Adaptation for Parallel Memetic Algorithm Based on Population Entropy

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

ABSTRACT
In this paper, we propose the island model parallel memetic algorithm with diversity-based dynamic adaptive strategy (PMA-DLS) for controlling the local search frequency and demonstrate its utility in solving complex combinatorial optimization problems, in particular large-scale quadratic assignment problems (QAPs). The empirical results show that PMA-DLS converges to competitive solutions at significantly lower computational cost when compared to the canonical MA and PMA. Furthermore, compared to our previous work on PMA using static adaptation strategy, it is found that the diversity-based dynamic adaptation strategy displays better robustness in terms of solution quality across the class of QAP problems considered without requiring extra effort in selecting suitable parameters.

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, population entropy

1. INTRODUCTION
One of the recent growing areas in Evolutionary Algorithm (EAs) research is *Memetic Algorithms* (MAs) [16]. MAs are population-based meta-heuristic search methods inspired by Darwinian's principles of natural evolution and Dawkins' notion of a meme. A meme is defined as a unit of cultural evolution that is capable of local refinements [5]. Hence, a memetic model of adaptation exhibits the plasticity of individuals that a strictly genetic model fails to capture. Recent studies on MAs have revealed their successes on a wide variety of optimization problems [9, 13, 18, 11, 21]. Some theoretical and empirical investigations on MAs can be found in [7, 9, 13, 18, 11, 21].

A well known strength of evolutionary algorithms (EAs) is the ability to partition the population of individuals or islands of EA subpopulations among multiple computing nodes. Recent extensions of MAs to parallel MAs (PMAs) have been proposed in [23, 24]. It is worth noting that a crucial aspect of MAs or PMAs is in defining an optimum balance between the extent of exploration provided by the GA, against the level of exploitation posed by the local search procedure throughout the memetic search. However, in canonical MAs or PMAs, it is common practice for the local search procedure to be applied on every individual/chromosome in the GA population(s). This is a very computationally intensive and inefficient search process. At the same time, exhaustive local search may lead to ineffective search due to premature fall in diversity during the PMA search. Our previous work [25] has proposed a selective local search strategy (PMA-SLS). However, it is a static adaptive approach based on pre-defined Gaussian function, which is less robust and requires tedious parameters tuning. Here, we consider a diversity-based dynamic adaptive strategy (PMA-DLS) for controlling the local search frequency and demonstrate its utility in solving complex combinatorial optimization problems, in particular large-scale quadratic assignment problems (QAPs).

The QAP is a class of NP-hard combinatorial optimization problems with many interesting practical applications. It was formulated by Koopmans and Beckmann [10] 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 $M=\{1,2,3,\ldots,n\}$, 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 which facility i is assigned.

Since QAP is NP-hard, only implicit enumeration approaches are known to solve them optimally. However, larger size problems (n>20) are generally considered to be 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) [14, 15], memetic algorithms [17, 18].

This paper presents an investigation on PMA-DLS for controlling the local search frequency in the PMA search. In contrast to canonical MAs and PMAs, the diversity-based dynamic adaptive approaches control the number of individuals undergoing the local search procedure throughout the PMA evolutionary search process. PMA-DLS controls the local search frequency adaptively based on changes of population diversity during the PMA search. The numerical results indicate that PMA-DLS converge to competitive solutions at significantly lower computational cost when compared to the canonical MA and PMA. Furthermore, the diversity-based dynamic adaptation strategy is shown to be more robust in terms of solution quality on the class of QAP problems considered compared to our previous work [25]. However, previous static adaptation strategy would require effort in selecting appropriate parameters to suit the problems in hand.

This paper is organized as follows. Section 2 provides a brief overview of the recent research activities on memetic algorithm. The proposed diversity-based dynamic adaptive approaches for controlling the local search frequency in the island model parallel memetic algorithm are described in Section 3. Section 4 presents the numerical results obtained from empirical study and provides a comprehensive quantitative/statistical comparison of PMA-DLS, PMA-SLS and PMA in the context of large scale QAPs. The search performances of the various algorithms in terms of solution quality, computational time, and solution precision are also reported in the section. Finally, we conclude the paper in Section 5.

2. RELATED WORK

Memetic algorithm may be regarded as a marriage between a population-based global search and the local improvement made by each of the individuals. This has the potential to exploit the complementary advantages of EAs (generality, robustness, global search efficiency), and problem-specific local search (exploiting application-specific problem structure, rapid convergence toward local minima). In recent years, a number of independent researchers have addressed several issues relating to the trade-off between exploration and exploitation in MAs. In this section, we present a brief overview on some of the typical issues considered in the literature as follows:

1) How often should local learning be applied, i.e., local search frequency?
2) On which solutions should the local learning be applied?
3) How long should the local learning be run, i.e., local search intensity?
4) Which local learning procedure or local search or meme to use?

