

Scalability in Island Models: computational experiments with the Social Spider Algorithm

Grasiele Regina Duarte^{a1}, Afonso Lemonge^a and Leonardo Goliatt^a

^aFederal University of Juiz de Fora, Juiz de Fora, MG, Brazil

Received on September 25, 2016 / accepted on October 19, 2016

Abstract

Bio-inspired metaheuristics are powerful optimization tools that often produce very good results for complex optimization problems. However, metaheuristics usually require high computational costs to obtain satisfactory solutions. An alternative to tackle this problem is to implement them with the use of the so-called Island Model. This model consists into partition the set of candidate solutions in sub-populations (islands) that evolve in parallel. The main feature of the island model is the migration of candidate solutions from one island to another along the iterations of the search process. A strategy to improve the convergence of an island-based metaheuristic is to increase the number of islands. However, as the number of islands increases, the increasing communication among them. In this paper we investigated the impact of increasing the number of islands on the quality of the result by applying a specific algorithm using the Island Model compared with the performance of isolated execution of the same algorithm. As optimization tool we have used the Social Spider Algorithm, inspired by the foraging behavior of social spiders that has produced competitive results in the literature. The experiments were conducted in a set of unconstrained optimization problems available in the literature and statistical metrics.

Keywords: Island Model, Social Spider Algorithm, Parallel Computing.

1 Introduction

Several bio-inspired algorithms were recently proposed in the literature to solve complex optimization problems from different areas. Many of these algorithms are population-based and the search for the optimal solutions occur iteratively throughout the evolution of a set of candidate solutions. In

¹E-mail Corresponding Author: grasiele.duarte@engenharia.ufjf.br

2015, inspired by the foraging strategy of social spiders was proposed in [11] the population Social Spider Algorithm (SSA) to solve global optimization problems. The SSA has shown competitive with other classical algorithms of literature and for this reason, is the object of study in this work.

A feature of the population bio-inspired algorithms, including SSA, is the high computational cost required for its execution. However, another feature of most of these algorithms is that they can be implemented for parallel execution relatively easily. With the increasing popularity of computers equipped with multi-cores processors and also the use of computer networks a parallelization strategy for iterative and population algorithms that has been receiving more and more attention of researchers is the Island Model.

In the Island Model, the set of candidate solutions is partitioned into sub-populations called islands that will be evolved by specific algorithms in a parallel architecture. A fundamental operation in the Island Model is the migration that is responsible for transferring candidate solutions between the islands periodically considering a set of user-defined parameters. The Island Model promotes speed up of the algorithms at the same time that improves the quality of the solution of the algorithms.

In this paper, we propose a non-hybrid Island Model composed by islands running an instance of SSA. Our aim is to investigate the impact that the Island Model can cause to the performance of the SSA. In addition, we have evaluated the influence of the number of islands in the performance of the model in order to verify to what extent it is interesting to invest in hardware for execution of non-hybrid Island Model with the aim to accelerate the execution and maintain or improve the quality of the result of the algorithms at the same time. For this study we used a set of unconstrained optimization problems available in the literature and the analysis of the results was directed by statistical metrics.

2 Island-based Social Spider Algorithm

2.1 Social Spider Algorithm

SSA [11] is a bio-inspired meta-heuristic based on the foraging behavior of social spiders. In SSA the search space is a hyper-dimensional artificial spider web and each position is a candidate solution. The quality of each solution represents the possibility to find a prey at the position [11]. A vibration produced in a position by an artificial spider will be received by others spiders with different intensities $\in (0, \infty)$ depending on the distances between each pair of spiders. In the SSA the vibration intensity generated

by the spider s at the source position \mathbf{P}_s is given by

$$I(\mathbf{P}_s, \mathbf{P}_s, t) = \log\left(\frac{1}{f(\mathbf{P}_s) - C} + 1\right) \quad (s = 1, 2, \dots, N), \quad (1)$$

where $f(\mathbf{P}_s)$ is the objective function value at position \mathbf{P}_s and C is a small constant such that all possible values $f(\mathbf{P}_s)$ are larger than C and N is the population size.

