

PARALLEL MEMETIC ALGORITHM WITH SELECTIVE LOCAL SEARCH FOR LARGE SCALE QUADRATIC ASSIGNMENT PROBLEMS

JING TANG, MENG HIOT LIM, YEW SOON ONG AND MENG JOO ER

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

Received September 2005; revised January 2006

ABSTRACT. *The extent of the application of local searches in canonical memetic algorithm is typically based on the principle of “more is better”. In the same spirit, the 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 a complete local search, we termed it as PMA-CLS. We show in this paper that instead of a complete local search, the island model PMA with selective application of local search (PMA-SLS) is effective in solving complex combinatorial optimization problems, in particular large-scale quadratic assignment problems (QAPs). A distinct feature of the PMA-SLS to be noted in our study is the sampling size. We make use of a normal distribution scheme to determine the sampling ratio. Empirical study on large scale QAPs with PMA-SLS and PMA-CLS are presented. It is shown that PMA-SLS arrives at solutions that are competitive to the PMA-CLS at significantly lower computation efforts on the diverse large scale QAPs considered. This we concluded is due mainly to the ability of the PMA-SLS to manage a more desirable diversity profile as the search progresses.*

Keywords: Combinatorial optimization, Quadratic assignment problem, Island model parallel memetic algorithm, Selective local search

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 [15] 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 $M = \{1, 2, \dots, n\}$, 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 denote an assignment as π with $\pi(i)$ being 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 [19], tabu search [1,8,28,30], ant colony optimization algorithms [25,29], simulated annealing [18,34,35], genetic algorithms (GA) [20,21], memetic algorithms [23,24] and others [13,14]. Among the work cited, [20,21] describe our work on a class of serial hybrid genetic algorithms based on the k -gene exchange local search for solving QAPs. The k -gene exchange relies on repeated exchanges of the alleles of at least 2 genes in a chromosome such that the overall objective value as a result of the exchange is reduced. For each generation, only the chromosomes improved by GA operators participate in the local search. This will serve as a baseline for comparison against the approach presented in our current work.

In recent decades, the role of local search in the context of genetic algorithms and the wider field of evolutionary computing has been widely discussed. Mascato and Norman [22,26] promoted the term “memetic algorithm” to describe genetic algorithm that relies heavily on local search. Some memetic algorithms have been applied quite successfully in a wide range of combinatorial optimization problems [17,23,24]. Much theoretical and empirical investigation on designing efficient memetic algorithm has been done by many researchers [10,12,16,17,27].

One pertinent issue of memetic algorithm is the usage of local search. While canonical memetic algorithm with complete local search helps improve the convergence rate significantly, poor diversity often results at the later stages due to excessive focus on local search. Intensive local search also incurs high computational 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. An intuitive approach to avoid redundant local search is to apply local search on the GA population selectively. This work represents an important departure from our previous work [20,21,32,33], where local search is applied only to chromosomes modified by the GA operators. At the same time, it departs from the canonical memetic algorithm (PMA-CLS) which relies heavily on local searches, i.e., local search is applied to all individuals in every MA generation.

In this paper, we present a study on island model parallel memetic algorithm with selective local search (PMA-SLS) for large scale QAPs. In particular, we consider complete local search at the initial stage, with decreasing intensity of local search as the search progresses. A well-known strength of evolutionary algorithm is the ability to partition populations or islands of individuals among multiple computing nodes. In addition, parallel EAs have been shown in many studies to produce good quality solutions even on highly complex problems [32,33]. Here, a grid-enabled solver is also used to facilitate the implementation of our parallel MA, such that islands of MA individuals are executed across several computing platforms in a networked environment.

This paper is organized as follows. Section 2 provides a brief overview of the standard memetic algorithm. The proposed island model parallel MA with selective local search is introduced in Section 3. 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) [20,21] and complete parallel memetic algorithm (PMA-CLS).

Analyses on the performance of the PMA-SLS in terms of solution quality, computational time, solution precision, and scalability are also presented in the section. Finally, Section 5 concludes this paper.

2. Memetic Algorithms. In the recent decades, the role of local search in the context of genetic algorithms and the wider field of evolutionary computing has become increasingly important. In the 80's, a new class of "Knowledge-augmented Genetic Algorithms (GAs)", sometimes called "hybrid-GAs", began to surface. Mascato and Norman [22,26] likened this to local refinement, and therefore promoted the term "memetic algorithm" to describe genetic algorithms that rely heavily on local search. Memetic algorithm is a marriage between a population-based global search and the local improvement made by each of the individuals. These methods are inspired by models of adaptation in natural systems that combine evolutionary adaptation of the population, with individual learning within a lifetime. Additionally, MAs are inspired by Richard Dawkin's concept of a meme, which represents a unit of cultural evolution that can exhibit local refinement [6]. Thus a memetic model of adaptation exhibits the plasticity of individuals that a strict genetic model fails to capture. A key difference exists between genes and memes [7]. Before a meme is passed on, it is typically adapted accordingly by the person who transmits, consistent with what that person thinks, understands and processes, whereas genes are passed on as a whole. In literature, MAs have also been referred to as hybrid genetic algorithms, genetic local searchers, Lamarckian genetic algorithms, Baldwinian genetic algorithms, etc. MAs have been applied in a number of different areas, for example operational research and optimization, automatic programming, machine and robot learning, the study and optimizing of models of economies, immune systems, ecologies, population genetics, the interaction between evolution and learning, and social systems, to name a few.

