

# Evolutionary Multi-objective Optimization for Multi-depot Vehicle Routing in Logistics

Xiaowen Bi, Zeyu Han & Wallace K. S. Tang

*Department of Electronic Engineering, City University of Hong Kong,  
Tat Chee Avenue, Kowloon, Hong Kong SAR  
{xiaowenbi2-c, zeyuhan2-c}@my.cityu.edu.hk, eekstang@cityu.edu.hk*

Received 29 March 2017

Accepted 30 August 2017

## Abstract

Delivering goods in an efficient and cost-effective way is always a challenging problem in logistics. In this paper, the multi-depot vehicle routing is focused. To cope with the conflicting requirements, an advanced multi-objective evolutionary algorithm is proposed. Local-search empowered genetic operations and a fuzzy cluster-based initialization process are embedded in the design for performance enhancement. Its outperformance, as compared to existing alternatives, is confirmed by extensive simulations based on numerical datasets and real traffic conditions with various customers' distributions.

**Keywords:** Multi-depot vehicle routing, multi-objective optimization, evolutionary algorithm, local search

## 1. Introduction

The Internet shopping enjoys a surge of popularity in recent years, supported by the advances of mobile communications and Internet technology. For a one-day global shopping event, it could generate gross merchandise value up to USD17.8 billions<sup>1</sup>. The huge amounts of transactions imply billions of goods need to be delivered to customers all over the world in an efficient way, which is an opportunity and also a challenge for the logistics industry.

In face of the challenge, many research efforts have been made, while vehicle routing problem (VRP)<sup>2,3</sup> is a related topic in operational research, focusing on domestic delivery. Given a depot, vehicles are to visit different sites of customers for once. VRP can be generalized as problem with multiple depots, referred as multi-depot vehicle routing problem (MDVRP)<sup>4,5</sup>. In MDVRP, customers are firstly assigned to one of the depots, and the vehicle at the

depot is deployed to serve the dedicated set of customers based on a designated routing path.

To better reflect the practical conditions, many variants of VRP and MDVRP have been suggested. For example, by considering the limited capacity of a vehicle, one obtains a classic capacitated vehicle routing problem<sup>2</sup>. In Ref. 4, two types of vehicles, delivery vehicle and installation vehicle, are employed, where the time window restriction on delivery and onsite service time of installation vehicle are considered. Heterogeneous depots have also been included according to commercial offers, motivated by the market segmentation, as described in Ref. 5. In Ref. 3, desynchronized arrivals are assumed in VRP, as a result, the queuing of parcels has to be managed to avoid congestion in delivery.

In this paper, we focus on the MDVRP with delivery service time. Since a customer site now serves as a collection point of parcels for a set of clients, the delivery service time becomes comparable to the

transportation time. Unlike the installation service time considered in Ref. 4, which is only included as an additional constraint, we incorporate the service time in the objective function, to truly reflect the total delivery period. In addition, as demonstrated in our results, the delivery service time would magnify the impact of customer distribution on algorithmic performance, making the conventional distance-based customer assignment methods less attractive.

MDVRP is proved to be NP-hard and usually involves multiple objectives. To cope with such a complex and difficult multi-objective problem (MOP), meta-heuristic algorithms, such as tabu search<sup>6,7</sup> and evolutionary algorithms<sup>8,9</sup>, are commonly adopted. Moreover, local search mechanisms have also been employed<sup>10,11</sup>, while, for further improvements, hybrid approaches combining evolutionary algorithms and local searching mechanisms<sup>12,13</sup> seem to be promising.

Here, a hybrid multi-objective evolutionary algorithm (HMOEA) is proposed. To enhance its searching capability, a fuzzy cluster-based initialization incorporating customers' density and distance, an ordered-based cut-and-paste operation and a two-phase local search are newly designed. The effectiveness of the design is also verified by extensive simulations based on real-world traffic data.

The rest of the paper is organized as follows. In Section 2, MDVRP with customer service time is formulated. The proposed HMOEA is then explained in Section 3 in details. Simulation results and discussions are made in Section 4, and finally, the work is concluded in Section 5.

## 2. Problem Formulation

In MDVRP, a set of  $N$  customers ( $\mathcal{V}_c$ ) is to be served by a set of  $M$  depots ( $\mathcal{V}_d$ ). Based on their locations, the transportation network and traffic conditions, a fully-connected directed graph,  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ , can be formulated, where  $\mathcal{V} = \mathcal{V}_c \cup \mathcal{V}_d$  is the set of nodes,  $\mathcal{E} = \{(i, j) | i, j \in \mathcal{V}, i \neq j\}$  is the set of edges and  $\mathcal{A} = \{(d_{ij}, \tau_{ij}) | (i, j) \in \mathcal{E}\}$  is the set of weights for the edges with  $d_{ij}$  and  $\tau_{ij}$  being the traveling distance and duration from node  $i$  to node  $j$ , respectively. The traveling duration is time-dependent re-

lated to real-time traffic conditions and the traveling distance is fixed. Also,  $d_{ij} \neq d_{ji}$  and  $\tau_{ij} \neq \tau_{ji}$ .

