



# A bi-level Voronoi diagram-based metaheuristic for a large-scale multi-depot vehicle routing problem

Wei Tu<sup>a</sup>, Zhixiang Fang<sup>b,c,\*</sup>, Qingquan Li<sup>a,b</sup>, Shih-Lung Shaw<sup>b,d</sup>, BiYu Chen<sup>b</sup>

<sup>a</sup> Shenzhen Key Laboratory of Spatial Smart Sensing and Services, Shenzhen University, Shenzhen 518060, China

<sup>b</sup> State Key Laboratory for Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, 129 Luoyu Road, Wuhan 430079, Hubei, China

<sup>c</sup> Engineering Research Center for Spatio-Temporal Data Smart Acquisition and Application, Ministry of Education of China, 129 Luoyu Road, Wuhan 430079, China

<sup>d</sup> Department of Geography, University of Tennessee, Knoxville, TN 37996-0925, USA

## ARTICLE INFO

### Article history:

Received 17 October 2012

Received in revised form 28 March 2013

Accepted 13 October 2013

Available online xxxx

### Keywords:

Multi-depot vehicle routing problem

Metaheuristic

Local search

Simulated annealing

Voronoi diagram

## ABSTRACT

In this paper, a bi-level Voronoi diagram-based metaheuristic is introduced to solve the large-scale multi-depot vehicle routing problem (MDVRP). The upper level of the Voronoi diagram, derived from the depots, is used to allocate customers to depots. The lower level of the Voronoi diagram, derived from the customers, limits the search space of reallocating customers among the depots and rearranging the customers among the routes from each depot to its Voronoi neighbors. The results of numerical experiments clearly indicate the benefits of this proposed bi-level Voronoi diagram approach for solving very large-scale MDVRPs while balancing the solution quality and the computational demand.

© 2013 Elsevier Ltd. All rights reserved.

## 1. Introduction

Over the past five decades, the vehicle routing problem (VRP), first formulated by Dantzig and Ramser (1959), has been one of the most studied optimization problems in transportation, logistics, and supply-chain management (Sheu and Talley, 2011). Solving large-scale VRPs (Qi et al., 2012) is critical to a wide range of large-scale delivery systems. In real-world applications, these systems usually involve a VRP with multiple depots, which is known as a multi-depot VRP (MDVRP). The MDVRP is a nondeterministic polynomial-time hard (NP-hard) problem (Polacek et al., 2004; Yu et al., 2010; Yu and Yang, 2011; Liu et al., 2011), and solving an MDVRP is time-consuming and computationally intractable (Ho et al., 2008; Ombuki-Berman and Hanshar, 2009). Two types of algorithms have been developed to solve MDVRPs. The first uses exact algorithms, such as mixed-integer linear programming (Dondo and Cerdá, 2007; Erdoğan and Miller-Hooks, 2012), branch-and-bound (Laporte et al., 1988), and set partitioning (Baldacci and Mingozzi, 2009). For instance, Laporte et al. (1984) formulated the symmetric MDVRP as an integer programming model and solved it with a branch-and-bound algorithm using a linear programming relaxation technique. Laporte et al. (1988) transformed the asymmetric MDVRP into a constraint assignment problem and found its optimal solution with a branch-and-bound approach. Baldacci and Mingozzi (2009) developed a set-partitioning formulation for the heterogeneous MDVRP, which was transformed into a heterogeneous VRP instance. However, these exact algorithms usually solve MDVRP instances with no more than 150 customers (Dondo and Cerdá, 2007; Creverier et al., 2007; Liu et al., 2011). This limitation reduces the applicability of these algorithms to real-world applications.

\* Corresponding author at: State Key Laboratory for Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, 129 Luoyu Road, Wuhan 430079, Hubei, China. Tel.: +86 27 6877 9889; fax: +86 27 6877 8043.

E-mail address: [zxfang@whu.edu.cn](mailto:zxfang@whu.edu.cn) (Z. Fang).

The second type of algorithms uses a heuristic or metaheuristic approach. Example heuristic approaches include simple construction and improvement heuristics (Tillman, 1969) and multiphase heuristics (Chao et al., 1993). Metaheuristic approaches include, for example, simulated annealing (SA; Lim and Zhu, 2006), Tabu search (TS; Cordeau et al., 1997), variable neighborhood search (VNS; Polacek et al., 2004; Hemmelmayr et al., 2009; Kuo and Wang, 2012), and evolutionary algorithms such as genetic algorithm (GA; Ho et al., 2008; Liu et al., 2009; Vidal et al., 2012), ant-colony optimization (ACO; Yu et al., 2010; Yu and Yang, 2011), and particle-swarm optimization (PSO; Potvin, 2009). These approaches have improved the ability to solve MDVRPs with small- and medium-scale customer bases, but they need additional improvements to solve very large-scale real-world MDVRPs.

