Memetic Algorithm with Heuristic Candidate List Strategy for Capacitated Arc Routing Problem

Haobo Fu, Yi Mei, Ke Tang and Yanbo Zhu

Abstract— **Capacitated Arc Routing Problem (CARP) has drawn much attention during the last few years because of its applications in the real world. Recently, we developed a Memetic Algorithm with Extended Neighborhood Search (MAENS), which is powerful in solving CARP. The excellent performance of MAENS is mainly due to one of its local search operators, namely the Merge-Split (MS) operator. However, the higher computational complexity of the MS operator compared to traditional local search operators remains as the major drawback of MAENS, especially when applying it to large-size instances. In this paper, we propose a heuristic candidate list strategy to sample the neighbors generated by the MS operator instead of enumerating or sampling them randomly, in order to avoid unnecessary callings of the MS operator during local search. Based on the strategy, an improved algorithm of MAENS, namely MAENS-II, is developed. Experimental results on benchmark instances showed that MAENS-II managed to obtain the same level of solution quality as MAENS with much less computational time. This should be credited to the utilization of the proposed heuristic strategy. On the other hand, in case both MAENS and MAENS-II were provided comparable computational time, MAENS-II outperformed MAENS in terms of solution quality.**

I. Introduction

The arc routing problem is a classic combinatorial optimization problem with various applications in the real world, such as road maintenances, garbage collection, mail delivery, school bus routing and meter reading [1], [2]. As one of the most classical forms of the arc routing problem, the Capacitated Arc Routing Problem (CARP) is investigated in this paper. CARP can be described as follows: consider a directed connected graph $G = (V, A, E)$, with the vertex set V, the arc (directed edge) set A and the edge set E. A fleet of identical vehicles, each with capacity Q, are based at a depot vertex $dep \in V$. Each arc or edge incurs a nonnegative traveling cost and a nonnegative demand: the edges or arcs with positive demands are called tasks, and those with zero demands are called non-tasks. A solution to CARP is a routing plan for the vehicles serving all tasks, and the objective is to minimize the total traveling cost under the following constraints:

- Each route starts and ends at the depot vertex dep.
- Each task is served exactly in one route.
- The total demands accumulated by each task in one route should not exceed the vehicle's capacity Q.

Since CARP is NP-hard, which was demonstrated in [3], exact algorithms are suitable only for small-size instances. Therefore, a lot of heuristic and meta-heuristic algorithms have been developed to solve CARP: [4], [5], [6], [7], [8], [9], [10].

Recently, we proposed a Memetic Algorithm with Extended Neighborhood Search (MAENS) [11] and demonstrated that it outperformed other approaches in terms of solution quality, especially for the large-size instances. Embedded in the general framework of memetic algorithms, the superior performance of MAENS should be credited to its local search operators. The local search process in MAENS is conducted in three main steps. First, a local optimum is identified in the neighborhoods defined by three traditional local search operators: single insertion, double insertion and swap. Then, the Merge-Split (MS) operator, which defines an extended-size neighborhood, is applied to the solution obtained in the first step. Finally, the first step is executed again on the solution obtained in the second step.

As demonstrated in [11], the MS operator is capable of escaping from local optima induced by traditional local search operators by merging several routes (a route group) into an unordered list of tasks and then splitting the unordered list. Although this strengthens the search ability of the algorithm, the MS operator is associated with high computational time complexity and it would be too time-consuming to enumerate all neighbors generated by MS. As a result, MAENS simply employs a random candidate list strategy to examine MS neighbors. However, this random candidate list strategy can be inefficient because the MS search potential of different route groups may vary largely. In this paper, we propose a heuristic candidate list strategy for the MS operator. A metric is first developed for this strategy, and then based on the metric, the route groups are sorted in terms of their MS search potential. Finally, incorporating the heuristic candidate list strategy into MAENS leads to an improved MAENS (MAENS-II). Experimental studies demonstrated the advantages of MAENS-II over MAENS from two aspects. First, MAENS-II managed to achieve the same solution quality with much less computational time. Second, the quality of the solutions obtained by MAENS-II is better than that of MAENS if both algorithms were provided comparable computational time.

The rest of the paper is organized as follows: the background is introduced in Section II, including the formal definition of CARP and the brief description of MAENS.

Haobo Fu, Ke Tang and Yi Mei are with the Nature Inspired Computation and Applications Laboratory, the Department of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China (emails: fuhaobo@mail.ustc.edu.cn, meiyi@mail.ustc.edu.cn, ketang@ustc.edu.cn).
Yanbo Zhu is with the School of Electronic and Information Engineering, Beihang University, Beijing, 100083, China.
Corresponding author: Ke Tang.

978-1-4244-8126-2/10/$26.00 ©2010 IEEE

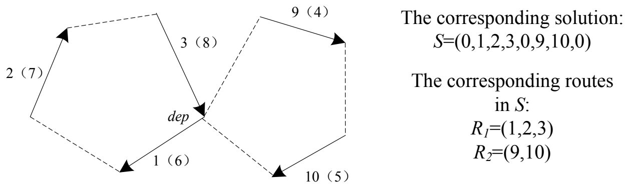

Fig. 1. An example of a CARP solution, numbers in brackets are the invs of the corresponding arcs

```
Begin
    Initialize a population P of solutions.
    Evaluate the population P.
      Repeat
        Set an intermediate population P1=P.
        Crossover (P1), resulting intermediate population P2.
        Localsearch (P2, pls), resulting intermediate population P3.
        Evaluate the population P3 ∪ P.
        Use stochastic ranking to sort solutions in P3 ∪ P.
        Set P = {the first pSize solutions in P3 ∪ P}.
      Until stopping criterion is met
End
Return the best feasible solution S ever encountered.
pls : the probability of executing local search.
pSize: the number of individuals in parent population.
```

Fig. 2. MAENS pseudo-code

In Section III, we describe the improved MAENS in details, such as the mathematical definition of *task distance within one route*, *task distance between two routes* and *route group distance*, and the pseudo-code of the improved MS operator. Then, experimental studies of the improved MAENS are presented in Section IV. Finally, conclusions and future work are discussed in Section V.