In this work, it is assumed that each depot can invoke at most one vehicle, while all the vehicles are homogeneous and have sufficient capacity. To reflect the transportation efficiency and customers' satisfaction, two primary criteria, namely the total traveling distance ( $f_1$ ) and maximum delivery duration ( $f_2$ ), are considered. A multi-objective MDVRP (MO MDVRP) is hence constructed as follows:

$$\text{Minimize } F(x) = (f_1(x), f_2(x)) \quad (1)$$

over  $x = \{x_{ijk} | x_{ijk} \in \{0, 1\}, i, j \in \mathcal{V}, k \in \mathcal{V}_d, i \neq j\}$  with

$$f_1(x) = \sum_k \sum_i \sum_j x_{ijk} d_{ij} \quad (2)$$

$$f_2(x) = \max_{k \in \mathcal{V}_d} \left\{ S|\mathcal{C}_k| + \sum_i \sum_j x_{ijk} \tau_{ij} \right\} \quad (3)$$

subject to

$$\sum_i x_{ijk} = \sum_j x_{jik}, \forall j \in \mathcal{V}, k \in \mathcal{V}_d \quad (4)$$

$$\sum_k \sum_i x_{ijk} = 1, \forall j \in \mathcal{V}_c \quad (5)$$

$$\sum_i \sum_j x_{ijk} \leq 1, \forall k \in \mathcal{V}_d \quad (6)$$

$$\sum_{i \in \mathcal{C}_k} \sum_{j \in \mathcal{C}_k} x_{ijk} \leq |\mathcal{C}_k| - 1, \forall k \in \mathcal{V}_d \quad (7)$$

and  $x_{ijk}$  is a binary decision variable. If the vehicle invoked by the  $k$ th depot takes the path from node  $i$  to node  $j$ ,  $x_{ijk} = 1$ ; Otherwise,  $x_{ijk} = 0$ . In (3),  $S$  indicates the length of serving time for a customer and  $\mathcal{C}_k$  represents the set of customers served by the  $k$ th depot. Constraint (4) ensures that the vehicle visiting and departing from a customer is the same one. Constraint (5) stipulates that each customer can only be served by one vehicle for once. Constraint (6) implies that a vehicle is invoked by only a particular depot. Constraint (7) is to eliminate sub-tour because each vehicle should visit all the assigned customers in a complete tour, rather than several sub-tours.

### 3. Design of Hybrid MOEA (HMOEA)

In order to effectively manage the MO MDVRP specified in Section 2, a hybrid multi-objective evolutionary algorithm (HMOEA) is proposed. Its flow diagram is given in Fig. 1 and the major designs are explained below.

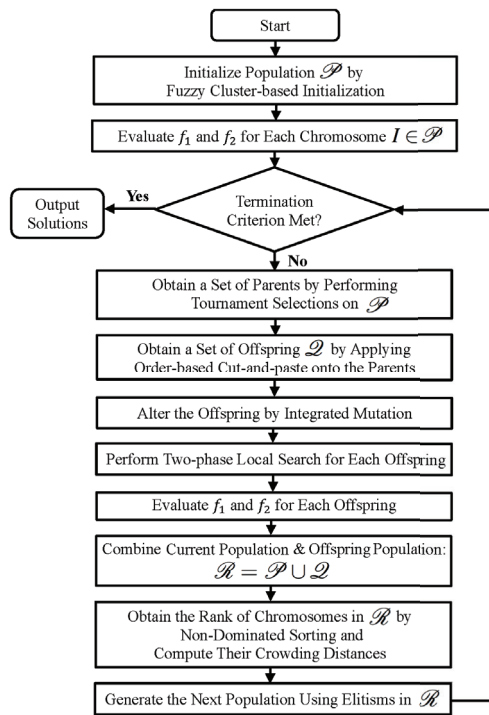


Fig. 1. Flow diagram of HMOEA.

#### 3.1. Chromosome representation

Encoding the solution by chromosome is always a key step in the design of an evolutionary algorithm (EA). In MDVRP, two sub-design tasks are involved. One is to assign customers to one of the depots while the other is to find the best route for serving the set of customers in each depot. Therefore, an ordered chromosome<sup>14</sup> is suggested.