One of the first issues pertinent to memetic algorithm design is considering how often the local search should be applied for, i.e., local search frequency? Hart [9] investigated the effect of local search frequency on MA search performance. Upon which, he investigated various configurations of the local search frequency at different stages of the MA search. Conversely, it was shown in K.W.C. Ku et al. [12] that it may be worthwhile to apply local search on every individual if the computational complexity of the local search is relatively low. Hart [9] also studied the issue on how to best select the individuals among the EA population that should undergo local search. In his work, fitness-based and distribution-based strategies were studied for adapting the probability of applying local search on the population of chromosomes in continuous parametric search problems. Land [13] extended his work to combinatorial optimization problems and introduced the concept of "sniff" for balancing genetic and local search or also known as the local/global ratio. In [7], Goldberg and Voessner provide a theoretical alternative for efficient global-local hybrid search and characterize the optimum local search time that maximizes the probability of achieving a solution of a specified accuracy. Recently, Bambha et al. [1] introduced a simulated heating technique for systematically integrating parameterized local search into evolutionary algorithms to achieve maximum solution quality under a fixed computational time budget.

It is worth noting that the performance of MA search is also greatly affected by the choice of neighborhood structures. Fitness landscape analysis [17] provided a way for identifying the structure of a given problem and thus a selection of local search algorithms. Krasnogor [11] 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 different local search intensity. The choice of multiple local learning procedure or memes during a memetic algorithm search in the spirit of Lamarckian learning, otherwise, known as meta-Lamarckian learning, on continuous optimization problems was also considered in Ong et al. [21]. For a detail taxonomy and comparative study on adaptive choice of memes in memetic algorithms, the reader may refer to [22].

A variety of parallel memetic algorithm (PMA) models that extends from canonical PGA have also been recently studied in the literature. These include the blackboard parallel asynchronous memetic algorithm proposed [2], master/slave PMA [6] and the island model PMA [4]. The issue on which individuals should local learning be applied was also recently considered in the context of island model parallel memetic algorithm [4]. From a survey of the literature, insignificant efforts have considered balancing global and local search in the context of parallel MA. In particular, little work in the literature has studied the effect of local search frequency on the diversity of PMAs. Hence, we present a study on the negative impact of excessive local search in island model PMA in the subsequent sections. In the event, we proposed diversity-based dynamic adaptive strategy for controlling the local search frequency in island model parallel memetic algorithms.

3. DIVERSITY-BASED DYNAMIC ADAPTIVE STRATEGY FOR ISLAND MODEL PMA

3.1 Canonical Island Model Parallel Memetic Algorithm (PMA)

In this paper, we focus on Island Model Parallel Memetic Algorithm (PMA) for solving large-scale combinatorial optimization problems. The pseudo code of a canonical PMA is outlined in Fig. 3-1.

In the first step, M subpopulations are randomly initialized. All individuals in the subpopulations then undergo the local search learning procedure in the spirit of Lamarckian learning. The local search procedure considered here is based on the k-gene exchange [23, 24, 14, 15]. Subsequently new subpopulations are created using the evolutionary operators, particularly, selection, mutation and crossover. For every P migration interval, K best performing individuals in each subpopulation migrate to its neighbouring subpopulation based on the one-way ring topology [24]. At the same time, it receives K individuals from a neighbouring subpopulation. The replacement scheme may be a random walk or alternatively, the worst performing K individuals are replaced with the K migrants from its neighbour. The entire procedure repeats until the stopping conditions are satisfied.

It is common knowledge that good diversity represents a core advantage of using island model parallel memetic algorithm for solving global optimization problems. Here, we consider using the 1) 2-island PMA and 2) 2-island PGA for solving the *sko100b* QAP benchmark optimization problem. Note that PGA represents a canonical parallel GA. In contrast to PMA, no form of local search is used throughout the PGA search. The diversity of each subpopulation may be measured by various means. A simple method is using entropy measure E:

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

BEGIN
Initialize M subpopulations of size N each
WHILE (termination condition not met)
FOR each subpopulation or island
Evaluate all individuals in the subpopulation
For each individual in the subpopulation
- 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 based on Selection, Mutation and Crossover.
END FOR
For every P migration interval
Send $K < N$ best individuals to a neighbouring subpopulation.
Receive K individuals from a neighbouring subpopulation.
Replace K individuals in the subpopulation.
END For
END WHILE
END

