

Solving Large Scale Combinatorial Optimization Using PMA-SLS

Jing Tang , Meng Hiot Lim, Yew Soon Ong, Meng Joo Er
Intelligent Systems Centre,
Nanyang Technological University,
Research TechnoPlaza, 7th Storey, BorderX Block,
Singapore 637553
{pg04159923, emhlim, asysong, emjer}@ntu.edu.sg

ABSTRACT

Memetic algorithms have become to gain increasingly important for solving large scale combinatorial optimization problems. Typically, the extent of the application of local searches in canonical memetic algorithm is based on the principle of “more is better”. In the same spirit, the island model parallel memetic algorithm (PMA) is an important extension of the canonical memetic algorithm which applies local searches to every transitional solutions being considered. For PMA which applies complete local search, we termed it as PMA-CLS. In this paper, we consider the island model PMA with selective application of local search (PMA-SLS) and demonstrate its utility in solving complex combinatorial optimization problems, in particular large-scale quadratic assignment problems (QAPs). Based on our empirical results, the PMA-SLS compared to the PMA-CLS, can reduce the computational time spent significantly with little or no loss of solution quality. This we concluded is due mainly to the ability of the PMA-SLS to manage a more desirable diversity profile as the search progresses.

Categories and Subject Descriptors

I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search – *Heuristic methods*; G.1.6 [Mathematics of Computing]: Optimization – *Global optimization*

General Terms

Algorithms, Performance, Experimentation

Keywords

combinatorial optimization, quadratic assignment problem, island model parallel memetic algorithm, selective local search

1. INTRODUCTION

In recent years, the role of local search in the context of genetic algorithms and the wider field of evolutionary computing has

been discussed. Mascato and Norman [12] promoted the term “memetic algorithm” to describe genetic algorithm that relies heavily on local search. While canonical memetic algorithm with complete local search helps improve the convergence rate significantly, poor diversity often results at later stage of search evolution due to excessive focus on local search. Intensive local search also incurs huge computation time. On the other hand, while using modest local search in PMA helps maintain good diversity of solutions throughout the search, the quality of solution precision is often compromised. In this paper, we consider the island model PMA with selective application of local search (PMA-SLS) and demonstrate its utility in solving complex combinatorial optimization problems, in particular large-scale quadratic assignment problems (QAPs).

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 [7] 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 π 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 . 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 π . $\pi(i)$ is the location to which facility i is assigned.

Since QAPs are NP-hard problems, only explicit 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 genetic algorithms (GA) [10, 11], memetic algorithms [13, 14].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'05, June 25–29, 2005, Washington, DC, USA.
Copyright 2005 ACM 1-59593-010-8/05/0006...\$5.00.

Two distinct features of the PMA-SLS to be noted in our study are the sampling size and the local search frequency. For the sampling size, we adopted a normal distribution scheme of determining the sampling ratio. As for the local search frequency, we imposed a complete local search at the start of evolution with a progressively decreasing frequency. In addition, parallel EAs have been shown in many studies to produce good quality solutions even on highly complex problems [17, 18]. Hence it is important that the intrinsic parallelism of EA is adopted in our work. A grid-enabled solver is also used to facilitate the implementation of our parallel MA, such that islands of MA individuals are executed across powerful computing resources within a distributed computing environment. Based on our empirical results, the PMA-SLS compared to the PMA-CLS, generally converged to better solution quality with a much lower computation time. This we concluded is due mainly to the ability of the PMA-SLS to manage a more desirable diversity profile as the search progresses.

This paper is organized as follows. Section 2 provides a brief overview of the memetic algorithm and parallel memetic algorithm. Section 3 focuses on the island model parallel MA in detail and a selective local search strategy is also introduced in the section. Section 4 presents the empirical results for selective local search based multi-island model parallel memetic algorithm (PMA-SLS), with comparisons to our previous work (PHGA) [10, 11], complete parallel memetic algorithm, (PMA-CLS). Analyses of the results obtained in terms of solution quality, computational time, solution precision, and scalability are also presented in the section. Finally, section 5 concludes the paper with some directions for future work.

2. MEMETIC ALGORITHMS AND PARALLEL MEMETIC ALGORITHMS

2.1 Memetic Algorithm

Memetic algorithm is a marriage between a population-based global search and the local improvement made by each of the individuals. Some memetic algorithms have been applied quite successfully in a wide range of combinatorial and continuous optimization problems [6, 9, 13, 14, 15]. In particular, large instances of many well-known combinatorial optimization problems have been solved to optimality where other meta-heuristics have failed to produce comparable results [13]. In standard memetic algorithms, a complete local search is performed on every individual in the MA's population. Goldberg and Voessner [4] presented a theoretical framework for discussing the balance between genetic search and local search. Hart [6] investigated several issues especially related to local search for designing efficient memetic algorithms for continuous optimization. Land [9] extended Hart's analysis to the combinatorial optimization domain where the balance between genetic search and local search was referred to as the local/global ratio. The balance can be also adjusted by the use of different neighbourhood structures. Krasnogor [8] investigated how to change the size and the type of neighbourhood structures dynamically in the framework of multimeme memetic algorithms where each meme had a different neighbourhood structure, a different acceptance rule and a different number of iterations of local search. Ong et al [15] also considered using multiple local methods or memes during a memetic algorithm search in the spirit

of Lamarckian learning on continuous optimization problems. In summary, most of the researchers in the literature address the following issues pertinent to memetic algorithms:

- (1) How often should local search be applied?
- (2) On which solutions should local search be used?
- (3) How long should local search be run?
- (4) How efficient does local search need to be?
- (5) What local search method or meme should be used for a particular problem or individual?

In this paper, we present a study on applying local search to solutions in the PMA population selectively so as to improve the efficiency of memetic algorithm. It is worth nothing that the present work represents a study relating to questions (1) and (2) listed above in the context of solving large scale QAPs.

2.2 Parallel Memetic Algorithm

Our previous work [17, 18] has shown that the success of the island model parallel hybrid GA. Similar to parallel GAs, parallel memetic algorithm 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 memetic algorithm in the literature, such as employing a blackboard mechanism in a parallel asynchronous memetic algorithm proposed [1], and the master/slave algorithm [3]. Here, we consider using an island model parallel memetic algorithm in our work.

3. ISLAND MODEL PARALLEL MEMETIC ALGORITHM

3.1 Algorithm Structure

As in our previous work [17, 18], the PMA studied in this paper still adopted the multi-island model. In the general multi-island model, all islands are identical except for the root island (RI) which holds additional administrative duties to effect the migration of individuals across the different 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. In our work, we utilize the island model to implement the PMA using selective local search strategy. The general template of the parallel memetic algorithm used in each subpopulation is shown in Figure 3-1. The local search procedure implemented in the memetic algorithm is a form of k -gene exchange [10, 11, 17, 18].

3.2 Selective Local Search Strategy

While the canonical memetic algorithm with complete local search has often shown great success on increasing search convergence rate, poor diversity may result due to excessive local search. On the other hand, while using modest local search in PMA helps maintain good diversity of solutions throughout the search, the quality of solution precision is often compromised [10, 11]. Here, we investigate a selective local search strategy in our multi-island PMA. We consider forcing a larger intensity of local search in the earlier stage than later stages of evolutionary search and apply local search with adaptively decreasing frequency, f , as the search progresses. Here, we model the intensity of local search using a Gaussian distribution defined as equation (2).

BEGIN

Initialize: Generate an initial MA subpopulation.

While (Stopping conditions are not satisfied)

Evaluation of all individuals in the subpopulation

For each individuals in the population

- Selectively apply local search to the individuals in the subpopulation
- Proceed with local improvement and replace the genotype in the subpopulation with the improved solution.

End For

Apply standard GA operators to create a new population; i.e., Selection, Mutation and crossover.

End While

END

Figure 3-1 The pseudo code for memetic algorithm with selective local search strategy

$$\gamma(\text{gen}; \mu, \sigma, \eta) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \exp\left(-\frac{1}{2} \left(\frac{\text{gen} - \mu}{\sigma}\right)^2\right) * \eta \quad (2)$$

where *gen* is the evolution generation ($\text{gen} \geq 0$), μ is the mean value, σ is the standard deviation, η represents a scaling factor on the number of chromosomes to apply local search. Our objective is to allocate candidates for local search based on a normal distribution over time. Then the number of chromosomes to apply local search (*Num*) is defined by the Gaussian function given in equation (2), evolution generation (*gen*) and subpopulation size (*Subpopulation_size*). It is given by:

$$\text{Num} = \gamma * \text{Subpopulation_size} \quad (3)$$

where *Num* denotes the number of candidates whereby local search is applied for generation *gen*.

Motivated by the work in [6], elitism mechanism identifies the best individual(s) in a population and ensures that they exist in the next generation. Therefore, these mechanisms induce a strong bias based on the rank of individuals in the generation. Introducing elitism will reduce the local search frequency since the elitism mechanisms provide some aids in the GA's competitive search. Considering the successfully employed elitism mechanism in our previous work [10, 11, 17, 18], so in the selective local search strategy discussed above, the selective candidates are sampled randomly from the PMA's subpopulation.

4. EMPIRICAL STUDY

For convenience, serial MA employing complete local search strategy in the whole population without using any parallel mechanisms is denoted as SMA. PMA-SLS denotes the island model PMA with the selective local search strategy. PMA-CLS refers to the island model PMA with the complete local search strategy, which means applying the local search to the whole subpopulation during every generation. PHGA [*] abbreviates the island model parallel hybrid GA referred to previous work [*] for comparison purpose.