There are two important spatial optimization issues in MDVRPs: customer allocation and routing optimization (Lim and Wang, 2005). The first optimization issue allocates customers to depots. This optimization has direct impacts on the solution structure of the MDVRP for different spatial configurations of customer locations, which in turn determines the final solution quality. Some studies (Renaud et al., 1996; Cordeau et al., 1997; Ho et al., 2008) simply allocated each customer to the nearest depot. In this case, the MDVRP is divided into multiple single-depot VRPs. This allocation strategy can generate local optimization solutions easily, but it does not necessarily lead to the globally optimized MDVRP solution.

The second issue is routing optimization. Some researchers employed an approximate brute-force exploration of neighborhood solutions to optimize vehicle routing (Renaud et al., 1996; Cordeau et al., 1997; Ho et al., 2008). To improve solution search speed, various spatial neighbor-reduction strategies such as granular neighborhood search (Toth and Vigo, 2003),  $k$ th nearest neighbor search (Li et al., 2005), Voronoi neighborhoods (Ouyang, 2007), and  $k$ -ring-shaped sets of Voronoi neighbors (Fang et al., 2012) have been used to reduce the search effort required for vehicle routing.

There are three important aspects that must be addressed when the above-mentioned approaches are used to solve very large-scale real-world MDVRPs: (i) these approaches need to be extended to solve MDVRPs with a large number of customers in real-world applications; (ii) the computation time must be reduced to make it useful for real-world MDVRP applications; and (iii) the spatial neighbor-reduction strategies should be extended to generate globally optimized MDVRP solutions by developing better strategies of allocating customers to depots. This paper therefore proposes a bi-level Voronoi diagram-based metaheuristic to solve very large-scale real-world MDVRPs. In particular, this study extends the one-level Voronoi diagrams (Fang et al., 2012) to bi-level Voronoi diagrams, which provides an efficient strategy of reallocating customers among the depots to improve MDVRP solution quality.

This paper is organized into seven sections. The next section offers a brief literature review of customer allocation strategies and metaheuristics for MDVRPs. Section 3 introduces the definition of  $k$ -ring Voronoi neighbors of a point and a line as a bi-level Voronoi diagram. Section 4 presents the details of the proposed algorithm, and Section 5 shows the computation results and algorithm performance. Section 6 discusses the results for Voronoi distance and the tradeoff of solution quality and computational effort. The final section provides concluding remarks and future research directions.

## 2. Literature review of MDVRP strategies

This literature review includes customer allocation strategies and metaheuristics for solving an MDVRP. Customer allocation is crucial to solving an MDVRP by dividing a large-scale MDVRP into multiple single-depot VRPs. This strategy reduces the computational complexity of an MDVRP algorithm. Several customer allocation strategies have been developed to perform this division: the nearest-depot approach, the cluster approach, and the border-customer approach. The nearest-depot approach assigns each customer to its nearest depot. Tillman (1969) first introduced this approach and used a saving heuristic to create an initial solution. Many researchers (Chao et al., 1993; Renaud et al., 1996; Cordeau et al., 1997; Ho et al., 2008) used this simple strategy to develop algorithms for solving MDVRPs. The cluster approach usually divides customers into clusters based on their geographical distribution and then allocates clusters to the appropriate depot. Thangiah and Salhi (2001) proposed an adaptive clustering method based on geometric shapes for the MDVRP. Giosa et al. (2002) adopted the “cluster first, route second” heuristic and introduced six algorithms to assign customers to clusters. Dondo and Cerdá (2007) gathered customers into clusters and assigned clusters to depots using a mixed-integer linear programming model. The third approach assigns border and non-border customers to depots by recognizing the importance of border customers for constructing globally optimized MDVRP solutions. Salhi and Sari (1997) divided customers into border and non-border customers according to the ratio of their distances to their nearest and second-nearest depots. Kuo and Wang (2012) used a probability function to represent the probability that a border or non-border customer is assigned to a depot.