Fig. 3-1 Pseudo code of the canonical island model PMA

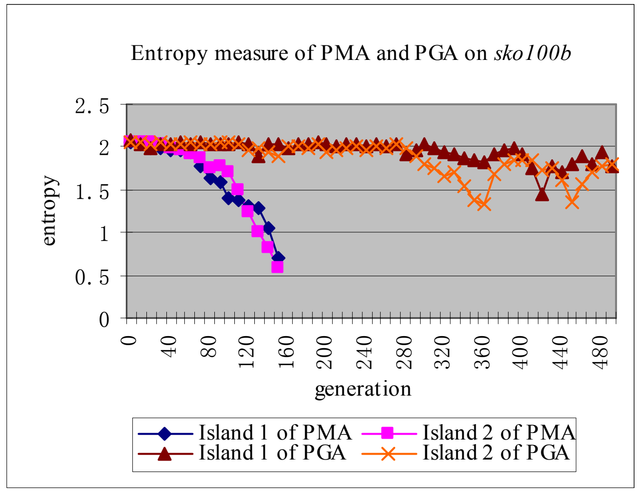


Fig. 3-2 Entropy measure for PMA and PGA on the sko100b QAP problem

where $p_j = \frac{|S_j|}{N}$ and N is the population size for each subpopulation [8]. Q is the number of the mutually exclusive subsets $S_1, S_2, \cdots, S_Q$ in each subpopulation. Each subset consists of individuals with the same fitness value. The number of individuals in each subset is $|S_1|$, $|S_2|, \cdots, |S_Q|$, respectively. Hence E indicates the degree of diversity in the subpopulation. For illustration, the diversity of each subpopulation in the 2-island PMA and PGA based on the entropy measure is depicted in Fig. 3-2.

From Fig. 3-2, it is worth highlighting the significant drop in the entropy measure of the PMA search in comparison to the PGA counterpart when searching on the *sko100b* benchmark problem. From these results, it appears that PMA loses search diversity much earlier than PGA due to possible excessive local searches. This significant drop in diversity for the PMA indicates the benefits of using local search in speeding up convergence rate of the search. However, it also implies the high risk of the PMA in losing search diversity prematurely as a result of the extensive local searches. This effect can also be observed to be more significant at the later stage of evolution search. To minimize the risk of premature convergence in the PMA, it is reasonable to ask whether the effects on performance might be reduced via adapting the local search frequency in the PMA search. Here, we present diversity-based dynamic adaptive strategy (PMA-DLS) for controlling the local search frequency in the PMA search.

3.2 Diversity-based Dynamic Adaptive Strategy (PMA-DLS)

In Fig. 3-2, it is noted that the population diversity degrades gradually with the evolving generation. Online entropy measure provides dynamic information about the stage of the evolutionary search process and the degree of diversity of each subpopulation. Since population diversity represents a crucial characteristic of the PMA, the approach considered here makes use of online entropy measure to adapting the local search frequency of the PMA, which is the diversity-based dynamic adaptive strategy or PMA-DLS in short. Hence, the dynamic local search frequency β in PMA-DLS can be defined based on online entropy ratio given by

$$\beta(gen) = 1 + \frac{E(gen) - E(gen-k)}{E(gen-k)} \tag{3}$$

where $E(gen)$ and $E(gen-k)$ $(gen \geq k)$ are the population entropy measure at the gen^{th} and $(gen\text{-}k)^{th}$ generation, respectively.

The PMA-DLS search thus begins by initializing all subpopulations randomly with a population size of ξ chromosomes, i.e., $\phi(0) = \xi$. Subsequently, the number of chromosomes that undergo local learning is defined by equation (4) that changes based on diversity level of the subpopulations.

$$\phi(gen) = \begin{cases} \xi, & gen = 0 \\ Min[\phi(gen-k) * \lfloor \beta(gen) \rfloor, \xi], & gen > 0 \end{cases} \tag{4}$$

4. EMPIRICAL STUDY

For comparison purpose, PMA-DLS denotes the island model PMA with diversity-based dynamic adaptive strategy. PMA-SLS denotes our previous work [25], i.e., the island model PMA with the selective local search strategy. PMA-FLS refers to the island model PMA with fixed local search strategy where local search is applied only to individuals that have undergone modification by the evolutionary operators [23, 24]. PMA denotes the island model PMA with complete local search strategy. MA abbreviates canonical memetic algorithm.

A grid-enabled solver [19, 20] is used to facilitate the implementation of the algorithms [23]. The algorithms were evaluated by averaging over 10 optimization runs. The configuration of the PMA control parameters is summarized in Table 4-1. The migration control parameters, stopping criterion and several other criteria have been defined to measure the performance as in [25].