The vibration attenuation over distance between the spiders a and b is defined by

$$I(\mathbf{P}_a, \mathbf{P}_b, t) = I(\mathbf{P}_a, \mathbf{P}_a, t) \times \exp\left(-\frac{D(\mathbf{P}_a, \mathbf{P}_b)}{\bar{\sigma} \times r_a}\right), \quad (2)$$

where $D(\mathbf{P}_a, \mathbf{P}_b)$ is the distance between the spiders a and b , calculated by L_1 -norm, $\bar{\sigma}$ is the average of the standard deviation along each dimension among all spider positions and $r_a \in (0, \infty)$ represents the attenuation rate.

In the initialization phase of SSA, each artificial spider occupies a random position in the web and its target vibration, namely v_s^{tar} , is set as the vibration in its current position. For a number of iterations *max_iter* the operations of SSA can be described by: (i) Fitness evaluation; (ii) Vibration generation; (iii) Mask changing; (iv) Random walk; (v) Constraint handling [11].

Each spider s receives N different vibrations among iterations of SSA and select the strongest (v_s^{best}). If v_s^{best} is larger than v_s^{tar} , the spider will store v_s^{best} as v_s^{tar} . A random walk is performed towards v_s^{tar} considering a binary vector (mask) \mathbf{m}_s associated to each spider. At each iteration each spider has a probability $1 - p_c$, $p_c \in (0, 1)$ to change its mask. If the mask should be changed then each bit has a probability $p_m \in (0, 1)$ to be assigned with 1 and $1 - p_m$ to be a 0.

Finally, each spider moves to a new position defined under moderated influences by random values $\in [0, 1]$, of the current position, of the previous position, of the source position of v_s^{tar} and also random positions which produce less intense vibrations. The pseudo-code of the SSA is available in in [11].

2.2 Island Model

Bio-inspired algorithms, including SSA, often require a high computational cost to solve an optimization problem. An alternative to tackle this drawback is to implement them applying a strategy to a parallel execution.

A parallelization strategy for these algorithms that has been receiving increasing attention of the researchers is the Island Model. An Island Model consists in a distributed implementation of population-based algorithms that enables speed up the execution and at the same time improves the quality of the obtained solutions.

In the Island Model, the set of candidate solutions is partitioned into several sub-populations called islands. Each island evolves its population with a specific algorithm in a parallel architecture that can be a computer equipped with a multi-core processor or several machines in a computer network. All islands can evolve their sub-populations with different instances of the same algorithm (*non-hybrid model*) or each island can run a distinct algorithm of the others (*hybrid model*). There is a fundamental operator in the Island Model called migration which transfers solutions from an island to another. In the migration process, good solutions (migrants) can be transferred to islands with solutions not so good, thus contributing to the evolutionary process of the model. Basically, the implementation of the Island Model involves decisions about the following parameters [2]:

- Number of islands: Number of instances of algorithms (or sub-populations);
- Migration topology: Defines how the islands are connected to others, which defines the paths available for migration. There are several possibilities reported in the literature [3, 4], however the most common is known as topology ring where the available connections are only between the two closest neighboring islands and the communication can be uni- or bi-directional.
- Migration rate: Indicates how many individuals migrate from the source island every occurrence of migration;
- Migration frequency: Defines how often the migration occurs.

