An Evolutionary Approach to the Multidepot Capacitated Arc Routing Problem

Article *in* IEEE Transactions on Evolutionary Computation · July 2010

DOI: 10.1109/TEVC.2009.2033578 · Source: IEEE Xplore

Lining Xing
National University of Defense Technology
26 PUBLICATIONS **88** CITATIONS
SEE PROFILE

Ying-Wu Chen
National University of Defense Technology
154 PUBLICATIONS **1,149** CITATIONS
SEE PROFILE

Xin Yao
Tianjin University
555 PUBLICATIONS **22,639** CITATIONS
SEE PROFILE

An Evolutionary Approach to the Multidepot Capacitated Arc Routing Problem

Lining Xing, Philipp Rohlfshagen, Yingwu Chen, and Xin Yao, *IEEE Fellow*

***Abstract*—The capacitated arc routing problem (CARP) is a challenging vehicle routing problem with numerous real world applications. In this paper, an extended version of CARP, the multidepot capacitated arc routing problem (MCARP), is presented to tackle practical requirements. Existing CARP heuristics are extended to cope with MCARP and are integrated into a novel evolutionary framework: the initial population is constructed either by random generation, the extended random path-scanning heuristic, or the extended random Ulusoy's heuristic. Subsequently, multiple distinct operators are employed to perform selection, crossover, and mutation. Finally, the partial replacement procedure is implemented to maintain population diversity. The proposed evolutionary approach (EA) is primarily characterized by the exploitation of attributes found in near-optimal MCARP solutions that are obtained throughout the execution of the algorithm. Two techniques are employed toward this end: the performance information of an operator is applied to select from a range of operators for selection, crossover, and mutation. Furthermore, the arc assignment priority information is employed to determine promising positions along the genome for operations of crossover and mutation. The EA is evaluated on 107 instances with up to 140 nodes and 380 arcs. The experimental results suggest that the integrated evolutionary framework significantly outperforms these individual extended heuristics.**

***Index Terms*—Capacitated arc routing problem, combinatorial optimization, evolutionary algorithms, time-limited service.**

I. Introduction

THE CAPACITATED arc routing problem (CARP) deals with undirected networks where each edge has a predefined traversal cost and some edges are required to be traversed (serviced) by some vehicle. There is a single depot which contains a fleet of identical vehicles, each with limited capacity. The objective of CARP is to determine a set of feasible vehicle trips of minimum total cost, such that every trip starts and ends at the depot, the total demand covered by each trip must not exceed the vehicle's capacity, and each required edge is serviced by a single trip.

The CARP has raised a growing interest in the last two decades because of important applications. As mentioned in the literature, CARP arises naturally in various industrial settings such as the planning of mail delivery [1] or school bus services [2], the routing of street sweepers [3], waste collection vehicles [4], gritting trucks [5] or snow plows [6], and the inspection of gas pipelines [7], oil pipelines [8] or electric power lines [9]. In order to make this paper more concrete and without loss of generality, examples are inspired by winter gritting.

To the best of our knowledge, Lacomme *et al.* [10] proposed the most recent version of CARP, the extended CARP (ECARP), in 2004. The authors considered the following four extensions: 1) mixed multigraph with different links (edges, arcs, and parallel links); 2) two distinct costs per link (deadheading and collecting); 3) prohibited turns and turn penalties; and 4) maximum trip length (an upper limit on the cost of trips). Here, deadheading means traveling without service. The number of vehicles is not specified and the problem definition potentially allows for an infinite number of vehicles as long as the single objective value is optimized [10]. We believe this to be impractical and chose to impose a limit on this number. This change, among multiple additional modifications, leads to the multidepot capacitated arc routing problem (MCARP) which is the focus of this paper.

Similar to ECARP, MCARP is modeled as a mixed multigraph with different links. Two distinct costs are predefined for each link, prohibited turns, and turn penalties are considered, no demand may exceed the vehicle's capacity, all vehicles are homogeneous, and no split gritting is allowed. MCARP differs from ECARP as it considers the *maximum service time* as well as *multiple depots*. In marginal winter climates, precautionary gritting ought to be finished by the highway management departments before the morning traffic [11]. For this reason, the maximum service time should be considered to guarantee successful service. As mentioned in [10], the maximum trip length is predefined for any trip, whilst the cost of every trip comprises collecting costs and deadheading

Manuscript received January 22, 2009, revised April 23, 2009 and July 15, 2009. This work was supported by the China Scholarship Council Grant No. 2 007 103 861, the National Natural Science Foundation of China Grant Nos. 70 971 131, 70 703 036, 70 601 035, 70 801 062, and 70 901 074, and the Engineering and Physical Sciences Research Council Grant No. EP-E058884-1 for "Evolutionary Algorithms for Dynamic Optimization Problems: Design, Analysis, and Applications."

L. Xing is with the College of Information Systems and Management, National University of Defense Technology, Changsha 410073, China (e-mail: xinglining@gmail.com).

P. Rohlfshagen is with the Center of Excellence for Research in Computational Intelligence and Applications, School of Computer Science, University of Birmingham, Birmingham B15 2TT, U.K. (e-mail: p.rohlfshagen@cs.bham.ac.uk).

Y. Chen is with the Department of Management Science and Engineering, College of Information Systems and Management, National University of Defense Technology, Changsha 410073, China (e-mail: chen.nudt@gmail.com).

X. Yao is with the Center of Excellence for Research in Computational Intelligence and Applications (CERCIA), School of Computer Science, University of Birmingham, Birmingham B15 2TT, U.K. (e-mail: x.yao@cs.bham.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2009.2033578

costs. It is difficult to convert this mixed cost to service time. In MCARP, service time is predefined simply as the vehicle travel time (this is equivalent to the total deadheading cost of trips). This restriction on the service time places hard constraints on the problem: if the road network is large or the service time is small, some required roads cannot be serviced because of their long distance from the depot. In order to guarantee the availability of services, the extension to multiple depots is taken into account in MCARP.

The additional considerations introduce further complexities to an already complex problem. For instance, the introduction of multiple depots significantly enlarges the solution space: suppose the number of vehicles is a, then there are a trips in most CARP solutions [10]. Further suppose the number of optional solutions of the ith trip to be Q_i, and there are generally $\prod_{i=1}^{a} Q_i$ solutions in the solution space of CARP/ECARP. If there are b depots, each trip can start from any of the depots, so the number of solutions of the ith trip is bQ_i, then there are normally $b^a(\prod_{i=1}^{a} Q_i)$ solutions in the solution space of MCARP. In total, the solution space of ECARP is identical to that of CARP whereas the solution space of MCARP is about b^a times than that of CARP. Furthermore, the complicated constraints of MCARP render a large number of potential solutions infeasible; the violation of any of the numerous constraints (such as, the vehicle's capacity, the number of vehicles, or the maximum service time) would result in an infeasible solution. Finally, it should be noted that MCARP is NP-hard. This follows from the fact that the basic CARP is NP-hard [10] and that it is possible to reduce MCARP to the basic CARP.

Unfortunately, good quality solutions to MCARP cannot be found efficiently by existing algorithms. In this paper, we examine the effectiveness of existing classical heuristics as well as metaheuristics. First, three classical heuristics, originally developed for CARP, were extended to MCARP. For some of the MCARP instances considered in this paper, these extended heuristics cannot guarantee to find a feasible solution. For example, one of the heuristics, extended augment–merge, finds a feasible solution for only 15 out of 87 MCARP instances considered. Furthermore, the significant size of the infeasible region of the search space makes it difficult to find high-quality solutions by using existing metaheuristics directly. In particular, using a standard genetic algorithm (GA) on MCARP, it proved rather difficult to generate feasible individuals (solutions) as the random initialization used by the starting population and the poor quality of the initial population would subsequently influence the performance of the GA. Therefore, a novel evolutionary approach (EA) is constructed in this research: some classical heuristics are integrated into a canonical evolutionary framework and two distinct kinds of heuristic information are learned from the near-optimal MCARP solutions that are obtained throughout the execution of the algorithm. This information is subsequently employed to guide the evolutionary process.

The remainder of this paper is organized as follows. Section II reviews some recent work related to CARP. Section III presents the formulation of MCARP and several constructive heuristics are proposed for MCARP in Section IV. Section V introduces the proposed EA in detail. The performance of the proposed methods is evaluated in Section VI, and some concluding remarks and directions for future work are summarized in Section VII.

II. Literature Review

Some representative variations of the CARP may be summarized as follows [12]: vehicle routing problem (VRP) [13], Chinese postman problem (CPP) [14], rural postman problem (RPP) [15], stochastic arc routing problem [16], capacitated Chinese postman problem [17], traveling salesman problem [18], general routing problem [19], and periodic arc routing problem [20]. The performance of classical heuristics and metaheuristics is generally evaluated by lower bounds (LBs). LBs for CARP are usually estimated by: matching LB [12], node scanning LB [17], node duplication LB [21], matching node scanning LB [22], multiple cuts node duplication LB (MCNDLB) [23], and linear programming formulations [24]. MCNDLB performs at least as well as the existing bound for CARP [23].

Exact methods for CARP such as branch-and-bound [25] are still limited to 20–30 edges [10]. The large instances need to be tackled by heuristics like augment–merge [12], path-scanning [26], construct-and-strike [27], Ulusoy's tour splitting method [28], and augment–insert [29]. Final experimental results suggest that heuristics produce fairly good solutions. Nevertheless, better solutions can be obtained by metaheuristics, i.e., tabu search [30–32], variable neighborhood descent method [33], guided local search [34], EA [20], [35], memetic algorithms (MA) [10], [36], simulated annealing [37], GA [38], and ant colony optimization [39].

Hierarchical and integrated approaches are different ways to solve CARP instances. In hierarchical approaches, cluster and route are treated separately [40]. A route first cluster second algorithm [40] builds an Euler tour over all arcs with positive demands. If the obtained tour is one connected graph, then the CPP algorithm is applied to produce an optimal set of vehicle routes. Otherwise, the RPP algorithm is applied to produce the final solution. In a cluster first route second algorithm [41], arcs are clustered into groups by considering the weight constraint and routes calculated using the CPP algorithm. The integrated approach was applied by considering cluster and route at the same time. Branda and Eglese [31] constructed the deterministic tabu search algorithm and obtained excellent solutions in reasonable computational time. Beullens *et al.* [34] developed a guided local search and achieved some high-quality solutions in the limited computational time. Finally, Lacomme *et al.* [38] constructed the multiobjective GA and got excellent performance both in terms of solution quality and computational efficiency.

In terms of multiple depots, many scholars have paid more attention to multidepot vehicle routing problems (MDVRP), a problem formulated by Sumichrast and Markham [42] in 1995. A number of heuristic approaches were developed to deal with the MDVRP [43–47]. Nevertheless, contrary to the VRP, in which products are delivered to client nodes in the network, the CARP consists of visiting a subset of edges (arcs), and

TABLE I

GLOSSARY OF MATHEMATICAL SYMBOLS USED IN THIS PAPER

Symbol	Meaning
$G_U = (V, E)$	Undirected connected graph.
$G_D = (V, A)$	Directed connected graph.
V	Set of N_V nodes.
E	Set of N_E edges.
A	Set of N_A arcs.
R	Set of N_R required links $R \subseteq A$ ($R \subseteq E$).
$d(u)$	Deadheading (traveling without gritting) cost of link u.
$s(u)$	Additional service (gritting) cost of required link u.
$q(u)$	Gritting demand of required link u.
$pen(u, v)$	Penalty occurs from arc u to arc v.
N_1	Number of depots.
N_2	Number of trucks.
Q	Capacity of trucks.
S	Average speed of trucks.
T	Maximal service time.
C	Maximal trip length.
α	Deadheading cost conversion coefficient.
L	Set of vehicle routes $L = \{l_1, l_2, \ldots, l_{N_2}\}$.
l_j	Vehicle route $l_j = \{\mu_{1j}, \mu_{2j}, \ldots, \mu_{(m_j)j}\}$.
m_j	Number of links in route l_j.
μ_{ij}	ith link in vehicle route l_j.
$f_j(\mu_{ij})$	Service indicator of link μ_{ij}.
$D(l_j)$	Total deadheading cost of route l_j.
$S(l_j)$	Total service cost of route l_j.
$P(l_j)$	Total penalty of route l_j.

hence the heuristic algorithm proposed to the MDVRP cannot be directly applied to deal with the MCARP. Research on MCARP (or similar extensions) is considerably less. Amberg *et al.* [49] proposed a tabu search using capacitated trees to solve the multiple center CARP. However, in their MCARP model, only a simplified cost is predefined to every edge (arc), prohibited turns and turn penalties are not considered, and the maximum service time is not taken into account either.

This brief review suggests that existing approaches cannot be employed directly to deal with MCARP. It is thus necessary to develop novel approaches to tackle MCARP and this motivates the work presented in this paper.

III. PROBLEM FORMULATION OF MCARP

The previous sections of this paper informally described CARP, ECARP, and MCARP. In this section, we will formally describe each problem and elaborate further on the attributes unique to MCARP. We have listed the attributes of different problems in itemized form to allow a direct comparison among these three problem variants. Table I provides the summary of symbols used in this paper.

The classical CARP is characterized as follows.

1) Inputs
   a) An undirected connected graph is $G = (V, E)$.
   b) Each arc $u \in E$ incurs a cost $d(u)$.
   c) Each arc $u \in R$ has a demand $q(u)$.
   d) One depot is available.
   e) N_2 identical trucks are available.
   f) The capacity of trucks is Q.
2) Outputs
   a) An optimal set of vehicle routes is L.
   b) Service indicators are $f_j(\mu_{ij})$.
3) Objectives

$$\min \sum_{j=1}^{N_2} D\left(l_j\right). \tag{1}$$

4) Constraints
   a) Start and return the same depot.
   b) Demand constraints.
   c) No partial service.

The ECARP, an extended version of CARP, has the following attributes.

1) Inputs
   a) A directed connected graph is $G = (V, A)$.
   b) Each arc $u \in A$ incurs a deadheading cost $d(u)$.
   c) Each arc $u \in R$ has an additional service cost $s(u)$.
   d) Each arc $u \in R$ has a demand $q(u)$.
   e) $pen(u, v)$ occurs from arc $u \in A$ to arc $v \in A$.
   f) One depot is available.
   g) N_2 identical trucks are available.
   h) The capacity of trucks is Q.
   i) The maximum trip length is C.
2) Outputs
   a) An optimal set of vehicle routes is L.
   b) Service indicators are $f_j(\mu_{ij})$.
3) Objectives

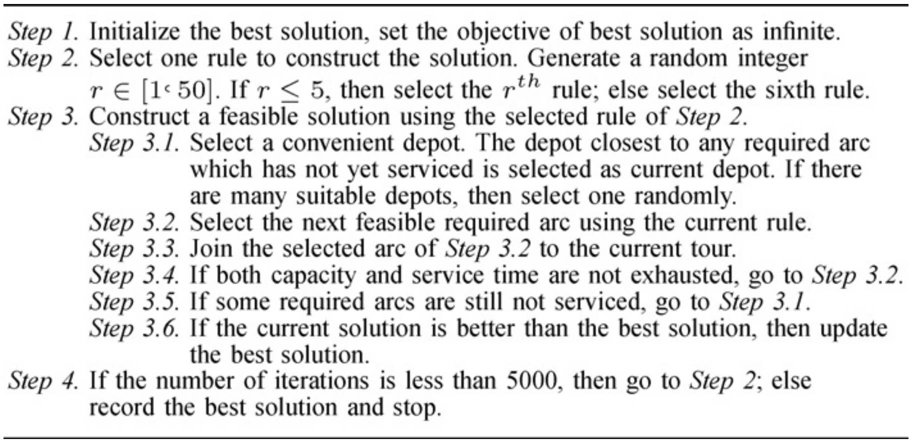
$$\min \sum_{j=1}^{N_2} \left\{ D\left(l_j\right) + S\left(l_j\right) + P\left(l_j\right) \right\}. \tag{2}$$

4) Constraints
   a) Start and return the same depot.
   b) Demand constraints.
   c) No partial service.
   d) Trip length constraints.

Finally, MCARP, which further extends ECARP, may be summarized as follows.

1) Inputs
   a) A directed connected graph is $G = (V, A)$.
   b) Each arc $u \in A$ incurs a deadheading cost $d(u)$.
   c) Each arc $u \in R$ has an additional service cost $s(u)$.
   d) Each arc $u \in R$ has a demand $q(u)$.
   e) $pen(u, v)$ occurs from arc $u \in A$ to arc $v \in A$.
   f) N_1 identical depots are available.
   g) N_2 identical trucks are available.
   h) The capacity of trucks is Q.
   i) The average speed of trucks is S.
   j) The maximal service time is T.
   k) Deadheading cost conversion coefficient α.
2) Outputs
   a) An optimal set of vehicle routes is L.
   b) Service indicators are $f_j(\mu_{ij})$.
   c) There is a start depot of each vehicle route.
3) Objectives

$$\min \sum_{j=1}^{N_2} \left\{ D\left(l_j\right) + S\left(l_j\right) + P\left(l_j\right) \right\}. \tag{3}$$



```
Step 1. Initialize the best solution, set the objective of best solution as infinite.
Step 2. Select one rule to construct the solution. Generate a random integer
        $r \in$ [1‹ 50]. If $r \leq 5$, then select the $r^{th}$ rule; else select the sixth rule.
Step 3. Construct a feasible solution using the selected rule of Step 2.
        Step 3.1. Select a convenient depot. The depot closest to any required arc
                  which has not yet serviced is selected as current depot. If there
                  are many suitable depots, then select one randomly.
        Step 3.2. Select the next feasible required arc using the current rule.
        Step 3.3. Join the selected arc of Step 3.2 to the current tour.
        Step 3.4. If both capacity and service time are not exhausted, go to Step 3.2.
        Step 3.5. If some required arcs are still not serviced, go to Step 3.1.
        Step 3.6. If the current solution is better than the best solution, then update
                  the best solution.
Step 4. If the number of iterations is less than 5000, then go to Step 2; else
        record the best solution and stop.
```

Fig. 1. Pseudocode of extended random path-scanning heuristic.

