# Towards Probabilistic Memetic Algorithm: An Initial Study on Capacitated Arc Routing Problem

Liang Feng, Yew-Soon Ong *Member, IEEE*, Quang Huy Nguyen and Ah-Hwee Tan *Member, IEEE*

***Abstract*— Capacitated arc routing problem (CARP) has attracted much attention due to its generality to many real world problems. Memetic algorithm (MA), among other meta-heuristic search methods, has been shown to achieve competitive performances in solving CARP ranging from small to medium size. In this paper we propose a formal probabilistic memetic algorithm for CARP that is equipped with an adaptation mechanism to control the degree of global exploration against local exploitation while the search progresses. Experimental study on benchmark instances of CARP showed that the proposed probabilistic scheme led to improved search performances when introduced into a recently proposed state-of-the-art MA. The results obtained on $24$ instances of the capacitated arc routing problems highlighted the efficacy of the *probabilistic scheme* with $9$ new best known solutions established.**

## I. Introduction

Capacitated Arc Routing defines the problem of servicing a set of street networks using a fleet of capacity constrained vehicles located at the central depot. The objective of the problem is to minimize the total routing cost involved. In practice, the CARP and its variants are found to be in abundance across applications involving the serving of street segments instead of specific nodes or points. Typical examples of CARP would include the applications of urban waste collection, winter gritting and post delivery [1]. Theoretically, CARP has been proven to be NP-hard with only explicit enumeration approaches known to solve them optimally. However, large scale problems are generally computationally intractable due to the poor scalability of most enumeration methods. From a survey of the literature, many heuristic approaches have played an important role in algorithms capable of providing good solutions within tractable computational time. In [2], Lacomme *et al.* presented the basic components that have been embedded into memetic algorithms (MAs) for solving the extended version of CARP (ECARP). Lacomme's MA (LMA) was demonstrated to outperform all known heuristics on three sets of benchmarks. Recently, Mei *et al.* [3] extended Lacomme's work by introducing two new local search methods, which successfully improved the solution quality of LMA. In addition, a memetic algorithm with extended neighborhood search was also proposed for CARP in [4]. Nevertheless, it is worth noting that majority of the works are designed based on heuristics that comes with little theoretical rigor.

Liang Feng, Yew-Soon Ong, Quang Huy Nguyen and Ah-Hwee Tan are with the School of Computer Engineering, Nanyang Technological University, 50 Nanyang Avenue, 639798, Singapore (email: FENG0039@ntu.edu.sg, ASYSOng@ntu.edu.sg, qhnguyen@ntu.edu.sg, ASAHTan@ntu.edu.sg).

In this paper, we present a formal probabilistic memetic algorithm or PMA in short for CARP. In contrast to earlier works demonstrated in the context of non-linear programming problem, using the theoretical framework introduced in [5], an upper bound for local search intensity is estimated and subsequently used to govern the level of evolutionary versus lifetime learning[1] for solving CARP. The rest of the paper is organized as follows: the detailed introduction of CARP and memetic algorithm (MA) are presented in section II, while the proposed probabilistic memetic approach is described in section III. Section IV presents and analyzes the experimental results on typically used CARP benchmarks. Finally, section V summarizes the paper with some conclusions.

## II. Preliminary

### A. Problem Definition

The CARP, first proposed by Golden and Wong [6], can be formally stated as follows: Given a connected undirected graph $G = (V, E)$, where vertex set $V = \{v_i\}, i = 1 \ldots n$, $n$ is the number of vertices, edge set $E = \{e_i\}, i = 1 \ldots m$ with $m$ denoting the number of edges. Consider a demand set $D = \{d(e_i)|e_i \in E\}$, where $d(e_i) > 0$ implies edge $e_i$ requires servicing, a travel cost vector $C_t = \{c_t(e_i)|e_i \in E\}$ with $c_t(e_i)$ representing the cost of traveling on edge $e_i$, a service cost vector $C_s = \{c_s(e_i)|e_i \in E\}$ with $c_s(e_i)$ representing the cost of servicing on edge $e_i$.

*Definition 1*: Given a depot node $v_d \in V$, a travel circuit $\mathcal{C}$ starting and ending at $v_d$ is considered **valid** if and only if the total load $\sum_{\forall e_i \in \mathcal{C}} d(e_i) \leq W$, where $W$ is the capacity of each vehicle. The cost of a travel circuit is then defined by the total service cost for all edges that needed service together with the total travel cost of the remaining edges that formed the circuit:

$$cost(\mathcal{C}) = \sum_{e_i \in \mathcal{C}_s} c_s(e_i) + \sum_{e_i \in \mathcal{C}_t} c_t(e_i) \tag{1}$$

where $\mathcal{C}_s$ and $\mathcal{C}_t$ are edge sets that required servicing and those that do not, respectively.