II. Background

A. Capacitated Arc Routing Problem

Subject to the three constraints mentioned in Section I, CARP is about determining a set of routes for vehicles while all edge tasks, $E_R \subseteq E$, and arc tasks, $A_R \subseteq A$, in a given graph $G = (V, E, A)$ (V: the vertex set, A: the arc set, E: the edge set) are to be served. In our method, we assign one ID for each arc task and two IDs for each edge task (An edge (v_i, v_j) is viewed as two arcs: $< v_i, v_j >$ and $< v_j, v_i >$). If an arc induced by an edge is assigned an ID, say t, $inv(t)$ is used to indicate the ID for the other arc which is induced by the same edge. Besides, the IDs are set to positive integers for the sake of convenience, except that 0 is set to represent a special task–dummy task. Both the head and the tail of the dummy task are the depot vertex dep. Besides, the demand and the traveling cost of the dummy task are zeros. Finally, each ID is associated with four features:

$\forall t, t \in$ the set of IDs

- $head(t)$: the head vertex of the corresponding arc.
- $tail(t)$: the tail vertex of the corresponding arc.
- $tc(t)$: the traveling cost of the corresponding arc.
- $demand(t)$: the demand of the corresponding arc.

Using such notations, we can represent a solution to CARP as an ordered list of IDs (Solution S is denoted as $S = (t_1, t_2, t_3, ..., t_n)$. t_i stands for an ID). Fig. 1 gives a simple illustration for such a solution representation.

Because of the solution representation, we can calculate the total cost of a solution S ($S = (t_1, t_2, t_3, ...t_n)$), $TC(S)$, this way:

$$TC(S) = \sum_{k=1}^{n} tc(t_k) + \sum_{k=1}^{n-1} sp(t_k, t_{k+1})$$

where n is the length of solution S ($length(S) = n$), and $sp(t_k, t_{k+1})$ denotes the traveling cost of the shortest path from $head(t_k)$ to $tail(t_{k+1})$.

In order to obtain the route plan for vehicles, we rewrite solution S containing m routes in its route form: $S = (0, R_1, 0, R_2, 0, R_3, ..., R_m, 0)$, where R_i represents the ith route, and R_i is a sequence of tasks ($R_i = (t_{i1}, t_{i2}, t_{i3}, ...)$, $t_{ij} \in$ the set of IDs excluding dummy task ID 0).

Finally, with such solution representation and three constraints mentioned in Section I, CARP can be stated as follows:

$$\begin{aligned} \min \quad & TC(S) = \sum_{k=1}^{length(S)} tc(t_k) + \sum_{k=1}^{length(S)-1} sp(t_k, t_{k+1}) \\ s.t.: \quad & \forall\ task\ t_k\ acc(t_k) = 1,\ if\ t_k \in A_R \\ & \forall\ task\ t_k\ acc(t_k) + acc(inv(t_k)) = 1,\ if\ t_k \in E_R \\ & \forall\ route\ R_i \sum_{k=1}^{length(R_i)} demand(t_{ik}) \leq Q \end{aligned}$$

where $acc(t)$ represents the times that task t appears in solution S.

B. Memetic Algorithm with Extended Neighborhood Search

Embedded in the general framework of memetic algorithms, MAENS first initializes a population of solutions. At each generation, crossover and local search are used to generate the offspring population. Then, stochastic ranking [12] is applied to sort individuals in the combined population of parents and offsprings, resulting the next parent population from which the $pSize$ best individuals are selected. A fitness function, which is defined as the weighted sum of solution's total cost and solution's total violation with an adaptive penalty parameter, is employed to guide the local search and the stochastic ranking process. Besides, identical individuals are not allowed to exist in the same population through the whole process of MAENS. The pseudo-code of MAENS is provided in Fig. 2, and interested readers can refer to [11] for details. The components of MAENS are briefly introduced below:

Initialization: At most $pSize$ non-clone solutions are constructed in the initial population because whenever

Input: Solution S.

Output: Solution S^*.

Step1: $S^* := S$.

(1) Conduct Single Insertion, Double Insertion and Swap local search on S^*. Find the best solution S_1, S_2 and S_3 in the corresponding neighborhoods $N_1(S^*)$, $N_2(S^*)$ and $N_3(S^*)$.

(2) Set S_4 = the best solution in set $\{S_1, S_2, S_3\}$.

If $(f(S_4) < f(S^*))$

$S^* := S_4$.

End If

Step2: flag := false.

Conduct the Merge-Split local search on S^*. Find the best solution S_{ms} in the corresponding neighborhood $N_{ms}(S^*)$.

If $(f(S_{ms}) < f(S^*))$

$S^* := S_{ms}$.

flag := true.

End If

Step3: If (flag == true)

Conduct Step1 again.

End If

Terminate, output solution S^*.

f: the fitness function used in MAENS.

Fig. 3. The local search process in MAENS

Input: Solution S: containing n routes, p: a parameter controlling what proportion of all possible route groups should be examined by the MS operator ($0 < p < 1$).

Output: Solution S^*.

Merge: (1) Randomly select 100*p (p is set to be a constant value (p=1.0) in the original MAENS) route groups (G_1, G_2, G_3, …) from solution S (all possible route groups will be examined if $C_n^{\text{merNum}} <= 100*p$).

(2) Merge these 100*p route groups to form 100*p unordered lists of tasks respectively.

Split: (1) Employ path-scanning heuristic with five rules and Ulusoy's splitting algorithm to sort and then split these 100*p unordered lists. Thus, five new route groups are obtained for each unordered list and the best route group of the five is reserved. Therefore, 100*p (G_1*, G_2*, G_3*, …) new route groups are produced.