Furthermore, the migration process can be *synchronous* (the exchange of solutions occurs at the same time for all islands) or *asynchronous* (each island promotes migration as soon as it is ready to do this). In addition, an important parameter is the the distribution of migrants solutions (or migration policy). The distribution can be *point to point* (solutions of a source island migrate to exactly one destination island) or *broadcast* (solutions of a source island are sent to all its neighboring islands). Much of the research of the Island Model aims at assessing the impact of the parameter values in the performance of the model [5, 6, 7, 8, 9, 2, 10], however, there are few

are 40 random combinations of the three parameters of SSA ($r_a \in [0.2, 3.0]$, $p_c \in [0.4, 0.9]$ and $p_m \in [0.1, 0.3]$). In each experiment, the algorithms were chosen based on the number of islands. For example, the model with 10 islands employs the 10 first algorithms (SSA-1 to SSA-10). The total number of iterations of each SSA was set to 1250 divided according to the occurrences of the migration process. The parameters of the Island Model were set as follows: (i) Migration topology: Ring Topology; (ii) Migration rate: 10% of the individuals of an island; (iii) Migration frequency: 25 - islands migrate solutions at each 50 (1250/25) iterations. A greedy strategy was adopted for the selection of migrants solutions. The best solutions of the source island were chosen to replace the worst ones on the destination island.

The proposed model was submitted to the resolution of the first 10 problems of the 15 proposed in [12], identified in this paper as F_i ($i = 1, 2, \dots, 10$). We have set the number of dimensions equal to 50. For each problem, were performed 30 independent runs for each Island Model and 10 independent runs of each of the 40 instances of the sequential SSA. Thus, in both cases, it was considered 500000 (1250×400) evaluations as the maximum number of function evaluations as defined in [12]. The experiments were performed on computer equipped with CPU AMD OpteronTM Processor 6272 (64 cores of 2.1GHz and cache memory of 2MB), RAM of 250GB and operational system Linux Ubuntu 14.04.4 LTS.

3 Results and Discussion

Figure 1 presents the boxplots of the distribution of the best objective function values found in the 30 independent runs by each model according to the number of islands. In the figure the model composed by only one island represents the best results obtained in the 10 runs by any sequential implementation of SSA (SSA-1, SSA-2, ... or SSA-40). The influence of the number of islands is similar for all problems: increasing the number of islands leads to an overall improvement in the final results. This is evident by observing the reduction of the median in the boxes in the most of subfigures. Migration has a positive impact on the algorithm performance.

Figures 1(b), 1(g) and 1(j) show that for problems F2, F7 and F10, increasing number of islands can even worse results. Possibly this is due to the reduction of the size of the local population of each island, which reduces local diversity and may cause premature convergence. This result may be related to the nature of the problem, in 75% of cases the increase the

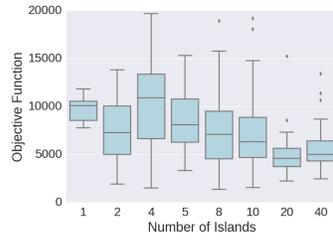
number of islands caused a positive effect on the final result. Thus, based on the values used in this work for the number of islands, it is interesting to use a large number of islands to improve the quality of the solution produced by the SSA.

Figure 2 shows that the speedup obtained is not linear. The super-linear speedup obtained probably is due to the reduction of the set of candidate solutions in the islands. The portion of the set stored in cache memory increases and the use of this memory accelerates the process. Observing the Figure 2 on can see that speedup rate is reduced as the number of islands increases, suggesting the growth of the cost of communication (overhead) for a larger number of islands. The results from this figure show that the non-hybrid Island Model SSA reduces considerably the computational cost and improves the final solutions. It is important to state that the hardware resources used for the experiments were not dedicated and there were other processes running simultaneously.

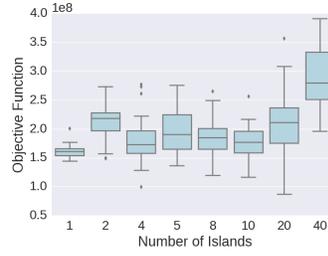
4 Conclusions

In the paper we have proposed a non-hybrid Island Model Social Spider Algorithm. Our model was evaluated with different numbers of islands in order to verify when is interesting to invest in hardware for a parallel execution of SSA, considering supercomputers with multi-cores processors up to 64 cores. Based on the experiments conducted here, we conclude that the proposed model have improved the performance of the SSA considering the final solutions and runtime. The increasing number of islands affected positively the model performance, showing that it is interesting invest in resources for parallel execution of the algorithm in order to obtain better results and higher speed compared with its sequential execution. As a future work we intend to experiment new alternatives regarding the parameters of the Island Model in order to evaluate the influence on the convergence of the SSA. We also intend to try a larger numbers of islands aiming executions in architectures based on computer networks such as clusters and grids where hundreds or thousands of processors can be used.