4) Constraints
   a) Start and return the same depot.
   b) Demand constraints.
   c) No partial service.
   d) Service time constraints.

It should be noted that if each truck has to return to the nearest available depot, then it will produce another interesting variant of the MCARP. After some small modifications (only the split procedure needs to be modified), our methods could still cope with this new variant. This novel variant will be farther studied in the future research. Additional MCARP central concepts are further elaborated and summarized in Appendix A.

In MCARP, there are multiple different costs for each vehicle route. The total deadheading (i.e., traversal without service) cost $D(l_j)$ is computed as

$$D(l_j) = \sum_{i=1}^{m_j} d(\mu_{ij}). \tag{4}$$

The total service cost $S(l_j)$ is determined as

$$S(l_j) = \sum_{i=1}^{m_j} s(\mu_{ij}) f_j(\mu_{ij}) \tag{5}$$

and finally, the total penalty $P(l_j)$ is

$$P(l_j) = \sum_{i=1}^{m_j - 1} pen(\mu_{ij}, \mu_{(i+1)j}). \tag{6}$$

There are five kinds of constraints considered in this paper. These are as follows.

1) Any route must start and return the same depot.
2) The total demand processed by each trip must not exceed the capacity Q. For every route l_j

$$\sum_{i=1}^{m_j} q(\mu_{ij}) f_j(\mu_{ij}) \leq Q, 1 \leq j \leq N_2. \tag{7}$$

3) Each required arc must be serviced by one truck exactly once (i.e., no partial service).
4) The total length of each trip must not exceed the maximum trip length C

$$\max_{1 \leq j \leq N_2} \left\{ D\left(l_j\right) + S\left(l_j\right) \right\} \leq C. \tag{8}$$

5) The total service time of every truck must not exceed the maximum service time T

$$\max_{1 \leq j \leq N_2} \left\{ \sum_{i=1}^{m_j} d\left(\mu_{ij}\right) \right\} \leq \frac{S \times T}{\alpha}. \tag{9}$$

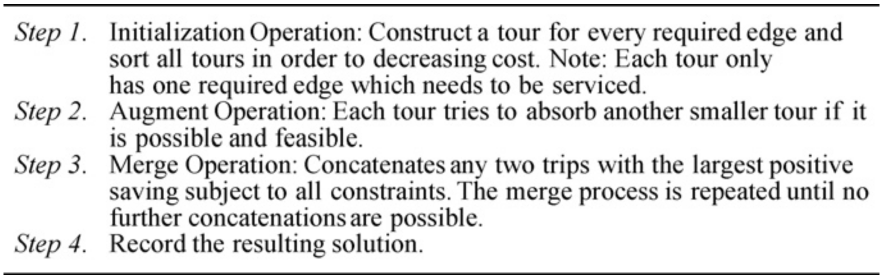
IV. Constructive Heuristics for the MCARP

The following classical heuristics, path-scanning [26], augment–merge [12], and the Ulusoy's heuristic [28] not only deal efficiently with CARP, but also have been extended to deal with ECARP [12]. In this paper, we further extend these heuristics to tackle MCARP. As mentioned in [12], the extended arc-to-arc distance matrix is employed to handle forbidden turns and Dijkstra's algorithm [50] is applied to precompute the shortest feasible path between all pairs of arcs.

A. Extended Random Path-Scanning

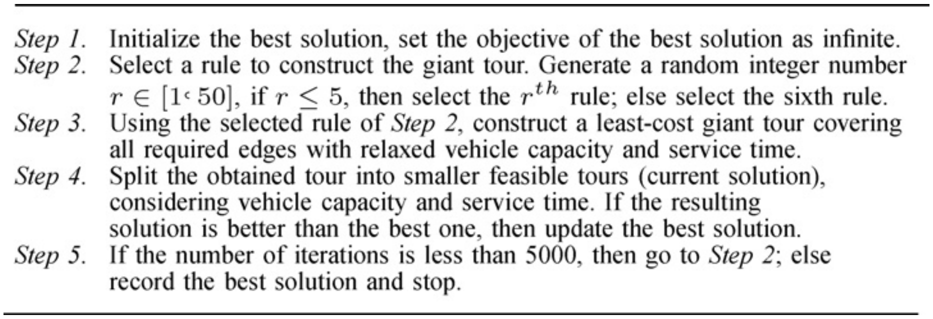
The extended random path-scanning (ERPS) constructs one tour at a time. When constructing a tour, it joins the most promising arc to the sequence of arcs until capacity Q or service time T is exhausted. For a sequence ending at arc u, it determines the set M of arcs closest to u, not serviced and feasible for Q and T. The following six rules are employed to select the next arc v in M.

1) *R1*: Dearest arc, maximize $d(v) + s(v) - pen(u, v)$.
2) *R2*: Cheapest arc, minimize $d(v) + s(v) + pen(u, v)$.
3) *R3*: Maximize the yield $q(v)/(d(v) + s(v) + pen(u, v))$.
4) *R4*: Minimize the yield $q(v)/(d(v) + s(v) - pen(u, v))$.
5) *R5*: Use R1 if the vehicle is less than half-full, else use R2.
6) *R6*: Use the above five rules randomly.



```
Step 1. Initialization Operation: Construct a tour for every required edge and
        sort all tours in order to decreasing cost. Note: Each tour only
        has one required edge which needs to be serviced.
Step 2. Augment Operation: Each tour tries to absorb another smaller tour if it
        is possible and feasible.
Step 3. Merge Operation: Concatenates any two trips with the largest positive
        saving subject to all constraints. The merge process is repeated until no
        further concatenations are possible.
Step 4. Record the resulting solution.
```

Fig. 2. Pseudocode of extended augment–merge heuristic.



```
Step 1. Initialize the best solution, set the objective of the best solution as infinite.
Step 2. Select a rule to construct the giant tour. Generate a random integer number
        $r \in$ [1‹ 50], if $r \leq 5$, then select the $r^{th}$ rule; else select the sixth rule.
Step 3. Using the selected rule of Step 2, construct a least-cost giant tour covering
        all required edges with relaxed vehicle capacity and service time.
Step 4. Split the obtained tour into smaller feasible tours (current solution),
        considering vehicle capacity and service time. If the resulting
        solution is better than the best one, then update the best solution.
Step 5. If the number of iterations is less than 5000, then go to Step 2; else
        record the best solution and stop.
```

Fig. 3. Pseudocode of extended random Ulusoy's heuristic.

The first four rules are individual rules, while the last two rules are compound rules. Rule R6 is constructed as follows:

$$\text{R6} = \begin{cases} \text{R1}, & 0 \leq r_i < 0.2 \\ \text{R2}, & 0.2 \leq r_i < 0.4 \\ \text{R3}, & 0.4 \leq r_i < 0.6 \\ \text{R4}, & 0.6 \leq r_i < 0.8 \\ \text{R5}, & 0.8 \leq r_i < 1 \end{cases} \tag{10}$$

where $r_i \in [0, 1)(1 \leq i \leq N_R)$ denotes a randomly chosen number. It other words, rule R6 randomly selects one rule from R1 to R5 at each decision point. The ERPS can quickly obtain one near-optimal solution thanks to compensation effects of these different rules. The pseudocode of ERPS is displayed in Fig. 1.

B. Extended Augment-Merge

Initialization, augment, and merge are three phases in the EAM heuristic. The initialization phase constructs N_R tours (one per required arc) and sorts the resulting tours in decreasing cost order. In the augment phase, tour T_i absorbs another smaller tour T_j if it satisfies the constraint of capacity Q and all the required arcs of T_j are on the deadheading path of T_i. In the merge phase, the heuristic evaluates possible concatenations of any two tours and joins the two tours with the largest positive saving subject to constraints of maximum service time T and capacity Q. This merge operation is repeated until no further concatenation is possible. EAM is a deterministic approach and its pseudocode is displayed in Fig. 2.

C. Extended Random Ulusoy's Heuristic

The extended random Ulusoy's heuristic (ERUH) temporarily relaxes capacity Q and service time T to construct a least-cost tour C covering all required arcs (a giant tour). This phase is implemented by executing ERPS with a big value of Q and T. The (giant) tour C is optimally split into some small feasible tours (the proposed split procedure is explained later). In order to achieve better results, users can keep all tours obtained by ERPS, split them, and return the best solution. The pseudocode of ERUH is displayed in Fig. 3.

V. Proposed Evolutionary Framework

This proposed evolutionary framework is characterized by the integration of the three extended heuristics discussed in previous sections. As mentioned before, the significant size of the infeasible region in the MCARP search space poses the difficulty of finding good quality feasible solutions using the EA alone. The integration of classical heuristics (*heuristics*) into the evolutionary framework addresses this issue. In addition, the near-optimal solutions obtained throughout the search are analyzed to extract information that can be used to guide the subsequent search (*heuristic information*). We subsequently label our approach as *HHEA*. The computational flow of HHEA is shown in Fig. 4. In the initialization phase, most individuals are generated by ERPS and ERUH, while in the mutation phase, the 2-Opt heuristic is employed to implement the mutation on one or two trips. The 2-Opt heuristic eliminates two edges and reconnects the two resulting paths in a novel way to obtain a new tour.

Many papers have reported on applications of the interaction between evolution and learning [51–55]. These approaches keep useful features of previous individuals to improve the performance of current individuals and exploit profitable individuals from the previous population to populate the memory (train a learning module). According to [55], such approaches outperform traditional evolutionary algorithms on several

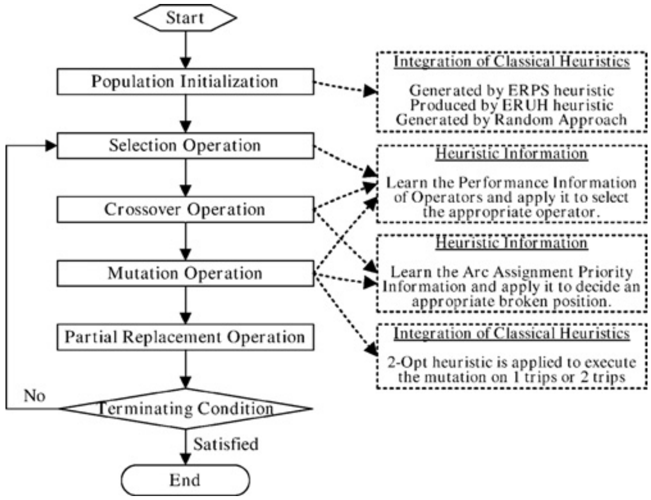

Start
Population Initialization
Selection Operation
Crossover Operation
Mutation Operation
Partial Replacement Operation
Terminating Condition
No
Satisfied
End
Integration of Classical Heuristics
Generated by ERPS heuristic
Produced by ERUH heuristic
Generated by Random Approach
Heuristic Information
Learn the Performance Information of Operators and apply it to select the appropriate operator.
Heuristic Information
Learn the Arc Assignment Priority Information and apply it to decide an appropriate broken position.
Integration of Classical Heuristics
2-Opt heuristic is applied to execute the mutation on 1 trips or 2 trips

Fig. 4. Computational flow of this proposed evolutionary framework.

benchmarks (e.g., flexible job shop scheduling problem). In a similar fashion, HHEA extracts instance-specific information from near-optimal MCARP solutions, obtained throughout the search process, to guide the subsequent evolution. This process is elaborated further in the subsequent sections.

A. Heuristic Information

In order to improve the performance of HHEA, various operators are employed to execute the operations of selection, crossover, and mutation. One cannot expect a single operator to perform equally well on a range of different instances, and therefore, it is necessary to discover which operators are efficient for the current instance. We call this information the performance information of operators (PIO). The success of each operator is judged by its ability to produce an individual that can be inserted into the current population (the insertion policy is explained in Section V-H). In HHEA, six distinct mutation operators are employed. Suppose $N(i)$ denotes the accumulated number of successful mutations by the ith mutation operator, then $N(i)$ may be considered as the performance of the ith mutation operator. If the current mutation operation is successful and it was executed using the ith mutation operator, and then $N(i)$ will be increased by 1. For the next mutation, the mutation operator is selected randomly according to following probability:

$$P(i) = \frac{N(i)}{\sum_{i=1}^{6} N(i)} \tag{11}$$

where $P(i)$ denotes the probability of selecting the ith operator.

The second measure is called the arc assignment priority information (AAPI) which is applied to establish a beneficial order for the given arcs. A matrix M_a with size $N_A \times Dim$ is defined for the AAPI, where Dim denotes the predefined row dimension of matrix M_a. A simple example of an AAPI matrix ($N_A = 4$ and $Dim = 6$) is shown in Table II. In this case, the first three ($0.5 \times Dim$) rows of each line denote the three closest arcs to the corresponding arc. For example, the number 1 in the first row of the second line denotes that the closest arc to arc 2 (corresponding to the second line) is arc 1 (each arc is indexed from 1 to N_A), whereas number 3 in the third row of the second line indicates that the third closest arc to arc 2 is arc 3. The last three rows of each line indicate the total number of appearances of a given sequence among the near-optimal solutions. For example, number 6 in the fourth row of the second line denotes the total number of times of the sequence (2, 1) appears among the near-optimal solutions obtained throughout the search (sequence (2, 1) means that arc 2 is serviced before arc 1).

TABLE II
EXAMPLE OF ARC ASSIGNMENT PRIORITY INFORMATION MATRIX

Arcs	Row 1	Row 2	Row 3	Row 4	Row 5	Row 6
1	2	3	4	12	8	10
2	1	4	3	6	11	13
3	1	4	2	9	10	11
4	3	2	1	5	9	16

In order to preserve some promising subsequences among current individuals, the AAPI is employed to select appropriate break points for crossover and mutation. If the number of times a given sequence appears is large, then the break points for this sequence is selected with a small probability. For example, if the current individual is 4 → 3 → 2 → 1, the appearance times of sequences (4, 3), (3, 2), and (2, 1) are 5, 11, and 6, respectively (obtained from Table II), and the broken position between sequences (4, 3), (3, 2), or (2, 1) is selected with a probability of 0.5, 0.07, or 0.43, respectively. The probabilities are computed as follows:

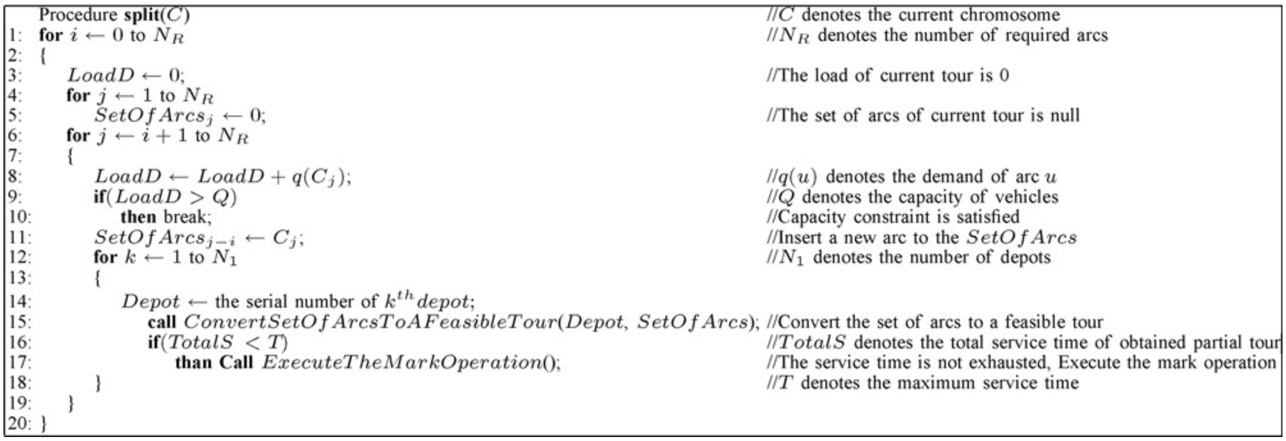
$$\begin{array}{ccc} 5 & 11 & 6 \\ 4 \rightarrow 3 & \rightarrow 2 & \rightarrow 1 \\ 0.50 & 0.07 & 0.43 \end{array}$$

Appearance times:	5	11	6
Inverse number (−):	−5	−11	−6
Make it positive (+12):	7	1	6
Probability computation (÷14)	0.5	0.07	0.43.

According to our experimental results, an appropriate value for row dimension of AAPI matrix is 20, and the AAPI is obtained from the top 30 individuals of the current population.