*Definition 2*: A set of travel circuits $\mathcal{S} = \{\mathcal{C}_i\},\ i = 1 \ldots k$ is a **valid solution** to the CARP if and only if:

1) $\forall i \in [1, k]$, $\mathcal{C}_i$ is valid.
2) $\forall e_i \in E\ and\ d(e_i) > 0$, there exists one and only one circuit $\mathcal{C}_i \in \mathcal{S}$ such that $e_i \in \mathcal{C}_i$.

[1]In this paper, the term lifetime learning is often used interchangeably with local search

The objective of CARP is then to find a valid solution $\mathcal{S}$ that minimizes the total cost:

$$C_{\mathcal{S}} = \sum_{\forall \mathcal{C}_i \in \mathcal{S}} cost(\mathcal{C}_i) \tag{2}$$

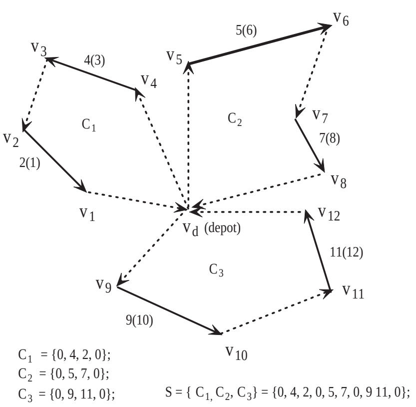


Fig. 1. An example of CARP

The example of a CARP is illustrated in Fig. 1, with $v_d$ representing the depot, full line denoting edges that require servicing (otherwise known as tasks) and dashed lines representing edges that do not require servicing. Each task is assigned a unique integer number (e.g., 2 is assigned to the task from $v_2$ to $v_1$), the integer numbers enclosed in brackets denoting the inversion of each task (i.e., direction of edge) accordingly. In Fig. 1, three feasible solution circuits $\mathcal{C}_1 = \{0, 4, 2, 0\}$, $\mathcal{C}_2 = \{0, 5, 7, 0\}$, and $\mathcal{C}_3 = \{0, 9, 11, 0\}$ can be observed, each composing of two tasks. A '0' index value is assigned at the beginning and end of circuits to initialize each circuit to start and end at the depot. According to equations 1 and 2, the total cost of a feasible solution $S = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3\}$ is then obtained as sum of the service costs for all tasks and the travel costs for all edges involved.

### B. *Memetic Algorithm*

Memetic algorithm (MA) has materialized as a form of population based search with lifetime learning as a separate process capable of local refinement for accelerating search. Recent studies on MAs have demonstrated that they converge to high quality solutions more efficiently than their conventional counterparts [7], [8], [9], [10], [11], [12], [13], [14], [15], [16] on many real world applications. To date, many dedicated MAs have been crafted to solve domain-specific problems more efficiently. In a recent special issue dedicated to MA research [17], several new design methodologies of memetic algorithms [17], [18], [19], [20], [21], [22], [23], and specialized memetic algorithms designed for tackling the permutation flow shop scheduling [20], optimal control systems of permanent magnet synchronous motor [21], VLSI floor planning [24], quadratic assignment problem [25], [26], gene/feature selection [27], have been introduced. From a survey of the area, it is now well established that potential algorithmic improvement can be achieved by considering some important issues of MA [5], [28], [29]:

1) Local search frequency, hereby denoted as $f_{il}$: defines how often should local learning be applied. $f_{il}$ can be represented as a percentage of the population, i.e., the percentage of individuals in the population that undergoes local learning, or the ratio of evolutionary to local search, i.e., in how many generations of global search should local learning be conducted. Alternatively, $f_{il}$ can be replaced with the local search probability, $P_{il}$, which defines the probability at which each individual in the population should undergo local learning.
2) Local search intensity, $t_{il}$: defines how much computational budget should be allocated to each local learning process. $t_{il}$ may be represented in terms of number of the objective function evaluations or time budget.
3) Subset of solution undergoing local search, $\Omega_{il}$: represents the subset of the solution population that undergoes local learning.
4) Local search method: which among a given set of available local learning strategies should be employed on a given problem at hand.

While the above issues have been studied extensively in the literature, for examples, Hart [30] and Ku [31] on the local search frequency, Land [32] on selecting appropriate individuals among the EA population that should undergo local search, Goldberg and Voessner [33] on local search intensity, Ong [7] and Kendall [34] on the selection of local search; it is worth noting that the works only consider the design issues separately. The recent work by Nguyen *et al.* [5] on the other hand proposed a theoretic probabilistic memetic framework (PrMF) that unifies the local search frequency, intensity and selection of solutions undergoing local search under a single theme. The proposed algorithm was demonstrated to exhibit superior performances on a set of continuous benchmark problems. However, it is worth noting that PrMF was designed specifically for handling of continuous optimization problem and the extension of the framework to combinatorial optimization is non-trivial, due to the lack of generic definitions of distance and basin of attraction in the context of combinatorial problem, since the topology of search space greatly depends on the variation operators considered. In this paper, we attempt to extend the formal probabilistic memetic framework for capacitated arc routing problem, which is described in the next section.

