

# Study of Migration Topology in Island Model Parallel Hybrid-GA for Large Scale Quadratic Assignment Problems

J Tang, M H Lim, Y S Ong and M J Er

Intelligent Systems Centre,  
Nanyang Technological University,  
Research TechnoPlaza, 7th Storey, BorderX Block,  
Singapore 637553  
{pg04159923, emhlim, asysong, emjer}@ntu.edu.sg

## Abstract

This paper extends our previous work [1] on the island model parallel hybrid-genetic algorithm (PHGA) for large scale Quadratic Assignment Problems (QAPs). Some issues on the control parameters of the migration process and how they affect the quality of the solutions and the efficiency of algorithm deserve further evaluative study. In this paper, we investigate the effect of migration topology on the performance of the PHGA. Two topologies, one-way ring topology and random topology, are studied and analyzed. The empirical results show that the PHGA with ring topology is better able to achieve an appropriate tradeoff between exploration and exploitation and hence more helpful to improve the performance of PHGA for solving large scale QAPs.

## 1. Introduction

Among the many classes of combinatorial optimization problems, the quadratic assignment problems (QAPs) are among the hardest with many interesting practical applications. It was formulated by Koopmans and Beckmann [2] for location planning of economic activities. To formulate a QAP mathematically, consider  $n$  facilities to be assigned to  $n$  locations with minimum cost. The QAP can be described by two  $n \times n$  matrices  $A = [a_{ij}]$  and  $B = [b_{ij}]$ . The goal is to find a permutation  $\pi$  of the set  $M = \{1, 2, 3, \dots, n\}$ , which minimizes the objective function  $C(\pi)$  as in Eq.(1).

$$C(\pi) = \sum_{l=1}^n \sum_{t=1}^n a_{lt} b_{\pi(l)\pi(t)} \quad (1)$$

In the above equation, matrix  $A$  can be interpreted as a distance matrix, i.e.  $a_{ij}$  denotes the distance between location  $i$  and location  $j$ , and  $B$  is referred to as the flow matrix, i.e.  $b_{ij}$  represents the flow of materials from facility  $i$  to facility  $j$ . We represent an assignment by the vector  $\pi$ .  $\pi(i)$  is the location to which facility  $i$  is assigned.

Since QAPs are NP-hard problems, only implicit enumeration approaches are known to solve them optimally. However, the large scale problems ( $n > 20$ ) are usually intractable due to the poor scalability of the enumeration methods. From literature survey, many heuristic approaches have played an important role in algorithms capable of providing good solutions within tractable computational time, such as greedy randomized search [3], genetic algorithms (GA) [4, 5] and so on.

Although GA can solve QAPs quite efficiently, the computational cost rises largely with the increase of problem size. As parallel computers become more commonplace in scientific computing, it becomes more feasible to harness their power for use with genetic algorithms. Our previous work [1] has conducted island model parallel hybrid-GA in a distributed computing environment for large scale QAPs with significant success.

In the island model parallel GA, four control parameters pertaining to the migration operation, which have a significant effect on performance are migration interval, migration rate, migration policy and migration topology. This paper extends our previous work [1] on the PHGA to empirically investigate the impact of migration topology of the island model on the performance when solving large scale QAPs. We compare the two different topologies, ring topology and random topology in terms of both solution quality and computational time. Considering the powerful computing resource provided conveniently in a distributed computing environment, a grid-enabled solver is used to facilitate the implementation of the PHGA, as our previous work [1].

This paper is organized as follows. Section 2 provides an introduction to PGA, giving a brief overview of the different forms of parallelism used in GAs and of the island model in particular. Section 3 focuses on some different control parameters and strategies used in island model parallel GA, especially the various migration topology. Our empirical study of migration topology in multi-island model is provided in section 4, and we compare the simulation results between the PHGA using ring topology and random topology, additionally giving the analysis in terms of

solution quality and computational time. Section 5 concludes the paper with some directions for future work.

## 2. Parallel Genetic Algorithm

Genetic algorithm (GA) or parallel genetic algorithm (PGA) is a popular heuristic algorithm, widely used to solve complex combinational optimization problems. PGA is more prominent because of its distributed and flexible features. It is also easy to implement and has great potential for substantial improvement in search performance. There are a variety of models for parallelizing GAs in the literature. They have been separated in three main categories [6]: global single-population master-slave GAs, single-population fine-grained GAs, and multiple-population coarse-grained GAs.