4.1 Results Comparison – PMA-DLS *vs.* PMA-SLS

For the selective local search strategy in PMA-SLS [25] based on the Gaussian distribution function, the subpopulation size is a constant for certain number of islands in the PMA and μ is set to zero. The other two parameters (σ, η) were tuned in order to adjust the local search frequency for each generation *gen*. To decide on the appropriate configuration, significant effort was expended on parameters tuning in order to achieve a desirable level of performance. Various parameters setting for the Gaussian function were experimented to configure the PMA-SLS. For example, in Fig. 4-1, three Gaussian functions denoted as γ_1, γ_2 and γ_3 with different parameters setting are shown. The

Table 4-1 Parameters setting for the PMA

MA parameters	Multi-island PMA
Population size	240
Subpopulation size	240/*M*
Elite size	2 (*M*=2)
	1 (*M*≥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: number of islands (processing nodes)

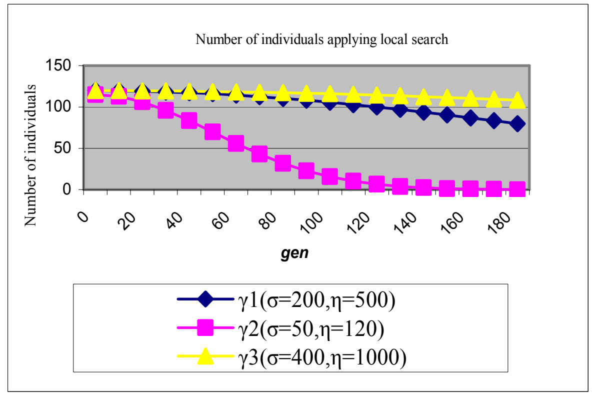


Fig. 4-1 Gaussian functions for PMA-SLS

corresponding number (*Num*) of individuals where local search is applied can be determined accordingly. Supposing that the number of island M is 2, the subpopulation size equals 120. Population sampling for local search was carried out every 10 generations. According to Fig. 4-1, application of γ_1, γ_2 and γ_3 will result in different local search frequency applied in the PMA. Based on the application of γ_1, γ_2 and γ_3, the corresponding PMA-SLS-1, PMA-SLS-2, and PMA-SLS-3 were derived. Meanwhile, PMA-DLS is more straightforward, with fewer parameters to set. Only parameter *k* is required for Equation (4). Here, *k* is set to 10.

We first carried out experimental study to gauge the effect of the choice of Gaussian function on the performance of PMA-SLS. The results presented in Table 4-2 are based on comparison of PMA-SLS and PMA-DLS on one particular benchmark. This experiment shows that PMA-DLS could produce good solutions with 0.08% average gap, consuming 859.40 seconds of CPU time. In comparison, the 3 variants of PMA-SLS vary in terms of solution quality and CPU time. In terms of CPU time, PMA-SLS-3 requires as much as 903.20 seconds while PMA-SLS-2 takes up 563.80 seconds of CPU time. On solution quality, the average gap of the PMS-SLS with the three configurations falls into the range of 0.08% to 0.16%. This may be due to the different number of individuals undergoing local search in PMA-SLS, especially at the later stages of the evolution process. For example, the number of individuals whereby local search is applied in PMA-SLS-3 is much larger than that for PMA-SLS-2 (when *gen*>100). Consequently, PMA-SLS-3 produced better solution (0.08%) quality than PMA-SLS-2 (0.16%). However, PMA-SLS-3 takes up more computational time. Thus, it appears that γ_1 ($\sigma = 200, \eta = 500$) produced the most competitive results in terms of solution quality and computational cost.

4.2 Results Comparison – PMA-DLS *vs.* PMA

To demonstrate the advantage of PMA-DLS, a specific comparison among PMA-DLS and PMA on the two-island model for the same benchmark, *sko100b*, is shown in Table 4-3. In Table 4-3, PMA-DLS produces competitive solutions although the frequency of local search of PMA-DLS never exceed that of PMA which maintain the highest local search frequency throughout the evolution process. The diversity of each subpopulation for PMA-DLS and PMA, measured by the entropy, was traced in our simulation and shown in Fig. 4-2.

Table 4-2 Comparison of PMA-DLS and PMA-SLS with γ_1, γ_2 and γ_3

			CPU time	Generation	TG	Average	Average gap	Best	Gap	Success rate
sko100b	2-island	PMA-SLS-1	875.20	113.60	168.40	154012.80	0.08%	153904	0.01%	0.00%
153890		PMA-SLS-2	563.80	134.10	174.70	154114.60	0.16%	153962	0.05%	0.00%
		PMA-SLS-3	903.20	119.20	171.30	154016.60	0.08%	153904	0.01%	0.00%
		PMA-DLS	**859.40**	**125.30**	**169.00**	**154020.80**	**0.08%**	**153920**	**0.02%**	**0.00%**

Table 4-3 Comparison between PMA-DLS and PMA

sko100b	2-island	CPU time	Generation	TG	Solution	Gap
153890	PMA	1461	81	151	153954	0.04%
	PMA-DLS	761	64	134	153942	0.03%