The use of metaheuristics is a realistic approach to solve an MDVRP in a reasonable time and usually generates near-optimal solutions (Gendreau et al., 2008). Two often-used metaheuristics for MDVRPs are local search and evolutionary algorithms. Local search improves an MDVRP solution by moving from the current solution to a neighborhood solution in the search space of candidate solutions by making local changes. The search space and the local changes are crucial to the efficiency of MDVRP algorithms and the quality of their solutions. Several reduction strategies for the search space have been introduced to save computing time:  $k$ th nearest neighbor (Li et al., 2005),  $k$ -ring Voronoi neighbors (Fang et al., 2012), and granular neighborhoods (Toth and Vigo, 2003). Local changes are largely dependent on neighborhood structures and include 1–0 exchange, 1–1 exchange, 2-opt, 3-opt, or-opt, the three-point move (Groër et al., 2010; Zachariadis and Kiranoudis, 2010), generalized insertion (GENI) (Cordeau et al., 1997), and others. However, a local search can be easily trapped in local minima. To escape local minima, several strategies—such as Tabu search (TS) (Renaud et al., 1996; Cordeau et al., 1997; Toth

and Vigo, 2002; Crevier et al., 2007; Laporte, 2009), simulated annealing (SA) (Osman, 1993; Kirkpatrick et al., 1983; Wu et al., 2002; Lim and Zhu, 2006) and variable neighborhood search (VNS) (Mladenović and Hansen, 1997; Hansen and Mladenović, 2001; Polacek et al., 2004; Ergun et al., 2006; Ropke and Pisinger, 2006; Kytöjoki et al., 2007; Lee et al., 2008; Milthers, 2009; Fleszar et al., 2009; Imran et al., 2009; Kuo and Wang, 2012)—have been used for MDVRPs by accepting worse-case solutions in the local search. These local search strategies require thorough knowledge of the spatial distribution of customers to limit local searches efficiently and exactly.

Evolutionary algorithms, including GA (Bae et al., 2007; Ho et al., 2008; Berman and Hanshar, 2009), ACO (Yu et al., 2010), and PSO (Marinakos et al., 2010), have been widely used to solve VRPs. These three types of evolutionary algorithms have demonstrated a powerful ability to solve MDVRPs. To demonstrate the usefulness of Voronoi diagrams for solving MDVRPs with very large-scale customer bases, this paper proposes a bi-level Voronoi diagram-based metaheuristic and integrates it into an SA algorithm (Kirkpatrick et al., 1983; Osman, 1993) to test its performance. Moreover, this proposed metaheuristic can be further used to improve the performance of evolutionary algorithms for MDVRPs.

### 3. Bi-level Voronoi diagrams

This section presents bi-level Voronoi diagrams by introducing the concept of the  $k$ -ring Voronoi neighbors of a point and a line, which are used in the MDVRP algorithm described below.

#### 3.1. $k$ -Ring Voronoi neighbors of a point and a line

Based on the definitions of Voronoi diagram, Voronoi region, and Voronoi edges (Aurenhammer, 1991; Okabe et al., 2000; Chen et al., 2000), the  $k$ -ring Voronoi neighbors of a point and a line are defined below respectively:

**Definition 1.** A Voronoi distance (Duczmalski et al., 2011; Fang et al., 2012) between points  $p_i$  and  $p_j$  is the number of Voronoi edges crossed when moving from  $p_i$  to  $p_j$  and is denoted as  $vd(p_i, p_j)$ . If  $p_i$  and  $p_j$  are the same point, then  $vd(p_i, p_j) = 0$ . For example, the Voronoi distance between the two points ( $P_1$  and  $P_2$ ) in Fig. 1a is 2.

**Definition 2.** The  $k$ -ring Voronoi neighbors of a point  $p_i$  (Fang et al., 2012) are points that have a Voronoi distance  $k$  to point  $p_i$ , which can be defined as:

$$KVN(p_i, k) = \{x | vd(p_i, x) = k, x \in P, k \in N\}, P = \{p_1, p_2, \dots, p_q\} \quad (1)$$

A  $k$ -ring Voronoi neighbor set  $P'$  of a point  $p_i$  includes points with the maximum Voronoi distance  $k$  to point  $p_i$ , which can be defined as:

$$KVN(P', k) = \{x | \min vd(p_i, x) = k, x \in P, p_i \in P', P' \subset P, k \in N\}. \quad (2)$$

For example, Fig. 1b illustrates the 1-ring and 2-ring Voronoi neighbors of the seed (point).