The algorithms were evaluated by averaging over 10 optimization runs. The configuration of the PMA control parameters is summarized in Table 4-1. We set the elite size of the serial algorithm to be 4, therefore when $M=2$, the elites for each

Table 4-1 Parameters setting for the PMA

MA parameters	Multi-island PMA
Population size	240
Subpopulation size	240/ <i>M</i>
Maximum number of generations	180
Fitness scaling factor <i>Sf</i>	3
Crossover probability <i>Pc</i>	0.8
Mutation probability <i>Pm</i>	0.05
Zerofit threshold constant <i>Kz</i>	5

M: number of islands (processing nodes)

Table 4-2 Migration control parameters of PMA

Control Parameter	Description
Migration Interval	Every 10 generations
Migration Rate	One chromosome
Migration Policy	Simple elitist strategy
Migration Topology	One-way ring strategy

subpopulation is 2, and when *M* is more than 3, the elites for each subpopulation is kept at 1. In addition, based on the observation of the quick convergence speed of MA, we set up the maximum number of generations to control the termination of SMA, PMA-SLS and PMA-CLS as 180, shorter than the 500 used in previous work pertaining to PHGA [17, 18]. Based on our previous study [17, 18], the migration control parameters are set as in Table 4-2. Several criteria have been defined to measure the performance as listed in Table 4-3. The search 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

As for the parameters pertaining to the selective local search strategy of PMA-SLS, based on preliminary computational experiments, we set μ as 0, σ as 200, and η equals to 500. We use a graduated scheme to manage the periodicity of local search denoted as *f*. For $\text{gen} < 50$, we set *f* to be 10, meaning local search is carried out every 10 generations. For $\text{gen} \geq 50$, *f* is set to 30.

Table 4-3 Criteria for measuring performance shown in Table 4-7 to 4-11

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 simulation 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 simulation 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 simulation runs.

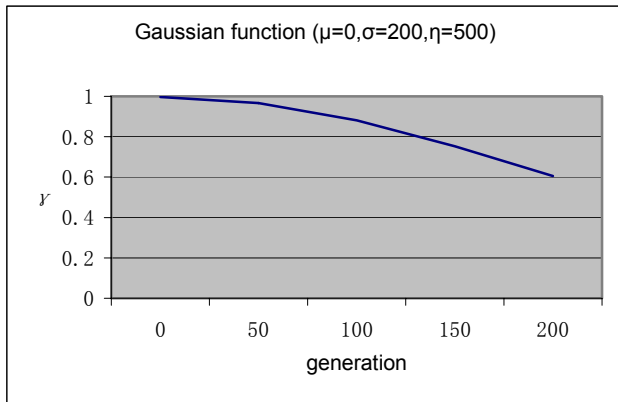


Figure 4-1 Configured Gaussian function

Since the migration interval is set as 10 generations in Table 4-2, when PMA-SLS invokes the local search according to the frequency f , the local search is carried out in every generation during the migration interval. The Gaussian function configured by the above parameters is shown in Figure 4-1.

Firstly, we demonstrate the advantage of memetic algorithm with selective local search compared to complete local search. We test both PMA-CLS and PMA-SLS on the two-island model for the same benchmark, *sko100b*. The results are shown in Table 4-4.

Particularly, we monitor closely the change in diversity of each subpopulation for PMA-SLS and PMA-CLS by measuring the entropy of each subpopulation shown in Figure 4-2. Consider Q to be mutually exclusive subsets $S_{i1}, S_{i2}, \dots, S_{iQ}$ in the t^{th} population. The number of individuals in each subset is $|S_{i1}|, |S_{i2}|, \dots, |S_{iQ}|$, respectively. Then the entropy E_t of the t^{th} population can be calculated as follows [5]:

$$E_t = -\sum_{j=1}^Q p_j \log(p_j) \quad (4)$$

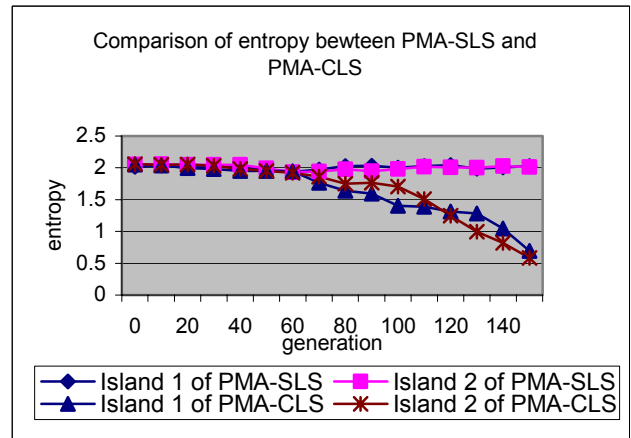


Figure 4-2 Comparison of entropy between PMA-SLS and PMA-CLS

where $p_j = \frac{|S_{ij}|}{N}$ with N being the population size.