## III. Probabilistic Memetic Algorithm for Capacitated Arc Routing Problem

In [5], the theoretical upper bound on local search intensity of an MA was derived as:

$$t_{upper} = \frac{t_g}{n} \frac{ln(1 - p_2^{(k)})}{ln(1 - p_1^{(k)})} \tag{3}$$

where $t_g$ denotes the function evaluations incurred in a generation, $n$ denoting the population size, $p_1^{(k)}$ and $p_2^{(k)}$ representing the probabilities of an individual, in generation

$k$, hitting the global optimum or falling within the basin of attraction of the global optimum, respectively. Based on Taylor series expansion and with $t_g$ configured to $n$, the above equation was simplified to:

$$t_{upper} = \frac{p_2^{(k)}}{p_1^{(k)}} \tag{4}$$

Subsequently, the upper bound is used to determine whether the current individual should undergo local search and/or how much computational budget should be allocated to the local search phase.

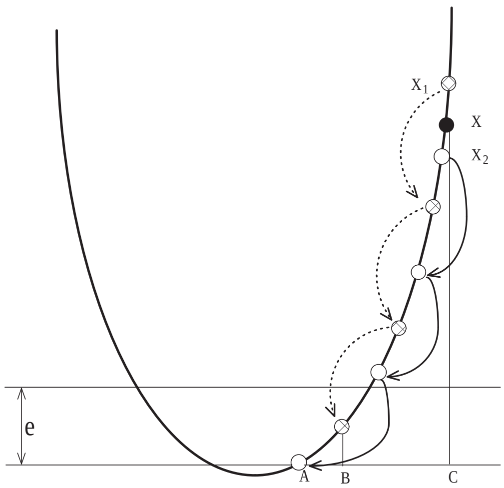


Fig. 2. A depiction on the estimation of $t_{upper}$ in the probabilistic memetic framework for non-linear programming

Fig. 2 presents a depiction on the estimation of $t_{upper}$ in the probabilistic memetic framework for non-linear programming, with $e$ denoting the precision accuracy for convergence to global optimum, $X$ representing the current individual of interest, while $X_1$ and $X_2$ are the nearest neighbors of $X$ (based on simple Euclidean distance) selected from the database of solution vectors previously encountered and archived while the search progresses. Note that the local search traces for $X_1$ and $X_2$ are also depicted in the figure. From here, the best solution $A$ found in the neighborhood of $X$ and the furthest search point $B$ within the range of $e$ from $A$ then to approximate $p_1 = \frac{|AB|}{volume\ of\ search\ space}$, and $p_2 = \frac{|AC|}{volume\ of\ search\ space}$. The upper bound for local search intensity, $t_{upper}$, on $X$ is subsequently determined based on equation 4. On the other hand, the expected local search intensity, $t_{expected}$, required to reach the local optimum of $X$ is then defined as the estimated average length of local search traces of $X_1$ and $X_2$. Finally, local search is performed on the current individual only when $t_{expected} \leq t_{upper}$.

Due to the discrete nature of combinatoric problems, Euclidean distance does not serve as a suitable measure on solution closeness or similarity. The key challenge when extending the PrMF for continuous optimization problems in [5] to combinatoric context lies in the appropriate definition of distance metrics. Several candidates available in the context of combinatorial optimization include the *Hamming Distance*, *Minkowski-r-distance* [35], *exact match distance* [36], *deviation distance* [36], *edit distance* [35] or the *Jaccard's similarity coefficient* [37]. Here, we study the *Jaccard's similarity coefficient* since it was considered in the context of Vehicle Routing Problem (VRP) [37] to measure the similarity between two sets. The similarity coefficient is defined as the cardinality of the intersection of the two sets divided by the cardinality of union of the two:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{5}$$

where $A$ and $B$ are two sets containing elements of the same types. From the formulation above, it can be shown that if $A = B$, then $J(A, B) = 1$; if $|A \cap B| = 0$, then $J(A, B) = 0$, and $J \in [0, 1]$.

Here, Based on the idea of *Jaccard's similarity coefficient*, the closeness measure between two solutions of CARP is defined as follows:

$$Dis(A, B) = |A \cup B| - |A \cap B| \tag{6}$$

where $A$ and $B$ denote the sets of arcs for two solutions in CARP. The distance defined in Eq. 6 gives us an integer value in the range $[0, 2 \times |A|]$ which serves our algorithm better than the one defined in Eq. 5 which results a real number in the range $[0, 1]$.

With a distance metric defined, it is now possible to estimate the values for $p_1$ and $p_2$. To begin, the $q$ nearest neighbors of a given individual of interest is first identified. Since the definition of "nearest" relationship is generally loose in the combinatoric context, a single nearest neighbor of the current individual is found to be sufficient for accurate estimation.

Let the current solution be $X$. The nearest neighbor of $X$, denoted as $X_{nber}$, is then identified from database $\Omega$ which archives all previous local search traces. The local optimum reached, starting from $X_{nber}$, is then labeled here as $X_A$, while the solution individual found along the search trace before converging to $X_A$ is labeled as $X_B$, as depicted in Fig. 3 .