B. Chromosome Structure and Evaluation

Each chromosome C is defined simply as a sequence of the N_R required arcs, without trip delimiters, and with implicit shortest paths between consecutive arcs and may be viewed as a giant tour ignoring the capacity and service time constraints. The split procedure is then employed to obtain a feasible MCARP solution. The fitness $F(C)$ of individual C is acquired by the split procedure, which minimizes the total cost subject to the sequence of arcs defined by individual C and the number of trips. The number of trips is variable but cannot exceed the number of trucks N_2. The pseudocode of the split procedure is displayed in Fig. 5. This procedure runs in $O(N_R N_2)$ space and its complexity is $O(N_R^2 N_2)$. This algorithm enumerates all subsequences $(C_i, \ldots, C_j)$ of C that correspond to feasible trips and computes their total



```
Procedure split($C$)                                             //$C$ denotes the current chromosome
for $i \leftarrow 0$ to $N_R$                                     //$N_R$ denotes the number of required arcs
{
    $LoadD \leftarrow 0$;                                        //The load of current tour is 0
    for $j \leftarrow 1$ to $N_R$
        $SetOfArcs_j \leftarrow 0$;                              //The set of arcs of current tour is null
    for $j \leftarrow i+1$ to $N_R$
    {
        $LoadD \leftarrow LoadD + q(C_j)$;                       //$q(u)$ denotes the demand of arc $u$
        if($LoadD > Q$)                                          //$Q$ denotes the capacity of vehicles
            then break;                                          //Capacity constraint is satisfied
        $SetOfArcs_{j-i} \leftarrow C_j$;                        //Insert a new arc to the $SetOfArcs$
        for $k \leftarrow 1$ to $N_1$                             //$N_1$ denotes the number of depots
        {
            $Depot \leftarrow$ the serial number of $k^{th}depot$;
                call $ConvertSetOfArcsToAFeasibleTour(Depot,\ SetOfArcs)$; //Convert the set of arcs to a feasible tour
                if($TotalS\ < T$)                                //$TotalS$ denotes the total service time of obtained partial tour
                    than Call $ExecuteTheMarkOperation()$;       //The service time is not exhausted, Execute the mark operation
        }                                                        //$T$ denotes the maximum service time
    }
}
```

Fig. 5. Proposed split procedure working on a chromosome.

cost and total deadheading cost, while it enumerates all depots in the third loop; the split procedure first satisfies the constraint imposed by the maximum number of trucks and then finds the minimum total cost of the solutions. It should be noted that the operation *ExecuteTheMarkOperation*(), as shown in Fig. 5, is performed by the multimark algorithm, a simple example of which may be found in Appendix B.

C. Population Initialization

In the proposed evolutionary framework, both feasible and infeasible solutions are allowed to exist in the population. In order to penalize infeasible solutions, their fitness is doubled (MCARP is a minimization problem). The population consists exclusively of unique phenotypes and clones (identical solutions) are never accepted. Since the verification of unique phenotypes is computationally very expensive, an approximate but faster method is applied to guarantee all individuals among the population have different costs. Suppose L_b and U_b denote the LB and upper bound on the total cost of solutions, respectively. Here, L_b is approximated by MCNDLB [23], and U_b is empirically estimated via $U_b = 5L_b$. We experimentally established the interval $[L_b, U_b]$ to be divided into 10 000 levels, and only the best solution in each subinterval is inserted into the current population.

The population is initialized using three different methods: random generation, ERPS heuristic, and ERUH heuristic. The quality of the initial population is improved significantly by ERPS and ERUH. However, many attempts may be required to add a new distinct chromosome to the population when the population size (P_S) is large or the problem size is small. Therefore, the number of attempts to obtain one new individual is upper bounded by N_I: if it turns out to be too difficult to achieve a sufficient number of initial individuals, the process of population initialization will be terminated after N_I unsuccessful attempts. In this case, the value of P_S will be reduced to the number of unique individuals generated already. The fitness evaluations in the population initialization phase are counted into the total number of fitness evaluations.

D. Selection Operation

In an iteration of the HHEA, two distinct chromosomes are selected to undergo crossover and mutation. The resultant individual is inserted into the current population if possible. The following two types of selection are used.

1) *Binary Tournament*: Selects two different chromosomes randomly and the least-cost one is kept.
2) *Rank Order Selection*: Selects the chromosome of rank i (individuals are ranked according to their fitness) with probability

$$P(i) = \frac{2(P_S - i + 1)}{P_S(P_S + 1)}. \tag{12}$$

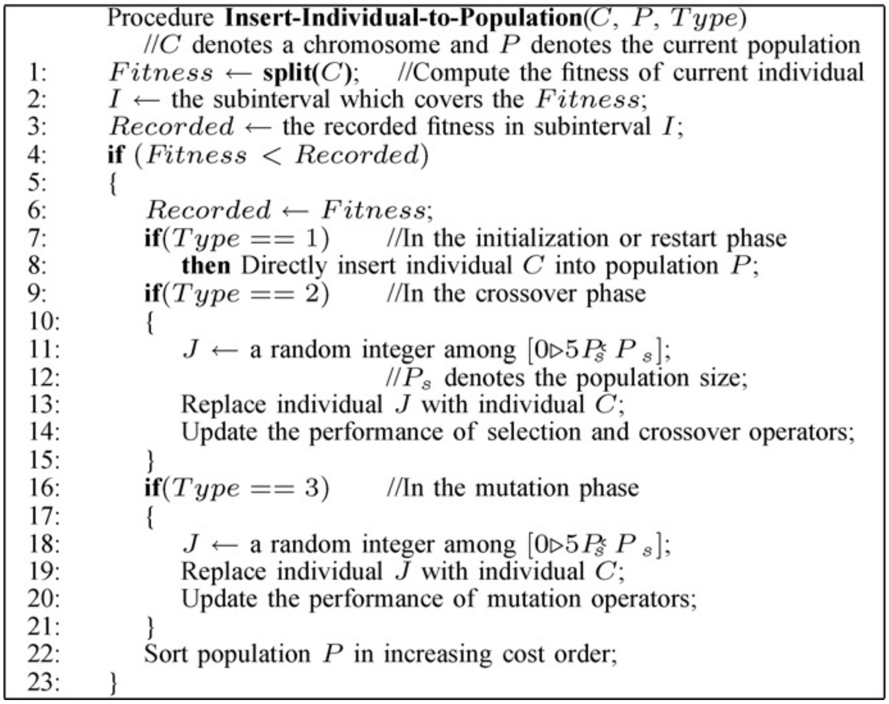
E. Crossover Operation

The fitness (solution quality) of newly produced offspring is used to adjust the genetic operators, and hence to influence the search. However, since selection alone does not generate any novel individuals, its success rate is established once the individuals have been crossed over (which takes place immediately after selection). HHEA makes use of two different crossover operators, order crossover (OX) and linear order crossover (LOX). LOX is designed specifically for linear chromosomes and OX is used for circular permutations. These offspring resulting from the crossover event are evaluated by the proposed split procedure.

F. Mutation Operation

It has been demonstrated that the effectiveness of canonical GAs can be improved significantly by using a memetic approach: the algorithm's genetic operators are complimented by local search for improving overall performance [10]. With some probability, each offspring is converted into a solution to undergo *mutation* by means of local search, which scans pairs of arcs to evaluate the moves listed below where x (y) denotes the arc serviced after arc u (v) in trip T_1 (T_2).

1) *Move 1*: Invert arc u if it is a bidirectional arc.
2) *Move 2*: Move u after v.
3) *Move 3*: Move adjacent arcs (u, x) after arc v.
4) *Move 4*: Swap arcs u and v.
5) *Move 5*: 2-Opt moves on one trip [10].



```
Procedure Insert-Individual-to-Population($C$, $P$, $Type$)
    //$C$ denotes a chromosome and $P$ denotes the current population
$Fitness \leftarrow$ split($C$);    //Compute the fitness of current individual
$I \leftarrow$ the subinterval which covers the $Fitness$;
$Recorded \leftarrow$ the recorded fitness in subinterval $I$;
if ($Fitness < Recorded$)
{
    $Recorded \leftarrow Fitness$;
    if($Type == 1$)      //In the initialization or restart phase
        then Directly insert individual $C$ into population $P$;
    if($Type == 2$)      //In the crossover phase
    {
        $J \leftarrow$ a random integer among $[0\triangleright 5P_s^{,} P_s]$;
                              //$P_s$ denotes the population size;
        Replace individual $J$ with individual $C$;
        Update the performance of selection and crossover operators;
    }
    if($Type == 3$)      //In the mutation phase
    {
        $J \leftarrow$ a random integer among $[0\triangleright 5P_s^{,} P_s]$;
        Replace individual $J$ with individual $C$;
        Update the performance of mutation operators;
    }
    Sort population $P$ in increasing cost order;
}
```

Fig. 6. Computational flow of inserting one individual into the population.

TABLE III

STRATEGY SETTINGS OF THREE DIFFERENT VERSIONS OF THE HHEA

Phases	HHEA-1	HHEA-2	HHEA-3
Selection	S1	S2	S2
Crossover	S1 + S3	S2 + S3	S2 + S4
Mutation	S1 + S3	S2 + S3	S2 + S4

TABLE IV

PARAMETER SETTINGS OF HHEA

Name	Role	Value
P_S	Population Size	200
N_I	Maximum number of attempts to get each initial chromosome	100
R_M	Local search rate in main phase and restarts	0.05
N_L	Fixed Number of attempts in each Local search	5
N_F^m	Maximum number of fitness evaluations in main phase	2000
N_R	Maximum number of restarts	5
N_S	Number of solutions replaced in each restart	20
N_F^r	Maximum number of fitness evaluations per restart	600

6) *Move 6*: 2-Opt moves on two trips [10].

The mutation operator then selects a type of move randomly. It then randomly produces N_L optional moves and computes the improved objective of each move. Finally, the move that results in the best improvement is performed.

TABLE V

SUMMARY OF ABBREVIATIONS USED

Symbols	Signification
SN	Serial number of examples.
$Name$	Name of examples.
N_P	Number of penalties.
L_b	Lower bound of examples.
$Depots$	Set of predefined depots.
F^*	Known optimal result of examples.
F_A	Average obtained result of examples.
F_D	Standard deviation of obtained results.
F_A^E	Error between F_A and L_b.
F_A^*	Error between F_A and F^*.
T_A	Average computational time.

G. Partial Replacement Operation

According to [10], many CARP instances can be solved by adding a fixed number of short restarts based on the partial replacement procedure (PRP) [56]. It has been demonstrated experimentally that PRP never degrades the worst cost, preserves the best solution, and obtains a better final solution [56]. In this paper, the same number of restarts and the same length of each restart are allocated to all examples.

H. Insertion Policy

The computational flow of inserting an individual (which has to be unique) into the population is displayed as Fig. 6. In the initialization or restart phase, the new chromosome is inserted directly into population. In the crossover or mutation phase, the chromosome, whose total cost is larger than the median cost of the current population, is selected randomly from the population and is replaced by the new individual.

TABLE VI
ATTRIBUTES OF MODIFIED MCARP INSTANCES GENERATED BY BENCHMARK FILES (TOTAL 87 EXAMPLES)

SN	Name	N_V	N_A	N_P	L_b	Depots	α
C1	E-GDB1	12	44	44	544.8	1	0.6
C2	E-GDB2	12	52	44	616.4	1	0.6
C3	E-GDB3	12	44	34	520.8	8	0.6
C4	E-GDB4	11	38	44	523.2	1	0.6
C5	E-GDB5	13	52	50	704.8	1	0.6
C6	E-GDB6	12	44	44	592	7	0.6
C7	E-GDB7	12	44	56	557.6	1	0.6
C8	E-GDB8	27	92	50	489.2	5, 17	0.6
C9	E-GDB9	27	102	50	542	26	0.6
C10	E-GDB10	12	50	30	528.8	12	0.6
C11	E-GDB11	22	90	36	718.4	1	0.6
C12	E-GDB12	13	46	12	786.4	5	0.6
C13	E-GDB13	10	56	2	1035.6	1	0.6
C14	E-GDB14	7	42	40	201.6	4	0.6
C15	E-GDB15	7	42	16	112.8	5	0.6
C16	E-GDB16	8	56	40	241.2	1	0.6
C17	E-GDB17	8	56	28	170.4	1	0.6
C18	E-GDB18	9	72	88	325.6	1	0.6
C19	E-GDB19	8	22	6	91.6	5	0.6
C20	E-GDB20	11	44	40	222.8	1	0.6
C21	E-GDB21	11	66	60	313.2	1	0.6
C22	E-GDB22	11	88	108	391.6	1	0.6
C23	E-GDB23	11	110	114	457.6	1	0.6
C24	E-KSHS1	8	30	14	20 006	5	0.05
C25	E-KSHS2	10	30	12	14 608.8	1	0.05
C26	E-KSHS3	6	30	18	16 463.6	1	0.05
C27	E-KSHS4	8	30	24	17 767.6	3	0.05
C28	E-KSHS5	8	30	30	20 335.6	2	0.05
C29	E-KSHS6	9	30	30	20 390	1	0.05
C30	E-VAL1A	24	78	4	294.4	13	2
C31	E-VAL1B	24	78	4	294.4	5	2
C32	E-VAL1C	24	78	4	346.4	5	2
C33	E-VAL2A	24	68	12	382	15	2
C34	E-VAL2B	24	68	12	386	20	2
C35	E-VAL2C	24	68	12	444.4	15	2
C36	E-VAL3A	24	70	10	130	11	2
C37	E-VAL3B	24	70	10	131.6	11	2
C38	E-VAL3C	24	70	10	145.2	11	2
C39	E-VAL4A	41	138	4	690.8	32	2
C40	E-VAL4B	41	138	4	714.4	21	2
C41	E-VAL4C	41	138	4	735.6	26	2
C42	E-VAL4D	41	138	4	784.4	27	2
C43	E-VAL5A	34	130	10	737.2	12	2
C44	E-VAL5B	34	130	10	742	21	2
C45	E-VAL5C	34	130	10	781.6	2	2
C46	E-VAL5D	34	130	10	862.8	6	2
C47	E-VAL6A	31	100	8	383.2	3	2
C48	E-VAL6B	31	100	8	403.2	12	2
C49	E-VAL6C	31	100	8	448	11	2
C50	E-VAL7A	40	132	8	498	1	2
C51	E-VAL7B	40	132	8	498	1	2
C52	E-VAL7C	40	132	8	533.6	1	2
C53	E-VAL8A	30	126	76	698.8	1, 28	2
C54	E-VAL8B	30	126	76	698.8	1, 24	2
C55	E-VAL8C	30	126	76	748.8	1, 17	2
C56	E-VAL9A	50	184	22	576	18, 34	2
C57	E-VAL9B	50	184	22	564	1, 38	2
C58	E-VAL9C	50	184	22	574.4	1, 35	2
C59	E-VAL9D	50	184	22	622.8	1, 37	2
C60	E-VAL10A	50	194	28	755.2	13	2
C61	E-VAL10B	50	194	28	771.2	37	2
C62	E-VAL10C	50	194	28	802.4	28	2
C63	E-VAL10D	50	194	28	814.4	31	2
C64	E-EGL-E1-A	77	196	6	3376.6	19	0.2
C65	E-EGL-E1-B	77	196	6	3720.6	12	0.2
C66	E-EGL-E1-C	77	196	6	4611.2	76	0.2
C67	E-EGL-E2-A	77	196	4	4357.6	19	0.2
C68	E-EGL-E2-B	77	196	4	5775.6	9	0.2
C69	E-EGL-E2-C	77	196	4	5369.8	44	0.2
C70	E-EGL-E3-A	77	196	2	5297.4	19	0.2
C71	E-EGL-E3-B	77	196	2	7233.6	14	0.2
C72	E-EGL-E3-C	77	196	2	7541.8	10	0.2
C73	E-EGL-E4-A	77	196	20	5278	1, 20, 41	0.2
C74	E-EGL-E4-B	77	196	20	5866	23, 44, 76	0.2
C75	E-EGL-E4-C	77	196	20	6209.2	1, 13, 20, 40	0.2
C76	E-EGL-S1-A	140	380	2	4718	79	0.2
C77	E-EGL-S1-B	140	380	2	5373.4	85	0.2
C78	E-EGL-S1-C	140	380	2	7325.6	81	0.2
C79	E-EGL-S2-A	140	380	8	8647	1, 63	0.2
C80	E-EGL-S2-B	140	380	8	10 291.2	1, 139	0.2
C81	E-EGL-S2-C	140	380	8	10 680	72, 123	0.2
C82	E-EGL-S3-A	140	380	8	8626	1, 57	0.2
C83	E-EGL-S3-B	140	380	8	10 640	1, 43	0.2
C84	E-EGL-S3-C	140	380	8	12 278.6	1, 66	0.2
C85	E-EGL-S4-A	140	380	4	10 070.8	2, 11, 110	0.2
C86	E-EGL-S4-B	140	380	4	10 835.2	34, 111, 118	0.2
C87	E-EGL-S4-C	140	380	4	13 325.2	6, 89, 112	0.2

TABLE VII
ATTRIBUTES OF MCARP INSTANCES WITH KNOWN OPTIMA (TOTAL 20 EXAMPLES)

SN	Name	N_V	N_A	N_P	L_b	F^*
E1	CASE-C101	40	132	0	164	180
E2	CASE-C102	40	132	0	218	262
E3	CASE-C103	40	132	0	174	210
E4	CASE-C104	40	132	0	218	262
E5	CASE-C201	49	168	0	224	240
E6	CASE-C202	49	168	0	296	324
E7	CASE-C203	49	168	0	224	272
E8	CASE-C204	49	168	0	296	324
E9	CASE-C301	70	246	0	328	360
E10	CASE-C302	70	246	0	424	492
E11	CASE-C303	70	246	0	348	412
E12	CASE-C304	70	246	0	424	492
E13	CASE-C401	91	324	0	432	480
E14	CASE-C402	91	324	0	556	668
E15	CASE-C403	91	324	0	464	532
E16	CASE-C404	91	324	0	556	668
E17	CASE-C501	100	360	0	484	540
E18	CASE-C502	100	360	0	630	754
E19	CASE-C503	100	360	0	514	550
E20	CASE-C504	100	360	0	630	754

TABLE VIII
EXPERIMENTAL RESULTS OF EAM HEURISTIC

SN	Name	L_b	F_A	F_A^E (%)	$T_A(S)$
C2	E-GDB2	616.4	784.4	27.26	3.27
C3	E-GDB3	520.8	594.4	14.13	1.78
C5	E-GDB5	704.8	859.2	21.91	2.7
C6	E-GDB6	592	660	11.49	1.73
C7	E-GDB7	557.6	860	54.23	1.66
C12	E-GDB12	786.4	891.2	13.33	0.7
C14	E-GDB14	201.6	226	12.1	3.17
C15	E-GDB15	112.8	122.4	8.51	4.11
C17	E-GDB17	170.4	179.2	5.16	7.7
C19	E-GDB19	91.6	121.2	32.31	0.52
C34	E-VAL2B	386	486	25.91	2.89
C47	E-VAL6A	383.2	455.2	18.79	8.36
C66	E-EGL-E1-C	4611.2	5679.8	23.17	1.51
C72	E-EGL-E3-C	7541.8	9992.6	32.5	3.66
C77	E-EGL-S1-B	5373.4	7274.2	35.37	2.25
Average:				22.41	3.07

When inserting a new distinct chromosome C into the current population P, we need to locate the subinterval I which covers the fitness of C. If the fitness of C is better than the recorded fitness of I, then the recorded fitness of I is updated and C is inserted into P. An operation (e.g., selection, crossover, or mutation) is considered successful if C can be inserted into P and unsuccessful otherwise.