From Figure 4-2, it is observed that PMA-SLS can consistently maintain a good level diversity as the evolution progresses. On the other hand, the entropy of PMA-CLS is reduced very sharply, indicating that local search has a tendency to speed up convergence significantly. From an evolutionary process point of view, PMA-CLS shows poorer diversity due to excessively localized search, especially in the later stage of evolution. This indicates a likelihood of higher ratio of duplicate solutions in the population. As a result, there is a higher level of redundancy in local searches being applied to the whole subpopulation over many generations. In addition, as shown in Table 4-4, both PMA-CLS and PMA-SLS achieved almost the same level of solution quality, but PMA-CLS usually incurs higher computational cost due to the intensive local search. Thus, we can observe that PMA-SLS can reduce the computational time spent significantly with

Table 4-4 Comparison between PMA-SLS and PMA-CLS

sko100b	2-island	CPU time	Generation	TG	Solution	Gap
153890	PMA-CLS	1461	81	151	153954	0.04%
	PMA-SLS	823	84	154	153950	0.04%

little or no lost of solution quality. This is mainly attributed to its capability to maintain a higher level of population diversity.

Tables 4-5 to 4-9 summarize the empirical results of testing on a diverse set of large scale QAP benchmarks. The benchmark problems considered in the present study are classes of synthetic problems randomly generated or created to study the robustness of algorithms for solving QAPs [2]. The values in the first column of Tables 4-5 to 4-9 are the best-known values of the respective benchmark problems. The results of PMA-SLS are highlighted in bold for ease of comparison. Tables 4-5 and 4-6 present a detailed

comparison study between PMA-SLS and results from our previous work [17, 18] on *sko100b* and *tai100b* benchmarks, respectively. Tables 4-7, 4-8 and 4-9 show the simulation results on the other three classes QAPs, namely, *tai100a*, *sko100** and *wil100*, respectively.

4.1 Comparison and Analysis

An inspection of the experimental results indicates that the island model parallel memetic algorithm with selective local search strategy can significantly improve the effectiveness and efficiency

Table 4-5 Results of testing on *sko100b* benchmark

		CPU time	Generation	TG	Average	Average gap	Best	Gap	Success rate
sko100b	SMA	3096.50	127.30	160.50	153955.60	0.04%	153890	0.00%	20.00%
153890	2-island PMA-SLS	875.20	113.60	168.40	154012.80	0.08%	153904	0.01%	0.00%
	PMA-CLS	1350.00	94.70	145.90	153950.40	0.04%	153890	0.00%	20.00%
	PHGA[17]	183.60	171.80	241.80	154215.00	0.21%	153924	0.02%	0.00%
	3-island PMA-SLS	1084.80	136.40	173.90	153975.40	0.06%	153934	0.03%	0.00%
	PMA-CLS	1045.30	111.30	158.40	153936.20	0.03%	153890	0.00%	20.00%
	PHGA[18]	188.80	185.60	255.60	154441.00	0.36%	153960	0.05%	0.00%
	4-island PMA-SLS	898.00	137.10	178.10	153990.80	0.07%	153902	0.01%	0.00%
	PMA-CLS	1445.90	122.20	174.60	153952.20	0.04%	153898	0.01%	0.00%
	PHGA[18]	174.50	282.50	352.50	154213.80	0.21%	153952	0.04%	0.00%
	6-island PMA-SLS	429.40	130.20	168.20	153985.00	0.07%	153890	0.00%	10.00%
	PMA-CLS	694.30	104.80	154.50	153925.40	0.02%	153890	0.00%	20.00%
	PHGA[18]	148.80	213.30	283.30	154254.60	0.24%	154074	0.12%	0.00%
	8-island PMA-SLS	568.30	119.30	166.90	153942.80	0.04%	153894	0.00%	0.00%
	PMA-CLS	484.00	118.80	162.40	153937.60	0.03%	153890	0.00%	10.00%
	PHGA[18]	136.90	173.20	243.20	154295.60	0.26%	153910	0.01%	0.00%
	10-island PMA-SLS	289.30	95.20	148.80	153987.80	0.06%	153890	0.00%	10.00%
	PMA-CLS	439.00	111.20	144.40	153942.60	0.04%	153890	0.00%	30.00%
	PHGA[18]	119.60	150.80	220.80	154195.80	0.20%	153936	0.03%	0.00%