Some memetic algorithms have been applied quite successfully in a wide range of combinatorial and continuous optimization problems [12,17,23,24,27]. 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 (see for example [23] for a comparison of MAs against other approaches for the quadratic assignment problem). In standard memetic algorithms, a complete local search is performed on every individual in the MA's population. Goldberg and Voessner [10] presented a theoretical framework for discussing the balance between genetic search and local search. Hart [12] investigated several issues especially related to local search for designing efficient memetic algorithms for continuous optimization. Land [17] 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 [16] 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 and Keane [27] also considered using multiple local methods or memes during a memetic algorithm search in the spirit of Lamarckian learning on continuous optimization problems. To summarize, there has been increasing efforts to address the following issues pertinent to memetic algorithms recently:

- (1) How often should a local search be applied?
- (2) Which and how many solutions should undergo a local search?
- (3) How long should local search be conducted?
- (4) How efficient does a local search need to be?
- (5) What local search method or meme should be used for a particular problem or individual?

Our focus in this paper is on the second issue. We present a study on applying a local search to individuals of the PMA population selectively so as to improve the efficiency of the memetic algorithm. We further consider its effect on the overall diversity of the population which consists of several islands of sub-population evolving concurrently.

3. Island Model PMA with Selective Local Search.

3.1. Island model PMA. A multi-island parallel memetic algorithm showing the inter-islands relationship is depicted in Figure 1.

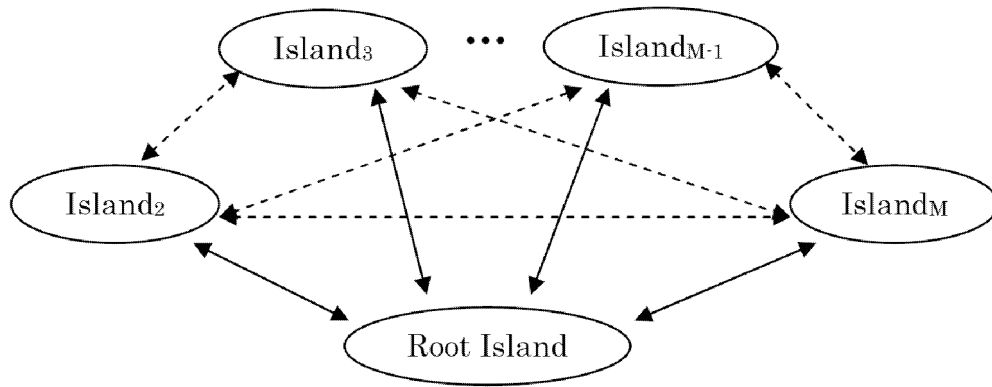


FIGURE 1. Inter-islands relationships of multi-island PMA model

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 sub-populations. 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 sub-populations. 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.

The pseudo-code shown in Figure 2 gives a schematic description of the island model parallel memetic algorithm.

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 sub-population is shown in Figure 3.

In each sub-population, a local search method is employed to improve the selected chromosome based on the Lamarckian learning strategy, i.e., replacement of genotype in

```

initialize  $M$  subpopulations of size  $N$  each
generation number := 1
WHILE termination condition not met do
    FOR each subpopulation do in parallel
        evaluate and select individuals by fitness
    IF migration interval is met then
        send  $K < N$  best individuals to a neighbouring subpopulation
        receive  $K$  individuals from a neighbouring population
        replace  $K$  individuals in the subpopulation
    END IF
    produce new individuals by crossover, mutation and local search
END PARALLEL DO
    generation number := generation number + 1
END WHILE

```

FIGURE 2. Pseudo code of the island model parallel memetic algorithm

```

BEGIN
    While (Stopping conditions are not satisfied or migration interval is not met)
        Evaluate all individuals in the subpopulation
        For each individual 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
        Create a new population (Selection, Mutation and Crossover).
    End While
END

```

FIGURE 3. Pseudo codes for memetic algorithm with selective local search strategy in each sub-population of PMA

the sub-population with the locally improved solution. The local search procedure implemented in the memetic algorithm is a form of k -gene exchange [20,21,32,33]. Standard GA operations are then used to create the next sub-population. The strategy used to selectively apply local search to the individuals in the sub-population is introduced in the following subsection.

3.2. Selective local search strategy.

3.2.1. Entropy and diversity of population on PMA-CLS. One of the key issues for an evolutionary algorithm is its convergence rate which can be measured by its population diversity. Consider Q to be mutually exclusive subsets $S_{t1}, S_{t2}, \dots, S_{tQ}$ in the t^{th} population. The number of individuals in each subset is $|S_{t1}|, |S_{t2}|, \dots, |S_{tQ}|$, respectively. Then the entropy E_t of the t^{th} population can be calculated as follows [11]:

$$E_t = - \sum_{j=1}^Q p_j \log(p_j) \quad (2)$$

where

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

Particularly, we monitor closely the change in diversity of each sub-population for PMA-CLS by measuring the entropy of each sub-population shown in Figure 4.