The chromosome is encoded as an integer string with a length of  $(N + M - 1)$ . Without loss of generality, the numbers 1 to  $N$  denote the customers while  $(N + 1)$  to  $(N + M - 1)$  serve as separators. An illustrative example is given in Fig. 2. The chromosome in Fig. 2 (a) consists of 12 customers and 3 separators. It can thus define the assignment of customers

to 4 depots  $\{A, B, C, D\}$  and also the route of each vehicle as indicated in Fig. 2 (b).

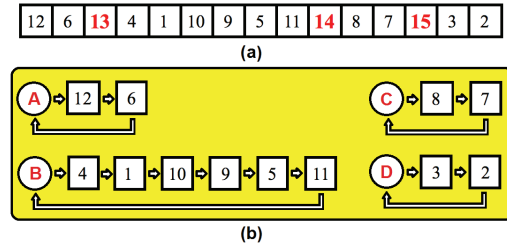


Fig. 2. (a) Ordered chromosomes for 12 customers and 4 depots (b) Resultant route for each depot.

#### 3.2. Population initialization

After defining the chromosome, the first step in HMOEA is to generate the initial population. Although randomly generated population is common, the preference of having a better set of initialized population has also been discussed in EA communities. In Ref. 12, chromosome is configured by assigning customers to the nearest depot. Such a binary decision is modified into a fuzzy one in Ref. 15, and a customer can have different levels of belonging to all the depots, governed by the following membership function (MF)

$$u_{ik} = \frac{(d_{ki})^{-\frac{1}{f-1}}}{\sum_{j=1}^M (d_{ji})^{-\frac{1}{f-1}}} \quad (8)$$

where  $d_{ki}$  is the distance between depot  $k$  and customer  $i$ , and  $f > 1$  is a constant indicating the level of cluster fuzziness.

Although such a distance-based initialization method would facilitate the search of solutions, the impact of customer distribution is totally ignored. It can be easily imagined that, if too many customers are assigned to a depot, it may not be a good choice.

Hence, a new MF is proposed as below:

$$u_{ik} = \alpha \frac{(d_{ki})^{-\frac{1}{f-1}}}{\sum_{j=1}^M (d_{ji})^{-\frac{1}{f-1}}} + (1 - \alpha) \frac{(\rho_k)^{-\frac{1}{f-1}}}{\sum_{j=1}^M (\rho_j)^{-\frac{1}{f-1}}} \quad (9)$$

where  $\alpha \in [0, 1]$  is a weighting factor and  $\rho_k$  is the density of customers in the region served by depot  $k$  which is defined as

$$\rho_k = \frac{|\mathcal{C}_k|}{r^2 \pi} \quad (10)$$

with  $r = 0.5(\max_{p,q \in \mathcal{C}_k} d_{pq})$ . The new MF in (9) is contributed by both the distance and the density of customers at a depot with weighting tuned by  $\alpha$ .

A new fuzzy cluster-based initialization (FCBI) process can then be designed to obtain a population  $\mathcal{P} \equiv \bigcup_{i=1}^w \mathcal{P}_i$  where  $\mathcal{P}_i$  (of size  $P_i$ ) is generated as given in Table 1 based on the factor  $\alpha_i$ , with  $i = 1, 2, \dots, w$ .

Table 1. Pseudo code for FCBI.

```

Set  $\mathcal{P}_i = \emptyset$ ;
Obtain  $\mathcal{C}_k$  for  $k = 1, 2, \dots, M$  by assigning customers to
the closest depot;
Compute  $\rho_k$  for  $k = 1, 2, \dots, M$  with  $\alpha = \alpha_i$  using (10);
for  $j = 1$  to  $N$ 
  Compute  $u_{jk}$  for  $k = 1, 2, \dots, M$  using (9);
  Find  $k'$  s.t.  $k' = \arg \max_{k \in \mathcal{C}_d} u_{jk}$ ;
  Update  $\mathcal{C}_k, \mathcal{C}_{k'}$  s.t.  $\mathcal{C}_k = \mathcal{C}_k \setminus \{j\}$  and  $\mathcal{C}_{k'} = \mathcal{C}_{k'} \cup \{j\}$ ;
  Update  $\rho_k$  for  $k = 1, 2, \dots, M$  using (10);
end
for  $p = 1$  to  $P_i$ 
  for each  $\mathcal{C}_k$  with  $k = 1, 2, \dots, M$ , generate routing path
  by sequential sampling algorithm probabilistically16;
  Encode the solution as a chromosome  $I$ ;
  Update  $\mathcal{P}_i$  by  $\mathcal{P}_i = \mathcal{P}_i \cup \{I\}$ ;
end

```

### 3.3. Selection process

After generating the initial population and evaluating the objective functions  $f_1$  and  $f_2$  for each chromosome, genetic cycles are repeated until termination criteria are met. In each cycle, parents are selected from the current population and used to generate offspring via some *genetic operations*. Offspring will further be improved by a *two-phase local search*. The elites in the current population and offspring population are then preserved to form the next population.