(2) Choose G_{best}*. Such G_{best}* results in best solution quality improvement, if it replaces the corresponding old route group G_{best} in solution S. Ties are broken arbitrarily if several new route groups satisfy such condition.

(3) Replace the corresponding old route group G_{best} in solution S with the new route group G_{best}*, producing the final output solution S^* of the MS operator.

Fig. 4. The detailed steps of the MS operator

$maxTrial$ tries fail to produce a non-clone solution from solutions already generated, the initialization process ends. A solution is initialized this way: an empty route is first constructed. Then, the construction iteratively connects the end of current route to the closest tasks whose demands do not violate the vehicle's capacity if added to the current route's demand (ties are broken arbitrarily if more than one task satisfies the condition). If no task satisfies the condition, the current route is closed, and a new empty route is generated. Once all tasks are served, the initialization of one solution ends.

Crossover: Two parent solutions are selected randomly from the current parent population to generate one offspring solution by the sequence based crossover (SBX) operator. Given solution S_1 and solution S_2, SBX first randomly selects two routes R_1 and R_2 from S_1 and S_2 respectively. Then R_1 and R_2 are broken into two sub-routes at random points respectively, i.e. $R_1 = (R_{11}, R_{12})$ and $R_2 = (R_{21}, R_{22})$. The result offspring solution S_3 is obtained by replacing R_{12} in solution S_1 with R_{22} in solution S_2. Finally, tasks duplicated are deleted, and tasks missed are inserted randomly. For SBX's details, readers can refer to [13].

Local search: The local search process in MAENS involves two kinds of local search operators: the small-step operators (single insertion, double insertion and swap) and the large-step operator (the Merge-Split operator). We illustrate the local search process in Fig. 3. The first and last step of the local search is conducted in the neighborhoods induced by the small-step local search operators, while the second step is conducted in the neighborhood induced by the MS operator. The MS operator works as follows:

The local search process of the MS operator is constituted of two sub-processes, i.e. the Merge sub-process and the Split sub-process, sequentially. The pseudo-code of the MS operator is illustrated in Fig.4. Given a solution S with L routes ($S = (0, R_1, 0, R_2, 0, R_3, ..., R_L, 0)$, R_i is the ith route), the MS operator on solution S can be described as follows:

- Merge: In this step, MS randomly selects $merNum$ routes (a route group) from solution S and combines them into an unordered list of tasks. With regard to the solution $S = (0, R_1, 0, R_2, 0, R_3, ..., R_L, 0)$, there are C_L^{merNum} possible ways to select route groups.
- Split: In this step, firstly, MS sorts the unordered list of tasks obtained in the Merge step using a path-scanning heuristic with five different rules [4]. Therefore, five ordered lists of tasks are generated for every unordered list. Secondly, Ulusoy's splitting algorithm [5] is used to split the five ordered lists into new routes, respectively. Finally, five new solutions are produced by replacing the old selected route group with the five new route groups respectively, and the best solution is selected as the output of the MS operator.

III. Memetic Algorithm with Heuristic Candidate List Strategy

As demonstrated in [11], the outstanding performance of MAENS is largely due to the utilization of the MS operator. Unlike traditional local search operators, which are thought to be only capable of searching neighborhoods of small size, the MS operator alters solutions significantly and often results in extraordinarily different solutions. However, because of the high computational complexity of the MS operator compared to traditional local search operators, it is too time-consuming to Merge-Split all route groups when the current solution contains a large number of routes. In MAENS, 100 route groups are randomly selected for the MS operator if the size of all possible route groups is larger than 100. In this way, all route groups share the same probability to be examined by the MS operator. In fact, this is not efficient since different route groups possess different MS search potential. Since MS employs path-scanning heuristic to sort tasks, tasks which are close to each other are likely to appear in the same route, and it might be useless to call for the MS operator on route groups in which routes are distinctively separated from each other in the graph. Aware of that, we develop a metric in this paper to measure the MS search potential of route groups. Based on the metric, an improved algorithm of MAENS is proposed.

A. The Metric for the Merge-Split Search Potential

As discussed above, one kind of route groups (*entangled route groups*), in which task distances within routes are larger compared to task distances between routes, are more likely to result in solution quality improvements through the MS operator than another kind of route groups (*separated route groups*), in which task distances within routes are smaller compared to task distances between routes. As a result, the metric which aims at ranking route groups' MS search potential should distinguish *entangled route groups* from *separated route groups* appropriately. With the consideration of characteristics of *entangled route groups* and *separated route groups* mentioned above, the metric which we developed for ranking route groups' MS search potential measures the rate of average task distances between routes to average task distances within routes.

We name the metric for ranking route groups' MS search potential as *route group distance* (RD). First, we define the *task distance within one route* (T) and the *task distance between two routes* (D): considering route R_u, the *task distance within one route* is:

$$T_u = \frac{1}{2|R_u|^2 - 2|R_u|} \sum_{v_i, v_i^*} d_{v_i v_i^*}$$

Considering route R_u and route R_v, the *task distance between two routes* is:

$$D_{uv} = \frac{1}{2|R_u| * 2|R_v|} \sum_{v_i, v_j} d_{v_i v_j}$$

- v_i and v_i^*: vertex induced by the tasks in route R_u. Besides, when calculating $d_{v_i v_i^*}$, v_i and v_i^* do not belong to the same task.
- v_j: vertex induced by the tasks in route R_v.
- $d_{v_i v_j}$: the shortest distance from vertex v_i to vertex v_j in the graph.
- $|R_i|$: the number of tasks in route R_i.