Table 4-6 Results of testing on *tai100b* benchmark

		CPU time	Generation	TG	Average	Average gap	Best	Gap	Success rate
tai100b	2-island PMA-SLS	782.40	106.70	134.00	1186275856.50	0.02%	1185996137	0.00%	40.00%
1185996137	PHGA[17]	186.90	175.30	245.30	1188882832.20	0.24%	1186007112	0.00%	0.00%
	3-island PMA-SLS	1042.40	89.50	138.50	1186201737.10	0.02%	1185996137	0.00%	20.00%
	PHGA[18]	191.70	179.00	249.00	1189426579.70	0.29%	1187378490	0.12%	0.00%
	4-island PMA-SLS	647.50	92.60	102.30	1186007361.40	0.00%	1185996137	0.00%	80.00%
	PHGA[18]	178.10	268.80	332.00	1187539521.00	0.13%	1186007112	0.00%	0.00%
	6-island PMA-SLS	356.60	88.30	104.90	1186058956.40	0.01%	1185996137	0.00%	70.00%
	PHGA[18]	160.10	233.70	296.70	1187892570.00	0.16%	1185996137	0.00%	10.00%
	8-island PMA-SLS	407.80	73.00	89.80	1186081947.80	0.01%	1185996137	0.00%	70.00%
	PHGA[18]	153.50	261.10	318.30	1187905557.00	0.16%	1185996137	0.00%	10.00%
	10-island PMA-SLS	220.70	68.70	82.70	1186053344.20	0.00%	1185996137	0.00%	80.00%
	PHGA[18]	148.00	250.00	320.00	1187927883.00	0.16%	1186052259	0.00%	0.00%

Table 4-7 Results of testing on *tai100a* benchmark

		CPU time	Generation	TG	Average	Average gap	Best	Gap	Success rate
tai100a	2-island PMA-SLS	860.00	127.20	164.60	21458262.60	1.58%	21382118	1.22%	0.00%
21125314	PHGA[17]	222.80	238.50	308.50	21464686.20	1.61%	21335594	1.00%	0.00%
	3-island PMA-SLS	1056.30	119.90	161.50	21430953.00	1.45%	21341252	1.02%	0.00%
	4-island PMA-SLS	889.60	140.20	170.90	21420954.60	1.40%	21352956	1.08%	0.00%
	6-island PMA-SLS	451.40	152.50	180.00	21373508.00	1.17%	21270370	0.69%	0.00%
	8-island PMA-SLS	582.90	146.90	178.40	21363905.00	1.13%	21342280	1.03%	0.00%
	10-island PMA-SLS	309.60	123.60	169.50	21382655.00	1.21%	21295312	0.80%	0.00%

Table 4-8 Results of testing on *sko100 benchmarks**

			CPU time	Generation	TG	Average	Average gap	Best	Gap	Success rate
sko100a 152002	2-island	PMA-SLS	883.60	133.80	175.10	152188.20	0.12%	152042	0.03%	0.00%
		PHGA[17]	194.00	203.40	273.40	152322.80	0.21%	152122	0.08%	0.00%
	3-island	PMA-SLS	1084.30	128.80	172.90	152186.80	0.12%	152056	0.04%	0.00%
	4-island	PMA-SLS	885.20	142.40	176.80	152119.00	0.08%	152058	0.04%	0.00%
	6-island	PMA-SLS	431.90	138.90	176.90	152109.40	0.07%	152067	0.04%	0.00%
	8-island	PMA-SLS	560.80	122.70	169.30	152103.00	0.06%	152036	0.02%	0.00%
	10-island	PMA-SLS	283.20	98.10	156.00	152102.80	0.06%	152042	0.03%	0.00%
sko100c 147862	2-island	PMA-SLS	939.30	121.80	168.40	147934.80	0.05%	147862	0.00%	10.00%
		PHGA[17]	184.40	205.80	275.80	148140.40	0.18%	148050	0.13%	0.00%
	3-island	PMA-SLS	1104.10	129.60	171.40	147943.00	0.05%	147868	0.00%	0.00%
	4-island	PMA-SLS	845.90	111.40	160.50	147908.20	0.03%	147862	0.00%	10.00%
	6-island	PMA-SLS	416.80	106.40	151.80	147885.60	0.02%	147862	0.00%	20.00%
	8-island	PMA-SLS	531.60	103.60	150.90	147885.80	0.01%	147862	0.00%	10.00%
	10-island	PMA-SLS	284.20	107.90	151.00	147895.40	0.02%	147862	0.00%	10.00%
sko100d 149576	2-island	PMA-SLS	883.00	111.00	166.90	149803.60	0.15%	149618	0.03%	0.00%
		PHGA[17]	232.10	259.90	327.40	150036.80	0.31%	149732	0.10%	0.00%
	3-island	PMA-SLS	1077.80	121.30	166.10	149788.00	0.14%	149692	0.08%	0.00%
	4-island	PMA-SLS	881.20	146.70	180.00	149752.00	0.12%	149630	0.04%	0.00%
	6-island	PMA-SLS	436.80	135.80	173.80	149699.40	0.08%	149578	0.00%	0.00%
	8-island	PMA-SLS	595.60	111.20	169.60	149717.40	0.10%	149640	0.04%	0.00%
	10-island	PMA-SLS	312.60	120.10	167.20	149681.60	0.07%	149584	0.01%	0.00%
sko100e 149150	2-island	PMA-SLS	845.40	121.00	166.70	149205.80	0.04%	149150	0.00%	10.00%
		PHGA[17]	235.50	252.90	322.90	149642.20	0.33%	149198	0.03%	0.00%
	3-island	PMA-SLS	1037.20	113.20	161.90	149207.40	0.04%	149166	0.01%	0.00%
	4-island	PMA-SLS	898.50	114.30	164.50	149202.60	0.04%	149150	0.00%	10.00%
	6-island	PMA-SLS	452.10	113.70	156.90	149179.20	0.02%	149150	0.00%	30.00%
	8-island	PMA-SLS	521.50	115.20	145.90	149167.00	0.01%	149150	0.00%	30.00%
	10-island	PMA-SLS	274.20	91.20	130.00	149176.40	0.02%	149150	0.00%	40.00%
sko100f 149036	2-island	PMA-SLS	888.40	104.60	153.70	149232.80	0.13%	149126	0.06%	0.00%
		PHGA[17]	206.50	214.80	284.80	149496.60	0.31%	149228	0.13%	0.00%
	3-island	PMA-SLS	1168.50	135.20	174.10	149210.80	0.12%	149136	0.07%	0.00%
	4-island	PMA-SLS	872.10	126.10	166.70	149150.40	0.08%	149036	0.00%	10.00%
	6-island	PMA-SLS	451.70	136.10	172.30	149205.40	0.11%	149078	0.03%	0.00%
	8-island	PMA-SLS	565.90	122.50	166.10	149178.00	0.10%	149058	0.01%	0.00%
	10-island	PMA-SLS	300.30	107.00	161.40	149203.40	0.11%	149114	0.05%	0.00%