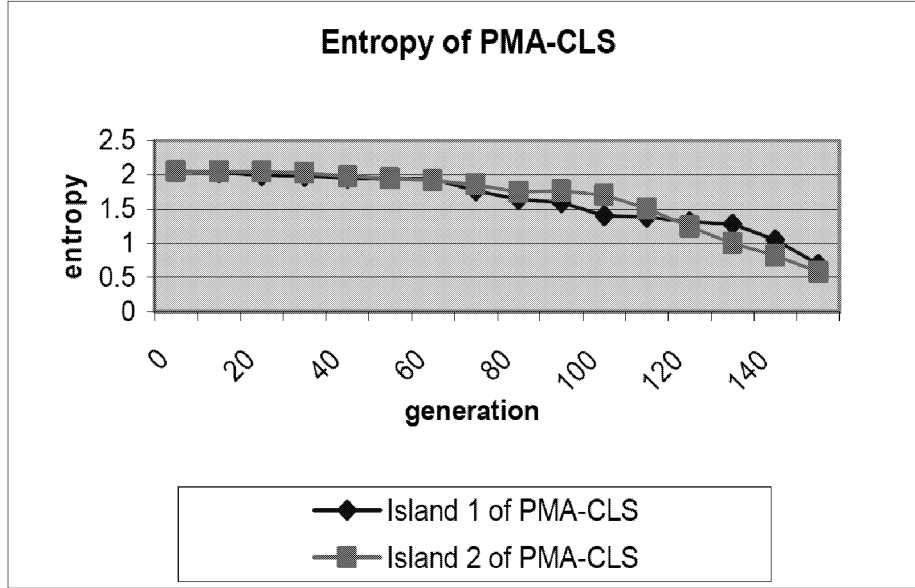


FIGURE 4. Entropy of PMA-CLS

Figure 4 illustrates the significant drop in entropy for the PMA-CLS, indicating that local search has a tendency to speed up convergence significantly. From an evolutionary process point of view, PMA-CLS results in poorer diversity due to excessive localized searches, especially at the later stage of evolution. This indicates a likelihood of higher ratio of duplicate solutions in the population. As a result, there is a high level of redundancy in local searches being applied to the whole sub-population over many generations. Consequently, PMA-CLS usually incurs higher computational cost due to the intensive local search. Thus, to reduce the tendency of applying local search on duplicates, we propose the selective local search strategy.

3.2.2. Selective local search strategy. The selective local search strategy is achieved by incorporating a selection function which dictates the number of individuals to be selected for local search. In our approach, we use a selection function characterized by the right half of the Gaussian distribution function. The number of chromosomes selected with respect to the overall population size is determined based on the selection ratio obtained from the right half Gaussian function as in Eq.(3).

$$\gamma(gen; \mu, \sigma, \eta) = \frac{1}{\sqrt{2\pi} * \sigma} \exp\left(-\frac{1}{2}\left(\frac{gen - \mu}{\sigma}\right)^2\right) * \eta \quad (3)$$

where gen is the evolution generation ($gen \geq 0$), μ and σ are the mean and standard deviation of the objective value for chromosomes in the population, respectively. η represents a scaling factor on the number of chromosomes the local search is applied. This

results in a progressively decreasing number of chromosomes whereby local search is being applied, as the number of evolutionary cycles increase. Based on Eq.(3), the number of chromosomes selected (Num) is determined according to the population size of each island, as per Eq.(4) below.

$$Num = \gamma * Subpopulation_size \quad (4)$$

4. Empirical Study. In this section, we present the empirical study of the island model parallel memetic algorithm based on the selective local search strategy, comparing with parallel hybrid GA [32,33], serial MA and PMA, both with complete local search. In particular, we evaluate the performance of different algorithms both in terms of computation time and solution quality. The statistical significance based on t -test for PMA-SLS compared with PMA-CLS is evaluated for the reduction of computation cost. The comparison in terms of scalability for the PMA is investigated as well. For convenience, the abbreviations for the different algorithms studied in the paper are as follows:

SMA – Serial memetic algorithm;

PMA-SLS – Island model parallel memetic algorithm with selective local search strategy;

PMA-CLS – Island model parallel memetic algorithm with complete local search strategy;

PHGA[*] – Island model parallel hybrid GA based on our previous work in [32] or [33].

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 simulation runs and the algorithms were evaluated based on their average performance.

In our empirical study, the configuration of the PMA control parameters is summarized in Table 1.

TABLE 1. Parameters setting for the PMA

MA parameters	Multi-island PMA
Population size	240
Sub-population size	$240/M$
Elite size	2 ($M=2$)
	1 ($M \geq 3$)
Maximum number of generations	180
Fitness scaling factor S_f	3
Crossover probability P_c	0.8
Mutation probability P_m	0.05
Zerofit threshold constant K_z	5

$M \equiv$ number of islands (processing nodes)

Except for S_f and K_z , all the above are standard GA parameters. S_f is the scaling factor to adjust the fitness values of the general population in order to avoid any unintentional bias. The approach for scaling is based on the commonly used linear scaling model [9].

The parameter K_z is a multiplication constant. It is akin to the upper bound value used in many deterministic search techniques. As far as the implementation of the GA goes, the K_z parameter shows up in the calculation of the fitness value. It specifies the upper bound as K_z times the lower bound value. The lower bound value is usually taken to be the known optimum solution or the best-known lower bound value of a QAP benchmark.