I. Three Variants of the HHEA

In this paper, three different versions of HHEA are designed for comparison. Each variant shares the same overall computational flow but employs different strategies for the operations of selection, crossover, and mutation. The variants are designed to determine whether there exists a significant difference in employing one strategy over the other. The choice of strategies is summarized in Table III. Variants HHEA-1 and HHEA-2 are employed to compare the impact of strategies S1 and S2, while HHEA-2 and HHEA-3 are employed to compare the impact of strategies S3 and S4. These strategies are explained as follows.

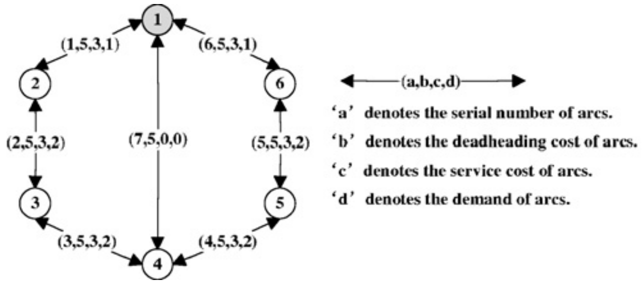

1
2
3
4
5
6
(1,5,3,1)
(6,5,3,1)
(2,5,3,2)
(7,5,0,0)
(5,5,3,2)
(3,5,3,2)
(4,5,3,2)
(a,b,c,d)
'a' denotes the serial number of arcs.
'b' denotes the deadheading cost of arcs.
'c' denotes the service cost of arcs.
'd' denotes the demand of arcs.

Fig. 7. Simple example of an MCARP instance. In this example, there are six nodes (node 1 is the depot node), seven arcs, and two available trucks. The capacity of trucks is six. For the sake of simplicity, the constraint of maximum service time is not considered in this example.

1) *S1 (Uniform Selection of Operators):* In the selection phase, rank order selection and binary tournament are chosen randomly with equal probability.
2) *S2 (Empiristic Selection of Operators Using Heuristic Information):* This strategy adopts the PIO to empirically select an appropriate mutation operator (see Section V-A).
3) *S3 (Uniform Selection of One Broken Position):* This strategy randomly selects a broken position for crossover or mutation with equal probability.
4) *S4 (Empiristic Selection of the Broken Position Using Heuristic Information):* In this case, AAPI is used to empirically decide one broken position (see Section V-A).

As mentioned previously, the overall computational flow, and the main and restart phase in particular, is identical across the three variants. In the main phase, selection, crossover, and mutation are the three primary operations. The main phase of HHEA is terminated when the defined number of fitness evaluations is exhausted. The partial replacement procedure is employed to maintain or increase the diversity of current population at the beginning of the restart phase. The searching process of restart phase is similar to the main phase except the different parameters.

VI. EXPERIMENTAL RESULTS

The experiments were designed to compare HHEA-3 with ERPS, ERUH, HHEA-1, HHEA-2, and MA [10]. Each algorithm is allowed to run for 5000 fitness evaluations on each of the 107 problem instances which have up to 140 nodes and 380 arcs. Out of those instances, 87 instances are produced from existing CARP instances. As the global optima of the modified instances is unknown, the LB is obtained by MCNDLB [23] and is used as a reference value to estimate the solution quality of different approaches. Furthermore, an additional 20 instances with known optima are constructed which may be used directly to obtain the computational error of the different algorithms.

A. Parameter Settings

The HHEA was implemented using Visual C++ language, and executed on a personal computer with the 1.8 GHz processor and 512 MB memory. We established a favorable choice

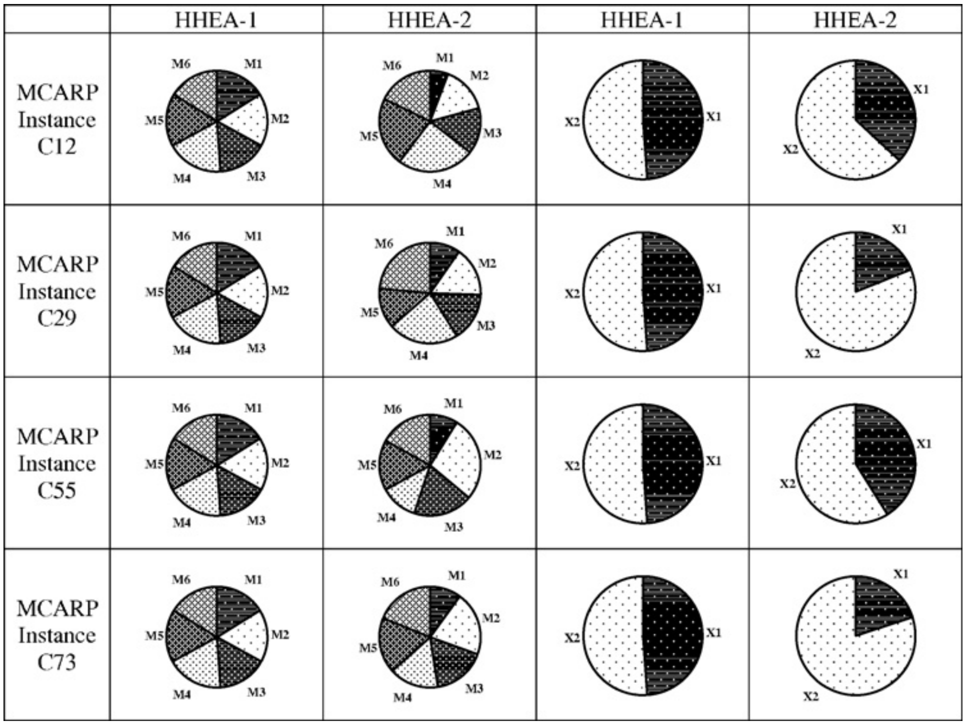
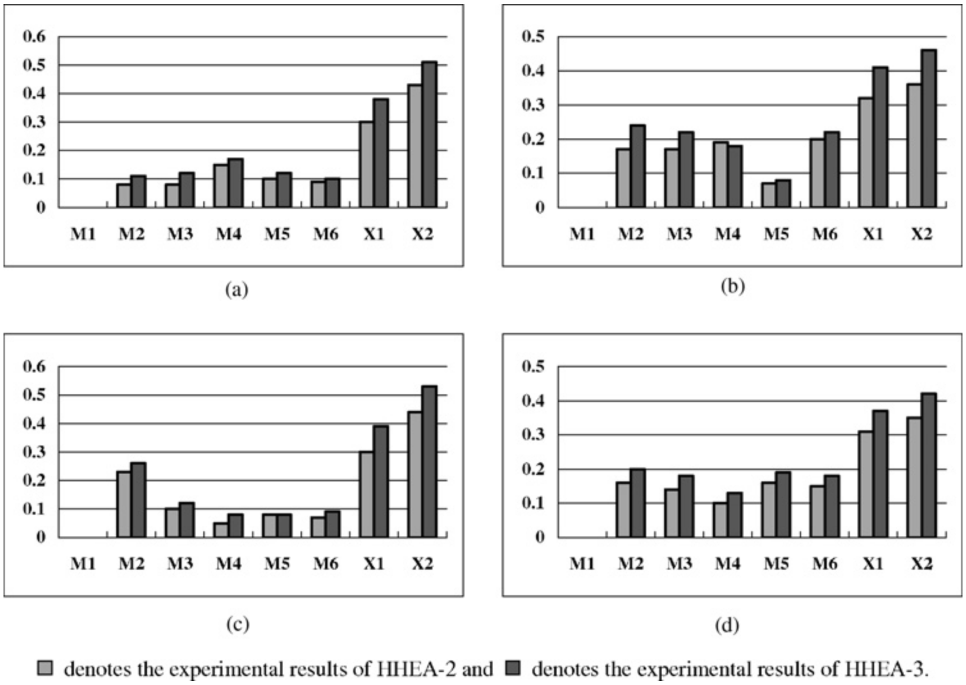


Fig. 8. Performance of the different operators in HHEA-1 and HHEA-2. Symbols M1, M2, . . ., M6 denote the first, second, and sixth mutation operator, respectively. The symbol X1 denotes the OX operator and symbol X2 denotes the LOX operator. Ratio of selection denotes the ratio of selecting each operator throughout the whole evolution process.



(a)

(b)

(c)

(d)

Fig. 9. Performance of different operators in HHEA-2 and HHEA-3. (a) Comparative results from instance C12. (b) Comparative results from instance C29. (c) Comparative results from instance C55. (d) Comparative results from instance C73. Axis *y* denotes the ratio of improvement and it means the probability of solution improvement by the given operator throughout the whole evolution process. Symbols M1, M2, . . ., M6 denote the first, second, and sixth kind of mutation operators, respectively. Symbol X1 denotes the OX operator and symbol X2 denotes the LOX operator.

TABLE IX

EXPERIMENTAL RESULTS OF ERPS AND HHEA-3 AVERAGED OVER 50 TRIALS (MODIFIED MCARP INSTANCES)

	ERPS				HHEA-3					ERPS				HHEA-3			
SN	F_A	F_D	F_A^E(%)	T_A(S)	F_A	F_D	F_A^E(%)	T_A(S)	SN	F_A	F_D	F_A^E(%)	T_A(S)	F_A	F_D	F_A^E(%)	T_A(S)
C1	559	19.6	2.6	69.3	562.7	18.8	3.3	215.2	C45	867.6	2.8	11	78.9	**817.3**	5	4.6	589.6
C2	633.7	31.2	3.3	71.6	622.9	8.1	1.1	223.5	C46	1017.2	6.6	17.9	86.9	**933.7**	8.6	8.2	515.3
C3	555.7	4.8	7.4	68.8	**532.7**	3.6	2.3	226.2	C47	426.9	1.5	11.4	73.3	**400.8**	4.8	4.6	605.9
C4	583.4	7.4	12.2	69	**544.3**	6.3	4	207.5	C48	447.2	2	10.9	74.5	**420.1**	4.8	4.2	532
C5	764	9.2	8.9	72.1	**710.6**	9	0.8	227.9	C49	527.7	2.9	17.8	80.2	**486.3**	4.2	8.5	328.7
C6	652.4	4.4	10.8	68.1	**595.5**	6.5	0.6	223.8	C50	544.8	2.5	9.4	75.9	**525.5**	2	5.5	795.7
C7	617.3	5.9	11.2	68.5	**591.5**	6.6	6.1	211.8	C51	557.3	3.6	11.9	77.8	**527.3**	5.7	5.9	632.2
C8	645.3	9.6	32.6	90.8	**537.2**	6.7	9.8	385.1	C52	643.5	3.8	20.6	85.9	**600.2**	7.1	12.5	409.1
C9	628.7	7.9	16.7	92.7	**611.8**	13.4	12.9	377.5	C53	757.5	9.4	8.4	75.2	**738.6**	4.2	5.7	1248.1
C10	564.2	4.3	7.4	68.2	**535.9**	4.4	1.3	241.3	C54	789.6	6.4	13	76.5	**748.9**	5.4	7.2	952.5
C11	795.3	6.2	11.1	72.4	**751.3**	4.3	4.6	335.8	C55	945	9.7	26.2	84.5	**854.8**	20.1	14.2	541.4
C12	908.3	7.3	16.0	70.7	**815.2**	5.1	3.7	219.1	C56	615.2	1.4	6.8	80.7	**586.4**	5.2	1.8	2911.2
C13	1059.4	48.1	2.3	69.8	1053.9	24.1	1.8	220	C57	624.4	3.5	10.7	83.3	**596.6**	3.7	5.8	2064.9
C14	212.3	1.2	5.3	67.2	**205.7**	1.1	2	199.3	C58	642.8	3.3	11.9	84.9	**603.8**	4.1	5.1	1683.6
C15	117	1.1	3.7	69.2	**113.8**	0.3	0.9	211.4	C59	736.2	8.5	18.2	96.2	**649.4**	4.7	4.3	950.3
C16	251.6	1.2	4.3	68.8	**249.9**	1.2	3.6	227.7	C60	802.8	5.3	6.3	84.1	**789.1**	5.9	4.5	1809.1
C17	174.5	1	2.4	68.8	**170.8**	0.5	0.2	227	C61	818.2	2.5	6.1	86.1	**796**	2.6	3.2	1385.4
C18	346.4	1.4	6.4	70	**332.6**	2.4	2.1	282.2	C62	854.6	2.6	6.5	89.8	**819.9**	4.4	2.2	1349.5
C19	93.5	1.6	2.1	64.2	**92.6**	1.4	1.1	167.2	C63	914.6	5.1	12.3	99.6	**864.7**	4.1	6.2	775.2
C20	237.7	2.1	6.7	69.2	**233.1**	2.9	4.6	215.9	C64	4615.8	25.8	36.7	86	**4237.6**	41.6	25.5	522.3
C21	322	1	2.8	70.5	**315.2**	3.1	0.6	226.2	C65	5249.8	66.4	41.1	90.5	**4733**	61.5	27.2	538.8
C22	400.6	1	2.3	73.7	**396.1**	2.8	1.1	283.1	C66	6086.8	57.5	32	97.4	**5461**	69.6	18.4	491.6
C23	472.2	1.2	3.2	78.6	**470.4**	2.1	2.8	334	C67	5943.8	64.5	36.4	98.4	**5491.6**	33.8	26	688.5
C24	21 566	189.1	7.8	66.2	**20 943**	102.5	4.7	199.3	C68	7733.5	63.8	33.9	109.6	**6884.1**	115.4	19.2	731.4
C25	15 763	373.4	7.9	66	**15 027**	92.8	2.9	193.8	C69	7882.9	35.7	46.8	114.9	**7500.8**	220.2	39.7	565.1
C26	17 616	100.3	7	65.4	**17 016**	173.5	3.4	194.6	C70	6891.9	99.9	30.1	106.1	**6765.3**	185.7	27.7	859.1
C27	21 517	378.1	21.1	66	**19 925**	217.9	12.1	196.5	C71	9208.4	46	27.3	125.8	**8754.5**	297.8	21	940.4
C28	23 447	284.6	15.3	65.8	**22 849**	552.8	12.4	203.7	C72	11 049	60	46.5	132.2	**9800.1**	96.3	29.9	802.7
C29	23 061	128.2	13.1	65.9	**22 380**	110.2	9.8	200.6	C73	7737.5	70.2	46.6	108.3	**6953.3**	396	31.7	1862.8
C30	316.5	1.2	7.5	69.5	**304.5**	2.8	3.4	536.4	C74	9526.4	79.4	62.4	123.7	**8423.8**	188.3	43.6	1289.1
C31	328.6	1.5	11.6	70.5	328.3	4.3	11.5	465.2	C75	12 319	125.1	98.4	157.3	**9838.2**	328.1	58.4	1391
C32	415.7	3.1	20	75.9	**386.3**	8.3	11.5	311.1	C76	6076.8	89	28.8	127.4	**5465**	43.4	15.8	1058.9
C33	398.8	5.1	4.4	68.9	**390.3**	3	2.2	423.4	C77	7796.8	90.1	45.1	147.7	**6551.1**	110.9	21.9	981.3
C34	423.4	5.9	9.7	69.5	**397.4**	3.5	3	340.6	C78	9691.8	124.8	32.3	163.6	**9237.9**	190.2	26.1	1010.4
C35	543.5	5.5	22.3	74.9	**539.3**	5.4	21.4	262.5	C79	12 728	118.9	47.2	221.5	**11 300**	780.3	30.7	2912.3
C36	137.8	1.2	6	68.6	**133.4**	0.4	2.6	433.9	C80	15 334	103.4	49	240.3	**14 149**	248.9	37.5	2048.6
C37	141.9	1	7.8	69.7	**135.4**	0.8	2.9	332	C81	17 344	78.8	62.4	300.9	**15 247**	314.7	42.8	1995.9
C38	177.9	1.8	22.5	75.4	**174.9**	3.6	20.5	257.2	C82	12 974	64.7	50.4	213.9	**10 373**	78.6	20.3	2897.2
C39	737.8	4.6	6.8	77.2	**725**	3.9	5	915	C83	19 365	165.5	82	341.7	**13 698**	302.9	28.7	2656.9
C40	770.1	4.8	7.8	79.3	**737.3**	2.6	3.2	750.8	C84	18 897	150.5	53.9	322.3	**17 081**	332.5	39.1	2499
C41	815	4.9	10.8	83.3	**775.6**	6.1	5.4	628.7	C85	16 627	249.6	65.1	322.1	**12 759**	869.3	26.7	2818.4
C42	898.1	5.9	14.5	87.5	**822.5**	3.8	4.9	520.2	C86	18 311	122.4	69	300.2	**15 052**	269.3	38.9	2044
C43	×	×	×	×	**772.2**	8.1	4.7	810.8	C87	25 278	226.7	89.7	420.1	**18 467**	188.6	38.6	2166.3
C44	795.4	6.4	7.2	77	**767.7**	5.8	3.5	639.7									

Symbol "×" indicates that no feasible solution can be obtained adopting the given approach. Values in bold face indicate that the performance of HHEA-3 is significantly better than that of ERPS (the confidence degree is 0.95) on the given instance.

of parameters, as listed in Table IV, by means of systematic experimentation. The Wilcoxon signed ranks test [57] is employed for the comparison of the different approaches. In this paper, the final experimental results were averaged over 50 trials. A summary of abbreviations used is given in Table V.