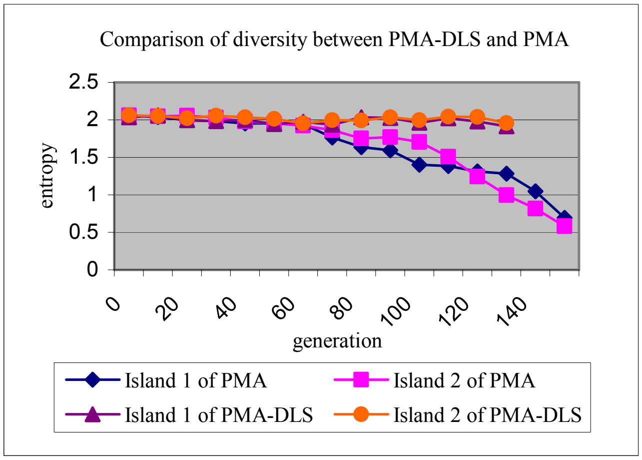


Fig. 4-2 Comparison of diversity between PMA-DLS and PMA

According to Fig. 4-2, PMA-DLS can consistently maintain a good level of diversity as the evolution progresses. However, the diversity of PMA shows a significant drop in entropy, especially at the later stages, indicating that local search has a tendency to speed up convergence significantly. From an evolutionary process point of view, PMA results in poorer diversity due to excessive localized searches, especially at the later stage of evolution. On the other hand, PMA-DLS adjusts the local search frequency based on changes in population diversity. The number of individuals to apply local search is then adjusted dynamically, enabling PMA-DLS to maintain a consistent level of population diversity. This in turn enhances the capacity of PMA-DLS to produce good solutions. A significant observation from Table 4-3 is that PMA-DLS and PMA achieved almost the same level of solution quality, with PMA incurring higher computational cost due to intensive local search. PMA-DLS therefore shows a potential for reducing computational time significantly with little or no lost of solution quality. This is mainly attributed to its capability to maintain a higher level of population diversity.

4.3 Full Comparison of Results and Analysis

Tables 4-4 to 4-7 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 [3]. The values in the first column of Tables 4-4 to 4-7 are the best-known values of the respective benchmark problems. Tables 4-4 and 4-5 present a detailed comparison study on *sko100b* and *tai100b* benchmarks, respectively. Tables 4-6 and 4-7 show the simulation results on the other two classes QAPs, namely, *tai100a* and *sko100** respectively.

An inspection of the experimental results indicates that PMA-DLS can significantly improve the efficiency in solving large scale QAPs. The higher success rate of PMA-DLS also indicates improved solution quality due to the higher level of diversity maintained during the evolution process. Furthermore, PMA-DLS can reduce the computational time significantly with little or no lost in solution quality compared to PMA. The comparison among PMA-SLS, PMA-DLS, PMA and PMA-FLS on *sko100b* benchmark is shown in Fig. 4-3.

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

			CPU time	Generation	TG	Average	Average gap	Best	Gap	Success rate
sko100b		MA	3096.50	127.30	160.50	153955.60	0.04%	153890	0.00%	20.00%
153890	2-island	PMA-DLS	859.40	125.30	169.00	154020.80	0.08%	153920	0.02%	0.00%
		PMA-SLS	875.20	113.60	168.40	154012.80	0.08%	153904	0.01%	0.00%
		PMA-FLS[6]	266.80	182.70	252.70	154494.20	0.39%	154160	0.18%	0.00%
		PMA	1350.00	94.70	145.90	153950.40	0.04%	153890	0.00%	20.00%
	4-island	PMA-DLS	885.70	131.40	170.00	153977.40	0.06%	153900	0.01%	0.00%
		PMA-SLS	898.00	137.10	178.10	153990.80	0.07%	153902	0.01%	0.00%
		PMA-FLS[7]	174.50	282.50	352.50	154213.80	0.21%	153952	0.04%	0.00%
		PMA	1445.90	122.20	174.60	153952.20	0.04%	153898	0.01%	0.00%
	6-island	PMA-DLS	413.80	126.80	164.20	153951.20	0.04%	153890	0.00%	20.00%
		PMA-SLS	429.40	130.20	168.20	153985.00	0.07%	153890	0.00%	10.00%
		PMA-FLS[7]	148.80	213.30	283.30	154254.60	0.24%	154074	0.12%	0.00%
		PMA	694.30	104.80	154.50	153925.40	0.02%	153890	0.00%	20.00%