We define the *route group distance* based on the definition of *task distance within one route* and the *task distance between two routes*. Here, we only consider route groups which own exactly two routes, since $merNum$ has value 2 in the original MAENS. The *route group distance* of the route group containing only route R_u and route R_v is:

$$RD_{uv} = \frac{D_{uv}}{T_u} * \frac{D_{uv}}{T_v}$$

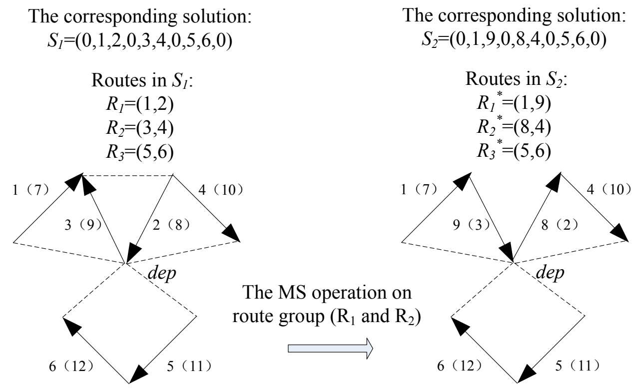

Fig. 5. Illustration of *entangled route group* and *separated route group*. Numbers in brackets are the invs of the corresponding arcs

With regard to the definition of the metric, *entangled route groups* tend to have smaller *route group distance* than *separated route groups*, and the MS operator will be focused on *entangled route groups*. A simple illustration is displayed in Fig. 5. In this example, every arc or edge incurs 1 unit traveling cost, and every task incurs 1 unit demand with vehicle's capacity of 2 units. The *route group distance* RD_{12}, RD_{13} and RD_{23} in solution S_1 are $4/9$, $25/6$ and $25/6$, respectively. Therefore, route group ($R1$ and $R2$) is selected for the MS operator, if only one route group can be examined by the MS operator. The result solution S_2, which cannot be found if route group ($R1$ and $R3$) or route group ($R2$ and $R3$) was selected for the MS operator, is the global optimum solution with total cost of 10 units.

B. Merge-Split Operator with Heuristic Candidate List Strategy

After the definition of *route group distance*, we modify the MS operator by ranking all MS neighbors according to the metric, which can be viewed as a heuristic candidate list strategy. We give the pseudo-code for the modified MS operator in Fig. 6.

The newly added portion for the MS operator is printed in boldface in Fig. 6 and the improved MAENS is obtained by combining the modified MS operator with all other components of the original MAENS. Since the resulting algorithm is basically an improved version of MAENS, it is named MAENS-II.

We notice that the time complexity of calculating the *route group distance* of all possible route groups in one solution S is O(n^2) (n is the number of routes in solution S), when $merNum$ takes value 2. Thus, we can expect that the calculation of *route group distance* might not induce too much additional computational time to MAENS.

IV. Experimental Studies

To evaluate the efficacy of the heuristic candidate list strategy, we compared the performance of MAENS-II to that of MAENS. All parameter settings and program realizations are identical for both MAENS-II and MAENS, except that

Input: Solution S: containing n routes, p: a parameter controlling what proportion of all possible route groups should be examined by the MS operator ($0 < p < 1$).

Output: Solution S^*.

The calculation of *route group distance*:

(1) Calculate the *route group distances* of all C_n^{merNum} route groups in solution S.

(2) Rank these route groups in a candidate list according to their *route group distances* (the smaller a route group's *route group distance,* the higher its ranking).

Merge: **(1) Select 100*p route groups (G_1, G_2, G_3, …) which rank in the front of the candidate list (all possible route groups would be examined if $C_n^{\text{merNum}} <= 100 *p$).**

(2)Merge these 100*p route groups to form 100*p unordered lists of tasks respectively.

Split: (1) Employ path-scanning heuristic with five rules and Ulusoy's splitting algorithm to sort and then split these 100*p unordered lists. Thus, five new route groups are obtained for each unordered list and the best route group of the five is reserved. Therefore, 100*p (G_1*, G_2*, G_3*, …) new route groups are produced.

(2) Choose G_{best}*. Such G_{best}* results in best solution quality improvement, if it replaces the corresponding old route group G_{best} in solution S. Ties are broken arbitrarily if several new route groups satisfy such condition.

(3) Replace the corresponding old route group G_{best} in solution S with the new route group G_{best}*, producing the final output solution S^* of the MS operator.

Fig. 6. The detailed steps of the modified MS operator

MAENS-II executes the heuristic candidate list strategy in the MS operator, while MAENS does not.

A. Experimental Setup

Computational experiments were carried out on three sets of benchmark CARP instances. The first set, named gdb, contains 23 instances, which were generated by DeArmon [14]. The second set, named val, consists of 34 instances based on 10 different graphs from Benavent et al [15]. The final set, named egl, was generated by Eglese [16] containing 24 instances.

The parameter settings are exhibited in Table I. The settings are the same as that in MAENS, except that one more parameter p is added. The parameter p is used to control what proportion of all possible route groups should be examined by the MS operator. By introducing the parameter p, we can test the performance of MAENS-II against that of MAENS when different numbers of route groups are examined by the MS operator. With consideration of both simplicity and comprehensiveness of the experimental studies, we set p to six different values ($p = 0.1, 0.3, 0.5, 0.7, 0.9, 1.0$). For the practical use of p, readers can refer to Fig.4 and Fig.6. All these experiments were executed 30 independent runs for both MAENS-II and MAENS.

B. Comparing MAENS-II with MAENS

Since we have already compared MAENS to a number of the state-of-the-art algorithms in [11], we demonstrate the efficacy of MAENS-II by only comparing it to MAENS. We present a brief description of the contents in Tables II to VI as follows:

- The columns headed "|R|", "LB", "BK" and "best" denote the number of tasks, the lower bounds found so far, the best known results obtained so far and the best results of MAENS-II at different p values during 30 independent runs, respectively. The column headed "MAENS" lists results which were produced by MAENS, while the "MAENS-II" contains results obtained with MAENS-II. p is the parameter which controls what proportion of all possible route groups should be examined by the MS operator.
- The lower bounds (LB) were collected from [15], [17], [18] and [19].
- Table II, Table III and Table IV present the average costs of the solutions obtained by MAENS-II and MAENS in the 30 runs. MAENS-II is compared to MAENS under six different levels settings of p: 0.1, 0.3, 0.5, 0.7, 0.9 and 1.0. Besides, two rows are added in the bottom: the row headed "mean" indicates the average of the corresponding column; the row headed "win" counts the number of wins between "MAENS" and "MAENS-II" under the same value of p ("MAENS" wins against "MAENS-II" only if the average solution cost in "MAENS" is lower than that in "MAENS-II", and vice versa). Two-sided Wilcoxon's rank sum test, performed at the 0.05 significance level, was used to compare the results of 30 runs in MAENS to those in MAENS-II and the results in boldface are those which are significantly better than the others. The best solutions obtained by MAENS-II, which are no worse than the best known results, are presented in boldface, and new best results are denoted in boldface with *.
- Table V provides the Average Percentage Deviation (APD) from the lower bounds for both MAENS-II and MAENS on the three benchmark sets.
- Table VI presents the average runtime of MAENS and MAENS-II. MAENS and MAENS-II are coded in c language, and the only difference between MAENS-II and MAENS about the realization is the heuristic candidate list strategy in MAENS-II. Besides, all experiments were run on an Intel(R) Xeon(R) E5506 @2.13GHz CPU.

From the "mean" rows and the "win" rows in Table II, Table III and Table IV, we can see that MAENS-II performs better than MAENS. In the benchmark set gdb, MAENS-II performs as well as MAENS: they both reached the lower bounds in almost 30 runs, which might be due to the fact that the solution space of gdb instances are relatively small and it might be easy to reach the lower bounds even without the Merge-Split operator. In the benchmark set val, the advantage of heuristic candidate list strategy becomes more obvious than that in gdb as can easily be seen in the "win" rows in Table III. This difference might be explained by the fact that instances in val possess more tasks, and thus have larger solution spaces than instances in gdb. Finally,

TABLE I
PARAMETER SETTINGS OF MAENS AND MAENS-II

NAME	EXPLANATION	VALUE
$pSize$	parent population size	30
$opSize$	offspring population size	180
p_{ls}	probability of executing local search	0.2
$maxTrial$	max number of trials for generating non-clone initial individuals	50
$maxGen$	max number of generations	500
$merNum$	number of routes merged in the MS operator	2
p	proportion of all possible route groups which are examined by the MS operator	0.1, 0.3, 0.5, 0.7, 0.9 and 1.0

we can see that MAENS-II's advantage over MAENS was reflected most clearly in the benchmark set egl. This can be demonstrated by the "win" rows in Table IV and the results obtained with two-sided Wilcoxon's rank sum test. There are 2 pair results, 6 pair results and 36 pair results in which MAENS-II is significantly better than MAENS in gdb, val and egl, respectively, though there exist 1 pair in val and 1 pair in egl in which MAENS-II is significantly worse than MAENS. Table V further confirms the advantage of MAENS-II.

Furthermore, we can see that, in Table II, Table III and Table IV, 14 new best results (3 in val, another 11 in egl) were obtained. Additionally, the advantage of MAENS-II over MAENS can be demonstrated most when combing Table V and Table VI. We interpret this in the following two respects.

- When using comparable computational time, i.e., under the same value of p, the solution quality of MAENS-II is better than that of MAENS. We can check this by comparing MAENS-II to MAENS in Table V with regards to the corresponding computational time in Table VI.
- The efficacy of MAENS-II can also be demonstrated by comparing the computational time required to achieve a given level of solution quality. For instance, when p takes value 0.3, MAENS-II achieved APDs of 0.09, 0.27 and 1.03 on the three benchmark sets. On the other hand, when p takes value 1.0, the APDs obtained by MAENS are 0.04, 0.36 and 1.09, respectively. In the former case, the runtime of MAENS-II were 9.93, 73.69 and 570.45 seconds on the three benchmark sets. In the latter case, MAENS spent 11.32, 105.35 and 821.62 seconds on the three benchmark sets, respectively. This showed that MAENS-II managed to achieve better solutions than MAENS with less computational time.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we investigated a powerful memetic algorithm for CARP, named MAENS, which is capable of escaping from local optima induced by traditional local search operators because of the use of MS operator in MAENS. One major drawback of MAENS is its high computational time largely caused by the MS operator. In the original MAENS, the MS operator simply employs a random candidate list strategy to examine MS neighbors, without consideration that some neighbors may result in little improvements in solution quality. Hence there might exists a waste of computational time, and probably some promising route groups may have no chance to undergo local search using the MS operator.

In order to improve the efficacy of the MS operator and to enable the MS operator to focus more on promising route groups, we developed a heuristic candidate list strategy. In this strategy, route groups are ranked based on a newly defined metric, called *route group distance*. Route groups that possess smaller *route group distances* are considered to be more promising and, thus, receive higher priority to be examined by the MS operator. As a result, the search process is more efficient in the improved algorithm, which we call it MAENS-II.

Experimental results showed that the heuristic candidate list strategy developed in this paper significantly increases the efficacy of MAENS. Either the same solution quality can be maintained in much less time with MAENS-II or better solutions can be discovered using comparable computational time. Further, 14 new best results were obtained among all the 81 instances. Since the only difference between MAENS-II and MAENS is the incorporation of the heuristic candidate list strategy, the superior performance of MAENS-II evidenced the efficacy of the proposed candidate list strategy.

Though we limited the metric based on route groups containing exactly two routes, the metric can be extended easily for route groups consisted of more than two routes by selecting two representative routes to calculate the *route group distance*. In near future, we will extend the heuristic candidate list strategy to situations where more than two routes are selected for the MS operator.

ACKNOWLEDGMENT

This work is partially supported by the National High Technology Development 863 Program of China (Grant No. 2006AA12A103) and the National Natural Science Foundation of China (Grant No. 60802036). The authors are grateful to Dr. Thomas Weise for proofreading the paper.