In Fig. 3, a "step" denotes a solution jumps or transition to a higher quality solution from the initial point. The dashed line represents the trace of $X_{nber}$ generated by the local search process. Probability $p_1$ is then derived as:

$$p_1 = \frac{Dis(X_B, X_A)}{volume\ of\ search\ space} \tag{7}$$

Probability $p_2$, on the other hand, is derived as:

$$p_2 = \frac{Dis(X, X_A)}{volume\ of\ search\ space} \tag{8}$$

Subsequently, the upper bound is then derived as:

$$t_{upper} = \frac{p_2}{p_1} = \frac{Dis(X, X_A)}{Dis(X_B, X_A)} \tag{9}$$

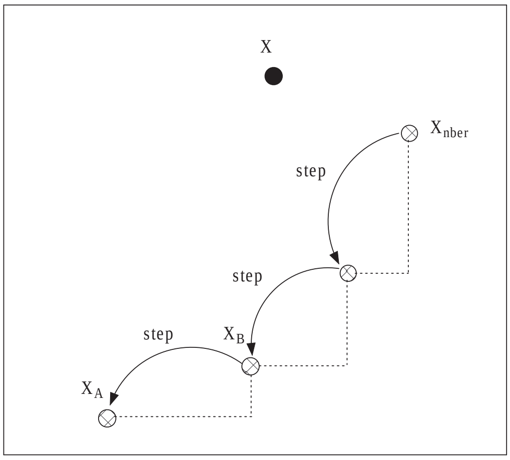


Fig. 3. A depiction on estimation of $t_{upper}$ in probabilistic memetic algorithm for combinatoric problem

Further, the upper bound is considered only if the current solution $X$ is sufficiently close to its nearest neighbor, that is:

$$Dis(X, X_{nber}) \leq \alpha \times Dis(X_{nber}, X_A) \tag{10}$$

where $\alpha$ is a parameter to scale the range. Thus the search will proceed with lifetime learning if Eq. 10 is valid.

The expected local search intensity on the other hand denotes the number of local search steps needed for $X$ to reach $X_A$. Here, we summarize the basic steps of the probabilistic memetic framework as follows:

**Algorithm 1** Outline of the probabilistic memetic framework

**Begin**:

**Step 1**: Identify the nearest neighbor $X_{nber}$ of the given individual $X$ from database $\Omega$

**Step 2**: Identify local optimum $X_A$ of $X_{nber}$ and solution $X_B$ in the local search trace of $X_{nber}$

**Step 3**: Compare the distance value $Dis(X, X_{nber})$ with the predefined neighborhood range $\alpha \times Dis(X_{nber}, X_A)$

**If** $(Dis(X, X_{nber}) > \alpha \times Dis(X_{nber}, X_A))$

$X$ will undergo local search

**Else**

1. Estimate the upper bound for lifetime learning or local search

$$t_{upper} = \frac{Dis(X, X_A)}{Dis(X_B, X_A)}$$

2. Estimate the expected value for lifetime learning or local search

$$t_{expected} = \; number\ of\ steps\ from\ X_{nber}\ to\ X_A.$$

3. if $t_{expected} \leq t_{upper}$, local search will take place; otherwise, proceed with global exploration

**End If**

**End**

TABLE I
SUMMARY OF THE PARAMETER SETTING IN OUR EXPERIMENT

| Description | Value |
|---|---|
| Population Size | 30 |
| Maximum number of restart | 20 |
| Local search probability (ILMA) | 0.1 in the 1st restart and 0.2 subsequently |
| Local search probability (PMA) | 0.1 in the 1st restart and adaptive subsequently |
| $\alpha$ in PMA | 1 |
| maximum solution length | 250 |
| Independent runs | 30 |

## IV. EXPERIMENT AND DISCUSSION

In this section, an experimental study on CARP benchmarks is conducted using the proposed probabilistic memetic algorithm (PMA). The performance efficacy of PMA is subsequently compared to the recently proposed improved memetic algorithm proposed in [3] for handling capacitated arc routing problem (ILMA), which forms the baseline for comparison.

### A. Detailed Setting

*1) Data Set:* The well-established *egl* CARP benchmark is used in the present experimental study. The data set was generated by Eglese based on data obtained from the winter gritting application in Lancashire [38], [39], [40]. It consists of 24 instances based on two graphs, each with a distance set of required edges and capacity constraints. Generally, *egl* consists of two groups of instances, the first with problem names started with E* while the second group's names started with S*. All instances in group E* has 77 vertices and 98 edges, while those in group S* possess 140 vertices and 190 edges. Hence, instances in group S* are deemed to be more complex than the counterparts in group E*.

