achieve good solutions. The increase of the selection pressure in small models can

[6] A non-dominated set is a group of solutions such that all them are better than the others for at least one objective.

[7] The Pareto front is composed of all non-dominated solutions from the feasible solution set.

explain its good performance. Moreover, it has been shown that long migration gaps and naive migratory flows (Random and Ring) improve the ICM performance. The reason for that is likely to be the population diversity and adequate selection pressure.

Although the CM and DE classic are important versions of this algorithm to solve optimization problems in continuous space, the results have been demonstrated the significantly out-performance of ICM comparing to the other studied approaches. The experimental analysis of the results presents evidence about the ICM performance reinforcing its ability to improve the EAs performance.

Concluding this study, it has accomplished its aim producing knowledge about the ICM approach for EAs leading with Large-Scale Global Optimization problems. Furthermore, this article will assist future studies of EAs distributed approaches and their applications.

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.

REFERENCES

[1] E. Alba, M. Giacobini, M. Tomassini, and S. Romero. 2002. Comparing Synchronous and Asynchronous Cellular Genetic Algorithms. In *Proceedings of the 7th International Conference on Parallel Problem Solving from Nature*. Springer-Verlag, London, UK, UK, 601–610.

[2] T. Bäck. 1996. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms.* Oxford University Press, Oxford, UK.

[3] S. P. Brooks and B. J. T. Morgan. 1995. Optimization using simulated annealing. *The Statistician: Journal of the Institute of Statisticians* 44 (1995), 241–257.

[4] T. Burczynski and W. Kus. 2004. *Optimization of Structures Using Distributed and Parallel Evolutionary Algorithms.* Springer Berlin Heidelberg, Berlin, Heidelberg, 572–579.

[5] T. Burczynski, W. Kus, A. Dlugosz, and P. Orantek. 2004. Optimization and defect identification using distributed evolutionary algorithms. *Engineering Applications of Artificial Intelligence* 17 (2004), 337–344.

[6] E. Cantú-Paz. 1998. A Survey of Parallel Genetic Algorithms. *Calculateurs Paralleles, Reseaux et Systems Repartis* 10 (1998).

[7] F. F. de Vega. 2016. *Evolutionary Algorithms: Perspectives on the Evolution of Parallel Models.* Springer International Publishing, Cham, 13–22.

[8] B. Dorronsoro and P. Bouvry. 2010. Differential Evolution Algorithms with Cellular Populations. In *Proceedings of the 11th International Conference on Parallel Problem Solving from Nature: Part II*. Springer-Verlag, Berlin, Heidelberg, 320–330.

[9] G. Folino, C. Pizzuti, and G. Spezzano. 2008. Training Distributed GP Ensemble With a Selective Algorithm Based on Clustering and Pruning for Pattern Classification. *IEEE Transactions on Evolutionary Computation* 12 (2008), 458–468.

[10] G. Folino and G. Spezzano. 2006. *P-CAGE: An Environment for Evolutionary Computation in Peer-to-Peer Systems.* Springer Berlin Heidelberg, Berlin, Heidelberg, 341–350.

[11] D. E. Goldberg and J. H. Holland. 1988. Genetic Algorithms and Machine Learning. *Machine Learning* 3, 2 (1988).

[12] Y. Gong, W. Chen, Z. Zhan, J. Zhang, Y. Li, Q. Zhang, and J. Li. 2015. Distributed evolutionary algorithms and their models: A survey of the state-of-the-art. *Applied Soft Computing* 34 (2015), 286–300.

[13] F. Herrera, M. Lozano, and C. Moraga. 1999. Hierarchical distributed genetic algorithms. *International Journal of Intelligent Systems* 14 (1999), 1099–1121.

[14] J. Kennedy and R. C. Eberhart. 1995. Particle swarm optimization. In *Proceedings of the 1995 IEEE International Conference on Neural Networks*, Vol. 4. Perth, Australia, IEEE Service Center, Piscataway, NJ, 1942–1948.

[15] F. Konietschke, L. A. Hothorn, and E. Brunner. 2012. Rank-based multiple test procedures and simultaneous confidence intervals. In *Electronic Journal of Statistics*, Vol. 6. The Institute of Mathematical Statistics and the Bernoulli Society, 738–759.

[16] X. Li, K. Tang, M. N. Omidvar, Z. Yang, and K. Qin. 2013. Benchmark Functions for the CEC"2013 Special Session and Competition on Large Scale Global Optimization. (2013).

[17] D. Lim, Y. Ong, Y. Jin, B. Sendhoff, and B. Lee. 2007. Efficient Hierarchical Parallel Genetic Algorithms using Grid computing. *Future Generation Computer Systems* 23 (2007), 658–670.

[18] R. A. Lopes, R. C. Pedrosa Silva, F. Campelo, and F. G. Guimarães. 2013. 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.* ACM, 173–174.

[19] R. A. Lopes, R. C. Pedrosa Silva, A. R.R. Freitas, F. Campelo, and F. G. Guimarães. 2014. A Study on the Configuration of Migratory Flows in Island Model Differential Evolution. In *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation.* ACM, New York, NY, USA, 1015–1022.

[20] S. Mahdavi, M. E. Shiri, and S. Rahnamayan. 2015. Metaheuristics in large-scale global continues optimization: A survey. *Information Sciences* 295 (2015), 407 – 428.

[21] D. C. Montgomery. 2006. *Design and Analysis of Experiments.* John Wiley & Sons.

[22] G. C. Onwubolu and D. Davendra. 2009. *Differential Evolution: A Handbook for Global Permutation-Based Combinatorial Optimization* (1st ed.). Springer Publishing Company, Incorporated.

[23] R. S. Prado, R. C. P. Silva, F. G. Guimar aes, and O. M. Neto. 2010. Using differential evolution for combinatorial optimization: A general approach. In *2010 IEEE International Conference on Systems, Man and Cybernetics.* 11–18.

[24] K. V. Price, R.N. Storn, and J. A. Lampinen. 2005. *Differential Evolution: A Practical Approach to Global Optimization.* Springer.

[25] R. Storn and K. Price. 1997. Differential Evolution: A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization* 11 (1997), 341–359.

[26] R. Subbu and A. C. Sanderson. 2004. Modeling and convergence analysis of distributed coevolutionary algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 34 (2004), 806–822.

[27] J. Tang, M. Lim, Y. Ong, and M. J. Er. 2004. 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.* IEEE, 2286–2291.

[28] D. K. Tasoulis, N.G. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis. 2004. Parallel Differential Evolution. In *In IEEE Congress on Evolutionary Computation.*

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