REFERENCES

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

[2] H. Handa, L. Chapman, and X. Yao, "Robust route optimization for gritting/salting trucks: a CERCIA experience," *IEEE Computational Intelligence Magazine*, vol. 1, no. 1, pp. 6–9, 2006.

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

[4] B. Golden, J. DeArmon, and E. Baker, "Computational experiments with algorithms for a class of routing problems," *Computers and Operations Research*, vol. 10, no. 1, pp. 47–59, 1983.

TABLE II

AVERAGE SOLUTION COSTS OVER 30 RUNS ON THE GDB BENCHMARK SET FOR BOTH MAENS AND MAENS-II

					p=0.1		p=0.3		p=0.5		p=0.7		p=0.9		p=1.0	
NAME	$\lvert R \rvert$	LB	BK	best	MAENS	MAENS-II	MAENS	MAENS-II	MAENS	MAENS-II	MAENS	MAENS-II	MAENS	MAENS-II	MAENS	MAENS-II
gdb1	22	316	316	**316**	316	316	316	316	316	316	316	316	316	316	316	316
gdb2	26	339	339	**339**	339	339	339	339	339	339	339	339	339	339	339	339
gdb3	22	275	275	**275**	275	275	275	275	275	275	275	275	275	275	275	275
gdb4	19	287	287	**287**	287	287	287	287	287	287	287	287	287	287	287	287
gdb5	26	377	377	**377**	377	377	377	377	377	377	377	377	377	377	377	377
gdb6	22	298	298	**298**	298	298	298	298	298	298	298	298	298	298	298	298
gdb7	22	325	325	**325**	325	325	325	325	325	325	325	325	325	325	325	325
gdb8	46	348	348	**348**	349.8	**349.1**	349.2	349.3	349.5	349.1	349	349	348.9	348.9	348.7	349.1
gdb9	51	303	303	**303**	307.1	**304.8**	304.4	305.0	304.4	304.2	303.8	303.7	303.9	304.1	304.0	304.1
gdb10	25	275	275	**275**	275	275	275	275	275	275	275	275	275	275	275	275
gdb11	45	395	395	**395**	395	395	395	395	395	395	395	395	395	395	395	395
gdb12	23	458	458	**458**	458	458	458	458	458	458	458	458	458	458	458	458
gdb13	28	536	536	**536**	540.5	540.3	540.2	539.9	539.5	538.1	539	538.5	538.6	538.1	537.5	538.4
gdb14	21	100	100	**100**	100	100	100	100	100	100	100	100	100	100	100	100
gdb15	21	58	58	**58**	58	58	58	58	58	58	58	58	58	58	58	58
gdb16	28	127	127	**127**	127	127	127	127	127	127	127	127	127	127	127	127
gdb17	28	91	91	**91**	91	91	91	91	91	91	91	91	91	91	91	91
gdb18	36	164	164	**164**	164	164	164	164	164	164	164	164	164	164	164	164
gdb19	11	55	55	**55**	55	55	55	55	55	55	55	55	55	55	55	55
gdb20	22	121	121	**121**	121	121	121	121	121	121	121	121	121	121	121	121
gdb21	33	156	156	**156**	156	156	156	156	156	156	156	156	156	156	156	156
gdb22	44	200	200	**200**	200	200	200	200	200	200	200	200	200	200	200	200
gdb23	55	233	233	**233**	234.1	234.3	233.8	233.8	233.3	233.3	233.7	233.5	233.3	233.3	233.3	233.4
Mean	29.4	253.8	253.8	253.8	254.3	254.2	254.1	254.1	254.1	254.0	254.0	254.0	254.0	254.0	253.9	254
win	-	-	-	-	1	3	2	1	0	3	0	3	1	1	4	0