Through a series of empirical study and based on results of previous work [32,33], the following control parameters of migration operation have been adopted in the PMA:

- Migration Interval – Migration occurs every 10 generations;
- Migration Rate – One chromosome per migration phase;
- Migration Policy – Elitist strategy, where the best individual in one subpopulation replaces the worst in the other;
- Migration Topology – One-way ring topology.

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.

As for the parameters pertaining to the selective local search strategy of PMA-SLS, based on preliminary computational experiments, the adopted Gaussian function was configured by the parameters, where μ , σ and η were set to 0, 200 and 500, respectively. Sampling for local search was carried out every 10 generations.

Several criteria defined to measure the performance are listed as follows:

- CPU time* – Average computation time in seconds upon termination of the algorithm;
- Generation* – Average number of generations elapsed before the occurrence of the best solution;
- TG* – Average number of generations elapsed before the algorithm terminates;
- Average* – Average objective value of the solutions obtained for all the simulation runs;
- Average gap* – Difference between the *Average* and the best-known value of the objective function;
- Best* – Best solution obtained among all the simulation runs;
- Gap* – Difference between the best-found value and the best-known value of a benchmark problem;
- Success rate* – Number of times the algorithm finds the best-known solution out of all the simulation runs.

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 algorithm.

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 2.

In particular, we monitor closely the change in diversity of each sub-population for PMA-SLS and PMA-CLS by measuring the entropy of each sub-population shown in Figure 5. From Figure 5, it is observed that PMA-SLS can consistently maintain a

TABLE 2. 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%

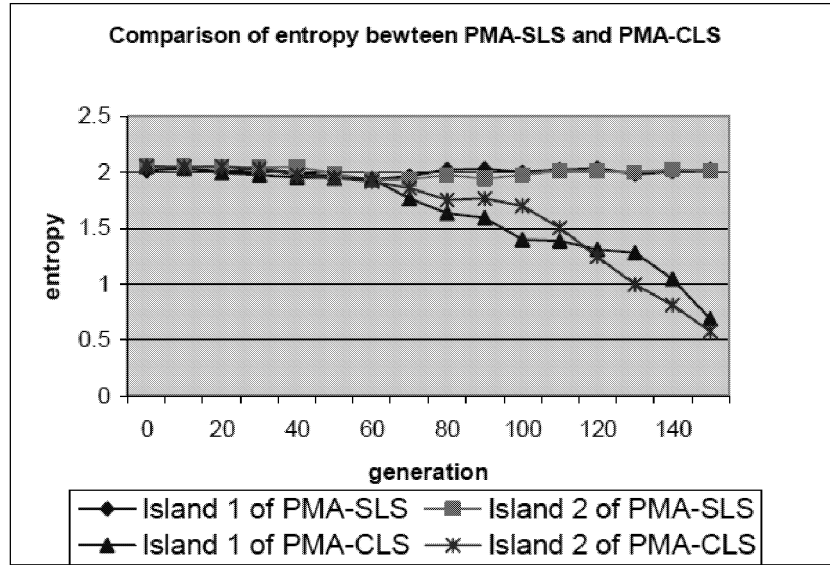


FIGURE 5. Comparison of entropy between PMA-SLS and PMA-CLS

good level diversity as the evolution progresses. On the other hand, the significant drop in entropy for the PMA-CLS indicates that the local search has a tendency to speed up convergence significantly. From an evolutionary process point of view, PMA-CLS results in poorer diversity due to excessive localized searches, especially at the later stages of evolution. This indicates a likelihood of higher ratio of duplicate solutions in the population. As a result, there is a high level of redundancy in local searches being applied to the whole sub-population over many generations. In addition, as shown in Table 2, 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 conclude that PMA-SLS is able to reduce the computational time spent significantly with little or no loss of solution quality. This is mainly attributed to its capability to maintain a higher level of population diversity.

Tables 3 to 8 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 [4]. The values in the first column of Tables 3 to 8 are the best-known cost values of the respective benchmark problems. The characteristics of these benchmark problems are summarized as follows:

- sko* – This group of benchmarks was proposed by J. Skorin-Kapov [28], the distance matrices of these problems are rectangular and the entries of the flow matrices are pseudo-random numbers.

tai – The instances *tai-a* are uniformly generated and was proposed in [30], while the instances *tai-b* were introduced in [31]. Problems of *tai-b* group are asymmetric and randomly generated.

wil – This group of benchmarks was proposed by M. R. Wilhelm and T. L. Ward [35], the distance matrices of these problems are rectangular.

tho – This group of benchmarks was proposed by U. W. Thonemann and A. Bolte [36], the distance matrices of these instances are rectangular.

The results of PMA-SLS are highlighted in bold for ease of comparison. Tables 3 and 4 present a detailed comparison study between PMA-SLS and some of our previous work [32,33] on *sko100b* and *tai100b* benchmarks, respectively. Tables 5 to 8 show the simulation results on the other classes QAPs, namely, *sko100**, *tai100a*, *wil100* and *tho150*, respectively.