To better cope with the MO MDVRP, we adopt the tournament selection as suggested in the non-dominated sorting algorithm (NSGA-II). The two randomly selected chromosomes in a tournament are compared using *Pareto dominance* and *crowding distance* and the better one is selected as parent. Their operational details can be found in Ref. 17 and are omitted here due to page limitation.

### 3.4. Genetic operations

#### 3.4.1. Order-based cut-and-paste

The cut-and-paste (CP) operation in JGGA<sup>18</sup> is well-known for its excellence in new solution exploration. Here, CP is modified for ordered chromosome and it is referred as order-based cut-and-paste (OCP). As shown in Fig. 3, a segment of genes is randomly selected from a parent, say Parent A, and inserted at a random site of another parent, Parent B. An offspring is then obtained by removing repeated genes. Another offspring can be obtained by swapping the roles of Parents A and B. In our work, the operational probability of OCP is governed by  $p_c$ .

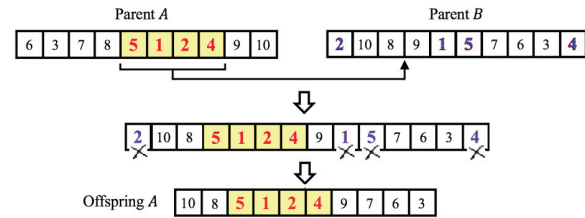


Fig. 3. An example of order-based cut-and-paste.

#### 3.4.2. Integration of mutation operators

To prevent trapping in local minima, three mutation operations are integrated and applied to the offspring. The first one is *gene swapping*, in which two genes are randomly selected and swapped. (Fig. 4(a)). The second one is referred as *inversion*. A portion of genes is randomly selected and their positions are inverted (Fig. 4(b)). The last one is called *self OCP*, which is to apply OCP onto the same offspring (Fig. 4(c)). In this work, same operational rate is used and given by  $p_m$ .

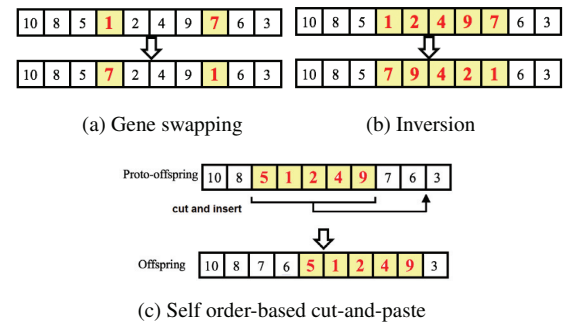


Fig. 4. Examples of the three mutation operators.

### 3.5. Two-phase local search

To effectively tackle with the MO MDVRP, two local searching mechanisms are applied to further improve each offspring, targeting the two key processes in the optimization, i.e. the path routing and the customer assignment, respectively.

#### 3.5.1. 2-opt solver

The first local search is based on 2-opt solver, which is originally suggested in Ref. 19 for the traveling salesman problem. It is to reduce routing path length by replacing two non-adjacent arcs with two other arcs. Its operational procedures are given in Table 2 and a simple illustration is depicted in Fig. 5. Consequently, the routing path is modified by 2-opt whenever there is a reduction of path distance.

Table 2. Pseudo code for 2-opt solver.

```

Extract  $M$  routing paths for  $M$  depots from chromosome  $I$ ;
for  $k = 1$  to  $M$ 
  for each  $(i, j)$ , where  $i, j \in \mathcal{C}_k$ ,  $prec(j) \neq i$ ,
    Let  $s = succ(i)$ ;  $t = prec(j)$ ;
    if  $(d_{it} + pL(t, s) + d_{sj} < pL(i, j))$ 
      Set  $succ(i) = t$ ,  $prec(j) = s$ ; Reverse  $p(s, t)$  to  $p(t, s)$ ;
    end
  end
end
Encode the improved solution and replace  $I$ ;

```

Note:  $prec(i)$  and  $succ(i)$  returns the preceding and succeeding nodes of  $i$  along the path.  $p(i, j)$  and  $pL(i, j)$  represent the path from  $i$  to  $j$  and its length, respectively.

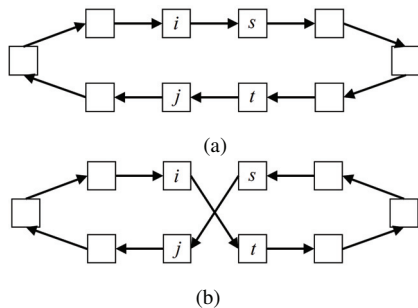


Fig. 5. Illustration of 2-opt solver. Modification of route due to the condition:  $pL(i, j)$  in (a)  $>$   $pL(i, j)$  in (b).