Table 4-9 Results of testing on *wil100* benchmark

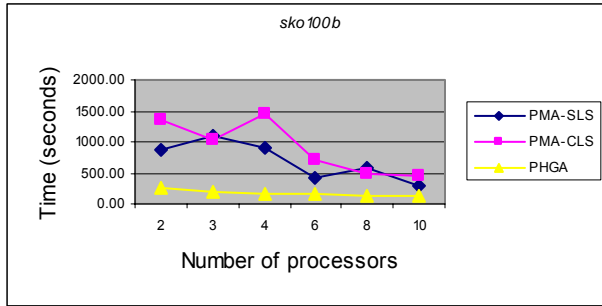
			CPU time	Generation	TG	Average	Average gap	Best	Gap	Success rate
wil100 273038	2-island	PMA-SLS	882.10	114.50	166.20	273198.80	0.06%	273054	0.01%	0.00%
		PHGA[17]	218.00	226.10	292.80	273458.20	0.15%	273236	0.07%	0.00%
	3-island	PMA-SLS	1111.20	132.50	168.40	273159.20	0.04%	273054	0.01%	0.00%
	4-island	PMA-SLS	895.20	115.20	165.60	273228.60	0.07%	273054	0.01%	0.00%
	6-island	PMA-SLS	445.00	99.30	164.30	273103.80	0.02%	273044	0.00%	0.00%
	8-island	PMA-SLS	560.10	115.90	160.30	273103.40	0.02%	273038	0.00%	10.00%
	10-island	PMA-SLS	295.20	93.90	161.20	273128.60	0.03%	273054	0.01%	0.00%

in solving large scale QAPs. The higher success rate of PMA-SLS also indicates the improved solution precision due to the higher level of diversity maintained during the evolution process for PMA-SLS.

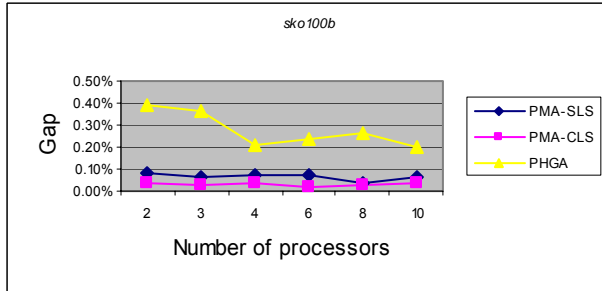
Firstly, from a solution quality point of view, both SMA and PMA achieved much better solution quality compared to PHGA. This is evident from the much improved solution gap and the higher success rate achieved. From the viewpoint of

computational time, compared to the serial MA, much shorter computational time is consumed by PMA-SLS and PMA-CLS to achieve almost the same level of solution quality, indicating the advantage of employing parallel memetic algorithms. The comparison between PMA-SLS, PMA-CLS and PHGA on *sko100b* benchmark is shown in Figure 4-3.