TABLE III

AVERAGE SOLUTION COSTS OVER 30 RUNS ON THE VAL BENCHMARK SET FOR BOTH MAENS AND MAENS-II

					p=0.1		p=0.3		p=0.5		p=0.7		p=0.9		p=1.0	
NAME	$\lvert R \rvert$	LB	BK	best	MAENS	MAENS-II	MAENS	MAENS-II	MAENS	MAENS-II	MAENS	MAENS-II	MAENS	MAENS-II	MAENS	MAENS-II
1A	39	173	173	**173**	173	173	173	173	173	173	173	173	173	173	173	173
1B	39	173	173	**173**	173	173	173	173	173	173	173	173	173	173	173	173
1C	39	245	245	**245**	245	245	245	245	245	245	245	245	245	245	245	245
2A	34	227	227	**227**	227	227	227	227	227	227	227	227	227	227	227	227
2B	34	259	259	**259**	259	259	259	259	259	259	259	259	259	259	259	259
2C	34	457	457	**457**	457.3	457.2	457.4	457	457	457	457.2	457	457	457.2	457	457
3A	35	81	81	**81**	81	81	81	81	81	81	81	81	81	81	81	81
3B	35	87	87	**87**	87	87	87	87	87	87	87	87	87	87	87	87
3C	35	138	138	**138**	138	138	138.0	138	138	138	138	138	138	138	138	138
4A	69	400	400	**400**	400	400	400.1	400.1	400	400	400	400	400	400	400	400
4B	69	412	412	**412**	412	412	412	412	412	412	412	412	412	412	412	412
4C	69	428	428	**428**	431.1	430.5	430.7	429.3	429.3	430.1	429.7	428.7	429.7	428.8	429.4	429.9
4D	69	526	530	**530**	535.6	535	534.8	533.7	535.0	533.3	533.5	533.5	533	532.8	533.2	533.1
5A	65	423	423	**423**	423	423	423	423	423	423	423	423	423	423	423	423
5B	65	446	446	**446**	446	446	446	446	446	446	446	446	446	446	446	446
5C	65	473	474	**474**	474	474	474	474	474	474	474	474	474	474	474	474
5D	65	573	577	**575***	586.7	587.6	585.4	584.6	584.7	583.5	583.1	584.5	584.6	583.7	583.8	583.6
6A	50	223	223	**223**	223	223	223	223	223	223	223	223	223	223	223	223
6B	50	233	233	**233**	233	233	233	233	233	233	233	233	233	233	233	233
6C	50	317	317	**317**	317.6	**317**	317.1	317	317	317	317	317	317.1	317.1	317	317
7A	66	279	279	**279**	279	279	279	279	279	279	279	279	279	279	279	279
7B	66	283	283	**283**	283	283	283	283	283	283	283	283	283	283	283	283
7C	66	334	334	**334**	334.2	334.1	334.1	334	334.1	334.1	334.1	334.0	334	334.0	334.1	334
8A	63	386	386	**386**	386	386	386	386	386	386	386	386	386	386	386	386
8B	63	395	395	**395**	395	395	395	395	395	395	395	395	395	395	395	395
8C	63	518	521	**521**	531.0	528.8	529.1	**527.1**	526.8	526.5	526.7	526.7	525.9	527.3	527	526.6
9A	92	323	323	**323**	323.3	323.3	323.3	323.3	323.2	323.1	323	323.0	323.0	323.0	323	323
9B	92	326	326	**326**	326.1	326	326	326.0	326	326	326	326	326	326	326	326
9C	92	332	332	**332**	332.2	332.3	332	332.1	332.1	332	332.1	332	332	332	332	332
9D	92	385	391	**390***	393.2	392.4	393.0	**391.7**	392.5	**391.5**	392.0	391.8	391.8	391.4	391.8	391.9
10A	97	428	428	**428**	**428**	428.2	428	428.1	428	428	428	428	428	428	428	428
10B	97	436	436	**436**	436.7	436.9	436.6	436.7	436.3	436.1	436.2	436.1	436.1	436.2	436.2	**436.0**
10C	97	446	446	**446**	446.4	446.4	446.2	446.3	446.1	446.1	446.1	446	446.2	446.0	446	446.0
10D	97	525	528	**526***	533.4	533.1	532.4	531.5	532.3	**531.0**	532.1	531.5	531.8	531.6	531.8	531.5
mean	63.3	343.8	344.4	344.3	345.6	345.5	345.4	345.2	345.2	345.1	345.1	345.1	345.1	345.1	345.1	345.1
win	-	-	-	-	4	9	4	9	1	8	1	8	3	6	2	6

TABLE IV

AVERAGE SOLUTION COSTS OVER 30 RUNS ON THE EGL BENCHMARK SET FOR BOTH MAENS AND MAENS-II

					p=0.1		p=0.3		p=0.5		p=0.7		p=0.9		p=1.0	
NAME	$\|R\|$	LB	BK	best	MAENS	MAENS-II	MAENS	MAENS-II	MAENS	MAENS-II	MAENS	MAENS-II	MAENS	MAENS-II	MAENS	MAENS-II
E1-A	51	3548	3548	**3548**	3548	3548	3548	3548	3548	3548	3548	3548	3548	3548	3548	3548
E1-B	51	4498	4498	**4498**	4511.1	4505.7	4504.0	4501.4	4507.4	4506.1	4503.4	4505.2	4505.1	4507	4503.1	4504.3
E1-C	51	5566	5595	**5595**	5609.7	5605.8	5608	5607	5604.7	5604.3	5603.9	5606.2	5603.4	5603.4	5603.8	5604.4
E2-A	72	5018	5018	**5018**	5018	5018	5018	5018	5018	5018	5018	5018	5018	5018	5018	5018
E2-B	72	6305	6317	**6317**	6336.6	6332.6	6334.4	6336.4	6330.1	6332.0	6331.8	6332.9	6337.1	6335.8	6331	6330.9
E2-C	72	8243	8335	**8335**	8360.0	8348.7	8354.2	8339.8	8345.2	8343.6	8346.2	8338.4	8337.8	8342.2	8345.8	8340.4
E3-A	87	5898	5898	**5898**	5907.4	5902.8	5898.9	5899.7	5898.9	5899.7	5898.9	5899.7	5898	5898	5898.5	5898
E3-B	87	7704	7775	**7775**	7808.8	**7792.2**	7798.4	7788.0	7795.0	7786.2	7792.8	7788.2	7786.9	7789.6	7798.8	7790.7
E3-C	87	10163	10292	**10292**	10328.4	10329.6	10326.4	10319.2	10324.6	10324.2	10325.1	10323.6	10325.8	10316.1	10321.4	10320.3
E4-A	98	6408	6456	**6444***	6474.4	6468.2	6470.1	6467.0	6465.0	6465.9	6464.5	6464.6	6465.8	**6462.4**	6469.2	6465.7
E4-B	98	8884	8998	**8962***	9034.0	9020.9	9020.7	9017.7	9024.5	**9009.3**	9022.5	9012.3	9015.6	9021.5	9014.2	9013.7
E4-C	98	11427	11561	**11550***	11683.6	11667.2	11688.9	**11648.2**	11671.7	11667.9	11669.6	11651	11654.2	11645.5	11652.7	11641.9
S1-A	75	5018	5018	**5018**	5020.57	5018	5018	5018	5018	5018	5018	5018	5018	5018	5018	5018
S1-B	75	6384	6388	**6388**	6411.7	6408.7	6416.2	6404.4	6405.0	6397.6	6405.8	6398.9	6411.7	6413.7	**6403.9**	6413.9
S1-C	75	8493	8518	**8518**	8526.9	8526	8520.2	8524.7	8518.7	8518	8518	8518	8518	8518	8518	8518
S2-A	147	9824	9895	**9889***	10021.8	**9978.4**	9995.1	**9953.7**	9997.0	**9964.2**	9957.2	9967.3	9965.2	9955.3	9974.4	9971.2
S2-B	147	12968	13147	**13101***	13273	**13236.5**	13246.8	**13204.5**	13253.9	13225.6	13235.5	**13202.4**	13234.4	13226	13229.5	**13199.4**
S2-C	147	16353	16430	**16430**	16595.3	**16545.6**	16560.2	16530.8	16568.1	**16517.2**	16574	**16513.1**	16547.9	16518.3	16567.1	**16528.5**
S3-A	159	10143	10257	**10227***	10334.8	**10308.3**	10314.5	10303.3	10316.1	**10290.8**	10305.6	10306.1	10296	10291	10304.9	10297.9
S3-B	159	13616	13749	**13695***	13901.1	**13830.7**	13882.6	**13813.9**	13859.3	13826.9	13832.4	13833.5	13857.2	13827.1	13846.1	**13803.2**
S3-C	159	17100	17207	**17194***	17391.6	**17334**	17381.8	**17301.3**	17365.7	**17312.6**	17370	**17307.5**	17357.1	**17314.7**	17341.4	**17311**
S4-A	190	12143	12341	**12297***	12464.9	**12434.5**	12437.9	12422.4	12420	12415.6	12423.8	**12396.2**	12419.3	12398.9	12414.9	12411
S4-B	190	16093	16337	**16283***	16472.1	**16436.5**	16459.8	**16426.5**	16464.1	**16429.1**	16442.5	**16404.5**	16439.2	16427	16449.4	**16402.9**
S4-C	190	20375	20538	**20521***	20804.8	20787.7	20757.5	20732.8	20799.3	**20699**	20775.7	**20699.6**	20763.5	**20683.1**	20773.4	20735
mean	109.9	9673.8	9754.8	9741.4	9826.6	9807.7	9815.0	9796.9	9813.3	9796.7	9807.6	9793.9	9805.1	9794.9	9806.1	9795.3
win	-	-	-	-	1	21	3	18	3	18	8	12	5	13	3	17