B. Modified MCARP Instances

A total of 87 modified MCARP instances were generated based on four sets of benchmark files [58] which may be downloaded from http://www.uv.es/~belengue/carp.html. Experimental results suggest that the conversion from CARP to MCARP is very quick and each CARP instance can be converted to a MCARP instance within a fraction of a second. The modified instances can be downloaded from http://xinglining.googlepages.com/home. The attributes of the modified MCARP instances are summarized in Table VI and the attributes of the underlying CARP instances are as follows.

1) *GDB*: 23 undirected CARP instances which were randomly constructed by Golden *et al.* [26]. These instances have 7–27 nodes and 11–55 edges and are denoted by gdb1, gdb2, ..., gdb23. In each instance of this set, all edges are required.
2) *KSHS*: Six standard CARP instances which were randomly generated by Kiuchi *et al.* [59]. All edges are required in each instance of this set. The instances (denoted by kshs1 to kshs6) have 6–10 nodes and 15 edges.
3) *BCCM*: 34 undirected CARP instances which were randomly generated by Benavent *et al.* [60] based on 10 different graphs. For each graph, some instances were generated (denoted $*$.A, $*$.B, ...) by changing the capacity of vehicles. In each instance of this set, all edges are required. The instances have 24–50 nodes and 34–97 edges.
4) *EGLESE*: 24 undirected instances constructed from real data by Li and Eglese [61]. These instances are very interesting because of their large size and real-world relevance. The data used to generate the instances represent road networks of two areas (east and south) of the county of Lancashire, U.K., which were digitized for the research of winter gritting. These EGLESE instances have sizes considerably greater than those in the preceding sets: 77–140 nodes, 98–190 edges, and up to 35 vehicles; also, many instances include nonrequired edges.

C. MCARP Instances with Known Optima

In order to further evaluate the HHEA framework, 20 MCARP instances were designed from scratch (these instances can be downloaded from http://xinglining.googlepages.com/home). The optima of these instances are known, and hence they can be applied directly to estimate the computational error of the different algorithms. Also, the gap between F_A and F^* (see Table V for an explanation of symbols), the gap

TABLE X

EXPERIMENTAL RESULTS OF HHEA-1 AND HHEA-2 AVERAGED OVER 50 TRIALS FOR THE MODIFIED MCARP INSTANCES

	HHEA-1				HHEA-2					HHEA-1				HHEA-2			
SN	F_A	F_D	F_A^E(%)	T_A(S)	F_A	F_D	F_A^E(%)	T_A(S)	SN	F_A	F_D	F_A^E(%)	T_A(S)	F_A	F_D	F_A^E(%)	T_A(S)
C1	564.6	6.9	3.6	210.7	**561.4**	16.9	3	213	C45	826.8	2.1	5.8	577.2	**821.4**	4.8	5.1	583.6
C2	635.1	3.8	3	218.8	**629.6**	4	2.1	221.2	C46	943.3	4.5	9.3	504.5	**937.4**	6.2	8.6	510.1
C3	542.6	1.9	4.2	221.5	**536**	3	2.9	223.9	C47	408.9	1.8	6.7	593.2	**405.3**	2.8	5.8	599.8
C4	558.4	2.2	6.7	203.1	**550.5**	4.4	5.2	205.4	C48	428.7	2.1	6.3	520.8	**421.9**	4.9	4.6	526.6
C5	720.2	6	2.2	223.1	**716.3**	6.3	1.6	225.6	C49	491.5	4.2	9.7	321.8	**489**	3.5	9.2	325.4
C6	603	7.1	1.9	219.1	**601.5**	5.1	1.6	221.5	C50	526.6	2.8	5.7	779	525.8	2.1	5.6	787.7
C7	598.4	3.4	7.3	207.4	597.8	4.1	7.2	209.7	C51	535.3	2	7.5	618.9	537.2	1.4	7.9	625.8
C8	548.7	4.8	12.2	377	**545.2**	4	11.4	381.2	C52	604.7	6.6	13.3	400.5	607.2	4.5	13.8	405
C9	620.6	15.9	14.5	369.6	**608.4**	20.8	12.3	373.7	C53	744.6	4.5	6.6	1221.9	**740.7**	6.4	6	1235.5
C10	543.1	3.6	2.7	236.2	**538.2**	3.4	1.8	238.9	C54	754.2	3.8	7.9	932.5	754.1	2.2	7.9	942.9
C11	755.2	5.2	5.1	328.7	**751.2**	4.9	4.6	332.4	C55	860.1	20.1	14.9	530	868.2	24.3	15.9	535.9
C12	819.1	8.6	4.2	214.5	**815.7**	5.2	3.7	216.9	C56	593.3	2.8	3	2850	**591.7**	1.8	2.7	2881.7
C13	1059.3	25.4	2.3	215.4	**1053.3**	13.3	1.7	217.8	C57	600.2	3.6	6.4	2021.5	599.6	1.6	6.3	2044
C14	210.7	1.2	4.5	195.1	**209.7**	1.4	4	197.3	C58	609.2	2.8	6.1	1648.2	**607.3**	3.1	5.7	1666.6
C15	118.4	1.2	5	207	**117.3**	1.5	4	209.3	C59	654.3	3	5.1	930.3	**649.8**	4.7	4.3	940.7
C16	254.8	1.3	5.6	222.9	**252.2**	1.8	4.6	225.4	C60	797.4	2.7	5.6	1771.1	**790.6**	4.3	4.7	1790.8
C17	175.8	1.4	3.2	222.2	**174.7**	1.7	2.5	224.7	C61	800.2	1.7	3.8	1356.3	799.3	2	3.6	1371.4
C18	337.6	1.7	3.7	276.3	**336.5**	1.6	3.3	279.3	C62	823.2	2.5	2.6	1321.1	823.6	2	2.6	1335.8
C19	97.1	1.4	6	163.7	**95.7**	1.4	4.5	165.5	C63	867.5	4.8	6.5	758.9	869.7	1.7	6.8	767.4
C20	238.8	2.2	7.2	211.4	**237.5**	1.5	6.6	213.7	C64	4215.2	51.3	24.8	511.3	**4197**	45.4	24.3	517
C21	318.7	1.9	1.8	221.5	**317.8**	1.5	1.5	223.9	C65	4719.6	101.4	26.9	527.5	4762.7	45.7	28	533.4
C22	402	2	2.7	277.2	**400.3**	1.7	2.2	280.2	C66	5534.5	24.4	20	481.3	**5429.7**	88.7	17.8	486.6
C23	476.5	1.1	4.1	327	**474.5**	1.2	3.7	330.6	C67	5488.6	63.3	26	674	**5435.7**	88.6	24.7	681.5
C24	20 950	109.1	4.7	195.1	**20 926**	111.9	4.6	197.3	C68	6901.2	70.7	19.5	716	**6814.1**	130.2	18	724
C25	15 014	94.1	2.8	189.7	**14 973**	7.3	2.5	191.8	C69	7586.3	113.9	41.3	553.2	7591.1	84.7	41.4	559.4
C26	17 118	172.7	4	190.5	17 103	187.1	3.9	192.6	C70	6868.2	130.1	29.7	841.1	**6780.2**	135	28	850.4
C27	19 957	92.2	12.3	192.4	**19 827**	166.6	11.6	194.5	C71	8814.6	139.6	21.9	920.6	**8586.9**	404	18.7	930.9
C28	23 277	337.6	14.5	199.4	**22 997**	599.3	13.1	201.6	C72	9850.5	44.7	30.6	785.8	9857.8	104.7	30.7	794.6
C29	22 414	89.6	9.9	196.4	**22 223**	176.4	9	198.6	C73	7003.1	298.8	32.7	1823.7	6994.9	363.6	32.5	1843.9
C30	311.1	1.7	5.7	525.1	**306.98**	2.9	4.3	531	C74	8297.4	168.9	41.4	1262	8372.4	261	42.7	1276.1
C31	332.1	3	12.8	455.4	334.7	2.4	13.7	460.5	C75	9502	320.6	53	1361.8	9951.2	165.7	60.3	1376.9
C32	387.9	7.2	12	304.6	390	3.1	12.6	308	C76	5476.9	48.5	16.1	1036.7	**5429.5**	64.5	15.1	1048.2
C33	398.1	2.3	4.2	414.5	**395.9**	2.7	3.6	419.1	C77	6599.8	37.9	22.8	960.7	6596.8	65.2	22.8	971.4
C34	407.8	2.4	5.6	333.4	**401**	3.9	3.9	337.2	C78	9261.5	145.8	26.4	989.2	**9141.3**	172.1	24.8	1000.2
C35	543.6	5.2	22.3	257	541.1	5.4	21.8	259.8	C79	11 333	662.2	31.1	2851.1	11 460	613.2	32.5	2882.8
C36	137.4	1	5.7	424.8	**135.9**	1	4.5	429.5	C80	14 066	268.1	36.7	2005.6	14 200	193.3	38	2027.9
C37	140.5	1.1	6.8	325	**138.5**	0.9	5.2	328.6	C81	15 274	343.2	43	1954	15 390	246	44.1	1975.7
C38	181	1.7	24.7	251.8	**173.4**	5	19.4	254.6	C82	10 422	52.2	20.8	2836.3	**10 362**	74.3	20.1	2867.9
C39	733.8	1.8	6.2	895.8	**726.8**	5.5	5.2	905.7	C83	13 683	286.2	28.6	2601.1	13 731	215.1	29.1	2630
C40	741.6	3.8	3.8	735	**740**	4.2	3.6	743.2	C84	17 079	193.1	39.1	2446.5	17 045	251.6	38.8	2473.7
C41	774.2	5.8	5.2	615.5	778.1	4.6	5.8	622.3	C85	12 759	834	26.7	2759.2	**12 591**	882.9	25	2789.9
C42	831.6	5.6	6	509.3	**826.8**	5.4	5.4	514.9	C86	14 960	249.9	38.1	2001.1	15 249	242.5	40.7	2023.3
C43	777.5	8.1	5.5	793.8	781.7	3.9	6	802.6	C87	18 141	271.5	36.1	2120.8	18 256	302.4	37	2144.4
C44	775.1	3.1	4.5	626.3	773	4.7	4.2	633.2									

Values in bold face indicate that the performance of HHEA-2 is significantly better than that of HHEA-1 (the confidence degree is 0.95).

between F^* and L_b, and the gap between F_A and L_b can be computed directly. The relationship between these gaps can then be further analyzed. The attributes of these MCARP instances are summarized in Table VII.

D. Experimental Results of EAM Heuristic

As mentioned before, the EAM heuristic is a deterministic approach and always obtains the same solution every time it is executed on the same instance. EAM can only find feasible solutions for 15 out of all modified MCARP instances as summarized in Table VIII. The average computational error of EAM is about 22% and the average computational time is about three seconds. Subsequently, EAM was not employed for further comparison because of its poor performance.

E. Comparison of ERPS and HHEA-3

Experimental results of ERPS and HHEA-3 are summarized in Table IX. Experimental results suggest that the performance of HHEA-3 is significantly better than that of ERPS. As mentioned before, ERPS was applied to generate some initial individuals in the main phase and restart phase of the HHEA-3. HHEA-3, however, then evolves these individuals into solutions that are at least as good as the initial ones. For this reason, HHEA-3 should be more powerful than ERPS and this was demonstrated by the final experimental results. For example, given the MCARP instance shown in Fig. 7, the total cost of optimal solutions is 58 and its optimal solution is as follows.

1) *Tour 1*: 1, 2, 3, 7 (arcs 1, 2, and 3 were serviced).
2) *Tour 2*: 6, 5, 4, 7 (arcs 6, 5, and 4 were serviced).

ERPS appends the most promising arc to the sequence of arcs until Q or T is exhausted. One optimal solution gained by ERPS (total cost is 68) is as follows.

1) *Tour 1*: 1, 2, 3, 4, 5, 6 (arcs 1, 2, 3, and 6 were serviced).
2) *Tour 2*: 6, 5, 4, 7 (arcs 5 and 4 were serviced).

ERPS is a stochastic algorithm with rather simple rules, and hence the greediness of its working mechanism results in inferior solutions. Different to ERPS, HHEA-3 generates the initial individuals using ERPS and ERUH, and improves the solution quality through crossover and mutation. The solution obtained by ERPS can be converted into a chromosome (1, 2, 3, 6, 5, 4). After obtaining its fitness by the split procedure, the actual optimal solution will be reached. In summary, ERPS cannot obtain high-quality solutions because of the greediness of its working mechanism, but this disadvantage can be overcome by HHEA through the fitness evaluations, crossover, and mutation.

F. Comparison of ERUH and HHEA-3

The experimental results of ERUH and HHEA-3 are listed in Table XI. The final results suggest that the performance of HHEA-3 is significantly better than that of ERUH. ERUH is a stochastic algorithm with simple rules but it can produce fairly good MCARP solutions. As mentioned previously, ERUH was used to generate some initial solutions in the main and