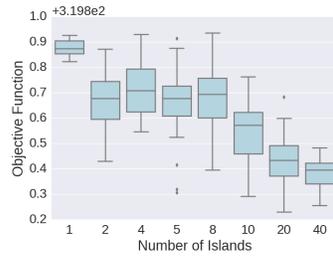
Acknowledgments. The authors thank the Graduate Program in Computational Modeling (UFJF) and Brazilian Agencies CNPq (grant 305099/2014-0) and FAPEMIG (grants TEC PPM 528/11, TEC PPM 388/14 and TEC APQ 01606/15) for the financial support.



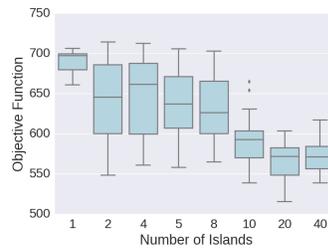
(a) F1.



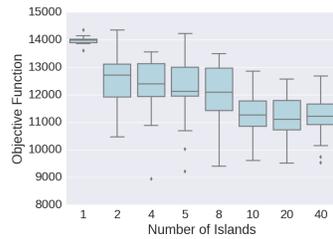
(b) F2.



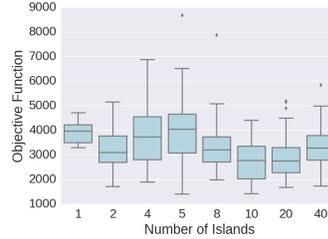
(c) F3.



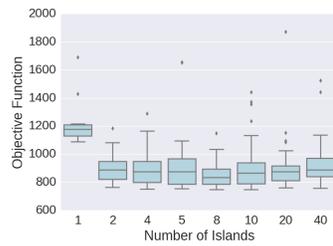
(d) F4.



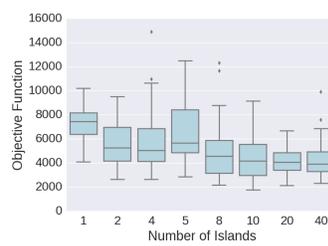
(e) F5.



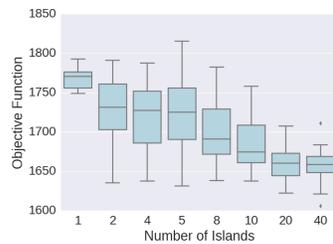
(f) F6.



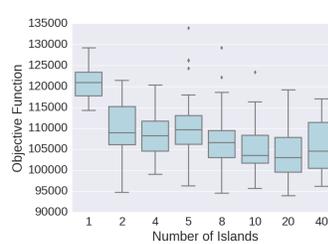
(g) F7.



(h) F8.



(i) F9.



(j) F10.

Figure 1: Boxplots of the distribution of best objective function values along the 30 runs of the Island Model.