TABLE 3. 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[32]	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[33]	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[33]	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[33]	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[33]	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[33]	119.60	150.80	220.80	154195.80	0.20%	153936	0.03%	0.00%

TABLE 4. 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[32]	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[33]	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[33]	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[33]	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[33]	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[33]	148.00	250.00	320.00	1187927883.00	0.16%	1186052259	0.00%	0.00%

TABLE 5. 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[32]	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[32]	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[32]	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[32]	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[32]	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 6. Results of testing on *tai100a* benchmark

			CPU time	Generation	TG	Average	Average gap	Best	Gap	Success rate
tai100a 21125314	2-island	PMA-SLS	860.00	127.20	164.60	21458262.60	1.58%	21382118	1.22%	0.00%
		PHGA[32]	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%

4.1. Results 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 in solving large scale QAPs. The higher success rate of PMA-SLS also indicates the improved solution precision due to the

TABLE 7. Results of testing on *wil100* benchmark

			CPU time	Generation	TG	Average	Average gap	Best	Gap	Success rate
wil100	2-island	PMA-SLS	882.10	114.50	166.20	273198.80	0.06%	273054	0.01%	0.00%
273038		PHGA[32]	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%

TABLE 8. Results of testing on *tho150* benchmark

			CPU time	Generation	TG	Average	Average gap	Best	Gap	Success rate
tho150	2-island	PMA-SLS	7290.30	140.30	174.00	8148332.60	0.18%	8142700	0.11%	0.00%
8133398		PHGA[32]	1428.50	308.50	368.20	8158144.60	0.30%	8140370	0.09%	0.00%
	3-island	PMA-SLS	6880.20	136.00	177.40	8145456.00	0.15%	8141754	0.10%	0.00%
		PHGA[33]	1168.40	180.20	250.20	8166026.60	0.40%	8148836	0.19%	0.00%
	4-island	PMA-SLS	5258.40	148.40	180.00	8144249.60	0.13%	8138428	0.06%	0.00%
		PHGA[33]	935.10	317.70	376.80	8162408.00	0.36%	8145990	0.15%	0.00%
	6-island	PMA-SLS	3267.40	156.20	180.00	8145297.20	0.15%	8142554	0.11%	0.00%
		PHGA[33]	885.30	332.50	386.00	8157363.67	0.29%	8151408	0.22%	0.00%
	8-island	PMA-SLS	3123.60	166.00	180.00	8143027.60	0.12%	8140248	0.08%	0.00%
		PHGA[33]	827.70	406.10	457.60	8161512.55	0.35%	8151436	0.22%	0.00%
	10-island	PMA-SLS	2004.20	136.00	177.20	8144993.20	0.14%	8142102	0.11%	0.00%
		PHGA[33]	605.10	418.50	462.40	8165464.60	0.39%	8154998	0.27%	0.00%

higher level of diversity maintained during the evolution process for PMA-SLS. In Tables 3 and 4, comparison among PMA-SLS, PMA-CLS and PHGA, shows that PMA-CLS and PMA-SLS can get much better quality than PHGA, and PMA-SLS can reduce the computational time significantly with little or no loss of solution quality compared to PMA-CLS.

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, which can be attributed to the powerful search capability of memetic algorithm. 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 6.

The plot in Figure 6(b) shows that PMA-SLS and PMA-CLS improve the solution quality significantly compared to PHGA. It 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 6(a), the greater reliance on local search makes PMA more time-consuming than the PHGA. In this respect, the island model paradigm of the parallel memetic algorithm and the distributed computing technology can help reduce the computational time significantly.

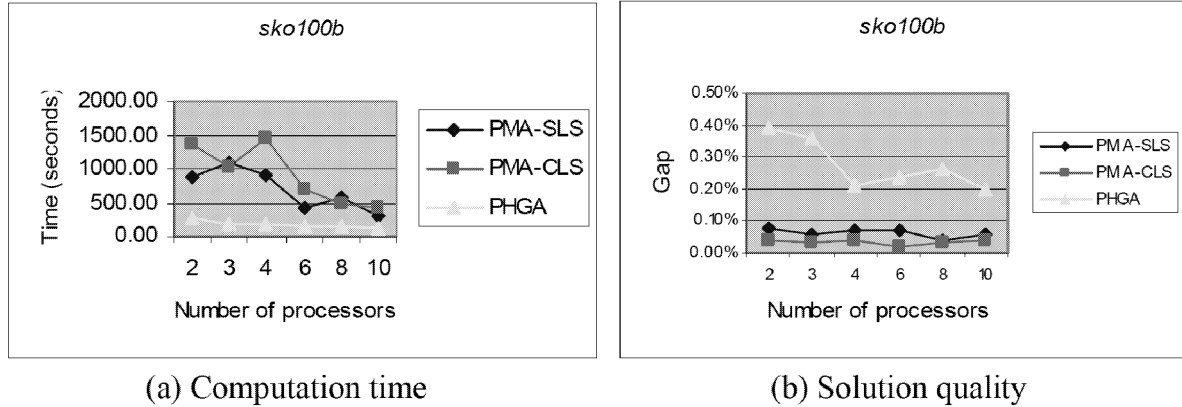


FIGURE 6. Comparison between PMA-SLS, PMA-CLS and PHGA on *sko100b* benchmark

Furthermore, the selective local search strategy used in PMA-SLS improves the efficiency of the PMA remarkably.