TABLE XI

EXPERIMENTAL RESULTS OF ERUH AND HHEA-3 AVERAGED OVER 50 TRIALS FOR SOLVING THE MODIFIED MCARP INSTANCES

	ERUH				HHEA-3					ERUH				HHEA-3			
SN	F_A	F_D	$F_A^E(\%)$	$T_A(S)$	F_A	F_D	$F_A^E(\%)$	$T_A(S)$	SN	F_A	F_D	$F_A^E(\%)$	$T_A(S)$	F_A	F_D	$F_A^E(\%)$	$T_A(S)$
C1	575.7	6.5	5.7	188.5	**562.7**	18.8	3.3	215.2	C45	824.4	1.3	5.5	437.8	**817.3**	5	4.6	589.6
C2	636.6	3.6	3.3	190.5	**622.9**	8.1	1.1	223.5	C46	944	2.4	9.4	321.2	**933.7**	8.6	8.2	515.3
C3	537.9	4.3	3.3	185.4	**532.7**	3.6	2.3	226.2	C47	410.8	1	7.2	473.2	**400.8**	4.8	4.6	605.9
C4	570.3	1.2	9	174.8	**544.3**	6.3	4	207.5	C48	427.3	1.4	6	367.6	**420.1**	4.8	4.2	532
C5	736.6	3.7	4.5	190.9	**710.6**	9	0.8	227.9	C49	495.9	1.5	10.7	243.2	**486.3**	4.2	8.5	328.7
C6	619.8	4.5	4.7	186.2	**595.5**	6.5	0.6	223.8	C50	526.1	1.2	5.6	697.1	525.5	2	5.5	795.7
C7	603.7	3.3	8.3	183.5	**591.5**	6.6	6.1	211.8	C51	536.2	1.1	7.7	542.2	**527.3**	5.7	5.9	632.2
C8	548.5	5.6	12.1	285.5	**537.2**	6.7	9.8	385.1	C52	606.8	4.1	13.7	317.6	**600.2**	7.1	12.5	409.1
C9	617.8	11.7	14	240.6	611.8	13.4	12.9	377.5	C53	742.8	3.1	6.3	1043.6	**738.6**	4.2	5.7	1248.1
C10	540.7	2.6	2.3	199.6	**535.9**	4.4	1.3	241.3	C54	752.8	2.8	7.7	793.8	**748.9**	5.4	7.2	952.5
C11	748.8	6.1	4.2	279.5	751.3	4.3	4.6	335.8	C55	843.2	13.5	12.6	412.8	854.8	20.1	14.2	541.4
C12	824.8	5.7	4.9	179.3	**815.2**	5.1	3.7	219.1	C56	593.4	1	3	2476.5	**586.4**	5.2	1.8	2911.2
C13	1054	2.9	1.8	189	1053.9	24.1	1.8	220	C57	600.1	1.4	6.4	1766.2	**596.6**	3.7	5.8	2064.9
C14	210.6	1.3	4.5	180.4	**205.7**	1.1	2	199.3	C58	607.3	1.8	5.7	1391.4	**603.8**	4.1	5.1	1683.6
C15	118.1	1.4	4.7	188.5	**113.8**	0.3	0.9	211.4	C59	655.5	1.2	5.3	721.1	**649.4**	4.7	4.3	950.3
C16	252.6	1.9	4.7	196.5	**249.9**	1.2	3.6	227.7	C60	791.6	1.8	4.8	1458.1	789.1	5.9	4.5	1809.1
C17	175.4	1.7	2.9	200.4	**170.8**	0.5	0.2	227	C61	796.3	1.9	3.3	1064.5	796	2.6	3.2	1385.4
C18	341.3	1.1	4.8	235.9	**332.6**	2.4	2.1	282.2	C62	824.1	2.8	2.7	870.7	**819.9**	4.4	2.2	1349.5
C19	96.9	1.4	5.8	164.6	92.6	1.4	1.1	167.2	C63	871.2	1.9	7	486.3	**864.7**	4.1	6.2	775.2
C20	247.5	3.7	11.1	185.2	**233.1**	2.9	4.6	215.9	C64	4260.7	29.7	26.2	388	**4237.6**	41.6	25.5	522.3
C21	320.9	1.8	2.5	204.6	**315.2**	3.1	0.6	226.2	C65	4806.8	25.3	29.2	325.2	**4733**	61.5	27.2	538.8
C22	401.8	1.9	2.6	226	**396.1**	2.8	1.1	283.1	C66	5548.6	57.1	20.3	286.4	**5461**	69.6	18.4	491.6
C23	×	×	×	×	**470.4**	2.1	2.8	334	C67	5473.3	63.3	25.6	486.1	5491.6	33.8	26	688.5
C24	20 977	99.8	4.9	169.4	**20 943**	102.5	4.7	199.3	C68	6955.8	57.3	20.4	403.6	**6884.1**	115.4	19.2	731.4
C25	15 036	96.1	2.9	169.8	15 027	92.8	2.9	193.8	C69	×	×	×	×	**7500.8**	220.2	39.7	565.1
C26	17 560	355.4	6.7	167.5	**17 016**	173.5	3.4	194.6	C70	6867.5	1	29.6	561.4	**6765.3**	185.7	27.7	859.1
C27	20 618	116.8	16	167.5	**19 925**	217.9	12.1	196.5	C71	8394.4	66.1	16	449.1	8754.5	297.8	21	940.4
C28	24 253	663.1	19.3	×	**22 849**	552.8	12.4	203.7	C72	×	×	×	×	**9800.1**	96.3	29.9	802.7
C29	22 422	124.6	10	173.5	22 380	110.2	9.8	200.6	C73	6714	139.6	27.2	1480.7	6953.3	396	31.7	1862.8
C30	309.3	1	5.1	423.1	**304.5**	2.8	3.4	536.4	C74	×	×	×	×	**8423.8**	188.3	43.6	1289.1
C31	336.9	2.7	14.4	312.1	**328.3**	4.3	11.5	465.2	C75	×	×	×	×	**9838.2**	328.1	58.4	1391
C32	395.7	2.8	14.2	217.2	**386.3**	8.3	11.5	311.1	C76	5555.2	22.9	17.7	583.9	**5465**	43.4	15.8	1058.9
C33	397.8	1.8	4.1	369.8	**390.3**	3	2.2	423.4	C77	6686.9	56.8	24.4	473.6	**6551.1**	110.9	21.9	981.3
C34	402.8	3.9	4.4	296.5	**397.4**	3.5	3	340.6	C78	×	×	×	×	**9237.9**	190.2	26.1	1010.4
C35	×	×	×	×	**539.3**	5.4	21.4	262.5	C79	10 902	117.2	26.1	1645.9	11 300	780.3	30.7	2912.3
C36	137.6	1.5	5.8	374.5	**133.4**	0.4	2.6	433.9	C80	×	×	×	×	**14 149**	248.9	37.5	2048.6
C37	140.3	1.2	6.6	286.9	**135.4**	0.8	2.9	332	C81	×	×	×	×	**15 247**	314.7	42.8	1995.9
C38	175.9	3.7	21.1	209.4	174.9	3.6	20.5	257.2	C82	10 452	54.9	21.1	1799.1	**10 373**	78.6	20.3	2897.2
C39	723	5.4	4.7	767.5	725	3.9	5	915	C83	×	×	×	×	**13 698**	302.9	28.7	2656.9
C40	739.8	1.8	3.6	600.2	737.3	2.6	3.2	750.8	C84	×	×	×	×	**17 081**	332.5	39.1	2499
C41	780.5	1.2	6.1	486.3	**775.6**	6.1	5.4	628.7	C85	12 147	178.6	20.6	2694.3	12 759	869.3	26.7	2818.4
C42	832.5	1.7	6.1	341.7	**822.5**	3.8	4.9	520.2	C86	×	×	×	×	**15 052**	269.3	38.9	2044
C43	773.2	4.2	4.9	666.9	772.2	8.1	4.7	810.8	C87	×	×	×	×	**18 467**	188.6	38.6	2166.3
C44	774.9	1.6	4.4	513.2	**767.7**	5.8	3.5	639.7									

Symbol "×" indicates that no feasible solution can be obtained adopting the given approach. Values in bold face indicate that the performance of HHEA-3 is significantly better than that of ERUH (the confidence degree is 0.95) on the given instance.

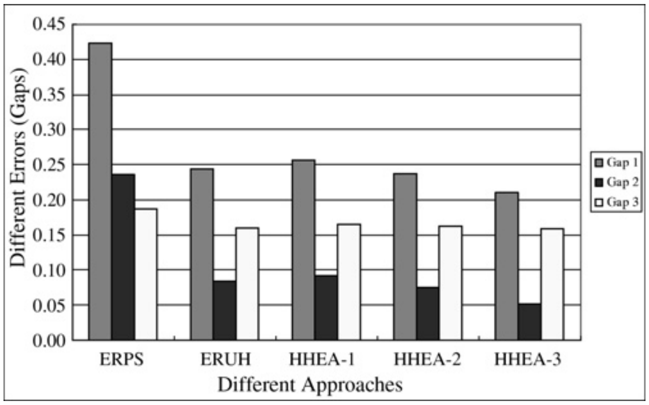


Fig. 10. Relationship between error F_A^E and error F_A^* ("Gap 1," "Gap 2," and "Gap 3") denotes error F_A^E, error F_A^*, and the additional error (gap between "Gap 1" and "Gap 2") of these different approaches.

restart phase of HHEA-3. As before, the fact that HHEA-3 evolves the initial individuals should guarantee solutions of improved quality. For this reason, HHEA-3 should be more powerful than ERUH and this was demonstrated by the final experimental results.

G. Comparison of HHEA-1 and HHEA-2

The experimental results of HHEA-1 and HHEA-2 are listed in Table X. According to the final results, the performance of HHEA-2 is significantly better than that of HHEA-1. HHEA-1 employs strategy S1 which selects an operator with equal probability. As displayed in Fig. 8, HHEA-1 selects each mutation operator with approximately equal probability (about 16%). HHEA-2 makes use of strategy S2 which selects each operator using the PIO. As shown in Fig. 8, HHEA-2 always selects the most promising mutation operator with high probability. For example, in the case of instance C12, the probability of improvement using the fourth mutation operator (15%) is higher than that of other mutation operators. Subsequently, HHEA-2 selects it with a high probability (25%). There are no bidirectional arcs in these MCARP instances and the first mutation operator only deals with the bidirectional arc, therefore, the probability of solution improvement of the first mutation operator is always 0. Also, HHEA-2 always selects an efficient crossover operator with high probability. For example, in the case of instance C12, the probability of improvement given LOX (42%) is higher than that of OX, and therefore, HHEA-2 selects LOX with high probability (63%). In summary, HHEA-2 always selects the more appropriate operator and the final results indicate that HHEA-2 outperforms HHEA-1.

H. Comparison of HHEA-2 and HHEA-3

The experimental results of HHEA-2 and HHEA-3 are listed in Table XII. According to the final results, the performance of HHEA-3 is significantly better than that of HHEA-2. In fact,

TABLE XII

EXPERIMENTAL RESULTS OF HHEA-2 AND HHEA-3 AVERAGED OVER 50 TRIALS FOR SOLVING MODIFIED MCARP INSTANCES

	HHEA-2				HHEA-3					HHEA-2				HHEA-3			
SN	F_A	F_D	F_A^E(%)	T_A(S)	F_A	F_D	F_A^E(%)	T_A(S)	SN	F_A	F_D	F_A^E(%)	T_A(S)	F_A	F_D	F_A^E(%)	T_A(S)
C1	561.4	16.9	3.0	213	562.7	18.8	3.3	215.2	C45	821.4	4.8	5.1	583.6	**817.3**	5	4.6	589.6
C2	629.6	4	2.1	221.2	**622.9**	8.1	1.1	223.5	C46	937.4	6.2	8.6	510.1	933.7	8.6	8.2	515.3
C3	536	3	2.9	223.9	**532.7**	3.6	2.3	226.2	C47	405.3	2.8	5.8	599.8	**400.8**	4.8	4.6	605.9
C4	550.5	4.4	5.2	205.4	**544.3**	6.3	4	207.5	C48	421.9	4.9	4.6	526.6	**420.1**	4.8	4.2	532
C5	716.3	6.3	1.6	225.6	**710.6**	9	0.8	227.9	C49	489	3.5	9.2	325.4	**486.3**	4.2	8.5	328.7
C6	601.5	5.1	1.6	221.5	**595.5**	6.5	0.6	223.8	C50	525.8	2.1	5.6	787.7	525.5	2	5.5	795.7
C7	597.8	4.1	7.2	209.7	**591.5**	6.6	6.1	211.8	C51	537.2	1.4	7.9	625.8	**527.3**	5.7	5.9	632.2
C8	545.2	4	11.4	381.2	**537.2**	6.7	9.8	385.1	C52	607.2	4.5	13.8	405	**600.2**	7.1	12.5	409.1
C9	608.4	20.8	12.3	373.7	611.8	13.4	12.9	377.5	C53	740.7	6.4	6	1235.5	**738.6**	4.2	5.7	1248.1
C10	538.2	3.4	1.8	238.9	**535.9**	4.4	1.3	241.3	C54	754.1	2.2	7.9	942.9	**748.9**	5.4	7.2	952.5
C11	751.2	4.9	4.6	332.4	751.3	4.3	4.6	335.8	C55	868.2	24.3	15.9	535.9	**854.8**	20.1	14.2	541.4
C12	815.7	5.2	3.7	216.9	815.2	5.1	3.7	219.1	C56	591.7	1.8	2.7	2881.7	**586.4**	5.2	1.8	2911.2
C13	1053.3	13.3	1.7	217.8	1053.9	24.1	1.8	220	C57	599.6	1.6	6.3	2044	**596.6**	3.7	5.8	2064.9
C14	209.7	1.4	4	197.3	**205.7**	1.1	2	199.3	C58	607.3	3.1	5.7	1666.6	**603.8**	4.1	5.1	1683.6
C15	117.3	1.5	4	209.3	**113.8**	0.3	0.9	211.4	C59	649.8	4.7	4.3	940.7	649.4	4.7	4.3	950.3
C16	252.2	1.8	4.6	225.4	**249.9**	1.2	3.6	227.7	C60	790.6	4.3	4.7	1790.8	789.1	5.9	4.5	1809.1
C17	174.7	1.7	2.5	224.7	**170.8**	0.5	0.2	227	C61	799.3	2	3.6	1371.4	**796**	2.6	3.2	1385.4
C18	336.5	1.6	3.3	279.3	**332.6**	2.4	2.1	282.2	C62	823.6	2	2.6	1335.8	**819.9**	4.4	2.2	1349.5
C19	95.7	1.4	4.5	165.5	**92.6**	1.4	1.1	167.2	C63	869.7	1.7	6.8	767.4	**864.7**	4.1	6.2	775.2
C20	237.5	1.5	6.6	213.7	**233.1**	2.9	4.6	215.9	C64	4197	45.4	24.3	517	4237.6	41.6	25.5	522.3
C21	317.8	1.5	1.5	223.9	315.2	3.1	0.6	226.2	C65	4762.7	45.7	28	533.4	**4733**	61.5	27.2	538.8
C22	400.3	1.7	2.2	280.2	**396.1**	2.8	1.1	283.1	C66	5429.7	88.7	17.8	486.6	5461	69.6	18.4	491.6
C23	474.5	1.2	3.7	330.6	**470.4**	2.1	2.8	334	C67	5435.7	88.6	24.7	681.5	5491.6	33.8	26	688.5
C24	20 926	111.9	4.6	197.3	20 943	102.5	4.7	199.3	C68	6814.1	130.2	18	724	6884.1	115.4	19.2	731.4
C25	14 973	7.3	2.5	191.8	15 027	92.8	2.9	193.8	C69	7591.1	84.7	41.4	559.4	7500.8	220.2	39.7	565.1
C26	17 103	187.1	3.9	192.6	**17 016**	173.5	3.4	194.6	C70	6780.2	135	28	850.4	6765.3	185.7	27.7	859.1
C27	19 827	166.6	11.6	194.5	19 925	217.9	12.1	196.5	C71	8586.9	404	18.7	930.9	8754.5	297.8	21	940.4
C28	22 997	599.3	13.1	201.6	22 849	552.8	12.4	203.7	C72	9857.8	104.7	30.7	794.6	**9800.1**	96.3	29.9	802.7
C29	22 223	176.4	9	198.6	22 380	110.2	9.8	200.6	C73	6994.9	363.6	32.5	1843.9	6953.3	396	31.7	1862.8
C30	307	2.9	4.3	531	**304.5**	2.8	3.4	536.4	C74	8372.4	261	42.7	1276.1	8423.8	188.3	43.6	1289.1
C31	334.7	2.4	13.7	460.5	**328.3**	4.3	11.5	465.2	C75	9951.2	165.7	60.3	1376.9	**9838.2**	328.1	58.4	1391
C32	390	3.1	12.6	308	386.3	8.3	11.5	311.1	C76	5429.5	64.5	15.1	1048.2	5465	43.4	15.8	1058.9
C33	395.9	2.7	3.6	419.1	**390.3**	3	2.2	423.4	C77	6596.8	65.2	22.8	971.4	**6551.1**	110.9	21.9	981.3
C34	401	3.9	3.9	337.2	**397.4**	3.5	3	340.6	C78	9141.3	172.1	24.8	1000.2	9237.9	190.2	26.1	1010.4
C35	541.1	5.4	21.8	259.8	539.3	5.4	21.4	262.5	C79	11 460	613.2	32.5	2882.8	**11 300**	780.3	30.7	2912.3
C36	135.9	1	4.5	429.5	**133.4**	0.4	2.6	433.9	C80	14 200	193.3	38	2027.9	14 149	248.9	37.5	2048.6
C37	138.5	0.9	5.2	328.6	**135.4**	0.8	2.9	332	C81	15 390	246	44.1	1975.7	15 247	314.7	42.8	1995.9
C38	173.4	5	19.4	254.6	174.9	3.6	20.5	257.2	C82	10 362	74.3	20.1	2867.9	10 373	78.6	20.3	2897.2
C39	726.8	5.5	5.2	905.7	725	3.9	5	915	C83	13 731	215.1	29.1	2630	13 698	302.9	28.7	2656.9
C40	740	4.2	3.6	743.2	**737.3**	2.6	3.2	750.8	C84	17 045	251.6	38.8	2473.7	17 081	332.5	39.1	2499
C41	778.1	4.6	5.8	622.3	775.6	6.1	5.4	628.7	C85	12 591	882.9	25	2789.9	12 759	869.3	26.7	2818.4
C42	826.8	5.4	5.4	514.9	**822.5**	3.8	4.9	520.2	C86	15 249	242.5	40.7	2023.3	**15 052**	269.3	38.9	2044
C43	781.7	3.9	6	802.6	772.2	8.1	4.7	810.8	C87	18 256	302.4	37	2144.4	18 467	188.6	38.6	2166.3
C44	773	4.7	4.2	633.2	**767.7**	5.8	3.5	639.7									

Values in bold face indicate that the performance of HHEA-3 is significantly better than that of HHEA-2 (the confidence degree is 0.95).