In a master-slave GA, there exists a single panmictic population (just as in a simple GA), but the evaluation of fitness is distributed among several processors. In fine-grained parallel GAs, selection and mating are restricted to a small neighborhood, but neighborhoods overlap permitting some interactions among all the individuals.

Multiple-population (or multiple-deme) GAs, also called island model PGA, are more sophisticated, as they consist of several subpopulations which exchange individuals occasionally. This exchange of individuals is called migration and it is controlled by the following parameters.

- Migration rate determines how many individuals migrate from a population;
- Migration frequency (migration interval) determines how often migration occurs;
- Migration topology determines the destination of the migrants;
- Migration policy determines which individuals migrate and which are replaced at the receiving deme.

Migration is a key feature of the island model. Since the control parameters on migration process affect the quality of the solutions and the efficiency of algorithm, the users should determine these parameters carefully. Among these control parameters, some issues deserved deeper consideration, such as migration topology, which is studied in the paper.

## 3. Migration Topology in Island Model Parallel GA

In our previous work [1], a multi-island PHGA showing the inter-islands relationship is depicted in Figure 3-1.

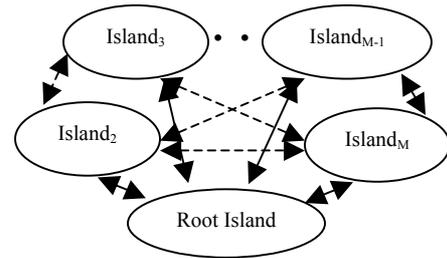


Figure 3-1 Inter-islands relationships of multi-island PHGA model

In the general multi-island model, all islands are identical except for the root island (RI) which holds the additional administrative duties to facilitate the migration of individuals across the different subpopulations. In Figure 1, the dotted line represents the migration of fit individuals between islands. The continuous line indicates that migration of individuals achieved through the RI, which then forwards these individuals to the respective subpopulations. By doing this, we attempt to facilitate a greater bias towards population diversity. This offers opportunities to explore a wider scope of the solution landscape, reducing the tendency for local minima attraction. With this framework, it is also easy to develop larger scale applications by incorporating more islands.

In the PHGA, the migration topology is decisive for the efficiency and the reliability [7]. If the topology has a dense connectivity (or a short diameter, or both) good solutions will spread fast to all the demes and may quickly take over the population. On the other hand, if the topology is sparsely connected (or has a long diameter), solutions will spread slower and the demes will be more isolated from each other, permitting the appearance of different solutions. These solutions may come together at a later time and recombine to form potentially better individuals.

Concerning this parameter, the island model PGA may be categorized as following either the *island model* (fully-connected demes) or the *stepping-stone model* (interaction restricted by customized logical or physical neighborhood). The choice of one arrangement against the other surely should observe the tradeoff “time spent in computation versus time spent in communication” [8]. As well, the deme connections may be either *static* (fixed *a priori*) or *dynamic* (reconfigured along the run).

Much research work in the literature has focused on this issue. Cantu-Paz [9, 10] theoretically investigated how the degree of the connectivity graph affects the chance that the desired solution is reached after two epochs. Cantu-Paz [11] considered that topology with different numbers of neighbors affects the distribution of fitness, and found that increasing the number of neighbors results in a higher selection intensity, and suggests that migration preserves more diversity as the number of neighbors increases.

In this paper, we study the performance of the PHGA with one-way ring topology and random topology, respectively. The ring topology is a loosely connected topology, where the communication is

limited to occur between the adjacent nodes, as depicted in Figure 3-2. The elitist emigrants (individuals) are sent

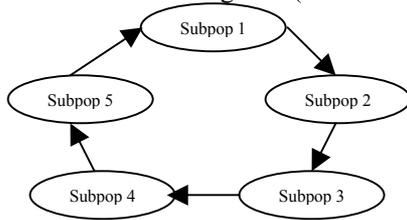


Figure 3-2 One-way ring topology

to the nearest neighbor in the ring. Then in the receiving subpopulations, the immigrants replace the least fit individuals.