#### 3.5.2. Customer grouping optimizer

The second local search is an inter-route operator called customer grouping optimizer (CGO). We de-

fine the delivery duration of a routing path of the vehicle at depot  $k$  as

$$DD_k = S|\mathcal{C}_k| + \sum_i \sum_j x_{ijk} \tau_{ij}. \quad (11)$$

CGO is designed to reduce  $\max_{k \in \mathcal{V}_d} (DD_k)$  by moving unbefitting customers from customer set  $\mathcal{C}_p$  with  $p = \arg \max_{k \in \mathcal{V}_d} DD_k$ , to smaller sets, without compromising the other objective, i.e. the total traveling distance. The general idea is illustrated in Fig. 6. Consider the customer  $t \in \mathcal{C}_p$ , which has the most significant impact in  $DD$ , i.e.  $(\tau_{st} + \tau_{tu} - \tau_{su})$  is maximum where  $s = prec(t)$  and  $u = succ(t)$ . If there exist  $i \in \mathcal{C}_q$  and  $j = succ(i)$ , such that  $(d_{it} + d_{tj} - d_{ij} \leq d_{st} + d_{tu} - d_{su})$ , customer  $t$  will be re-assigned to  $\mathcal{C}_q$ , s.t.  $t = succ(i) = prec(j)$ . This operation will be repeated for  $r$  times if CGO is carried out, governed by a probability of  $p_u$ . The choice of  $r$  is determined by compromising the speed and the performance. In our simulations,  $r$  is set as 15, and generally, the larger the searching space is, the larger  $r$  should be.

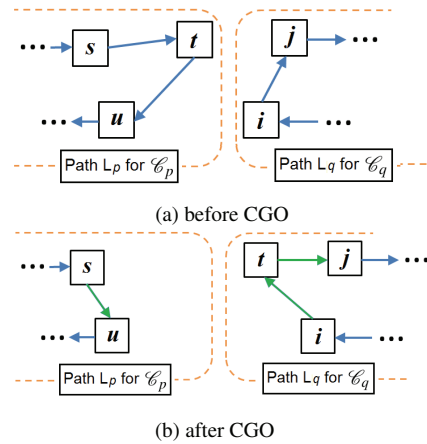


Fig. 6. Illustration of CGO.

## 4. Experimental Results

### 4.1. Testing cases and HMOEA parameters

In our simulations, two testing cases,  $\mathcal{G}_A$  and  $\mathcal{G}_B$ , are considered. For each case,  $M = 6$  depots of a major logistics company in HK, mainly located in Kowloon City and Yau Tsim Mong districts are selected.  $N = 100$  customers are served, with their locations distributed uniformly and non-uniformly for Cases  $\mathcal{G}_A$  and  $\mathcal{G}_B$ , respectively. Figures 7(a)

and 7(b) show the locations of customers and depots for the two cases. The traveling distance  $d_{ij}$  and traveling duration  $\tau_{ij}$  for any  $(i, j) \in \mathcal{E}$  are derived from Google Maps Distance API, to reflect the real time traffic condition. Also, the parameters used in HMOEA are listed in Table 3.

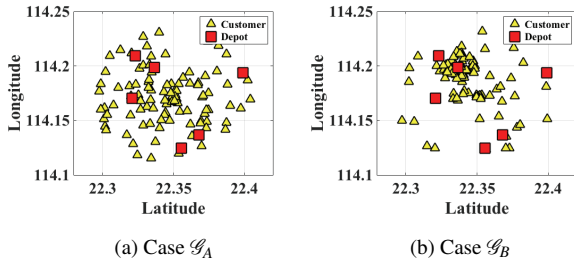


Fig. 7. Locations of customers and depots in  $\mathcal{G}_A$  and  $\mathcal{G}_B$ .

Table 3. General parameter settings of HMOEA.

Population size ( $ \mathcal{P} $ )	100
Probability of crossover ( $p_c$ )	0.9
Probability of mutation ( $p_m$ )	0.2
Probability of grouping optimizer ( $p_u$ )	0.5
Number of Offspring ( $ \mathcal{L} $ )	50

## 4.2. MO performance metric

In MO optimization, the goodness of a non-dominated solution set can be measured by two metrics, namely the convergence and the diversity<sup>18</sup>.

The convergence metric is defined as  $\beta = \sum_{i=1}^{|\mathcal{P}|} D_i / |\mathcal{P}|$  where  $D_i$  is the normalized Euclidean distance between the  $i$ th solution in the non-dominated solution set and its nearest true-Pareto-optimal solution. A smaller  $\beta$  indicates a better convergence.

