

A Study of Topology in Insular Parallel Genetic Algorithms

Leonardo G. Tavares, Heitor S. Lopes, Carlos R. Erig Lima

Bioinformatics Laboratory

Federal University of Technology Parana (UTFPR)

Av. 7 de setembro, 3165 80230-901, Curitiba (PR), Brazil

leonardo.tavares@up.edu.br, hslopes@pesquisador.cnpq.br, erig@utfpr.edu.br

Abstract— In this paper we study how the connectivity affects the performance of insular Parallel Genetic Algorithms (PGAs). Seven topologies PGAs were proposed, with growing number of connections. We used three instances of the well-known traveling salesman problem as benchmark. Each island of the PGA had different parameters and we established a fixed migration policy for all islands. Experiments were done and average results were reported. The effect of coevolution in PGAs was evidenced. The convergence time increased with the number of connections of the topology. The quality of solutions also increased in the same way. Although topologies with large connectivity increases the overall processing time, they take benefits to the quality of solutions found.

I. INTRODUCTION

Evolutionary Computation (EC) has been successfully used for many complex computational and engineering problems. There are many difficult problems that request the use of parallel implementations. Amongst the several EC paradigms, Genetic Algorithms (GA) are, probably, the most widely used, thanks to its simplicity, flexibility and efficiency.

A. Parallel Genetic Algorithms

Genetic Algorithm (GA) is a heuristic search method that has been widely used in applications where the size of the search space is very large. In essence, GAs are “search algorithms based on the mechanics of natural selection and natural genetics” [3]. That is, GAs are inspired by the principle of the survival of the fittest, where the fittest individuals are selected to produce offspring A for the next generation. In the context of search, individuals are candidate solutions to a given search problem. Hence, reproduction of the fittest individuals means reproduction of the best current candidate solutions. Genetic operators such as selection, crossover and mutation generate offspring from the fittest individuals. One of the advantages of GAs over traditional search methods is that the former performs a kind of global search using a population of individuals, rather than performing a local, hill-climbing search. Global search methods are less likely to get trapped into local maxima, in comparison with local search methods. Also, GAs do not work directly with the parameters of the problem. Instead, they deal with encoded solutions, thus making the method

more general and capable of being applied to several different domains.

Usually, GA implementations have several control parameters and its application to real-world problems requests fine-tuning such parameters in order to obtain the maximum performance. Although there are many self-adjusting approaches for this purpose, trial-and-error is still widely used. This is a motivation for using PGAs, because one can try many different sets of parameters in parallel. As a side-effect, diversity is greatly encouraged when using several different populations. Besides, one can take advantage of the co-evolution feature that may arise from PGAs. That is, GAs with different parameters can explore different regions of the search space and a PGA can combine this search empowering the overall algorithm.

Basically, there are three categories of PGAs: Master-Slave, Cellular, and Island PGA. The Master-Slave PGA works with a single global population managed by the master processor which is responsible for all tasks of the AG, but fitness computation. The slave processors are responsible only for computing the fitness of individuals. This type of PGA is usually employed for problems in which the computational cost of the fitness function is high. This PGA has the advantage of being easily implemented in a network of computers, as represented in Figure 1.

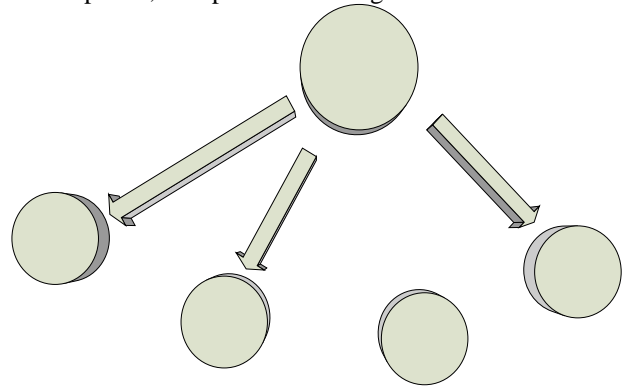


Figure 1. Master-Slave PGA.

The Cellular PGA, shown in Figure 2, uses a single population, which individuals are set over a planar grid, each

one managed by a processor. Genetic operations between individuals are performed only between neighbor individuals, according to the grid topology. The Cellular PGA is very sparsely reported in the literature because it requests special hardware, that is, massively parallel computers.