The random topology used in our work is a dynamic stepping stone scheme. Each recipient subpopulation receives the immigrants from certain donor subpopulation chosen at random for each migration phrase, such that each subpopulation received migrants from a single subpopulation other than itself.

#### 4. Empirical Study

For convenience, the PHGA with one-way ring topology is denoted as PHGA-ring, while PHGA-rand indicates the PHGA with random topology. In this section, we present the empirical study of the PHGA-ring and PHGA-rand on three large scale QAP benchmarks. In particular, we evaluate the PHGA-ring and PHGA-rand both in terms of computation time and solution quality. The comparison in terms of scalability for both algorithms is investigated as well.

The algorithms have been coded in C programming language and the simulations were carried out on a cluster of Pentium IV 1.9 GHz workstations. Each computing node is equipped with 256MB of RAM, running on Linux Redhat 7.0 operating system. For each QAP benchmark problem, we carried out 10 optimization runs and the algorithms were evaluated based on their average performance.

For our simulation, the configuration of the PHGA control parameters is summarized in Table 4-1. As for the elite size, we set the elite size of serial algorithm to

be 4, therefore when  $M=2$ , the elites for each subpopulation is 2, and when  $M$  is more than 3, the elites for each subpopulation is kept as 1.

GA parameters	Two-island PHGA
Population size	240
Subpopulation size	$240/M$
Maximum number of generations	500
Fitness scaling factor $S_f$	3
Crossover probability $P_c$	0.8
Mutation probability $P_m$	0.05
Zerofit threshold constant $K_z$	5

$M$ : number of islands (processing nodes)

Table 4-1 Parameters setting for the PHGA

Through a series of empirical study, the following control parameters of migration operation listed in Table 4-2 have been adopted in the PHGA. The search stops or terminates when either one of the following criteria is satisfied:

- i. Solution stalls for more than 70 successive generations;
- ii. Maximum number of generations has been reached.

The empirical results on the benchmarks are summarized in Tables 4-4 to 4-6. The sizes of all three problems considered in the study are relatively large. In addition, the values in the first column of Tables 4-4 to 4-6 are the best-known values of the respective benchmark problems [12]. The benchmarks where the PHGA-ring outperforms the PHGA-rand, in terms of solution quality are highlighted in bold. Several criteria have been defined to measure the performance. These are listed in Table 4-3. Among these criteria, *CPU time* is used to measure the computational cost of the algorithms in wall-clock time. *Generation* and *TG* provide a measure on the convergence rate of the algorithms in terms of the number of iterations rather than the wall-clock time. *Average*, *Average gap*, *Best*, *Gap* and *Success rate* serve as the criteria for measuring the solution quality of the algorithms.

Control Parameter	Description
Migration Interval	Migration occurs every 10 generations.
Migration Rate	One chromosome per migration phase.
Migration Policy	The simple elitist strategy, where the best individual in one subpopulation replaces the worst in the other.
Migration Topology	One-way ring topology for PHGA-ring; random topology for PHGA-rand.

Table 4-2 Migration control parameters of PHGA

Criterion	Definition
<i>CPU time</i>	Average computation time in seconds upon termination of the algorithm.
<i>Generation</i>	Average number of generations elapsed before the occurrence of the best solution.
<i>TG</i>	Average number of generations elapsed before the algorithm terminates.
<i>Average</i>	Average objective value of the solutions obtained for all the GA runs.
<i>Average gap</i>	Difference between the <i>Average</i> and the best-known value of the objective function. $AverageGap = (Average - bk) / (bk) * 100\%$ where <i>bk</i> is the best-known value of the objective function.
<i>Best</i>	Best solution obtained among all the GA runs.
<i>Gap</i>	Difference between the best-found value and the best-known value of a benchmark problem. $gap = (bf - bk) / (bk) * 100\%$ where <i>bf</i> is the best- found value of the objective function.
<i>Success rate</i>	Number of times the algorithm finds the best-known solution out of all the GA runs.