The diversity implies the non-uniformity and spread in the distribution of the non-dominated solution set. It is computed by  $\gamma = \frac{\sum_{j=1}^J D_j^e + \sum_{i=1}^{|\mathcal{P}|-1} |\delta_i - \bar{\delta}|}{\sum_{j=1}^J D_j^e + (|\mathcal{P}|-1) \cdot \bar{\delta}}$  where  $J$  is the number of objective functions (in our case,  $J = 2$ );  $D_j^e$  are Euclidean distances between the extreme solutions of the Pareto-optimal set and the boundary solutions of the non-dominated solution set;  $\delta_i$  is the distance between neighbor non-dominated solutions, and  $\bar{\delta}$  is the average of all  $\delta_i$ . Small  $\gamma$  indicates a wide spread of solutions, i.e. a better diversity.

Since the true Pareto optimal front is not available for our problem, a reference set is used instead. It is obtained by merging results from 5 runs of HMOEA (with 200,000 generations) and all the other compared methods.

## 4.3. Design Verification

### 4.3.1. Effectiveness of FCBI

We firstly investigate the effectiveness of the fuzzy cluster-based initialization (FCBI). It is compared to two common approaches, namely the distance-based (DBI) initialization and random initialization (RI).

In DBI, customers are assigned to the closest depot, while the routing path of each customer set is then generated by applying sequential sampling algorithm probabilistically. In RI, chromosomes are simply randomly generated.

FCBI, DBI and RI are used to generate the initial populations for NSGA-II, respectively, with order crossover and integrated mutation operator equipped. The average results over 20 runs are tabulated in Table 4. Since the influence of initialization tends to diminish against generations, the non-dominated solution sets obtained at the 4,000th generation are compared.

Table 4. Comparisons of different initialization methods based on performance metrics  $\beta$  and  $\gamma$ .  $\mu$  and  $\sigma$  are the mean and standard deviation of performance metrics, respectively, and the best is in bold.

		FCBI( $\mu, \sigma$ )	DBI( $\mu, \sigma$ )	RI( $\mu, \sigma$ )
$\mathcal{G}_A$	$\beta$	<b>(1.298, 0.203)</b>	(1.616, 0.468)	(3.480, 0.380)
	$\gamma$	<b>(0.822, 0.136)</b>	(0.992, <b>0.128</b> )	(1.062, 0.252)
$\mathcal{G}_B$	$\beta$	<b>(0.391, 0.201)</b>	(0.774, 0.280)	(0.957, 0.439)
	$\gamma$	<b>(0.907, 0.285)</b>	(1.045, <b>0.206</b> )	(1.159, 0.249)

As shown, FCBI performs the best in all cases, except that its  $\sigma_\gamma$  is slightly larger than that of DBI. However, since  $\mu_\gamma(FCBI)$  is smaller, it can still conclude that FCBI is better than DBI.

We also notice a significant improvement in Case  $\mathcal{G}_B$  when FCBI is employed. Comparing to RI, the improvements by DBI for Case  $\mathcal{G}_A$  and  $\mathcal{G}_B$  are 53.56% and 19.16%, respectively. While if FCBI is used, the improvements become 62.71% and 59.10%, respectively, confirming the ability of FCBI in managing various customer density distributions.

#### 4.3.2. Effectiveness of OCP

Under the framework of NSGA-II, we directly compare OCP with other conventional order-based crossovers, including order crossover (OX) and partially mapped crossover (PMX), for solving MO MDVRP. Table 5 summarizes the comparisons of results (average of 20 runs) of NSGA-II with these crossover operations, terminated at 150,000 generations. As shown, OCP performs the best, except that its  $\mu_\gamma$  is slightly larger than that of OX. However, its  $\sigma_\gamma$  is smaller, indicating a better consistency. PMX is the worst as it emphasizes absolute positions of genes, while for MDVRP (or general VRP), relative positions of genes are much more important.

Table 5. Comparison of different crossover operations

		OCP( $\mu, \sigma$ )	OX( $\mu, \sigma$ )	PMX( $\mu, \sigma$ )
$\mathcal{G}_A$	$\beta$	<b>(0.729, 0.264)</b>	(0.810, 0.279)	(0.989, 0.349)
	$\gamma$	(0.742, <b>0.079</b> )	<b>(0.736, 0.097)</b>	(0.800, 0.091)
$\mathcal{G}_B$	$\beta$	<b>(0.072, 0.023)</b>	(0.111, 0.084)	(0.218, 0.060)
	$\gamma$	<b>(0.720, 0.066)</b>	(0.738, 0.088)	(0.792, 0.079)

#### 4.3.3. Effectiveness of two-phase local search

Lastly, we investigate the effectiveness of the two-phase local search (TPLS). The non-dominated solution sets obtained by HMOEA with and without TPLS are plotted in Fig. 8.