*2) Numerical study:* Mei *et. al.* [3] proposed ILMA for handling of capacitated arc routing problem and shown to outperform several state-of-the-art algorithms on the *egl* CARP benchmark set. To exhibit the true efficacy on the theoretic rigor of the probabilistic memetic approach, we adopt the same global and local search operators as well as other algorithmic configurations, as described in ILMA in the numerical study. Further, in consistent with [3], the local search frequency for ILMA is fixed at 0.1 in the earlier stage and 0.2 subsequently. Alg.2 presents the ILMA equipped with the proposed probabilistic framework. For PMA, the local search frequency is also fixed at 0.1 before the first restart and subsequently adapted by the proposed algorithm. Further, for simplicity, here $\alpha$ is set to 1. All the experimental settings are summarized in Table I. Here, we made a prior assumption that the neighborhood structure of the local search conforms well with the modified Jaccard's coefficient defined in Eq. 6.

**Algorithm 2** Outline of ILMA equipped with the proposed probabilistic framework

**Begin**:
**Initialization**: Generate the initial population
**For** the first restart
    **While** (the termination criteria are not met)
    Select two chromosomes from the current population
    Perform crossover operator to generate offspring
    Apply the local search process with tracking capability on the generated offspring with a certain probability
    Update the current population with the newly generated offspring
**End For**
**For** each of the subsequently restart
**While** (stopping conditions are not satisfied)
    Select two chromosomes from the current population
    Perform crossover operator to generate offspring
    Apply probabilistic memetic framework to the offspring
    Update the current population with the newly generated offspring
**End While**
**End For**
**End**

TABLE II
NUMERICAL RESULTS OF ILMA AND PMA ON *egl* CARP BENCHMARK

| Data Set | $\|V\|$ | $\|E_R\|$ | $\|E\|$ | $LB$ | Best Known | ILMA $B.Cost$ | ILMA $Ave.Cost$ | ILMA $Std.Dev$ | PMA $B.Cost$ | PMA $Ave.Cost$ | PMA $Std.Dev$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1.E1-A | 77 | 51 | 98 | 3548 | 3548 | 3548 | 3548.0 | 0.0 | 3548 | 3548.0 | 0.0 |
| 2.E1-B | 77 | 51 | 98 | 4498 | 4498 | 4498 | 4512.3 | 12.3 | 4498 | **4500.7**†[2] | 8.2 |
| 3.E1-C | 77 | 51 | 98 | 5566 | 5595 | 5595 | 5602.1 | 9.9 | 5595 | **5597.1**† | 5.7 |
| 4.E2-A | 77 | 72 | 98 | 5018 | 5018 | 5018 | 5018.0 | 0.0 | 5018 | 5018.0 | 0.0 |
| 5.E2-B | 77 | 72 | 98 | 6305 | 6317 | 6317 | 6338.2 | 10.7 | 6317 | **6334.5** | 11.5 |
| 6.E2-C | 77 | 72 | 98 | 8243 | 8335 | 8335 | 8356.2 | 40.0 | 8335 | **8338.2**† | 12.2 |
| 7.E3-A | 77 | 87 | 98 | 5898 | 5898 | 5898 | 5918.8 | 33.9 | 5898 | **5906.8** | 22.9 |
| 8.E3-B | 77 | 87 | 98 | 7704 | 7777 | 7777 | 7797.8 | 25.0 | <u>**7775**</u> | **7789.3** | 17.7 |
| 9.E3-C | 77 | 87 | 98 | 10163 | 10292 | 10292 | 10314.2 | 23.7 | 10292 | **10313.7** | 38.9 |
| 10.E4-A | 77 | 98 | 98 | 6408 | 6456 | 6456 | 6472.4 | 16.0 | <u>**6444**</u> | **6464.8**† | 10.2 |
| 11.E4-B | 77 | 98 | 98 | 8884 | 8991 | 8991 | 9052.6 | 40.5 | <u>**8988**</u> | **9041.3** | 33.9 |
| 12.E4-C | 77 | 98 | 98 | 11427 | 11587 | **11587** | 11665.9 | 49.0 | 11594 | **11658.1** | 55.6 |
| 13.S1-A | 140 | 75 | 190 | 5018 | 5018 | 5018 | 5028.5 | 29.1 | 5018 | **5018.0** | 0.0 |
| 14.S1-B | 140 | 75 | 190 | 6384 | 6388 | 6388 | 6404.9 | 28.1 | 6388 | **6388.2**† | 1.1 |
| 15.S1-C | 140 | 75 | 190 | 8493 | 8518 | 8518 | 8567.0 | 37.8 | 8518 | **8554.4** | 35.7 |
| 16.S2-A | 140 | 147 | 190 | 9824 | 9911 | 9911 | 10025.9 | 57.0 | <u>**9908**</u> | **9989.1**† | 51.1 |
| 17.S2-B | 140 | 147 | 190 | 12968 | 13150 | **13150** | 13292.3 | 70.4 | 13153 | **13250.3**† | 75.1 |
| 18.S2-C | 140 | 147 | 190 | 16353 | 16489 | 16489 | 16621.3 | 69.7 | <u>**16442**</u> | **16568.8**† | 60.3 |
| 19.S3-A | 140 | 159 | 190 | 10143 | 10260 | 10261 | 10361.5 | 50.4 | <u>**10229**</u> | **10317.8**† | 50.9 |
| 20.S3-B | 140 | 159 | 190 | 13616 | 13807 | 13813 | 13948.3 | 64.4 | <u>**13711**</u> | **13843.0**† | 67.9 |
| 21.S3-C | 140 | 159 | 190 | 17100 | 17234 | 17288 | 17377.3 | 79.2 | **17244** | **17325.1**† | 71.0 |
| 22.S4-A | 140 | 190 | 190 | 12143 | 12341 | 12348 | 12484.5 | 67.2 | <u>**12323**</u> | **12449.4** | 71.4 |
| 23.S4-B | 140 | 190 | 190 | 16093 | 16345 | 16345 | 16553.1 | 71.9 | <u>**16331**</u> | **16449.6**† | 63.4 |
| 24.S4-C | 140 | 190 | 190 | 20375 | 20556 | **20556** | 20776.9 | 112.9 | 20578 | **20716.3**† | 79.1 |