Table 4-3 Criteria for measuring performance shown in Table 4-4 to 4-6

			CPU time	Generation	TG	Average	Average gap	Best	Gap	Success rate
sko100b 153890	3-island	PHGA-ring	188.80	185.60	255.60	154441.00	<b>0.36%</b>	153960	<b>0.05%</b>	0.00%
		PHGA-rand	156.80	103.80	173.80	154468.20	0.38%	154126	0.15%	0.00%
	4-island	PHGA-ring	174.50	282.50	352.50	154213.80	<b>0.21%</b>	153952	<b>0.04%</b>	0.00%
		PHGA-rand	145.40	206.50	276.50	154273.00	0.25%	154104	0.14%	0.00%
	6-island	PHGA-ring	148.80	213.30	283.30	154254.60	<b>0.24%</b>	154074	<b>0.12%</b>	0.00%
		PHGA-rand	139.10	184.80	248.00	154307.80	0.27%	154104	0.14%	0.00%
	8-island	PHGA-ring	136.90	173.20	243.20	154295.60	0.26%	153910	<b>0.01%</b>	0.00%
		PHGA-rand	169.00	263.40	333.40	154257.00	0.24%	154078	0.12%	0.00%
	10-island	PHGA-ring	119.60	150.80	220.80	154195.80	<b>0.20%</b>	153936	<b>0.03%</b>	0.00%
		PHGA-rand	124.20	180.80	250.80	154322.80	0.28%	154090	0.13%	0.00%

Table 4-4 Testing results of *sko100b* benchmark

			CPU time	Generation	TG	Average	Average gap	Best	Gap	Success rate
tai100b 1.186E+09	3-island	PHGA-ring	191.70	179.00	249.00	1189426579.70	0.29%	1187378490	0.12%	0.00%
		PHGA-rand	130.60	85.30	155.30	1189303083.40	0.28%	1186934079	0.08%	0.00%
	4-island	PHGA-ring	178.10	268.80	332.00	1187539521.00	<b>0.13%</b>	1186007112	<b>0.00%</b>	0.00%
		PHGA-rand	117.50	144.70	214.70	1187950822.00	0.17%	1186301510	0.03%	0.00%
	6-island	PHGA-ring	160.10	233.70	296.70	1187892570.00	<b>0.16%</b>	1185996137	0.00%	10.00%
		PHGA-rand	154.80	212.00	275.00	1187949722.00	0.17%	1185996137	0.00%	10.00%
	8-island	PHGA-ring	153.50	261.10	318.30	1187905557.00	<b>0.16%</b>	1185996137	0.00%	10.00%
		PHGA-rand	132.00	179.20	242.20	1188106218.00	0.18%	1185996137	0.00%	10.00%
	10-island	PHGA-ring	148.00	250.00	320.00	1187927883.00	<b>0.16%</b>	1186052259	<b>0.00%</b>	0.00%
		PHGA-rand	129.20	201.20	270.60	1188552881.00	0.22%	1186710060	0.06%	0.00%

Table 4-5 Testing results of *tai100b* benchmark

			CPU time	Generation	TG	Average	Average gap	Best	Gap	Success rate
tho150 8133864	3-island	PHGA-ring	1168.40	180.20	250.20	8166026.60	<b>0.40%</b>	8148836	<b>0.18%</b>	0.00%
		PHGA-rand	990.30	149.60	219.60	8166282.40	0.40%	8159792	0.32%	0.00%
	4-island	PHGA-ring	935.10	317.70	376.80	8162408.00	<b>0.35%</b>	8145990	<b>0.15%</b>	0.00%
		PHGA-rand	640.70	139.40	209.40	8171126.00	0.46%	8158848	0.31%	0.00%
	6-island	PHGA-ring	885.30	332.50	386.00	8157363.67	<b>0.29%</b>	8151408	0.22%	0.00%
		PHGA-rand	971.20	374.20	429.90	8164029.00	0.37%	8151152	0.21%	0.00%
	8-island	PHGA-ring	827.70	406.10	457.60	8161512.55	<b>0.34%</b>	8151436	<b>0.22%</b>	0.00%
		PHGA-rand	664.20	278.80	340.60	8168117.40	0.42%	8153310	0.24%	0.00%
	10-island	PHGA-ring	605.10	418.50	462.40	8165464.60	<b>0.39%</b>	8154998	<b>0.26%</b>	0.00%
		PHGA-rand	529.70	296.00	361.60	8171063.40	0.46%	8163262	0.36%	0.00%

Table 4-6 Testing results of *tho150* benchmark

#### 4.1 Analysis

From Tables 4-4 to 4-6, we can observe that the solution quality found by PHGA-ring is usually better than the

PHGA-rand. This is especially obvious as the number of islands is increased. Evaluation is based on the *Average*, *Average gap*, *Best* and *Gap* criteria on all the three large scale QAPs considered. As for the computational time

used in both migration topology schemes, it is noted that the PHGA-ring usually took longer than the PHGA-rand. This indicates that the PHGA-ring is able to carry on with the search longer than the PHGA-rand because it is able to maintain genetic diversity across the subpopulations more effectively and hence avoiding the tendency for premature convergence.

