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 servicing 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.

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 learning1 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 I: Given a depot node \( v_d \in V \), a travel circuit \( C \) starting and ending at \( v_d \) is considered valid if and only if the total load \( \sum_{e_i \in C} d(e_i) \leq W \), 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:

\[
\text{cost}(C) = \sum_{e_i \in C_s} c_s(e_i) + \sum_{e_i \in C_t} c_t(e_i)
\]

(1)

where \( C_s \) and \( C_t \) are edge sets that required servicing and those that do not, respectively.

Definition II: A set of travel circuits \( S = \{C_i\}, i = 1 \ldots k \) is a valid solution to the CARP if and only if:

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

1In this paper, the term lifetime learning is often used interchangeably with local search.
The objective of CARP is then to find a valid solution $S$ that minimizes the total cost:

$$C_S = \sum_{\forall C_i \in S} cost(C_i)$$  \hspace{1cm} (2)

Fig. 1. An example of CARP

The example of a CARP is illustrated in Fig. 1, with $v_0$ 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 $C_1 = \{0, 4, 2, 0\}, C_2 = \{0, 5, 7, 0\},$ and $C_3 = \{0, 9, 11, 0\}$ 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 $S = \{C_1, C_2, C_3\}$ 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 $f_{sl}$: defines how often should local learning be applied. $f_{sl}$ 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, $f_{sl}$ can be replaced with the local search probability, $P_{sl}$, which defines the probability at which each individual in the population should undergo local learning.

2) Local search intensity, $t_{sl}$: defines how much computational budget should be allocated to each local learning process. $t_{sl}$ may be represented in terms of number of the objective function evaluations or time budget.

3) Subset of solution undergoing local search, $\Omega_{sl}$: 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:

$$t_{upper} = \frac{t_{0} \ln(1 - P_{1}^{(k)})}{n \ln(1 - P_{2}^{(k)})}$$  \hspace{1cm} (3)

where $t_{0}$ 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:

\[
t_{\text{upper}} = \frac{p_2^{(k)}}{p_1^{(k)}}
\]

(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.

\[
\text{Dis}(X, X_A) = \text{Dis}(X_B, X_A)
\]

(9)

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:

\[
J(A, B) = \frac{|A \cap B|}{|A \cup B|}
\]

(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:

\[
\text{Dis}(A, B) = |A \cup B| - |A \cap B|
\]

(6)

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:

\[
p_1 = \frac{\text{Dis}(X_B, X_A)}{\text{volume of search space}}
\]

(7)

Probability \( p_2 \), on the other hand, is derived as:

\[
p_2 = \frac{\text{Dis}(X, X_A)}{\text{volume of search space}}
\]

(8)

Subsequently, the upper bound is then derived as:

\[
t_{\text{upper}} = \frac{p_2}{p_1} = \frac{\text{Dis}(X, X_A)}{\text{Dis}(X_B, X_A)}
\]

Fig. 2. A depiction on the estimation of \( t_{\text{upper}} \) in the probabilistic memetic framework for non-linear programming.
Further, the upper bound is considered only if the current solution $X$ is sufficiently close to its nearest neighbor, that is:

$$\text{Dis}(X, X_{\text{nbr}}) \leq \alpha \times \text{Dis}(X_{\text{nbr}}, X_A)$$ (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 is sufficiently close to its nearest neighbor, that is:

$$\text{Dis}(X, X_{\text{nbr}}) \leq \alpha \times \text{Dis}(X_{\text{nbr}}, X_A)$$ (10)

The expected local search intensity on the other hand is:

$$t_{\text{expected}} = \text{number of steps from } X_{\text{nbr}} \text{ to } X_A.$$ 3. if $t_{\text{expected}} \leq t_{\text{upper}}$, local search will take place; otherwise, proceed with global exploration

End If

End

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 theoretical 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.

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population Size</td>
<td>30</td>
</tr>
<tr>
<td>Maximum number of restart</td>
<td>20</td>
</tr>
<tr>
<td>Local search probability (ILMA)</td>
<td>0.1 in the 1st restart and 0.2 subsequently</td>
</tr>
<tr>
<td>Local search probability (PMA)</td>
<td>0.1 in the 1st restart and adaptive subsequently</td>
</tr>
<tr>
<td>$\alpha$ in PMA</td>
<td>1</td>
</tr>
<tr>
<td>maximum solution length</td>
<td>250</td>
</tr>
<tr>
<td>Independent runs</td>
<td>30</td>
</tr>
</tbody>
</table>
Algorithm 2 Outline of ILMA equipped with the proposed probabilistic framework

Begin:
  Initialisation: 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

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 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|”, “|V_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.

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 Compu-

\[ t \] value of 29 degree of freedom is significant at a 0.05 level of significance by \( t = t_{0.05} \).
tional 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