To determine the statistical significance of the reduced computational time by PMA-SLS, a t -test was used ($p < 0.05$), with the null hypothesis being that there is no difference between PMA-SLS and PMA-CLS. Based on t -test for statistical significance, the mean and the associated results of the one tail difference of two independent means, taken over 10 independent trials for *sko100b* between PMA-SLS and PMA-CLS, are presented in Table 9.

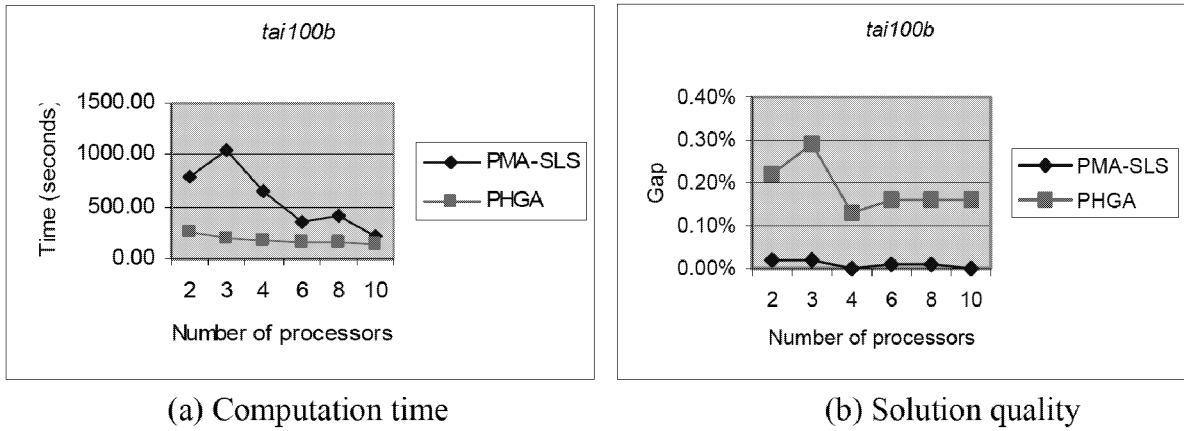
In Table 9, for 2-island, 4-island, 6-island and 10-island cases, the mean of PMA-SLS is smaller than that of PMA-CLS and the value $p < 0.05$ indicates that PMA-SLS indeed was able to reduce the computational time compared to PMA-CLS with high level of statistical significance. This validates the notion of PMA-SLS being able to search more efficiently than PMA-CLS to achieve comparable solution quality. There are only two exceptional cases, where the mean of PMA-SLS is larger than that of PMA-CLS. For the 3-island case, the result ($p > 0.05$) indicates the acceptance of null hypothesis that there is no statistical difference between PMA-SLS and PMA-CLS. For the 8-island case, the result ($p < 0.05$) indicates that PMA-SLS incurs higher computational time than PMA-CLS. Based on observations from this experiment, the reason for these two exceptional cases lies in the fact that given the sub-population size of the 3-island and 8-island, the reduced computation time cannot offset the increase in synchronization time overhead. This indicates that there is a tradeoff consideration in deciding on the size of sub-population for each island in PMA-SLS.

Similar to the *sko100b* benchmark, the effect of multiple islands processing is plotted as in Figure 7 for the *tai100b* benchmark. The results show that PMA-SLS can achieve much better solution quality with comparable computational time, especially for the case where the number of processor increases to 10 machines. It is also observed that the *tai100b* QAP 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 5 and 7, based on observations of the two criteria, *Gap* and *Success Rate*, the results of the different benchmarks (*sko100** and *wil100*) show that PMA-SLS can

TABLE 9. The mean and the one tail difference of t -test for statistical significance of *sko100b* between PMA-SLS and PMA-CLS

	PMA-SLS Mean	PMA-CLS Mean	p
2-island	875.2	1350	0.00010793
3-island	1084.8	1045.3	0.278106117
4-island	898	1445.9	1.0808E-11
6-island	429.4	694.3	1.331E-05
8-island	568.3	484	0.017846046
10-island	289.3	439	0.00010352

FIGURE 7. Comparison between PMA-SLS and PHGA on *tai100b* benchmark

significantly improve the solution quality in comparable computational time, especially so when the number of processor increases to 10 machines. The results indicate that PMA-SLS is superior to our previous PHGA for all benchmarks studied except for one instance, namely, *tai100a*, where marginal improvement in terms of solution quality was observed. This instance corresponds to the one that was randomly generated by Taillard using a uniform distribution. In [31], Taillard noted that for this type of randomly generated instance, finding good solutions (about 1% and 2%) is easy, but it is extremely difficult to find the optimum. This type of randomly generated instance is not that significant for practical applications of the QAP. As such, a set of non-uniformly generated random instances (*tai*b*) with the same characteristics as real-life problems was defined.

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 6 using the PMA-SLS is comparable to that found in [23]. 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 [23]. 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 [24] on the *sko100a* problem. As shown in Table 5, on the *sko100a* benchmark, the *Average gap* obtained in [24] was 0.096%, while we were able to reduce this value to 0.06%. As for the very large benchmark, *tho150*, the empirical results in Table 8 show that the solution quality on average is comparable to that found in [24].