TABLE XIII

EXPERIMENTAL RESULTS OF FIVE DIFFERENT APPROACHES AVERAGED OVER 50 TRIALS FOR SOLVING 20 INSTANCES WITH KNOWN OPTIMA

	ERPS				ERUH				HHEA-1				HHEA-2				HHEA-3			
SN	F_A	F_A^E(%)	F_A^*(%)	T_A(S)	F_A	F_A^E(%)	F_A^*(%)	T_A(S)	F_A	F_A^E(%)	F_A^*(%)	T_A(S)	F_A	F_A^E(%)	F_A^*(%)	T_A(S)	F_A	F_A^E(%)	F_A^*(%)	T_A(S)
E01	×	×	×	×	194	18.3	7.8	272.57	194	18.3	7.8	213.75	192.4	17.3	6.9	219.2	184	12.2	2.2	275
E02	322	47.7	22.9	89.62	270	23.9	3.1	222.14	270	23.9	3.1	178.02	266.8	22.4	1.8	179.63	262	20.2	0.0	178.84
E03	×	×	×	×	×	×	×	×	254	46.0	21.0	250.85	235.6	35.4	12.2	275.31	226	29.9	7.6	268.45
E04	296	35.8	13.0	67.98	270	23.9	3.1	222.5	270	23.9	3.1	176.96	269.2	23.5	2.7	179.73	266	22.0	1.5	182.48
E05	310	38.4	29.2	67.95	264	17.9	10.0	235.02	264	17.9	10.0	175.47	262	17.0	9.2	175.96	260	16.1	8.3	219.34
E06	396	33.8	22.2	82.53	350	18.2	8.0	195.49	350	18.2	8.0	154.39	348	17.6	7.4	153.12	342	15.5	5.6	153.57
E07	×	×	×	×	292	30.4	7.4	231.41	292	30.4	7.4	176.37	283.6	26.6	4.3	174.16	280	25.0	2.9	173.17
E08	388	31.1	19.8	69.19	350	18.2	8.0	193.56	350	18.2	8.0	149.6	347.2	17.3	7.2	153.2	340	14.9	4.9	154.81
E09	488	48.8	35.6	74.09	400	22.0	11.1	435.72	400	22.0	11.1	348.5	396.8	21.0	10.2	344.27	398	21.3	10.6	347.01
E10	585	38.0	18.9	82.46	530	25.0	7.7	318.98	530	25.0	7.7	260.52	526	24.1	6.9	254.74	520	22.6	5.7	254.19
E11	×	×	×	×	460	32.2	11.7	434.65	448	28.7	8.7	356.35	440.4	26.6	6.9	348.09	440	26.4	6.8	347.01
E12	583.6	37.6	18.6	75.84	530	25.0	7.7	318.13	530	25.0	7.7	252.69	527	24.3	7.1	254.73	520	22.6	5.7	254.9
E13	648	50.0	35.0	78.88	558	29.2	16.3	743.34	538	24.5	12.1	615.76	530.4	22.8	10.5	590.46	524	21.3	9.2	596.56
E14	815.2	46.6	22.0	85.57	707.6	27.3	5.9	515.88	716	28.8	7.2	430.35	709.4	27.6	6.2	416.22	702	26.3	5.1	416.59
E15	×	×	×	×	612	31.9	15.0	737.56	596	28.4	12.0	644.15	581.6	25.3	9.3	616.22	568	22.4	6.8	621.1
E16	820	47.5	22.8	83.7	707.6	27.3	5.9	516.18	718	29.1	7.5	416.95	710.6	27.8	6.4	413.95	700	25.9	4.8	418.05
E17	×	×	×	×	584	20.7	8.1	1039.28	580	19.8	7.4	904.35	569	17.6	5.4	871.8	562	16.1	4.1	882.18
E18	945	50.0	25.3	539.52	796.2	26.4	5.6	693.83	806	27.9	6.9	612.5	793.8	26.0	5.3	615.71	786	24.8	4.2	616.84
E19	×	×	×	163.13	610	18.7	10.9	1033.41	606	17.9	10.2	858.56	587.6	14.3	6.8	865.33	582	13.2	5.8	845.31
E20	924.6	46.8	22.6	532.34	798	26.7	5.8	698.31	804	27.6	6.6	611.94	789.6	25.3	4.7	620.68	788	25.1	4.5	616.14
Average:		42.5	23.7			24.4	8.4			25.1	8.7			23.0	6.9			21.2	5.3	

Symbol "×" indicates that none of the feasible solutions can be obtained by the given approach.

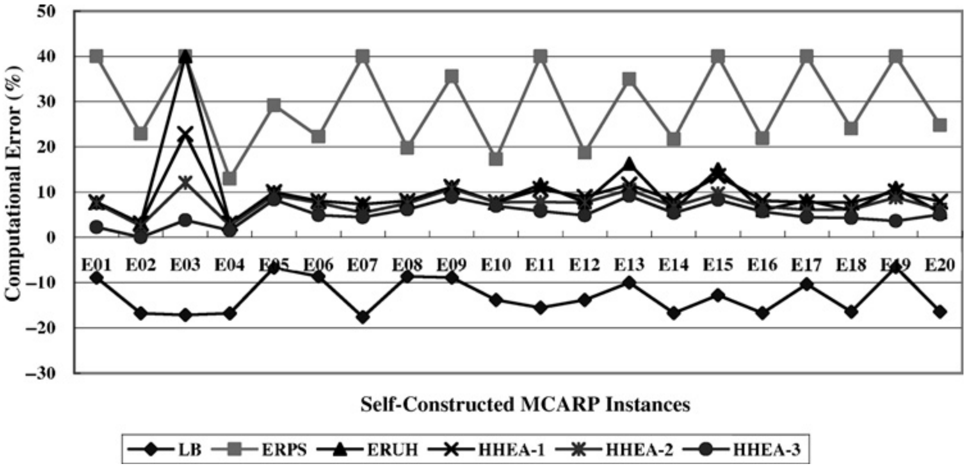


Fig. 11. Relationship between F_A, F^*, and L_b. For each instance with known optimum, F^* is regarded as the datum mark (x-axis), F_A is located above the datum mark, and L_b is located under the datum mark. In this figure, both F_A and L_b adopt the corresponding gap to the F^*. Above the datum mark, if the gap is reached to 40%, then it means that it cannot obtain a feasible solution of this instance by the given approach.

TABLE XIV
FINAL RESULTS OF MA AND HHEA-3 AVERAGED OVER 50 TRIALS FOR SOLVING 23 SIMPLE MCARP INSTANCES

	MA				HHEA-3			
SN	F_A	F_D	F_A^E(%)	T_A(S)	F_A	F_D	F_A^E(%)	T_A(S)
C1	573.7	23.5	5.3	191.3	**562.7**	18.8	3.3	215.2
C2	636.1	14.8	3.2	195.6	**622.9**	8.1	1.1	223.5
C3	535.9	6.9	2.9	189.3	**532.7**	3.6	2.3	226.2
C4	561.9	10.2	7.4	178.6	**544.3**	6.3	4	207.5
C5	734.4	11.7	4.2	188.7	**710.6**	9	0.8	227.9
C6	614.5	9.8	3.8	181.5	**595.5**	6.5	0.6	223.8
C7	603.9	11.3	8.3	187.9	**591.5**	6.6	6.1	211.8
C8	552.3	10.2	12.9	290.4	**537.2**	6.7	9.8	385.1
C9	612.5	15.9	13	285.3	611.8	13.4	12.9	377.5
C10	541	6.7	2.3	217.3	**535.9**	4.4	1.3	241.3
C11	752.2	6.9	4.7	273.4	751.3	4.3	4.6	335.8
C12	824.9	8.4	4.9	185.9	**815.2**	5.1	3.7	219.1
C13	1052.2	30.8	1.6	181.4	1053.9	24.1	1.8	220
C14	207	3.5	2.7	185.1	**205.7**	1.1	2	199.3
C15	114.9	1.2	1.9	183.6	**113.8**	0.3	0.9	211.4
C16	250.6	1.9	3.9	203.8	**249.9**	1.2	3.6	227.7
C17	172.3	1.2	1.1	199.6	**170.8**	0.5	0.2	227
C18	339	3.1	4.1	228.7	**332.6**	2.4	2.1	282.2
C19	92.8	1.4	1.3	168.2	92.6	1.4	1.1	167.2
C20	243.1	3.7	9.1	190.3	**233.1**	2.9	4.6	215.9
C21	318.5	3.8	1.7	213.5	**315.2**	3.1	0.6	226.2
C22	399	4.9	1.9	219.8	**396.1**	2.8	1.1	283.1
C23	474.5	3.5	3.7	320.3	**470.4**	2.1	2.8	334

Values in bold face indicate that the performance of HHEA-3 is significantly better than that of MA (the confidence degree is 0.95).

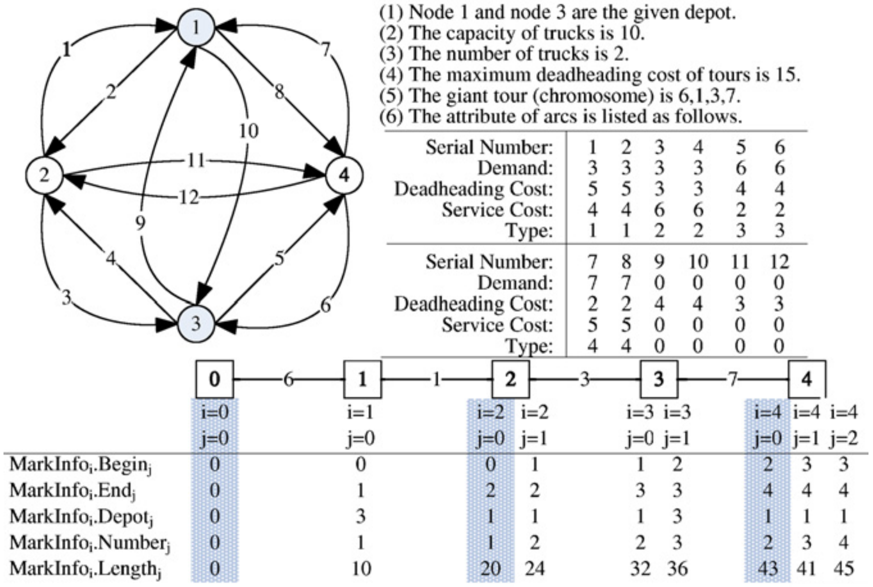
HHEA-2 uses strategy S3 which selects a broken position with the equal probability, while HHEA-3 makes use of strategy S4 which selects a broken position using the AAPI information. It follows that the probability of solution improvement by each operator in HHEA-3 should be higher than that in HHEA-2 and this is indeed the case as shown in Fig. 9. HHEA-3 selects a favorable broken position for each crossover (mutation) operation employing the AAPI information and thereby protecting promising subsequences in the current chromosome. The probability of solution improvement by each operator is thus largely improved. For example, executing HHEA-2 on instance C55, about 30% of crossovers of all crossovers executed by OX result in an improved solution, and about 44% of crossovers of all crossovers executed using LOX obtain an improved solution. Comparatively, among the whole evolutionary processes of HHEA-3, about 39% of OX crossovers and 53% of LOX crossovers result in an improved solution.

I. Comparison of MA and HHEA-3

In this section, 23 simple MCARP instances (C1–C23) were used to compare the performance of MA [10] and HHEA-3. As we consider a maximum service time, MA cannot be directly applied to deal with MCARP instances. Therefore, the split procedure in MA is replaced by the extended split procedure. Final experimental results of MA and HHEA-3 are listed in Table XIV. Experimental results suggest that the performance of HHEA-3 is significantly better than that of MA. In MA, most initial individuals are produced using random generation. In fact, the quality of the initial population will largely affect the final performance of optimization methods. In HHEA, most initial individuals are generated using ERPS and ERUH and PIO is used to select the appropriate operator for selection, crossover, and mutation and AAPI is applied to decide the appropriate broken position for crossover and mutation. For this reason, HHEA-3 should be more powerful than MA and it was demonstrated by the final results.

J. Experimental Results of Instances with Known Optima

The computational error of each optimization methods should be estimated by the gap F_A^* between F_A and F^*. Unfortunately, F^* is unknown for the modified MCARP instances and the computational error is, therefore, replaced by the gap F_A^E between F_A and L_b. However, there still exists a gap between F^* and L_b (i.e., L_b is not necessarily tight) and in order to completely understand the relationship between F_A^E and F_A^*, some experiments were implemented on the MCARP instances where F^* is known. The Three HHEA variants as well as the two extended heuristics (ERPS and ERUH) were used to solve these instances and the results are shown in Table XIII. As displayed in Fig. 10, there exists a large gap between F_A^E and F_A^*. For example, the error F_A^E of HHEA-3 is 21.2%, whereas the error F_A^* of HHEA-3 is 5.3%. Given the



(1) Node 1 and node 3 are the given depot.
(2) The capacity of trucks is 10.
(3) The number of trucks is 2.
(4) The maximum deadheading cost of tours is 15.
(5) The giant tour (chromosome) is 6,1,3,7.
(6) The attribute of arcs is listed as follows.

Serial Number:	1	2	3	4	5	6
Demand:	3	3	3	3	6	6
Deadheading Cost:	5	5	3	3	4	4
Service Cost:	4	4	6	6	2	2
Type:	1	1	2	2	3	3

Serial Number:	7	8	9	10	11	12
Demand:	7	7	0	0	0	0
Deadheading Cost:	2	2	4	4	3	3
Service Cost:	5	5	0	0	0	0
Type:	4	4	0	0	0	0

	i=0	i=1	i=2	i=2	i=3	i=3	i=4	i=4	i=4
	j=0	j=0	j=0	j=1	j=0	j=1	j=0	j=1	j=2
$\text{MarkInfo}_i.\text{Begin}_j$	0	0	0	1	1	2	2	3	3
$\text{MarkInfo}_i.\text{End}_j$	0	1	2	2	3	3	4	4	4
$\text{MarkInfo}_i.\text{Depot}_j$	0	3	1	1	1	3	1	1	1
$\text{MarkInfo}_i.\text{Number}_j$	0	1	1	2	2	3	2	3	4
$\text{MarkInfo}_i.\text{Length}_j$	0	10	20	24	32	36	43	41	45

Fig. 12. Simple example of the proposed multimark approach. After the operation, there are three different marks on virtual node 4. In fact, the best tradeoff is the leftmost one (total cost is 43, while the number of trucks is 2).

error F_A^E of HHEA-3, the actual error F_A^* is 5.3% (about 25%) and the additional error is 15.9% (about 75%). In summary, there is a gap between L_b and F^* and this gap (44–75%; Fig. 10) makes error F_A^E larger than error F_A^*.

Fig. 11 highlights the relationship between L_b, F^*, and F_A. From this figure, the following conclusions may be reached: First, there exists a large gap between L_b and F^*. Among these 20 instances, the average gap between L_b and F^* is 12.97% (the smallest is 6.7%, the largest is 16.8%). Second, the F_A obtained by HHEA-3 is very close to F^*. Among these 20 instances, the average gap between F_A and F^* is 5.19% (the smallest is 0 and the largest is 9.2%).

VII. Conclusion

This paper has proposed a novel EA called HHEA that employs classical heuristics as well as heuristic information to deal with the MCARP, an extension of the classical CARP. The experimental results demonstrate that HHEA significantly outperforms other heuristics. The main contributions of this paper may be summarized as follows: considering practical requirements, the MCARP was introduced. Both multidepots and maximum service time were taken into account to guarantee the availability of timely service. This paper is a step closer toward a more practical version of CARP problems. Also, three classical heuristics were extended to the MCARP and two of those, ERPS and ERUH, were used to initialize the population of HHEA. Furthermore, heuristic information was used to guide the evolutionary search of HHEA. As mentioned above, the initial individuals were generated using ERPS and ERUH. This not only improves the quality of the initial population but also ensures that feasible solutions are available. Furthermore, heuristic information was extracted from near-optimal solutions that are acquired throughout the search. This information was subsequently used to guide the subsequent evolution. In particular, the PIO was employed to select appropriate selection, crossover, and mutation operators and the AAPI was used to decide favorable positions for crossover and mutation.

There are many aspects that we intend to address in the near future. In particular, we would like to construct an improved approach for estimating the LB of MCARP instances. In this paper, the MCNDLB [23] was employed to estimate the LBs of MCARP instances. As mentioned before, the average gap between L_b and F^* is 12.97% among the 20 instances with known global optimum. Therefore, an improved algorithm should be developed for estimating the LB of MCARP instances.

Acknowledgment

The authors would like to thank all the members of the Evolutionary Algorithms for Dynamic Optimization Problems reading group (CERCIA) for their valued suggestions. In particular, they would like to thank T. T. Nguyen, Dr. H. Handa, and Dr. T. Ray for their constructive comments and suggestions. They are also grateful to the CERCIA group at the School of Computer Science, University of Birmingham, for providing a working environment and experimental facilities.

Appendix A
Additional MCARP Concepts

Additional MCARP concepts are listed as follows.

1) *Mixed Connected Graph:* The mixed graph allows for two kinds of links—undirected edges and directed arcs. In this paper, a mixed connected graph is transformed into a directed connected graph by replacing each edge by two arcs with opposing directions.