### *B. Result and Discussion*

Table II tabulates the performance of $PMA$ and $ILMA$ on several metrics. "B. Cost", "Ave.Cost" and "Std.Dev" indicate the the best result, averaged result and standard deviation obtained by the corresponding algorithms across 30 independent runs, respectively. In addition, the lower bound "LB" and current best known results known to date for each problem instance are obtained from [3]. In addition, columns "$|V|$", "$|E_R|$", and "$E$" of Table II describe the number of vertices, tasks and total edge tasks, of each problem instance, respectively. It is observed that problem instances 1 to 12 (E1-A to E4-C) are smaller in size (in terms of vertices) than the others (S1-A to S4-C). In the table, the algorithm, i.e., $PMA$ and $ILMA$, with superior performance with respect to "B. Cost" and "Ave.Cost", are highlighted in bold font, while all newly found best known solution by the PMA are then marked by underline.

From Table II, it is observed that both PMA and ILMA managed to converge to optimum solutions on problem instances E1-A, E1-B, E2-A, E3-A, and S1-A. Moreover, on the first half of the problem instances, both PMA and ILMA performed competitively in converging to the same best solutions. However, on the more complicated problem instances, PMA exhibits superior performance to ILMA by achieving some new best known results, i.e., lower cost than existing best known solution to date. Particularly, among the 24 problem instances, PMA was able to attain 9 new best known solutions that are significant improvements of the best known solution to date (as defined by Best Known).

In terms of average cost, PMA also outperforms ILMA on 22 out of total 24 problem instances considered. In particular, on S1-A, PMA constantly finds the optimum solution. Last but not least, Fig. 4 and Fig. 5 present the performance of PMA and ILMA against the lower bounds of each *egl* CARP problem instance. It can be observed that while PMA performs marginally better than ILMA on the easy problems (E1A - E4C), the performance gaps increase significantly on larger size problems, which highlights the efficacy on the theoretic rigor of the probabilistic memetic approach in balancing between global evolution and lifetime learning or local search on complex problems while the search progresses. Last but not least, the improved performance showed that our assumption on the agreement between the neighborhood structure and defined distance metric appears to be valid.

## V. Conclusion

In the last decades, Memetic algorithm has emerged as an important paradigm for solving CARP and shown to attained promising performances. The improved memetic algorithm for capacitated arc routing problem (ILMA) demonstrated plausible performances with specially designed local search schemes. By equipping the ILMA with a theoretic upper bound on local search intensity, a probabilistic memetic algorithm (PMA) is designed for handling CARP. PMA uses the upper bound to adaptively control the degree of local search to perform while the search progresses online. Experimental study presented on typical CARP benchmark problems highlighted the efficacy of PMA in converging to competitive or improved solutions when compared to the ILMA. More important, PMA was shown to attain new best known solutions to date on 9 of the *egl* problem instances.

[2]The $t$ value of 29 degree of freedom is significant at a 0.05 level of significance by $t - test$