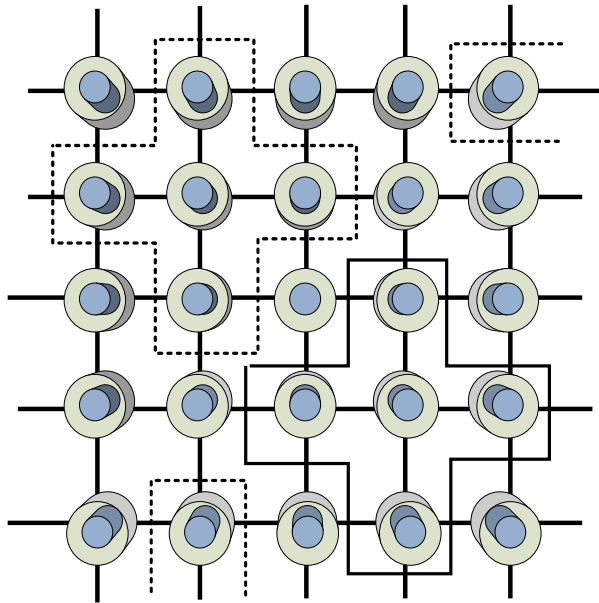


Figure 2. Cellular PGA

Island PGA is, possibly, the most frequently approach used for real-world problems. In this model, depicted in Figure 3, each island evolves an independent GA, usually with different running parameters. According to a predefined migration policy and connectivity, individuals of an island are sent out to another island. This model can be implemented in a network of computers, each one running an island, or simulated in a single computer. This last approach is frequently more efficient, since there is no overhead due to the communication between islands.

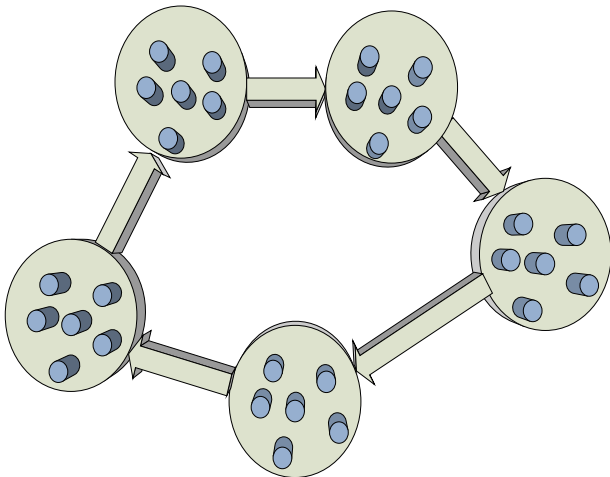


Figure 3. Island PGA.

The migration policy in Island PGAs includes the following issues:

- The topology that defines how islands are connected each other. This is a crucial issue for the performance of the PGA and it is the focus of this work. The topology has direct implication on the quality of the obtained solutions as well on the convergence of the GA.
- The migration rate controls the number of individuals that are able to migrate at each generation.
- The migration frequency determines the time interval (measured in generations) between each migration.
- The migration scheme specifies which individuals of a given population can migrate, as well as which individuals of another population will be substituted once a migrant arrives.

Some of the above issues of PGAs have been already investigated elsewhere [6],[8], whereas the objective of this work is to study specifically the effect of the topology and connectivity between islands of a PGA. For a further comprehensive study of several aspects of PGAs, see [2].

II. METHODOLOGY

The study of the influence of the topology of a PGA in its performance was carried out using a classical problem as case study.

A. The Traveling Salesman Problem

The Traveling Salesman Problem (TSP) is a classical problem of combinatorial optimization [1]. The TSP is, probably, the most widely studied problem of this category and has been used for testing many algorithms and metaheuristics, specially those related to evolutionary computation [4],[7]. TSP is the problem of finding the shortest closed tour (shortest Hamiltonian cycle) through a given set of n points visiting each point exactly once and returning to the starting point, minimizing the total cost of the tour. Variants of the TSP have many practical applications, such as in X-ray crystallography, integrated circuits design, vehicle routing, scheduling, mobile computing, robotics, etc.

B. Benchmarks

For this study, three instances of the TSPLIB benchmark were chosen [5]. These instances, shown in Table 1, have a growing number of cities. Since TSP is a NP-hard problem, the difficulty of the instance grows exponentially with the number of cities.

TABLE I. INSTANCES OF THE TSP USED AS BENCHMARK

Name	Number of cities	Optimal known solution
Berlin52	52	7542
St70	70	675
Lin105	105	14379

III. EXPERIMENTS

An Island PGA was implemented in C++ programming language and run in a desktop computer. For all experiments the PGA had 8 islands and the parallelism was simulated running a full generation in each population in turn. Then, the migration policy is executed, according to the predefined topology and connectivity between islands (populations). The crossover and mutation rates were set differently for each island of the PGA (see Table 2). The remaining running parameters of the PGA were set exactly the same for all islands, as follows:

- population size: 100 individuals,
- selection method: stochastic universal sampling,
- crossover operator: uniform order-based,
- stopping criterion: 500 generations without improvement in the best-so-far solution (convergence).