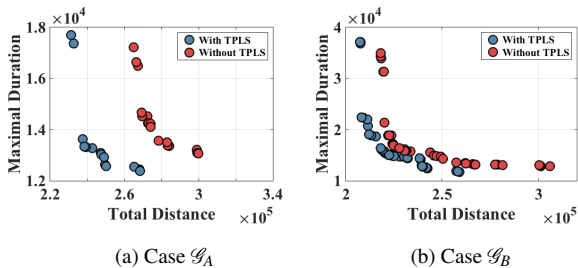


Fig. 8. Non-dominated solution sets obtained by HMOEA with and without TPLS. The sets are collected when there is no further update in last 1,000 generations.

In both cases, the inclusion of TPLS enhances the searching of non-dominated frontier. Moreover, as shown in Tables 4 and 5,  $\beta$  is much larger in Case  $\mathcal{G}_A$  than in Case  $\mathcal{G}_B$ , meaning that it is a more difficult problem. This is also confirmed in Fig. 8. It fails to locate non-dominated solutions in Case  $\mathcal{G}_A$  if TPLS is excluded. In contrast, improvement in non-dominated solution set is relatively small in Case  $\mathcal{G}_B$ .

#### 4.4. Comparisons of Algorithms

Finally, HMOEA is compared to NSGA-II and SPEA-II<sup>20</sup>, which are both well known for effectively approximating optimal solutions and maintaining the diversity of non-dominated frontier. The design of NSGA-II in Ref. 9 is adopted and it is equipped with DBI, one-point crossover and gene swapping mutation. For SPEA-II, RI, OX and gene swapping mutation are used. All other parameters of these three algorithms are kept the same for fair comparison.

Table 6 shows the average results of 10 runs of the above algorithms, each of which terminates after 150,000 generations. Both  $\mu_\beta$  and  $\mu_\gamma$  of HMOEA are the smallest in the two testing cases, indicating the best quality of the solutions. The standard derivations are slightly worse, but with the fact that  $\mu$  is much smaller, the outperformance of HMOEA is confirmed.

Table 6. Comparison of different evolutionary algorithms

		HMOEA( $\mu, \sigma$ )	NSGA-II( $\mu, \sigma$ )	SPEA-II( $\mu, \sigma$ )
$\mathcal{G}_A$	$\beta$	<b>(0.607, 0.235)</b>	(1.312, <b>0.132</b> )	(2.435, 0.405)
	$\gamma$	<b>(0.792, 0.161)</b>	(0.833, 0.104)	(1.093, <b>0.065</b> )
$\mathcal{G}_B$	$\beta$	<b>(0.067, 0.064)</b>	(0.168, 0.075)	(0.271, 0.313)
	$\gamma$	<b>(0.752, 0.166)</b>	(0.814, <b>0.059</b> )	(1.034, 0.143)

Due to the inclusion of local search, the computational time of HMOEA is longer. For one generation, HMOEA takes 16.84 ms while NSGA-II and SPEA-II need 4.43 ms and 3.49 ms, respectively, in a PC platform with Intel Core i5 (2.6 GHz) and 8G RAM. However, as shown in Fig. 9, when comparing the non-dominated solution sets obtained with a fixed run-time, HMOEA still performs superior than the others, confirming the effectiveness of our design.

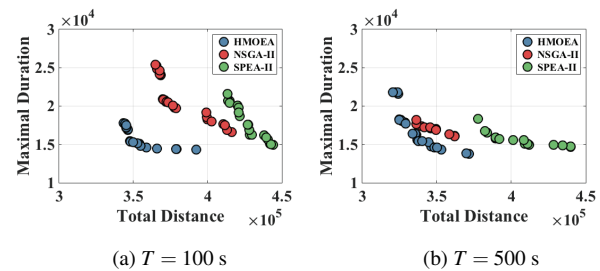


Fig. 9. Non-dominated solution sets obtained by different EAs with total run-time  $T$ . The sets are obtained by merging results of 10 runs.

## 5. Conclusion

In this paper, a hybrid multi-objective evolutionary algorithm, called HMOEA, is designed to address the multi-depot vehicle routing problem with delivery service time. New operations, including fuzzy cluster-based initialization, order-based cut-and-paste and two-phase local search, are proposed and implemented. Their effectiveness in exploration and handling various customer location distributions is confirmed by extensive simulations based on real traffic data. The simulation results also confirm that HMOEA outperforms other existing methods in solving the multi-depot vehicle routing problem.

## Acknowledgments

This paper was supported by grants from City University of Hong Kong (7004608 and 7004835).