2) *Type of Arcs:* Each edge can be regarded as a bidirectional arc and every arc can be regarded as one unidirectional arc. In order to distinguish these two types of arc, the 'type' attribute is defined for every arc. If the 'type' attribute of an arc is larger than 0, then it is a part of bidirectional arcs; else it is an unidirectional arc. For any two arcs corresponding to a required bidirectional arc, only one of them must be serviced in a feasible solution.
3) *Deadheading Cost:* Deadheading means traveling without service. No matter whether a link in the given route is serviced or not, it will incur the predefined deadheading cost.
4) *Initial Configuration:* The number of trucks in each depot (initial configuration) is not considered in this paper. That is, there are an abundant number of trucks in every depot, but the total number of available trucks across all depots is N_2.
5) *Service Time:* For the sake of simplicity, only the traveling time of vehicles is considered as the service time. There is a linear relationship between the deadheading cost and the service time and the deadheading cost may be transformed to the traveling distance of trucks by multiplying by a coefficient α.
6) *Service Indicator:* $f_j(\mu_{ij})$ is a Boolean variable such that if arc μ_{ij} is serviced in route l_j, then $f_j(\mu_{ij}) = 1$; else $f_j(\mu_{ij}) = 0$.

Appendix B
A Simple Example of the Multimark Operation

A simple example of the multimark operation is displayed in Fig. 12. In this example, there are four nodes, two depots (node 1 and node 3) and 12 arcs. For the sake of simplicity, a maximum deadheading cost of tours is employed to replace the constraint of maximum service time. From the final result of the mark operations, it can be seen that the total cost of the optimal solution, subject to the arc sequence of chromosome (6, 1, 3, 7), is 43. The corresponding optimal solution is listed as follows.

1) *Tour 1:* $8, 6, 4, 1$ (arcs 6 and 1 were serviced).
2) *Tour 2:* $10, 4, 11, 7$ (arcs 4 and 7 were serviced).

In these tours, every number corresponds to the serial number of one corresponding arc. If there are three trucks available, then the total cost of optimal solutions would be decreased to 41. The corresponding optimal solution is listed as follows.

1) *Tour 1:* $5, 6$ (arc 6 was serviced).
2) *Tour 2:* $2, 3, 9$ (arcs 2 and 3 were serviced).
3) *Tour 3:* $8, 7$ (arc 7 was serviced).

References

[1] J. Araoz, E. Fernandez, and C. Zoltan, "Privatized rural postman problems," *Comput. Oper. Res.*, vol. 33, no. 12, pp. 3432–3449, 2006.
[2] T. Bektas and S. Elmastas, "Solving school bus routing problems through integer programming," *J. Oper. Res. Soc.*, vol. 58, no. 12, pp. 1599–1604, 2007.
[3] G. A. Tobin and R. Brinkmann, "The effectiveness of street sweepers in removing pollutants from road surfaces in Florida," *J. Environ. Sci. Health (Part A)*, vol. 37, no. 9, pp. 1687–1700, 2002.
[4] E. Dijkgraaf and R. H. J. M. Gradus, "Fair competition in the refuse collection market," *Appl. Econ. Lett.*, vol. 14, no. 10, pp. 701–704, 2007.
[5] H. Handa, L. Chapman, and X. Yao, "Robust route optimization for gritting/salting trucks: A CERCIA experience," *IEEE Comput. Intell. Mag.*, vol. 1, no. 1, pp. 6–9, Feb. 2006.
[6] M. Dror, H. Stern, and P. Trudeau, "Postman tour on a graph with precedence relation on arcs," *Networks*, vol. 17, no. 3, pp. 283–294, 1987.
[7] H. S. Han, J. J. Yu, C. G. Park, and J. G. Lee, "Development of inspection gauge system for gas pipeline," *Korean Soc. Mech. Eng. Int. J.*, vol. 18, no. 3, pp. 370–378, Mar. 2004.
[8] J. Tao, P. W. Que, and Z. S. Tao, "Magnetic flux leakage device for offshore oil pipeline defect inspection," *Mater. Perform.*, vol. 44, no. 10, pp. 48–51, 2005.
[9] H. I. Stern and M. Dror, "Routing electric meter readers," *Comput. Oper. Res.*, vol. 6, no. 4, pp. 209–223, 1979.
[10] P. Lacomme, C. Prins, and W. Ramdane-Cherif, "Competitive memetic algorithms for arc routing problems," *Ann. Oper. Res.*, vol. 131, nos. 1–4, pp. 159–185, 2004.
[11] B. Smith. (2004 Jan.). *Highway Maintenance Network Management Policies and Standards*, Regulation of Environment and Transport, Cambridgeshire County Council, Cambridge, U.K.
[12] B. L. Golden and R. T. Wong, "Capacitated arc routing problems," *Networks*, vol. 11, no. 3, pp. 305–315, 1981.
[13] C. Rego, "A subpath ejection method for the vehicle routing problem," *Manage. Sci.*, vol. 44, no. 10, pp. 1447–1459, 1998.
[14] E. Minieka, "Chinese postman problem for mixed networks," *Manage. Sci.*, vol. 25, no. 7, pp. 643–648, 1979.
[15] L. A. Moreira, J. F. Oliveira, A. M. Gomes, and J. Soeiro Ferreira, "Heuristics for a dynamic rural postman problem," *Comput. Oper. Res.*, vol. 34, no. 11, pp. 3281–3294, Nov. 2007.
[16] G. Fleury, P. Lacomme, and C. Prins, "Evolutionary algorithms for stochastic arc routing problems," in *Applications of Evolutionary Computing*, Lecture Notes in Computer Science 3005. Berlin, Germany: Springer, 2004, pp. 501–512.
[17] A. A. Assad, B. L. Golden, and W.-L. Pearn, "The capacitated Chinese postman problem: Lower bounds and solvable cases," *Amer. J. Math. Manage. Sci.*, vol. 7, nos. 1–2, pp. 63–88, 1987.
[18] M. Fischetti and P. Toth, "A polyhedral approach to the asymmetric traveling salesman problem," *Manage. Sci.*, vol. 43, no. 11, pp. 1520–1536, 1997.
[19] A. Corberan, I. Plana, and J. M. Sanchis, "A branch and cut algorithm for the windy general routing problem and special cases," *Networks*, vol. 49, no. 4, pp. 245–257, 2007.
[20] P. Lacomme, C. Prins, and W. Ramdane-Cherif, "Evolutionary algorithms for periodic arc routing problems," *Eur. J. Oper. Res.*, vol. 165, no. 2, pp. 535–553, Sep. 2005.
[21] R. Hirabayashi, N. Nishida, and Y. Saruwatari, "Node duplication lower bounds for the capacitated arc routing problems," *J. Oper. Res. Soc. Jpn.*, vol. 35, no. 2, 119–133, 1992.
[22] W. L. Pearn, "New lower bounds for the capacitated arc routing problem," *Networks*, vol. 18, no. 12, pp. 181–191, 1988.
[23] S. Wohlk, "New lower bound for the capacitated arc routing problem," *Comput. Oper. Res.*, vol. 33, no. 12, 3458–3472, 2006.
[24] J. M. Belenguer and E. Benavent, "The capacitated arc routing problem: Valid inequalities and facets," *Comput. Optimizat. Applicat.*, vol. 10, no. 2, pp. 165–187, 1998.
[25] R. Hirabayashi, Y. Saruwatari, and N. Nishida, "Tour construction algorithm for the capacitated arc routing problem," *Asia-Pacific J. Oper. Res.*, vol. 9, no. 2, pp. 155–175, 1992.
[26] B. L. Golden, J. S. DeArmon, and E. K. Baker, "Computational experiments with algorithms for a class of routing problems," *Comput. Oper. Res.*, vol. 10, no. 1, pp. 47–59, 1983.
[27] W. L. Pearn, "Approximate solutions for the capacitated arc routing problem," *Comput. Oper. Res.*, vol. 16, no. 6, pp. 589–600, 1989.
[28] G. Ulusoy, "The fleet size and mix problem for capacitated arc routing," *Eur. J. Oper. Res.*, vol. 22, no. 3, pp. 329–337, Dec. 1985.
[29] W. L. Pearn, "Augment-insert algorithms for the capacitated arc routing problem," *Comput. Oper. Res.*, vol. 18, no. 2, pp. 189–198, 1991.
[30] A. Hertz, G. Laporte, and M. Mittaz, "A tabu search heuristic for the capacitated arc routing problem," *Oper. Res.*, vol. 48, no. 1, pp. 129–135, 2000.
[31] J. Branda and R. Eglese, "A deterministic tabu search algorithm for the capacitated arc routing problem," *Comput. Oper. Res.*, vol. 35, no. 4, pp. 1112–1126, 2008.
[32] Y. Mei, K. Tang, and X. Yao, "A global repair operator for capacitated arc routing problem," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 3, pp. 723–734, Jun. 2009.

[33] A. Hertz and M. Mittaz, "A variable neighbourhood descent algorithm for the undirected capacitated arc routing problem," *Transp. Sci.*, vol. 35, no. 4, pp. 425–434, 2001.

[34] P. Beullens, L. Muyldermans, D. Cattrysse, and D. Van Oudheusden, "A guided local search heuristic for the capacitated arc routing problem," *Eur. J. Oper. Res.*, vol. 147, no. 3, pp. 629–643, Jun. 2003.

[35] P. P. Repoussis, C. D. Tarantilis, and G. Ioannou, "Arc-guided evolutionary algorithm for the vehicle routing problem with time windows," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 624–647, Jun. 2009.

[36] K. Tang, Y. Mei, and X. Yao, "Memetic algorithm with extended neighborhood search for capacitated arc routing problems," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 1151–1166, Oct. 2009.

[37] R. W. Eglese, "Routing winter gritting vehicles," *Discrete Appl. Math.*, vol. 48, no. 3, pp. 231–244, 1994.

[38] P. Lacomme, C. Prins, and M. Sevaux, "A genetic algorithm for a bi-objective capacitated arc routing problem," *Comput. Oper. Res.*, vol. 33, no. 12, pp. 3473–3493, 2006.

[39] K. F. Doerner, R. F. Hartl, V. Maniezzo, and M. Reimann, "Applying ant colony optimization to the capacitated arc routing problem," in *Proc. Ant Colony Optimization Swarm Intell.*, Lecture Notes in Computer Science 3172. Berlin, Germany: Springer, 2004, pp. 420–421.

[40] H. A. Eiselt, M. Gendreau, and G. Laporte, "Arc routing problems. Part II: The rural postman problem," *Oper. Res.*, vol. 43, no. 3, pp. 399–414, 1995.

[41] G. Ghiani, F. Guerriero, G. Improta, and R. Musmanno, "Waste collection in southern Italy: Solution of a real-life arc routing problem," *Int. Trans. Oper. Res.*, vol. 12, no. 2, 135–144, 2005.

[42] R. T. Sumichrast and I. S. Markham, "A heuristic and lower bound for a multidepot routing problem," *Comput. Oper. Res.*, vol. 22, no. 10, pp. 1047–1056, 1995.

[43] J. Renaud, G. Laporte, and F. F. Boctor, "A tabu search heuristic for the multidepot vehicle routing problem," *Comput. Oper. Res.*, vol. 23, no. 3, pp. 229–235, 1996.

[44] S. Salhi and M. Sari, "A multilevel composite heuristic for the multidepot vehicle fleet mix problem," *Eur. J. Oper. Res.*, vol. 103, no. 1, pp. 95–112, Nov. 1997.

[45] W. Ho, G. T. S. Ho, P. Ji, and H. C. W. Lau, "A hybrid genetic algorithm for the multidepot vehicle routing problem," *Eng. Applicat. Artif. Intell.*, vol. 21, no. 4, pp. 548–557, Jun. 2008.

[46] R. Dondo and J. Cerda, "A cluster-based optimization approach for the multidepot heterogeneous fleet vehicle routing problem with time windows," *Eur. J. Oper. Res.*, vol. 176, no. 3, pp. 1478–1507, Feb. 2007.

[47] A. Lim and F. Wang, "Multidepot vehicle routing problem: A one-stage approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 2, no. 4, pp. 397–402, Oct. 2005.

[48] C. T. Su, "Dynamic vehicle control and scheduling of a multidepot physical distribution system," *Integr. Manuf. Syst.*, vol. 10, no. 1, pp. 56–65, 1999.

[49] A. Amberg, W. Domschke, S. Voß, "Multiple center capacitated arc routing problems: A tabu search algorithm using capacitated trees," *Eur. J. Oper. Res.*, vol. 124, no. 2, pp. 360–376, Jul. 2000.

[50] T. H. Cormen, C. L. Leiserson, and M. L. Rivest, "Single-source shortest paths," in *Introduction to Algorithms*. Cambridge, MA: MIT Press, 1990, pp. 491–527.

[51] C. J. Chung and R. G. Reynolds, "A testbed for solving optimization problems using cultural algorithm," in *Proc. 5th Annu. Conf. Evol. Program.*, Cambridge, MA: MIT Press, 1996, pp. 225–236.

[52] J. Branke, "Memory-enhanced evolutionary algorithms for dynamic optimization problems," in *Proc. Congr. Evol. Comput.*, Piscataway, NJ: IEEE Press, 1999, pp. 1875–1882.

[53] S. J. Louis and J. McDonnell, "Learning with case-injected genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 8, no. 4, pp. 316–328, Aug. 2004.

[54] R. S. Michalski, "Learnable evolution model: Evolution process guided by machine learning," *Mach. Learning*, vol. 38, no. 1, pp. 9–40, 2000.

[55] N. B. Ho, J. C. Tay, and E. M. K. Lai, "An effective architecture for learning and evolving flexible job-shop schedules," *Eur. J. Oper. Res.*, vol. 179, no. 2, pp. 316–333, Jun. 2007.

[56] B. K. S. Cheung, A. Langevin, and B. Villeneuve, "High performing evolutionary techniques for solving complex location problems in industrial system design," *J. Intell. Manuf.*, vol. 12, nos. 5–6, pp. 455–466, 2001.

[57] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learning Res.*, vol. 7, pp. 1–30, Jan. 2006.

[58] J. M. Belenguer and E. Benavent, "A cutting plane algorithm for the capacitated arc routing problem," *Comput. Oper. Res.*, vol. 30, no. 5, pp. 705–728, 2003.

[59] M. Kiuchi, Y. Shinano, R. Hirabayashi, and Y. Saruwatari, "An exact algorithm for the capacitated arc routing problem using parallel branch and bound method," in *Proc. Abstracts 1995 Spring Natl. Conf. Oper. Res. Soc. Jpn.*, pp. 28–29.

[60] E. Benavent, V. Campos, A. Corberán, and E. Mota, "The capacitated Chinese postman problem: Lower bounds," *Networks*, vol. 22, no. 7, pp. 669–690, 1992.

[61] L. Y. O. Li and R. W. Eglese, "An interactive algorithm for vehicle routing for winter-gritting," *J. Oper. Res. Soc.*, vol. 47, no. 2, pp. 217–228, 1996.

Lining Xing received two B.S. degrees in economics and science from Xi'an Jiaotong University, Shaanxi, China, in 2002. Currently, he is pursuing the Ph.D. degree from the College of Information Systems and Management, National University of Defense Technology, Changsha, China.

His research interests include management decision making, simulation optimization, and artificial intelligence.

Philipp Rohlfshagen received the B.S. degree in computer science and artificial intelligence from the University of Sussex, Brighton, U.K., in 2003, and the M.S. degree in natural computation and the Ph.D. degree in evolutionary computation from the University of Birmingham, Birmingham, U.K., in 2004 and 2007, respectively.

He is currently a Research Fellow at the Center of Excellence for Research in Computational Intelligence and Applications, School of Computer Science, University of Birmingham, where he is studying the theoretical aspects of evolutionary algorithms for dynamic optimization problems. His research interests include combinatorial optimization, computational complexity, evolutionary computation, global optimization, algorithm design, and molecular genetics.

Yingwu Chen received the B.S. degree in automation, the M.S. degree in system engineering, and the Ph.D. degree in engineering from the National University of Defense Technology (NUDT), Changsha, China, in 1984, 1987, and 1994, respectively.

He was a Lecturer from 1989 to 1994, and an Associate Professor from 1994 to 1999 at NUDT. Since 1999, he has been a Distinguished Professor and the Director of the Department of Management Science and Engineering, College of Information Systems and Management, NUDT, where he focuses on management theory and its applications. He has authored more than 60 research publications. His current research interests include assistant decision-making systems for planning, decision-making systems for project evaluation, management decisions, and artificial intelligence.

Dr. Chen is the Editor of the *Principle of System Engineering* (Press of National University of Defense Technology), and *Technology of Quantificational Analysis* (China Renmin University Press).

Xin Yao (M'91–SM'96–F'03) received the B.S. degree from the University of Science and Technology of China (USTC), Hefei, China, in 1982, the M.S. degree from the North China Institute of Computing Technology, Beijing, China, in 1985, and the Ph.D. degree from USTC in 1990, all in computer science.

He is currently a Chair (Full Professor) of Computer Science and the Director of the Center of Excellence for Research in Computational Intelligence and Applications, University of Birmingham, U.K. He is a Distinguished Lecturer of IEEE Computational Intelligence Society, a Distinguished Visiting Professor at USTC, and a Visiting Professor at Nanjing University, Xidian University and Northeast Normal University. He has been an invited keynote speaker at more than 50 international conferences, and has more than 300 refereed research publications in evolutionary computation and neural network ensembles. His research interests include evolutionary computation, neural network ensembles, global optimization, data mining, computational time complexity of evolutionary algorithms, and real-world applications. In addition to basic research, he works closely with many industrial partners on various real-world problems.

Dr. Yao was an Editor-in-Chief of IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION from 2003 to 2008, and is an Associate Editor or an Editorial Board Member of ten other international journals. He is the Editor of the World Scientific Book Series on Advances in Natural Computation, and a Guest Editor of several journal special issues.