The plot in Figure 4-3(b) shows that PMA-SLS and PMA-CLS improve the solution quality significantly compared to PHGA. It



(a) Computation time



(b) Solution quality

Figure 4-3 Comparison between PMA-SLS, PMA-CLS and PHGA on *sko100b* benchmark

is noted that the maximum number of generations for PHGA was set at 500. Instead, the maximum number of generations for PMA-SLS and PMA-CLS was set to 180. This is indicative of the powerful search capability and quick convergence speed of the PMA. As for the computational time shown in Figure 4-3(a), the greater reliance on local search makes PMA more time-consuming than the PHGA. In this respect, the island model parallel paradigm of the memetic algorithm and the distributed computing technology can help reduce the computational time significantly. Furthermore, the selective local search strategy used in PMA-SLS improves the efficiency of the PMA remarkably. This validates the notion of PMA-SLS being able to search more efficiently than PMA-CLS to achieve comparable solution quality.

The results of testing on *tai100b* benchmark in Table 4-6 also showed that PMA-SLS can achieve much better solution quality with comparable computational time, especially in the case where the number of processor increases to 10 machines. It is also observed that the *tai100b* benchmark shows a much higher *Success rate*, indicating that the PMA-SLS has greater success in locating the global optimum. This implies that the PMA is capable of locating the best-known solution more frequently than the PHGA.

In Tables 4-7 and 4-9, based on observations of the two criteria, *Gap* and *Success Rate*, the results of the different benchmarks (*sko100** and *will00*) show that PMA-SLS can significantly improve the solution quality in comparable computational time. The results indicate that PMA-SLS is superior to the PHGA for all benchmarks studied except for one instance, namely, *tai100a*, where marginal improvement in terms of solution quality was observed. Taillard [16] mentioned that for this type of randomly generated instances, finding good solutions (about 1% and 2%) is

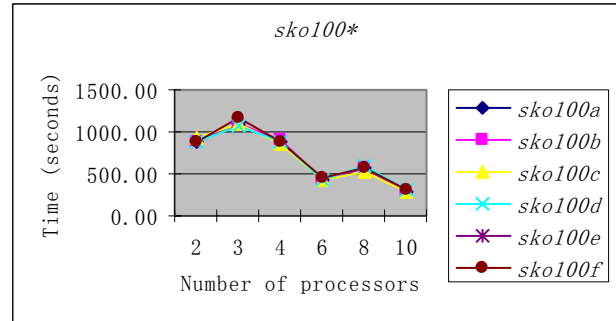


Figure 4-4 Scalability of *sko100 benchmarks by PMA-SLS**

easy, but it is extremely difficult to find the optimum.

When judged against existing results available in the literature, it is noted that the results of several instances using PMA-SLS is competitive or better than that of the MAs developed by other authors. For example, our results on the *tai100a* benchmark problem in Table 4-7 using the PMA-SLS is comparable to that found in [13]. The *Average gap* of *tai100a* was reported as 1.135%. The *Average gap* of our results on *tai100a* using 8 machines is 1.13%. Also the results of *tai100b* for PMA-SLS are much better than that shown in [13]. The *Average gap* of *tai100b* was reported as 0.026%, with the *Success rate* being less than 50%. On the other hand, *Average gap* of 0% was achieved by our PMA-SLS using 4 or 10 machines, and the *Success rate* is very commendable, being as high as 80%. Furthermore, it is worth nothing that the PMA-SLS is also capable of attaining search quality that is significantly better than that obtained in [14] on the *sko100a* problem. As shown in Table 4-8, on the *sko100a* benchmark, the *Average gap* obtained in [14] was 0.096%, while we were able to reduce this value to 0.06%.

4.2 Analysis on Scalability

Observing the trend of computational time of the PMA-SLS, we are motivated to further investigate the scalability of the PMA-SLS along with the increase in the number of islands. For example, Figures 4-4 present the empirical results of scalability on the *sko100** type benchmark problems studied above.

In Figures 4-4, the results show that the total CPU time of PMA-SLS reduces accordingly along with the increase in the number of computing nodes. This strongly demonstrates the high scalability of the island model PMA-SLS realized in a networked computing environment. However, the decrease in CPU time for the PMA-SLS does not follow a strict degressive trend when the number of islands is 3 and 8. An inspection of the figure shows that there are tubers at the points corresponding to 3 and 8 processors. Hence, one may concur that the capability of multi-island PMA-SLS could be related to the size of subpopulations. However, further studies are necessary to address this issue appropriately.