Table 4-5 Results of testing on *tai100b* benchmark

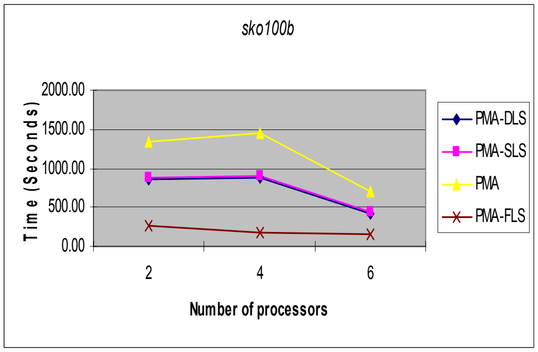
			CPU time	Generation	TG	Average	Average gap	Best	Gap	Success rate
tai100b	2-island	PMA-DLS	694.70	94.70	124.30	1186119285.20	0.01%	1185996137	0.00%	50.00%
1185996137		PMA-SLS	782.40	106.70	134.00	1186275856.50	0.02%	1185996137	0.00%	40.00%
		PMA-FLS[6]	186.90	175.30	245.30	1188882832.20	0.24%	1186007112	0.00%	0.00%
	4-island	PMA-DLS	633.40	105.00	122.00	1186121434.00	0.01%	1185996137	0.00%	80.00%
		PMA-SLS	647.50	92.60	102.30	1186007361.40	0.00%	1185996137	0.00%	80.00%
		PMA-FLS[7]	178.10	268.80	332.00	1187539521.00	0.13%	1186007112	0.00%	0.00%
	6-island	PMA-DLS	342.40	65.20	107.20	1186132401.50	0.01%	1185996137	0.00%	70.00%
		PMA-SLS	356.60	88.30	104.90	1186058956.40	0.01%	1185996137	0.00%	70.00%
		PMA-FLS[7]	160.10	233.70	296.70	1187892570.00	0.16%	1185996137	0.00%	10.00%

Table 4-6 Results of testing on *tai100a* benchmark

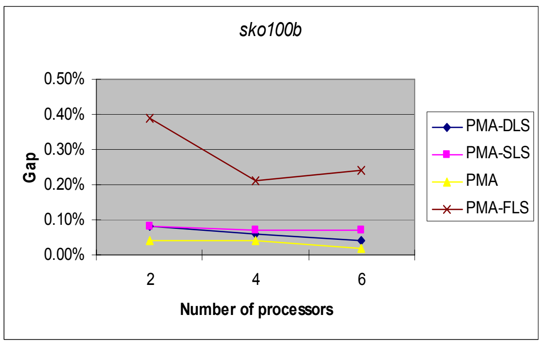
			CPU time	Generation	TG	Average	Average gap	Best	Gap	Success rate
tai100a	2-island	PMA-DLS	866.20	127.90	161.80	21442193.71	1.50%	21379594	1.18%	0.00%
21125314		PMA-SLS	860.00	127.20	164.60	21458262.60	1.58%	21382118	1.22%	0.00%
		PMA-FLS[6]	222.80	238.50	308.50	21464686.20	1.61%	21335594	1.00%	0.00%
	4-island	PMA-DLS	889.20	156.20	180.00	21380930.80	1.21%	21362016	1.12%	0.00%
		PMA-SLS	889.60	140.20	170.90	21420954.60	1.40%	21352956	1.08%	0.00%
	6-island	PMA-DLS	433.60	146.40	180.00	21368255.10	1.15%	21237278	0.53%	0.00%
		PMA-SLS	451.40	152.50	180.00	21373508.00	1.17%	21270370	0.69%	0.00%