TABLE II. SPECIFIC RUNNING PARAMETERS OF EACH ISLAND PGA

	#1	#2	#3	#4	#5	#6	#7	#8
pcrss	0.9	0.8	0.7	0.9	0.8	0.7	0.9	0.8
Pmut	0.03	0.03	0.03	0.05	0.05	0.05	0.08	0.08

The migration policy was defined as follows: each 10 generations, two individuals emigrate from each population: the best individual so far and a randomly chosen individual. The incoming individuals substitute two randomly chosen individuals of the current population. It is important to note that, actually, the immigrant is a copy of the original individual, not itself.

A total of seven different topologies were tested, as shown in Figure 4. The first topology (4a) is a star in which a single population receives immigrants from all other seven populations. The second topology (4b) is a single ring, where each population receives immigrants from only one neighbor, in the clockwise direction. The next topology (4c) is a double ring: each population receives immigrants from both neighbors. The fourth topology (4d) is a cube: all populations receive immigrants from the three closest neighbors. The next topologies (4e, 4f and 4g), each population is doubly-connected with five, six and seven neighbors, respectively. The direction of migration is not represented only for the sake of clarity. The total number of connections between islands grows according to the topology, respectively, 7, 8, 16, 24, 32, 48, and 56.

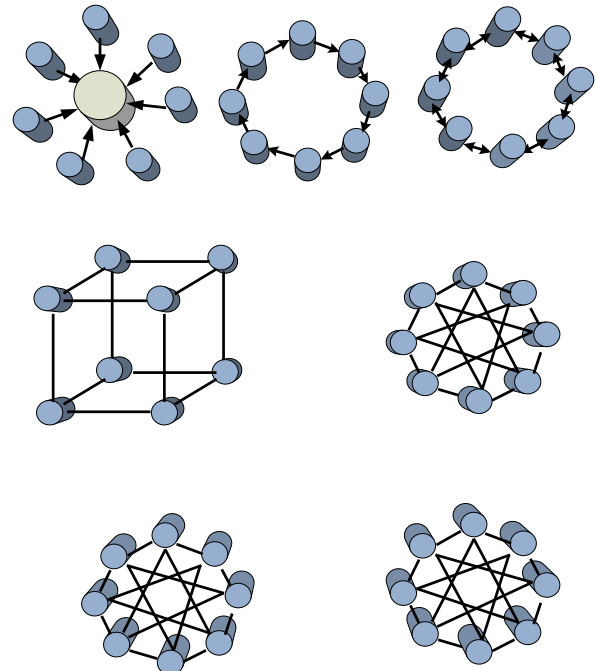


Figure 4. Topologies for the PGA.

IV. RESULTS

Due to the stochastic nature of a GA, every experiment was run for 50 times and numerical results presented below refer to average values. All experiments were done in a desktop computer with Athlon processor at 1,1 GHz, under Suse Linux 9.

A sequential GA (referred as SGA) was also run to compare with the parallel versions. The same convergence criterion of the PGAs was used for the SGA.

Figure 5 presents the average processing time until convergence (according to the stopping criterion) as function of the topology. Overlapped with the vertical bars, there are tendency lines represented by the linear interpolation of the values

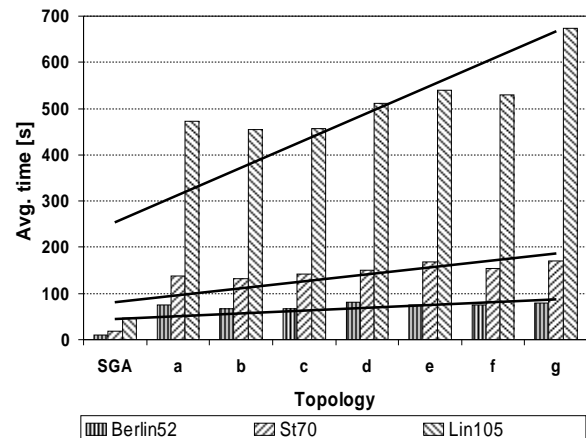


Figure 5. Average convergence time as function of topology.

The performance of the PGA is shown in Figure 6. Vertical bars represent the average results obtained relative to the optimal known value known (Table 1). It is important to notice that, in this work, we are not interested in achieving optimal results, but investigate the effect of the topology in the behavior of the island PGA. Therefore, no effort was done to fine-tune parameters of using special genetic operators. Tendency lines were not added for the sake of clarity of the plots.