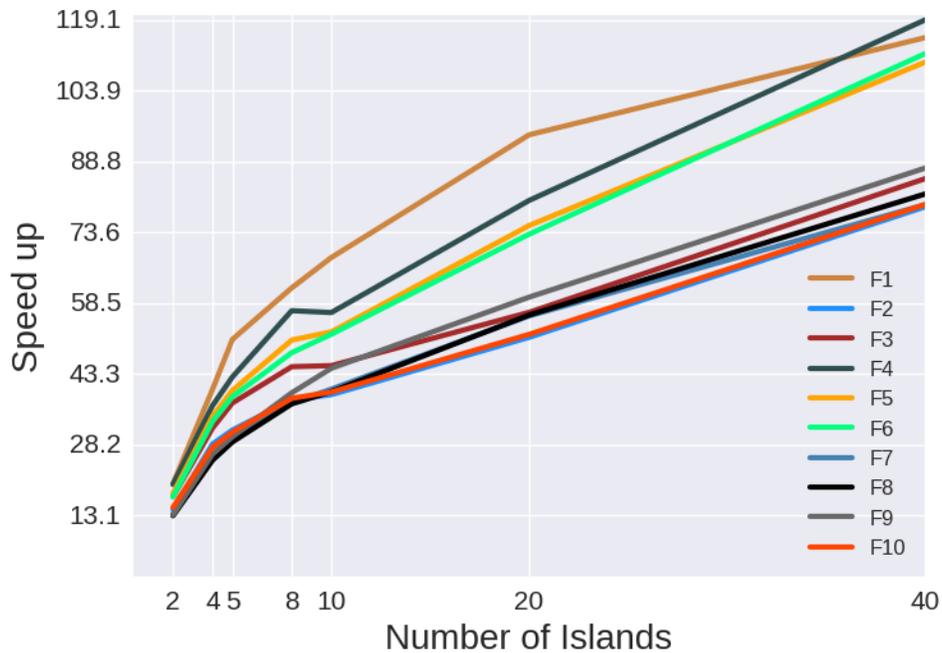


Figure 2: Speedup obtained for each problem.

References

- [1] Yu, J. J. Q. and Li, V. O. K., Parameter sensitivity analysis of social spider algorithm, In *2015 IEEE Congress on Evolutionary Computation (CEC)*, p. 32003205. ISBN 978-1-4799-7492-4, 2015a.
- [2] Ruciski, M., Izzo, D. and Biscani, F., On the impact of the migration topology on the island model, *Parallel Computing*, 36(1011):555571, ISSN 0167-8191, 2010, Parallel Architectures and Bioinspired Algorithms.
- [3] Tasoulis, D. K., Pavlidis, N. G., Plagianakos, V. P. and Vrahatis, M. N., Parallel differential evolution, In *Evolutionary Computation, 2004, CEC2004, Congress on*, volume 2, p. 20232029, IEEE, 2004.
- [4] Izzo, D., Rucinski, M. and Ampatzis, C., Parallel global optimisation metaheuristics using an asynchronous island-model, In *2009 IEEE Congress on Evolutionary Computation*, p. 23012308, 2009.

- [5] Cantú-Paz, E., A survey of parallel genetic algorithms, *Calculateurs paralleles, reseaux et systems repartis*, 10(2):141171, 1998.
- [6] Alba, E. and Troya, J. M., A survey of parallel distributed genetic algorithms, *Complexity*, 4(4):3152, ISSN 1099-0526, 1999.
- [7] Crainic, T. G. and Toulouse, M., *Parallel Strategies for Meta-Heuristics*, p. 475513, 2003, Springer US, Boston, MA. ISBN 978-0-306-48056-0.
- [8] Alba, E., *Parallel Metaheuristics: A New Class of Algorithms*, 2005, Wiley, ISBN: 978-0-471-67806-9.
- [9] Skolicki, Z. M., *An Analysis of Island Models in Evolutionary Computation*. PhD thesis, 2007, Fairfax, VA, USA. AAI3289714.
- [10] Lopes, R. A., Silva, R. C. P., Campelo, F. and Guimares, F. G., Dynamic selection of migration flows in island model differential evolution, In *Proceedings of the 15th annual conference companion on Genetic and evolutionary computation*, p. 173174, ACM, 2013.
- [11] Yu, J. J. and Li, V. O., A social spider algorithm for global optimization. *Applied Soft Computing*, 30:614627, ISSN 1568-4946, 2015.
- [12] Liang, J. J., Qu, B. Y., Suganthan, P. N. and Chen, Q., Problem definitions and evaluation criteria for the cec 2015 competition on learning-based real-parameter single objective optimization, Technical report, 2014, Nanyang Technological University.