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. Figure 8 presents the empirical results of scalability on all the three types of large benchmark problems studied above.

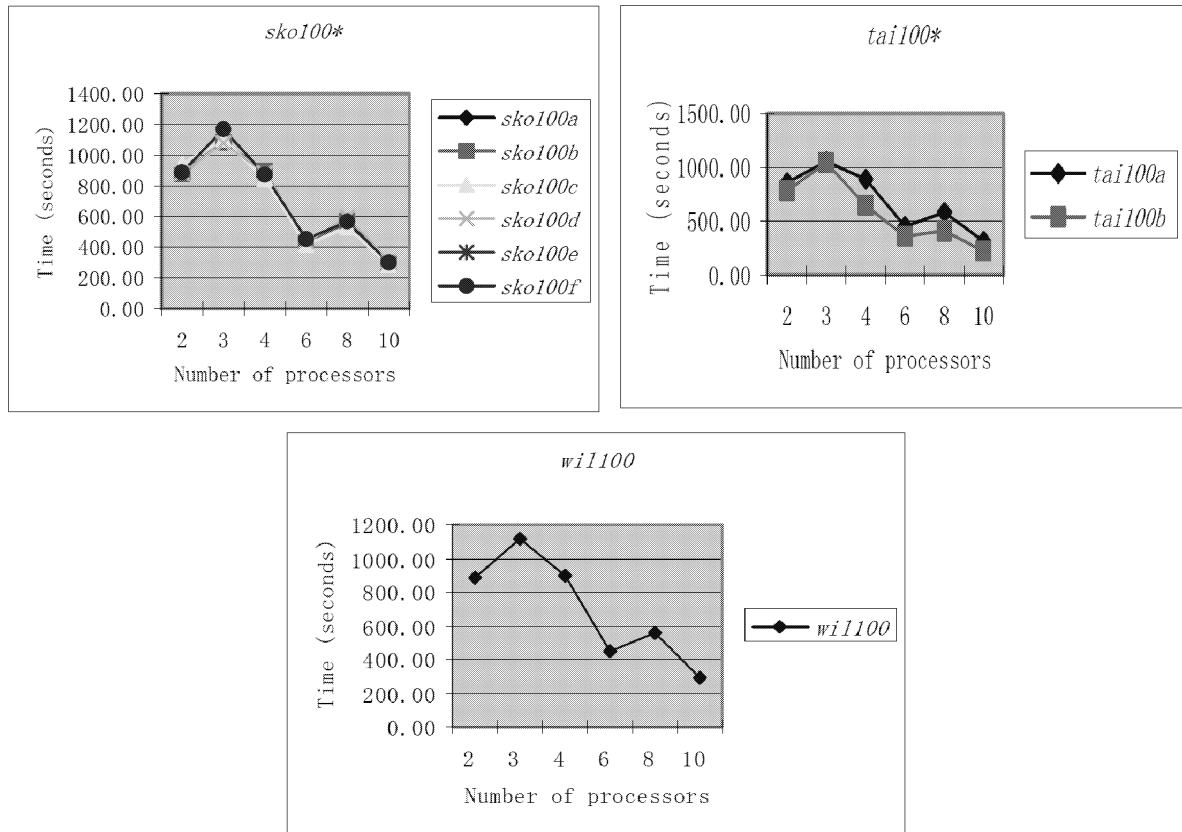


FIGURE 8. Scalability of *sko100**, *tai100** and *wil100* benchmarks by PMA-SLS

In Figure 8, the results show that for all the three types of large scale benchmark problems considered, 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 figures shows that there are tubers at the point of 3 and 8 in the number of processors. As shown in Table 9 above, one may concur that the capability of multi-island PMA-SLS could be related to the size of sub-populations.

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 literature. 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 improve the efficiency for large scale QAPs with little or no lost of solution quality. The improvement was shown to be statistically significant. The higher success rate of PMA-SLS coupled with improved solution precision can be attributed to the intrinsic parallelism and the higher level of diversity maintained during the evolutionary process of the PMA-SLS.

The results obtained have opened up several issues for future research. The issue on the effect of other parameters pertaining to the selective local search strategy on PMA-SLS's performance need to be explored, in order 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 progresses 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.

Acknowledgment. This work is funding supported by Singapore Technologies Engineering Pte Ltd. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

- [1] Battiti, R. and G. Tecchiolli, The reactive tabu search, *ORSA Journal on Computing*, vol.6, no.2, pp.126-140, 1994.
- [2] Bradwell, R. and K. Brown, Parallel asynchronous memetic algorithms, *Genetic and Evolutionary Computation Conference*, Evolutionary Computation and Parallel Processing Workshop, Orlando, Florida, 1999.
- [3] Burkard, R. E. and U. Derigs, Assignment and matching problems: Solution methods with FORTRAN-programs, *Lecture Notes in Economics and Mathematical Systems*, Springer-Verlag, Berlin, 1980.
- [4] Burkard, R. E., S. E. Karisch and F. Rendl, QAPLIB – A quadratic assignment problem library, *Journal of Global Optimization*, vol.10, pp.391-403, 1997. Available from < <http://www.opt.math.tu-graz.ac.at/qaplib/> >
- [5] Cantu-Paz, E., A survey of parallel genetic algorithms, *Calculateurs Paralleles, Reseaux et Systems Repartis*, vol.10, no.2, pp.141-171, Paris: Hermes, 1998.
- [6] Dawkins, R., *The Selfish Gene*, Oxford University Press, New York, 1976.
- [7] Digalakis, J. G. 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.
- [8] Drezner, Z., The extended concentric tabu for the quadratic assignment problem, *European Journal of Operational Research*, vol.160, no.2, pp.416-422, 2005.
- [9] Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, MA: Addison-Wesley: Reading, 1989.