From the results, one may conclude that the island model is well suited for managing diversification and intensification during the search process. Each deme might be seen as intensification in a particular region and a great number of demes provide some diversification in the global space. From this point of view, migration represents a tradeoff between exploration of new individuals and exploitation of highly fit individuals which have already been found. The logical relationship between subpopulations imposed by the topology of the island model PGA has an effect on this tradeoff as well. The ring topology used for the PHGA described in this paper ensures local communications between subpopulations. The benefit of this design is that migration occurs locally between adjacent subpopulations on the ring. This yields local exploitation of fit individuals, while globally the separate subpopulations are free to explore different types of strings independently. So this loosely connected topology help to overcome local optimum and improve the performance to some extent.

## 4.2 Analysis on Scalability

Observing the trend of computational time of the PHGA-ring and PHGA-rand, we are motivated to further investigate the scalability of the two algorithms. Figures 4-1 to 4-3 present the empirical results of scalability on the three large benchmark problems.

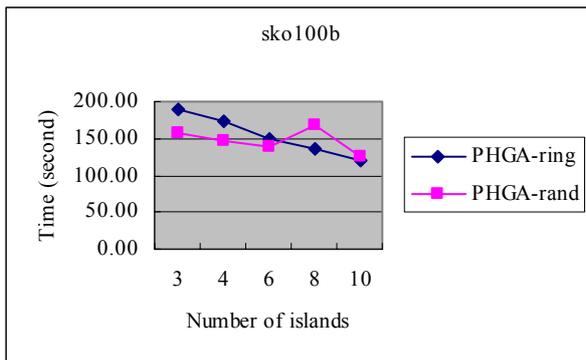


Figure 4-1 Total CPU time of *sko100b* benchmark

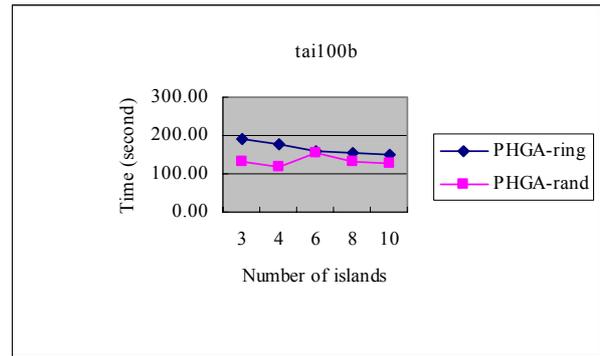


Figure 4-2 Total CPU time of *tai100b* benchmark

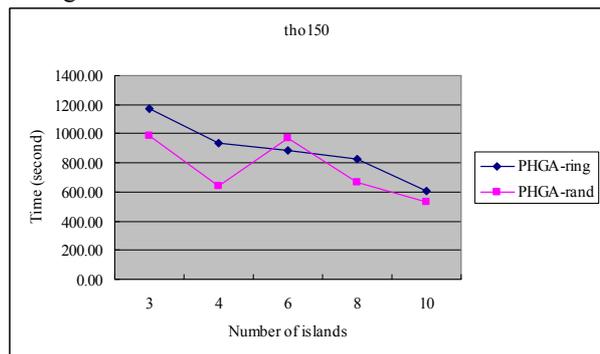


Figure 4-3 Total CPU time of *tho150* benchmark

In Figures 4-1 to 4-3, the results show that for the large scale benchmark problems considered, the total CPU time of PHGA-ring reduces accordingly along with the increase in the number of computing nodes. However, the decrease of CPU time of the PHGA-rand does not follow an obvious degressive trend when the number of islands increases. It indicates that for large scale QAPs, the scalability of the PHGA-ring is much better than the PHGA-rand due to more appropriate tradeoff between exploration of new individuals and exploitation of highly fit individuals found. This again demonstrates that the ring topology is more helpful for improving the performance of PHGA in solving large scale QAPs.