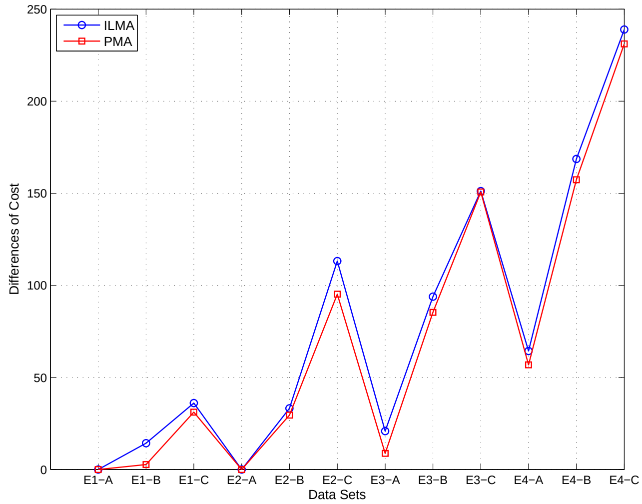


Fig. 4. Comparison of average cost between ILMA and PMA on E1-A to E4-C

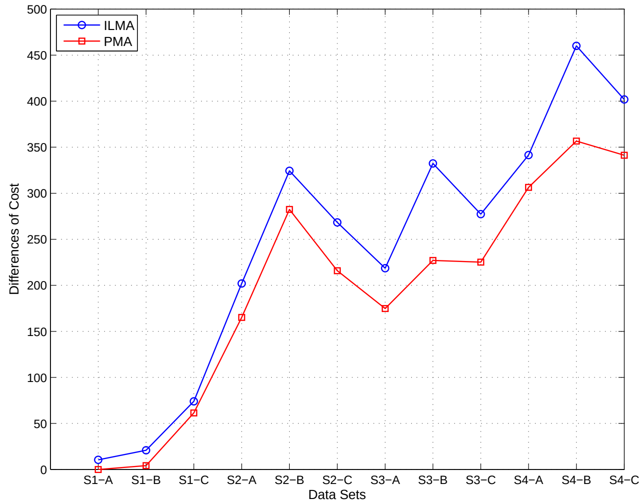


Fig. 5. Comparison of average cost between ILMA and PMA on S1-A to S4-C

The current work represents an initial effort to design formal memetic framework for solving discrete problems. In the future, we hope to pursue further research on suitable distance metrics that can take into account the topology of the search space induced by the variation operator used. In particular, through landscape analysis, the distribution of optima, basins of attractions and volume of the search space may be statistically estimated.

## Acknowledgment


This project is supported by the Singapore National Research Foundation Interactive Digital Media R&D Program under research Grant NRF2008IDM-IDM004-037. The authors would also like to acknowledge the Center for Computational Intelligence at the School of Computer Engineering, Nanyang Technological University for their technical support, the reviewers for their constructive comments on the earlier version of this paper as well as Yi Mei of USTC China, for providing the original source code of ILMA.

## REFERENCES


[1] M. Dror, *Arc routing. Theory, solutions and applications*. Boston: Kluwer Academic Publishers, 2000.

[2] P. Lacomme, C. Prins, and W. Ramdane-Chérif, "Competitive memetic algorithms for arc routing problem," *Annals of Operational Research*, vol. 141, no. 1–4, pp. 159–185, 2004.

[3] Y. Mei, K. Tang, and X. Yao, "Improved memetic algorithm for capacitated arc routing problem," *IEEE Congress on Evolutionary Computation*, pp. 1699–1706, 2009.

[4] K. Tang, Y. Mei, and X. Yao, "Memetic algorithm with extended neighborhood search for capacitated arc routing problems," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1159–1166, 2009.

[5] Q. H. Nguyen, Y. S. Ong, and M. H. Lim, "A probabilistic memetic framework," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 604–623, 2009.

[6] B. Golden and R. Wong, "Capacitated arc routing problems," *Networks*, vol. 11, no. 3, pp. 305–315, 1981.

[7] Y. S. Ong and A. J. Keane, "Meta-lamarckian learning in memetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 2, pp. 99–110, 2004.

[8] A. Torn and A. Zilinskas, "Global optimization," *Lecture Notes in Computer Science*, vol. 350, 1989.

[9] C. Houck, J. Joines, and M. Kay, "Utilizing lamarckian evolution and the baldwin effect in hybrid genetic algorithms," *NCSU-IE Technical Report 96-01, Meta-Heuristic Research and Applications Group, Department of Industrial Engineering, North Carolina State University*, 1996.

[10] A. Vicini and D. Quagliarella, "Airfoil and wing design using hybrid optimization strategies," *American Institute of Aeronautics and Astronautics Journal*, vol. 37, no. 5, pp. 634–641, 1999.

[11] Y. S. Ong, P. B. Nair, and A. J. Keane, "Evolutionary optimization of computationally expensive problems via surrogate modeling," *American Institute of Aeronautics and Astronautics Journal*, vol. 41, no. 4, pp. 687–696, 2003.

[12] M. N. Le, Y. S. Ong, Y. C. Jin, and B. Sendhoff, "Lamarckian memetic algorithms: local optimum and connectivity structure analysis," *Memetic Computing*, vol. 1, no. 3, pp. 175–190, 2009.

[13] M. G. Gong, C. Liu, L. C. Jiao, and G. Cheng, "Hybrid immune algorithm with lamarckian local search for multi-objective optimization," *Memetic Computing*, vol. 2, no. 1, pp. 47–67, 2009.

[14] C. K. Ting, C. F. Ko, and C. H. Huang, "Selecting survivors in genetic algorithm using tabu search strategies," *Memetic Computing*, vol. 1, no. 3, pp. 191–203, 2009.

[15] J. Shinkyu, S. Hasegawa, K. Shimoyama, and S. Obayashi, "Development and investigation of efficient ga/pso-hybrid algorithm applicable to real-world design optimization," *IEEE Computational Intelligence Magazine*, vol. 4, no. 3, pp. 36–44, 2009.

[16] A. Sattar and R. Seguier, "Hmoam: hybrid multi-objective genetic optimization for facial analysis by appearance model," *Memetic Computing*, vol. 2, no. 1, pp. 25–46, 2010.