Table 4-7 Results of testing on *sko100benchmarks**

			CPU time	Generation	TG	Average	Average gap	Best	Gap	Success rate
sko100a	2-island	PMA-DLS	852.80	118.80	164.50	152156.10	0.11%	152069	0.03%	0.00%
152002		PMA-SLS	883.60	133.80	175.10	152188.20	0.12%	152042	0.03%	0.00%
		PMA-FLS[6]	194.00	203.40	273.40	152322.80	0.21%	152122	0.08%	0.00%
	4-island	PMA-DLS	855.30	132.40	171.60	152104.10	0.07%	152059	0.04%	0.00%
		PMA-SLS	885.20	142.40	176.80	152119.00	0.08%	152058	0.04%	0.00%
	6-island	PMA-DLS	416.60	131.60	170.80	152126.20	0.08%	152044	0.03%	0.00%
		PMA-SLS	431.90	138.90	176.90	152109.40	0.07%	152067	0.04%	0.00%
sko100c	2-island	PMA-DLS	847.30	120.60	167.90	147928.60	0.05%	147862	0.00%	10.00%
147862		PMA-SLS	939.30	121.80	168.40	147934.80	0.05%	147862	0.00%	10.00%
		PMA-FLS[6]	184.40	205.80	275.80	148140.40	0.18%	148050	0.13%	0.00%
	4-island	PMA-DLS	826.30	112.20	163.00	147894.00	0.02%	147862	0.00%	20.00%
		PMA-SLS	845.90	111.40	160.50	147908.20	0.03%	147862	0.00%	10.00%
	6-island	PMA-DLS	401.20	124.20	173.00	147887.20	0.02%	147868	0.00%	30.00%
		PMA-SLS	416.80	106.40	151.80	147885.60	0.02%	147862	0.00%	20.00%
sko100d	2-island	PMA-DLS	869.60	136.10	170.80	149742.20	0.11%	149656	0.05%	0.00%
149576		PMA-SLS	883.00	111.00	166.90	149803.60	0.15%	149618	0.03%	0.00%
		PMA-FLS[6]	232.10	259.90	327.40	150036.80	0.31%	149732	0.10%	0.00%
	4-island	PMA-DLS	813.50	137.00	177.20	149729.20	0.10%	149648	0.05%	0.00%
		PMA-SLS	881.20	146.70	180.00	149752.00	0.12%	149630	0.04%	0.00%
	6-island	PMA-DLS	429.00	134.20	168.40	149707.60	0.09%	149620	0.03%	0.00%
		PMA-SLS	436.80	135.80	173.80	149699.40	0.08%	149578	0.00%	0.00%
sko100e	2-island	PMA-DLS	809.40	111.30	148.00	149198.20	0.03%	149150	0.00%	30.00%
149150		PMA-SLS	845.40	121.00	166.70	149205.80	0.04%	149150	0.00%	10.00%
		PMA-FLS[6]	235.50	252.90	322.90	149642.20	0.33%	149198	0.03%	0.00%
	4-island	PMA-DLS	864.80	119.80	173.60	149188.80	0.03%	149150	0.00%	10.00%
		PMA-SLS	898.50	114.30	164.50	149202.60	0.04%	149150	0.00%	10.00%
	6-island	PMA-DLS	425.00	107.80	144.80	149183.60	0.02%	149150	0.00%	40.00%
		PMA-SLS	452.10	113.70	156.90	149179.20	0.02%	149150	0.00%	30.00%
sko100f	2-island	PMA-DLS	825.70	98.90	155.20	149218.03	0.12%	149096	0.04%	0.00%
149036		PMA-SLS	888.40	104.60	153.70	149232.80	0.13%	149126	0.06%	0.00%
		PMA-FLS[6]	206.50	214.80	284.80	149496.60	0.31%	149228	0.13%	0.00%
	4-island	PMA-DLS	813.90	130.20	173.40	149144.20	0.07%	149036	0.00%	20.00%
		PMA-SLS	872.10	126.10	166.70	149150.40	0.08%	149036	0.00%	10.00%
	6-island	PMA-DLS	423.20	151.40	180.00	149145.20	0.07%	149092	0.04%	0.00%
		PMA-SLS	451.70	136.10	172.30	149205.40	0.11%	149078	0.03%	0.00%



(a) Computation time



(b) Solution quality

Fig. 4-3 Comparison among PMA-SLS, PMA-DLS, PMA and PMA-FLS on *sko100b* benchmark

The plot in Fig. 4-3(b) shows that PMA-DLS, PMA-SLS and PMA improve the solution quality significantly compared to PMA-FLS. It is noted that the maximum number of generations for PMA-FLS was set at 500. Instead, the maximum number of generations for PMA-DLS, PMA-SLS and PMA 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 Fig. 4-3(a), the greater reliance on local search makes PMA more time-consuming than the PMA-FLS. However, with the island model paradigm of the parallel memetic algorithm, distributed computing technology can help to reduce the computational time significantly. Furthermore, the diversity-based dynamic adaptive local search strategy used in PMA-DLS improves the efficiency of the PMA remarkably.

Similar to the *sko100b* benchmark, the effect of multiple islands processing for the *tai100b* benchmark shows that PMA-DLS can achieve much better solution quality with comparable computational time. It is also observed that the *tai100b* QAP benchmark shows a much higher *Success rate*, indicating that the PMA-DLS has greater success in locating the global optimum. This implies that the PMA-DLS is capable of locating the best-known solution more frequently than the PMA-FLS. In addition, the results in Table 4-6 show that PMA-DLS are even more superior compared to PMA-FLS, even for the seemingly difficult class of benchmarks, *tai100a*. Remarkable improvement in terms of solution quality was observed.

4.4 Comparison with Other Results

When judged against existing results available in the literature, it is noted that the results for PMA-DLS of several instances is much better than that of the MAs developed by other authors. For example, our results of *tai100b* for PMA-DLS are much better than that shown in [17]. The *Average gap* of *tai100b* was reported as 0.026%, with the *Success rate* being less than 50%. On the other hand, *Average gap* achieved by our PMA-DLS (0.01%) is much better, and the *Success rate* is very commendable, being as high as 80%. Furthermore, it is worth nothing that the PMA-DLS are also capable of attaining search quality that is significantly better than that obtained in [18] on the *sko100a* problem. As shown in Table 4-7, on the *sko100a* benchmark, the *Average gap* obtained in [18] was 0.096%, while we were able to reduce this value to 0.07%.