**Definition 3.** The  $k$ -ring Voronoi neighbors of a line are the points around the line with a Voronoi distance  $k$  from the line, which is defined as:

$$KVN(L, k) = \{x | vd(x, s) = k, s \in KVN(L, 0), k \in N; x, s \in V_c; k \geq 0\} \quad (3)$$

$$KVN(L, 0) = \{x | VR_x \cap L \neq \emptyset, x \in P\} \quad (4)$$

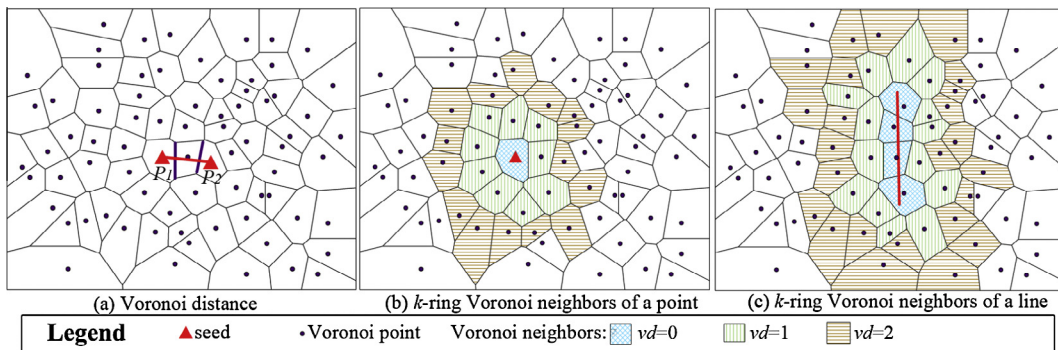


Fig. 1.  $k$ -ring Voronoi neighbors in a Voronoi diagram.

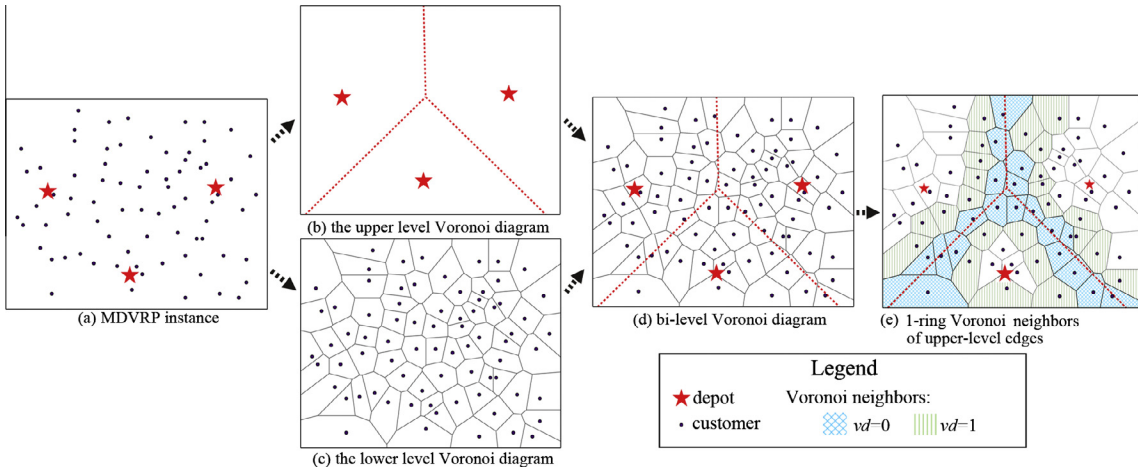


Fig. 2. Construction of a bi-level Voronoi diagram.

where  $L$  denotes the geometry of the line. The value  $KVN(L,0)$  represents the Voronoi neighbors crossed by  $L$ . For example, Fig. 1c illustrates the 1-ring and 2-ring Voronoi neighbors of the red<sup>1</sup> line.

### 3.2. Bi-level Voronoi diagrams

This study proposes a bi-level Voronoi diagram (Fig. 2) to help optimize customer allocation in an MDVRP. The upper-level Voronoi diagram (Fig. 2b) is derived from the depot set  $V_d$  in a MDVRP, while the lower-level Voronoi diagram (Fig. 2c) is derived from the customer set  $V_c$  in the MDVRP. The upper-level Voronoi diagram coarsely defines an initial coverage area for each depot. This diagram helps allocate customers to the right depot and limits the search space required for vehicle-routing optimization of each depot. The lower-level Voronoi diagram defines the Voronoi neighbors of customers, which helps to optimize the vehicle routing problem for each depot by limiting the local search space to the  $k$ -ring Voronoi neighbors. The two levels of Voronoi diagrams are generated by Fortune's (1987) sweep algorithm in  $O(N \log N)$  time. Fig. 2d illustrates the bi-level Voronoi diagram, which defines a basic customer partitioning strategy for each depot.