TABLE V

AVERAGE APD OVER 30 RUNS ON THE BENCHMARK SETS FOR BOTH MAENS AND MAENS-II

	p=0.1		p=0.3		p=0.5		p=0.7		p=0.9		p=1.0	
NAME	MAENS	MAENS-II	MAENS	MAENS-II	MAENS	MAENS-II	MAENS	MAENS-II	MAENS	MAENS-II	MAENS	MAENS-II
gdb	0.13	0.10	0.08	0.09	0.07	0.05	0.06	0.05	0.05	0.05	0.04	0.06
val	0.51	0.33	0.45	0.27	0.40	0.25	0.37	0.25	0.36	0.25	0.36	0.25
egl	1.28	1.12	1.18	1.03	1.15	1.03	1.10	1.01	1.09	1.02	1.09	1.02

TABLE VI

AVERAGE RUNTIMES (IN CPU SECONDS) OVER 30 RUNS ON THE BENCHMARK SETS FOR BOTH MAENS AND MAENS-II

	p=0.1		p=0.3		p=0.5		p=0.7		p=0.9		p=1.0	
NAME	MAENS	MAENS-II	MAENS	MAENS-II	MAENS	MAENS-II	MAENS	MAENS-II	MAENS	MAENS-II	MAENS	MAENS-II
gdb	9.72	8.97	10.09	9.93	9.61	9.48	11.75	11.96	12.93	12.88	11.32	14.55
val	62.72	66.51	68.22	73.69	78.08	80.25	85.80	86.70	99.64	100.41	105.35	101.06
egl	434.30	478.28	498.15	570.45	582.39	664.58	673.07	757.35	769.50	845.50	821.62	892.89

[5] G. Ulusoy, “The fleet size and mix problem for capacitated arc routing,” *European Journal of Operational Research*, vol. 22, no. 3, pp. 329–337, 1985.

[6] A. Hertz, G. Laporte, and M. Mittaz, “A tabu search heuristic for the capacitated arc routing problem,” *Operations Research*, vol. 48, no. 1, pp. 129–135, 2000.

[7] A. Hertz and M. Mittaz, “A variable neighborhood descent algorithm for the undirected capacitated arc routing problem,” *Transportation Science*, vol. 35, no. 4, pp. 425–434, 2001.

[8] P. Lacomme, C. Prins, and W. Ramdane-Chérif, “Competitive memetic algorithms for arc routing problem,” *Annals of Operational Research*, vol. 131, no. 1-4, pp. 159–185, 2004.

[9] J. Brandao and R. Eglese, “A deterministic tabu search algorithm for the capacitated arc routing problem,” *Computers and Operations Research*, vol. 35, no. 4, pp. 1112–1126, 2008.

[10] Y. Mei, K. Tang, and X. Yao, “A Global Repair Operator for Capacitated Arc Routing Problem,” *IEEE Transactions on System, Man and Cybernetics, Part B*, vol. 39, no. 3, pp. 723–734, 2009.

[11] 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. 1151–1166, 2009.

[12] T. P. Runarsson and X. Yao, “Stochastic ranking for constrained evolutionary optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284–294, 2000.

[13] J. Potvin and S. Bengio, “The vehicle routing problem with time windows part II: genetic search,” *INFORMS journal on Computing*, vol. 8, no. 2, p. 165, 1996.

[14] J. S. DeArmon, “A comparison of heuristics for the capacitated Chinese postman problem,” Master’s thesis, University of Maryland, College Park, MD, 1981.

[15] E. Benavent, V. Campos, A. Corberän, and E. Mota, “The capacitated arc routing problem: lower bounds,” *Networks*, vol. 22, no. 7, pp. 669–690, 1992.

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

[17] D. Ahr, “Contributions to multiple postmen problems,” Ph.D. dissertation, Rupercht-Karls-Universitat, Heidelberg, Germany, 2004.

[18] R. Baldacci and V. Maniezzo, “Exact methods based on node-routing formulations for undirected arc-routing problems,” *Networks*, vol. 47, no. 1, pp. 52–60, 2006.

[19] H. Longo, D. A. M. Poggi, and E. Uchoa, “Solving capacitated arc routing problems using a transformation to the CVRP.” *Computers and Operations Research*, vol. 33, no. 6, pp. 1823–1837, 2006.