- [10] Goldberg, D. and S. Voessner, Optimizing global-local search hybrids, *Proc. of the Genetic and Evolutionary Computation Conference*, Morgan Kaufmann, 1999.
- [11] Grefenstette, J. J., Incorporating problem specific knowledge into genetic algorithms, *Genetic Algorithms and Simulated Annealing*, Morgan Kaufmann Publishers, pp.42-60, 1987.
- [12] Hart, W. E., *Adaptive Global Optimization with Local Search*, Ph.D. Thesis, University of California, San Diego, USA, 1994.
- [13] Hasegawa, M., T. Ikeguchi, K. Aihara and K. Itoh, A novel chaotic search for quadratic assignment problems, *European Journal of Operational Research*, vol.139, no.3, pp.543-556, 2002.
- [14] Ishii, S. and M. Sato, Constrained neural approaches to quadratic assignment problems, *Neural Networks*, vol.11, pp.1073-1082, 1998.
- [15] Koopmans, T. C. and M. J. Beckmann, Assignment problems and the location of economic activities, *Econometrica*, vol.25, pp.53-76, 1957.
- [16] Krasnogor, N., *Studies on the Theory and Design Space of Memetic Algorithms*, Ph.D. Thesis, University of the West of England, Bristol, 2002.
- [17] Land, M. W. S., *Evolutionary Algorithms with Local Search for Combinatorial Optimization*, Ph.D. Thesis, University of California, San Diego, USA, 1998.
- [18] Laursen, P. S., Simulated annealing for the QAP – Optimal tradeoff between simulation time and solution quality, *European Journal of Operational Research*, vol.69, pp.238-243, 1993.
- [19] Li, Y., P. M. Pardalos, 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.
- [20] Lim, M. H., 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.
- [21] Lim, M. H., 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.
- [22] Mascato, P., 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.
- [23] Merz, P. and B. Freisleben, A comparison of memetic algorithms, tabu search, and ant colonies for the quadratic assignment problem, *Proc. of the 1999 International Congress of Evolutionary Computation*, IEEE Press, pp.2063-2070, 1999.
- [24] Merz, P. and B. Freisleben, Fitness landscape analysis and memetic algorithms for the quadratic assignment problem, *IEEE Transactions on Evolutionary Computation*, vol.4, no.4, pp.337-352, 2000.
- [25] Middendorf, M., F. Reischle, H. Schmeck, Multi colony ant algorithms, *Journal of Heuristics*, vol.8, pp.305-320, 2002.
- [26] Norman, M. and P. Moscato, A competitive and cooperative approach to complex combinatorial search, *Proc. of the 20th Joint Conference on Informatics and Operations Research*, Buenos Aires, Argentina, pp.3.15-3.29, 1991.
- [27] Ong, Y. S. and A. J. Keane, Meta-lamarckian in memetic algorithm, *IEEE Transactions on Evolutionary Computation*, vol.8, no.2, pp.99-110, 2004.
- [28] Skorin-Kapov, J., Tabu search applied to the quadratic assignment problem, *ORSA Journal of Computing*, vol.2, no.1, pp.33-45, 1990.
- [29] Solimanpur, M., P. Vrat and R. Shankar, Ant colony optimization algorithm to the inter-cell layout problem in cellular manufacturing, *European Journal of Operational Research*, vol.157, no.3, pp.592-606, 2004.
- [30] Taillard, E. D., Robust tabu search for the quadratic assignment problem, *Parallel Computing*, vol.17, pp.443-455, 1991.
- [31] Taillard, E. D., Comparison of iterative searches for the quadratic assignment problem, *Location Science*, vol.3, pp.87-105, 1995.

- [32] Tang, J., M. H. Lim and Y. S. Ong, A parallel hybrid GA for combinatorial optimization using grid technology, *Proc. of the Congress on Evolutionary Computation*, Canberra, Australia, vol.3, pp.1895-1902, 2003.
- [33] Tang, J., 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, *Proc. of the Eighth International Conference on Control, Automation, Robotics and Vision*, Special Session on Computational Intelligence on the Grid, Kunming, China, 2004.
- [34] Tian, P., J. Ma and D. M. Zhang, Application of the simulated annealing algorithm to the combinatorial optimisation problem with permutation property: An investigation of generation mechanism, *European Journal of Operational Research*, vol.118, no.1, pp.81-94, 1999.
- [35] Wilhelm, M. R. and T. L. Ward, Solving quadratic assignment problems by simulated annealing, *IIE Transaction*, vol.19, no.1, pp.107-119, 1987.
- [36] Thonemann, U. W. and A. Bolte, An improved simulated annealing algorithm for the quadratic assignment problem, Working paper, School of Business, Department of Production and Operations Research, University of Paderborn, Germany, 1994.