5. CONCLUSION

This paper proposes a diversity-based dynamic adaptive strategy in the island model parallel memetic algorithm with adaptive local search frequency. Based on the comparison between PMA-SLS and PMA-DLS, there are three advantages of PMA-DLS which determines the number of individuals undergoing local search based on online dynamic population diversity. First, the number of individuals to be selected for local search is made dynamic and adaptive to online fluctuation of population diversity. This diversity-adaptive approach avoids premature convergence resulting from fast decreasing population diversity, as well as reduces computational cost. In addition, for the island model PMA, the diversity-based dynamic adaptive local search is able to adjust the number of individuals for local search according to the different diversity fluctuation tendency in each island. Secondly, the PMA-DLS adjusts the local search frequency online, avoiding the laborious task of parameters tuning of PMA-SLS. For PMA-SLS, the Gaussian function used to decide on the local search frequency was problem specific. It was configured through trial-and-error experimentation without generalization or analysis of the characteristics of the PMA with respect to population diversity, an important characteristic indicative of the population convergence level. Therefore, PMA-DLS is desirable to produce more robust solution quality. Thirdly, an intrinsic characteristic of PMA-DLS is the Markovian property, in deciding the frequency of applying local search. Equation (4) computes the number of chromosomes that undergo local learning in the current generation based on the previous k generations and the current generation. This property is consistent with the theoretical foundation of various evolutionary algorithms, such as genetic algorithms and memetic algorithms.

6. ACKNOWLEDGMENTS

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

7. REFERENCES

[1] N.K. Bambha, S.S. Bhattacharyya, J. Teich and E. Zitzler, Systematic integration of parameterized local search into evolutionary algorithms, *IEEE Transactions on Evolutionary Computation*, 8 (2): 137-155, 2004.

[2] 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.

[3] 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/

[4] C. Cotta, A. Mendes, V. Garcia, P. Franca, and P. Moscato, Applying memetic algorithms to the analysis of microarray data, *Application of Evolutionary Computing*, G. Raidl et al. (eds.), Lecture notes in computer science, Springer-Verlag , 2611: 22-32, 2003.

[5] R. Dawkins, *The Selfish Gene.* Oxford University Press, New York, 1976.

[6] J.G. Digalakis and K.G. Margaritis, Performance Comparison of Memetic Algorithms, *Journal of Applied Mathematics and Computation*, Elsevier Science, 158 (25): 237-252, 2004.

[7] 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.

[8] J.J. Grefenstette. Incorporating problem specific knowledge into genetic algorithms. In *Genetic Algorithms and Simulated Annealing*, Morgan Kaufmann Publishers, pages 42-60, 1987.

[9] W.E. Hart. *Adaptive Global Optimization with Local Search*. Ph. D. Thesis, University of California, San Diego, 1994.

[10] T.C. Koopmans and M.J. Beckmann. Assignment problems and the location of economic activities. *Econometrica*, 25: 53-76, 1957.

[11] N. Krasnogor. *Studies on the Theory and Design Space of Memetic Algorithms*. Ph. D. Thesis, University of the West of England, Bristol, June 2002.

[12] K.W.C. Ku, M.W. Mak and W.C. Siu, A study of the Lamarckian evolution of recurrent neural networks, *IEEE Transactions on Evolutionary Computation*, 4 (1): 31-42, 2000.

[13] M.W.S. Land. *Evolutionary Algorithms with Local Search for Combinatorial Optimization*. Ph. D. Thesis, University of California, San Diego, 1998.

[14] 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.

[15] 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.

[16] P. Mascato. *On evolution, search, optimization, genetic algorithms and martial arts: Toward memetic algorithms*. Technical Report. Caltech Concurrent Computation Program, Report. 826, California Institute of Technology, Pasadena, California, USA, 1989.

[17] 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.

[18] P. Merz and B. Freisleben. Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. *IEEE Transactions on Evolutionary Computation*, 4 (4): 337-352, 2000.

[19] H.K. Ng, Y.S. Ong, T. Hung and B.S. Lee. Grid enabled optimization. European Grid Conference, *Lecture Notes on Science*, 3470: 296-304, 2005.

[20] H.K. Ng, D. Lim, Y.S. Ong, B.S. Lee, L. Freund, S. Parvez and B. Sendhoff. A multi-cluster grid enabled evolution framework for aerodynamic airfoil design optimization. International Conference on Natural Computing, 27-29 August 2005, *Lecture Notes on Science*, 3611: 1112-1121, Springer-verlag, L.P. Wang, K. Chen and , Y.S. Ong, Eds.

[21] Y.S. Ong and A.J. Keane. Meta-lamarckian in memetic algorithm. *IEEE Transactions on Evolutionary Computation*, 8 (2): 99-110, 2004.

[22] Y.S. Ong, M.H. Lim, N. Zhu and K.W. Wong. Classification of adaptive memetic algorithms: a comparative study. *IEEE Transactions on Systems, Man and Cybernetics - Part B*, 36(1): 141-152, 2006.

[23] 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.

[24] 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.

[25] J. Tang, M.H. Lim, Y.S. Ong and M.J. Er. Solving large scale combinatorial optimization using PMA-SLS. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'05),* Washington, DC, USA, June 25–29, 2005.