[17] Y. S. Ong, N. Krasnogor, and H. Ishibuchi, "Special issue on memetic algorithm," *IEEE Transactions on Systems, Man and Cybernetics - Part B*, vol. 37, no. 1, pp. 2–5, 2007.

[18] M. H. Lim and Y. L. Xu, "Application of hybrid genetic algorithm in supply chain management," *International Journal of Computers, Systems, and Signals*, vol. 6, no. 1, 2005.

[19] J. E. Smith, "Co-evolving memetic algorithms: a review and progress report," *IEEE Transactions on Systems, Man and Cybernetics - Part B*, vol. 37, no. 1, pp. 6–17, 2007.

[20] B. Liu, L. Wang, and Y. H. Jin, "An effective pso-based memetic algorithm for flow shop scheduling," *IEEE Transactions on Systems, Man and Cybernetics - Part B*, vol. 37, no. 1, pp. 18–27, 2007.

[21] A. Caponio, G. L. Cascella, F. Neri, N. Salvatore, and M. Sumne, "A fast adaptive memetic algorithm for online and offline control design of pmsm drives," *IEEE Transactions on Systems, Man and Cybernetics - Part B*, vol. 37, no. 1, pp. 28–41, 2007.

[22] D. Liu, K. C. Tan, C. K. Goh, and W. K. Ho, "A multiobjective memetic algorithm based on particle swarm optimization," *IEEE Transactions on Systems, Man and Cybernetics - Part B*, vol. 37, no. 1, pp. 42–50, 2007.

[23] S. M. K. Hasan, R. Sarker, D. Essam, and D. Cornforth, "Memetic algorithms for solving job-shop scheduling problems," *Memetic Computing*, vol. 1, no. 1, pp. 69–83, 2008.

[24] M. Tang and X. Yao, "A memetic algorithm for vlsi floorplanning," *IEEE Transactions on Systems, Man and Cybernetics - Part B*, vol. 37, no. 1, pp. 62–69, 2007.

[25] J. Tang, M. H. Lim, and Y. S. Ong, "Parallel memetic algorithm with delective local search for large scale quadratic assignment problems," *International Journal of Innovative Computing, Information and Control*, vol. 2, no. 6, pp. 1399–1416, 2006.

[26] ——, "Diversity-adaptive parallel memetic algorithm for solving large scale combinatorial optimization problems," *Soft Computing Journal*, vol. 11, no. 9, pp. 873–888, 2007.

[27] Z. Zhu, Y. S. Ong, and M. Dash, "Wrapper-filter feature selection algorithm using a memetic framework," *IEEE Transactions on Systems, Man and Cybernetics - Part B*, vol. 37, no. 1, pp. 70–76, 2007.

[28] 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: Cybernetics*, vol. 36, no. 1, pp. 141–152, 2006.

[29] M. Lozano, F. Herrera, N. Krasnogor, and D. Molina, "Real-coded memetic algorithms with crossover hill-climbing," *Evolutionary Computation*, vol. 12, no. 3, pp. 273–302, 2004.

[30] W. E. Hart, "Adaptive global optimization with local search," *PhD thesis, University of California, San Diego*, 1994.

[31] 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*, vol. 4, no. 1, pp. 31–42, 2000.

[32] M. W. S. Land, "Evolutionary algorithms with local search for combinatorial optimization," *Ph. D. Thesis, University of California, San Diego*, 1998.

[33] D. E. Goldberg and S. Voessner, "Optimizing global-local search hybrids," *The Genetic and Evolutionary Computation Conference*, vol. 1, pp. 220–228, 1999.

[34] G. Kendall, P. Cowling, and E. Soubeiga, "Choice function and rrandom hyperheuristics," *Fourth Asia-Pacific Conference on Simulated Evolution and Learning*, pp. 667–671, 2002.

[35] K. Sörensen, "Distance measures based on the edit distance for permutation-type representations," *J. Heuristics*, vol. 13, no. 1, pp. 35–47, 2007.

[36] S. Ronald, "More distance functions for order-based encodings," *IEEE Congress on Evolutionary Computation*, pp. 558–563, 1998.

[37] A. G. Najera and J. A. Bullinaria, "Bi-objective optimization for the vehicle routing problem with time windows: Using route similarity to enhance performance," *Proceedings of the 5th International Conference on Evolutionary Multi-Criterion Optimization*, pp. 275–289, 2009.

[38] R. W. Eglese, "Routing winter gritting vehicles," *Discrete Applied Mathematics*, vol. 48, no. 3, pp. 231–244, 1994.

[39] R. W. Eglese and L. Y. O. Li, "A tabu search based heuristic for arc routing with a capacity constraint and time deadline," *in Metaheuristics: theory and applications, I. H. Osman and J. P. Kelly, Eds. Boston: Kluwer Academic Publishers*, pp. 633–650, 1996.

[40] L. Y. O. Li and R. W. Eglese, "An interactive algorithm for vehicle routing for winter-gritting," *Journal of the Operational Research Society*, vol. 47, no. 2, pp. 217–228, 1996.