5. CONCLUSION AND FUTURE WORK

In this paper we propose and experimentally validate the island model parallel memetic algorithm with selective local search strategy for several large scale QAPs. The empirical results are evaluated both in terms of solution quality and computational time, and a comprehensive comparative study with PHGA, PMA-CLS and results of MA available in the other literature is conducted. The performance in terms of scalability of PMA-SLS

is investigated as well. According to the results, it can be concluded that the island model parallel memetic algorithm with selective local search strategy can reduce the computational time spent significantly with little or no loss of solution quality for large scale QAPs. The higher success rate of PMA-SLS also indicates the improved solution precision due to the higher level of diversity maintained during the evolutionary process.

The results obtained have opened up several issues for future research. In particular, based on the observation on the scalability of the PMA-SLS along with the increase in the number of islands, the inspection of some tubers in the scalability trend line may motivate us to further address the issue that the capability of multi-island PMA-SLS could be related to the size of subpopulations. In addition, the issue on the effect of other parameters pertaining to the selective local search strategy on PMA-SLS's performance need to be explored to achieve more optimal parameters setup for further improvement in performance.

The successful application of the PMA-SLS within a distributed computing environment in solving several large scale QAPs demonstrates its potential in solving other computationally demanding optimization problems. In our opinion, the potential of future algorithmic progress for the QAPs and other difficult optimization problems can be achieved by a marriage between the island model parallel memetic algorithm paradigm and the power that computational grids have to offer.

6. ACKNOWLEDGMENTS

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

7. REFERENCES

- [1] R. Bradwell and K. Brown. Parallel asynchronous memetic algorithms. In *Genetic and Evolutionary Computation Conference (GECCO'99), Evolutionary Computation and Parallel Processing Workshop*, Orlando, Florida, July 13-17, 1999.
- [2] R.E. Burkard, S.E. Karisch and F. Rendl. QAPLIB—A quadratic assignment problem library. *Journal of Global Optimization*, 10: 391-403, 1997. Available from <<http://www.opt.math.tu-graz.ac.at/qaplib/>>
- [3] J.G. Digalakis and K. G. Margaritis. A performance comparison of parallel genetic and memetic algorithms using MPI. *IntRep03*, University of Macedonia, Parallel Distributed Processing Laboratory, Thessaloniki, Greece, 2000.
- [4] D. Goldberg and S. Voessner. Optimizing global-local search hybrids. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*, W. Banzhaf, J. Daida, A. Eiben, M. Garzon, V. Honavar, M. Jakaiela, and R. Smith (Eds.), Morgan Kaufmann, 1999.
- [5] J.J. Grefenstette. Incorporating problem specific knowledge into genetic algorithms. In *Genetic Algorithms and Simulated Annealing*, Morgan Kaufmann Publishers, pages 42-60, 1987.
- [6] W.E. Hart. Adaptive Global Optimization with Local Search. Ph. D. Thesis, University of California, San Diego, 1994.
- [7] T.C. Koopmans and M.J. Beckmann. Assignment problems and the location of economic activities. *Econometrica*, 25: 53-76, 1957.
- [8] N. Krasnogor. Studies on the Theory and Design Space of Memetic Algorithms. Ph. D. Thesis, University of the West of England, Bristol, June 2002.
- [9] M.W.S. Land. Evolutionary Algorithms with Local Search for Combinatorial Optimization. Ph. D. Thesis, University of California, San Diego, 1998.
- [10] 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*, 15: 249-268, 2000.
- [11] M.H. Lim, Y. Yuan and S. Omatu. Extensive testing of a hybrid genetic algorithm for quadratic assignment problem. *Computational Optimization and Applications*, 23: 47-64, 2002.
- [12] P. Mascato. On evolution, search, optimization, genetic algorithms and martial arts: Toward memetic algorithms. *Tech. Rep. Caltech Concurrent Computation Program*, Report. 826, California Institute of Technology, Pasadena, California, USA, 1989.
- [13] P. Merz and B. Freisleben. A comparison of memetic algorithms, tabu search, and ant colonies for the quadratic assignment problem. In *Proceedings of the 1999 International Congress of Evolutionary Computation (CEC'99)*, IEEE Press, pages 2063-2070, 1999.
- [14] P. Merz and B. Freisleben. Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. *IEEE Transaction on Evolutionary Computation*, 4 (4): 337-352, 2000.
- [15] Y.S. Ong and A.J. Keane. Meta-lamarckian in memetic algorithm. *IEEE Transactions on Evolutionary Computation*, 8 (2) (2004) 99-110.
- [16] E.D. Taillard. Comparison of iterative searches for the quadratic assignment problem. *Location Science*, 3: 87-105, 1995.
- [17] 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)*, Canberra, Australia, December 8-12, 2003.
- [18] J. Tang, M. H. Lim, Y.S. Ong and M. J. Er. Study of migration topology in island model parallel hybrid-GA for large scale quadratic assignment problems. In *The Eighth International Conference on Control, Automation, Robotics and Vision (ICARCV2004)*, Special Session on Computational Intelligence on the Grid, Kunming, China, December 6-9, 2004.