## 5. Conclusion and Future Work

The work presented in this paper is undertaken to extend our previous work on PHGA for large scale QAPs. We investigate the effect of varying the migration topology on the performance of the PHGA. Two topologies, one-way ring topology and random topology are studied. The empirical results of PHGA-ring and PHGA-rand are evaluated both in terms of computation time and solution quality. The comparison in terms of scalability for both algorithms is investigated as well. It shows that the solution quality found by PHGA-ring is generally better than the PHGA-rand. The PHGA-ring is more effective in maintaining genetic diversity to prevent the premature convergence. This is especially obvious with the increase in the number of

islands. In addition, the scalability of the PHGA-ring is much better than the PHGA-rand.

From the results, it can be concluded that the PHGA with ring topology maintains a more appropriate tradeoff between exploration and exploitation and hence more helpful in improving the performance of PHGA for large scale QAPs. The obtained results have opened up several issues for future research. In particular, the issue on the effect of other control parameters of migration progress on the performance of island model PGA need to be explored to achieve more optimal parameters setup for the migration operation.

## Acknowledgement

The authors acknowledge the funding support of Singapore Technologies Engineering Pte Ltd in this work.

## References

- [1] J. Tang, M.H. Lim and Y.S. Ong, "A Parallel Hybrid GA for Combinatorial Optimization Using Grid Technology," in *Congress on Evolutionary Computation (CEC 2003)*, December 8-12, Canberra, Australia, 2003.
- [2] T.C. Koopmans and M.J. Beckmann, *Assignment problems and the location of economic activities*. *Econometrica* vol, 25, pp. 53-76, 1957.
- [3] Y. Li, P.M. Pardalos, and M.G.C. Resende. "A greedy randomized adaptive search procedure for the quadratic assignment problem," in *Quadratic Assignment and Related Problems*. P. Pardalos and H. Wolkowicz (Eds.), AMS: Providence, Rhode Island, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 16, pp. 173-187, 1994.
- [4] M.H. Lim, Y. Yuan, and S. Omatu, "Efficient Genetic Algorithms Using Simple Genes Exchange Local Search Policy for the Quadratic Assignment Problem," *Computational Optimization and Applications*, vol. 15, pp. 249-268, 2000.
- [5] M.H. Lim, Y. Yuan, and S. Omatu, "Extensive Testing of a Hybrid Genetic Algorithm for Quadratic Assignment Problem," *Computational Optimization and Applications*, vol. 23, pp. 47-64, 2002.
- [6] E. Cantu-Paz, "A Survey of Parallel Genetic Algorithms," *Calculateurs Paralleles, Reseaux et Systems Repartis*, vol. 10, no. 2, pp. 141-171, 1998.
- [7] E. Cantu-Paz, D.E. Goldberg, "Efficient Parallel Genetic Algorithms: Theory and Practice," *Computer Methods in Applied Mechanics and Engineering*, no.186(2-4), pp. 221-238, 2000.
- [8] E. Cantu-Paz. *Efficient and Accurate Parallel Genetic Algorithms*, Kluwer Academic Publisher, 2000.
- [9] E. Cantu-Paz, D.E. Goldberg, "On the Scalability of Parallel Genetic Algorithms," *Evolutionary Computation*. vol. 7, no. 4, pp. 429-449, 1999.
- [10] E. Cantu-Paz, "Topologies, Migration Rates, and Multi-Population Parallel Genetic Algorithms," *IlliGAL Report No. 99007*, 1999.
- [11] E. Cantu-Paz, "On the Effects of Migration on the Fitness Distribution of Parallel Evolutionary Algorithms," in *Workshop on Evolutionary Computation and Parallel Processing at GECCO-2000*, pp. 3-6. Las Vegas, 2000.
- [12] R.E. Burkard, S.E. Karisch and F. Rendl, "QAPLIB—A quadratic assignment problem library", *Journal of Global Optimization*, vol. 10, pp. 391-403, 1997. [URL: <http://www.opt.math.tu-graz.ac.at/qaplib/>]