## References

1. <http://www.atimes.com/article/alibabas-singles-day-smashes-sales-record-15-hours/>.
2. C. Archetti, L. Bertazzi, D. Laganà and F. Vocaturo, The undirected capacitated general routing problem with profits, *Eur. J. Oper. Res.* **257**(3) (2017) 822-833, DOI:10.1016/j.ejor.2016.08.001.
3. Z. Naji-Azimi, M. Salari, J. Renaud and A. Ruiz, A practical vehicle routing problem with desynchronized arrivals to depot, *Eur. J. Oper. Res.* **255**(1) (2016) 58-67, DOI:10.1016/j.ejor.2016.04.007.
4. H. Bae and I. Moon, Multi-depot vehicle routing problem with time windows considering delivery and installation vehicles, *Appl. Math. Model.* **40**(13) (2016) 6536-6549, DOI:10.1016/j.apm.2016.01.059.
5. L. Calvet, A. Ferrer, M. I. Gomes, A. Juan and D. Masip, Combining statistical learning with metaheuristics for the multi-depot vehicle routing problem with market segmentation, *Comput. Ind. Eng.* **94** (2016) 93-104, DOI:10.1016/j.cie.2016.01.016.
6. J. W. Escobar, R. Linfati, P. Toth and M. G. Baldoquin, A hybrid granular tabu search algorithm for the multi-depot vehicle routing problem, *J. Heuristics.* **20**(5) (2014) 483-509, DOI:10.1007/s10732-014-9247-0.
7. A. Martínez-Puras and J. Pacheco, MOAMP-Tabu search and NSGA-II for a real bi-objective scheduling-routing problem, *Know-Based. Syst.* **112** (2016) 92-104, DOI:10.1016/j.knosys.2016.09.001.
8. T. Vidal, T. G. Crainic, M. Gendreau and C. Prins, Implicit depot assignments and rotations in vehicle routing heuristics, *Eur. J. Oper. Res.* **237**(1) (2014) 15-28, DOI:10.1016/j.ejor.2013.12.044.
9. E. B. Alaïa, I. H. Dridi, H. Bouchriha and P. Borne, Genetic algorithm for multi-criteria optimization of multi-depots pick-up and delivery problems with time windows and multi-vehicles, *Acta. Polytech. Hung.* **12**(8) (2015) 155-174.
10. P. C. Guedes, W. P. Lopes, L. R. Rohde and D. Borenstein, Simple and efficient heuristic approach for the multiple-depot vehicle scheduling problem, *Optim. Lett.* **10**(7) (2016) 1449-1461, DOI:10.1007/s11590-015-0944-x.
11. A. Subramanian, E. Uchoa and L. S. Ochi, A hybrid algorithm for a class of vehicle routing problems, *Comput. Oper. Res.* **40**(10) (2013) 2519-2531, DOI:10.1016/j.cor.2013.01.013.
12. W. Ho, G. T. S. Ho, P. Ji and H. C. W. Lau, A hybrid genetic algorithm for the multi-depot vehicle routing problem, *Eng. Appl. Artif. Intel.* **21**(4) (2008) 548-557, DOI:10.1016/j.engappai.2007.06.001.
13. S. Wang, Z. Lu, L. Wei, G. Ji and J. Yang, Fitness-scaling adaptive genetic algorithm with local search for solving the multiple depot vehicle routing problem, *Simulation.* **92**(7) (2016) 601-616, DOI:10.1177/0037549715603481.
14. K. S. Tang, J. J. Yin and K. F. Man, A genetic-based optimization for multi-depot vehicle routing problems, *IEEE. Int. Symp. Ind. Elec.* (2010) 1545-1549.
15. H. Ewbank, P. Wanke and A. Hadi-Vencheh, An unsupervised fuzzy clustering approach to the capacitated vehicle routing problem, *Neural. Comput. Appl.* **27**(4) (2016) 857-867, DOI:10.1007/s00521-015-1901-4.
16. Z. H. Ahmed, A hybrid genetic algorithm for the bottleneck traveling salesman problem, *ACM. T. Embed. Comput. S.* **12**(1) (2013), DOI:10.1145/2406336.2406345.
17. K. Deb, A. Pratap, S. Agarwal and T. A. M. T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE. T. Evolut. Comput.* **6**(2) (2002) 182-197, DOI:10.1109/4235.996017.
18. T. M. Chan, K. F. Man, S. Kwong and K. S. Tang, A jumping gene paradigm for evolutionary multi-objective optimization, *IEEE. T. Evolut. Comput.* **12**(2) (2008) 143-159, DOI:10.1109/TEVC.2007.895269.
19. G. A. Croes, A method for solving traveling-salesman problems, *Oper. Res.*, **6**(6) (1958) 791-812, DOI:10.1287/opre.6.6.791.
20. E. Zitzler, M. Laumanns and L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm, *TIK-Report* **103** (2001).