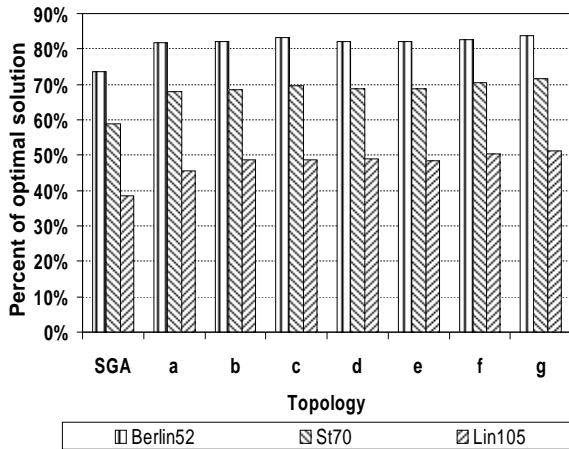


Figure 6. Average results as function of topology.

V. DISCUSSION AND CONCLUSION

Figure 5 shows that the SGA converged very quickly. Certainly, this is due to the loss of genetic diversity. It can be observed that all the remaining PGAs took much more time to reach convergence. This remarkable feature of PGAs is a direct consequence of the way islands explore the search space and stimulates genetic diversity for more time. Recall that different values of crossover and mutation rates were used for each island. The tendency lines for all experiments show that as the number of connections of the topology increase, so increase the convergence time.

As a consequence of the fast convergence of the SGA, the quality of its results were quite below those of the PGAs, as shown in Figure 6. Although not directly evident from the plots, the effect of coevolution in PGAs can be inferred: the improvement in quality of solutions in one population will induce improvements in other populations and so on. Figure 6 shows a small, but consistent, increase in quality of solutions as the connectivity increases. Again, we emphasize that the focus of the work is not on obtaining optimal solutions.

As expected, as the number of cities of the instance increases, the time needed to process each generation increases too. Taking Berlin52 as reference, St70 and Lin105 needed approximately 50% and 160% more time to process a generation, respectively. This relationship did not change as

the number of connections in the topology change. However, the time to process each generation did increase according to the topology. In the worst case (from topology 4a to 4g) there was an increase of 22% in time. This was due to the time needed to process the incoming migrants. In general, the analysis of results, for the sake of this work, were more consistent using larger instances. Therefore, future work will use instances such as Lin105 or larger.

The efficiency of PGAs is dependent on several running parameters, as well as on the migration policy. Other works in the literature focused some of these aspects, while the present work focused only the topology and connectivity of the PGA.

Appart from the obvious conclusions that PGAs is always more efficient than the same SGA version, and that PGAs can maintain genetic diversity for longer time than SGA, results of this work suggest that as the connectivity of the PGA increases, a small, but consistent increase in quality of solutions will take place.

Acknowledgment

This work was partially supported by a grant from the Brazilian National Research Council – CNPq to H.S.Lopes, no. 309262/2007-0.

REFERENCES

- [1] Applegate, D.L., Bixby, R.E., Chvatal, V. and Cook, W.J. The Traveling Salesman Problem - A Computational Study, Princeton University Press, Princeton, USA, 2006.
- [2] Cantú-Paz, E. Efficient and Accurate Genetic Algorithms. Springer, New York, USA, 2001.
- [3] Goldberg, D.E. Genetic Algorithms in Search, Optimization and Machine Learning. Reading: Addison-Wesley, 1989.
- [4] Lopes, H.S. and Coelho, L.S. Particle swarm optimization with fast local search for the blind travelling salesman problem. In Proceedings of 5th International Conference on Hybrid Intelligent Systems (HIS'2005) (Rio de Janeiro, Brazil, November 6-9). IEEE Press, 2005, 245-250.
- [5] Reinelt, G. TSPLIB – A traveling salesman problem library. ORSA Journal on Computing, 3, 1991, 375-384.
- [6] Skolicki, Z. and De Jong, K. The influence of migration sizes and intervals on island models. In Proceedings of Genetic and Evolutionary Computation Conference (GECCO'2005) (Washington, USA, June 25-29). ACM Press, New York, 2005, 1295-1302.
- [7] H-K. Tsai, J-M. Yang, Y. F. Tsai and C-Y. Kao. An evolutionary algorithm for large traveling salesman problems. IEEE Trans. Sys.Man Cybernetics, Part B: Cybernetics, 34(4), 2004, 1718-1729.
- [7] Whitley, D., Rana, S. and Heckendorn, R. The island model genetic algorithm: on separability, population size and convergence. Journal of Computing and Information Technology, 7, 1999, 33-47.