We should note that the  $\lambda$ -ring Voronoi neighbors of the Voronoi edge in the upper Voronoi diagram have a high probability of being reallocated among their nearby depots during the process of searching for the optimal solution. These Voronoi neighbors form a dynamic customer reallocation area, which is the critical area addressed by the MDVRP solution improvement operators in the proposed algorithm below. Fig. 2e illustrates the 1-ring Voronoi neighbors around the Voronoi edges of depots.

## 4. The proposed algorithm

Based on the bi-level Voronoi diagram (BVD), a bi-level Voronoi diagram-based metaheuristic (BVDH) algorithm for solving large-scale MDVRPs is proposed here. The BVDH algorithm is an extension of the standard simulated annealing. The key idea of the BVDH algorithm is to limit the optimization search space in simulated annealing by using the  $k$ -ring Voronoi neighbors in the BVDH.

The proposed BVDH algorithm consists of two phases. The first phase is to construct an initial solution. A bi-level Voronoi diagram is generated to divide the customers into groups served by different depots. Then, an initial routing for each depot is created using the single-depot "cluster-first, route-second" VRP algorithm proposed by Fang et al. (2012) with the constraints on the minimum increment of route lengths and the maximum capacity of vehicles. The second phase is to improve the solution using allocation modification and local search strategies. In this phase, the algorithm employs two strategies to adjust vehicle routings in the current solution. One is the reallocation of customers in the  $\lambda$ -ring Voronoi neighbors around the Voronoi edges of the upper-level diagram. This strategy aims at improving the vehicle routings that cross Voronoi regions of different depots. The other strategy is to rearrange customers among the routes of each depot using the  $k$ -ring Voronoi neighbors in local searches. These two strategies together can reduce the local search space, thereby speeding up the solution of large-scale MDVRPs. The details of this algorithm are described in the following two subsections.

<sup>1</sup> For interpretation of color in Fig. 1, the reader is referred to the web version of this article.



#### 4.1. The construction phase

This phase constructs an initial solution for the MDVRP. A customer is initially allocated to a depot if the customer is located in the Voronoi region of the depot in the upper-level Voronoi diagram. The initial vehicle routes are created for each depot. Routes for the customers allocated to a depot are generated using the single-depot “cluster-first, route-second” VRP algorithm (Fang et al., 2012). These routes represent the initial solution of the MDVRP.

#### 4.2. The improvement phase

This algorithm extends the SA algorithm (Kirkpatrick et al., 1983; Osman, 1993) to improve the current MDVRP solution. The improvement consists of two parts: routings within the space of one depot and routings between the depots.

To improve routings within the space of one depot, this algorithm rearranges customers from one route to another for each depot by limiting the local search space to the  $k$ -ring Voronoi neighbors and using three popular neighborhood structures of the 1–0 exchange move, the 1–1 exchange move, and the 2-opt move (Zachariadis and Kiranoudis, 2010). The three neighborhood structures are implemented for each customer and its  $k$ -ring neighbors in the nearby routes, only if these moves can reduce the total route length and meet the Metropolis et al.’s (1953) acceptance criterion stated in Fang et al. (2012).

To improve the routings between the depots, this algorithm uses only the 1–0 exchange move and the 1–1 exchange move to improve the solution. These two moves are implemented within the  $\lambda$ -ring Voronoi neighbors of Voronoi edges in the upper-level Voronoi diagram. The 2-opt move is not used because it could generate many routes across the depots, which could increase the total route length of the current solution. The 1–0 and 1–1 exchange moves can generate three kinds of possible inappropriate routes, namely, (i) a route has a customer that is located outside the service area of the route’s depot or outside the  $\lambda$ -ring Voronoi neighbors of its previous or next customer in the route; (ii) two routes belonging to different depots have edges and customers that are too close to each other; (iii) two routes belonging to different depots cross each other. These three kinds of inappropriate routes have a strong effect on the solution quality. This proposed algorithm therefore identifies these inappropriate routes and modifies the between-depot routes to improve the solution. These three kinds of inappropriate routes can be defined as follows:

- (i) An inappropriate route occurs when a route has a customer that is located outside the service area of a depot or outside the  $\lambda$ -ring Voronoi neighbors of its previous or next customer in this route. This kind of inappropriate routes can be defined as:

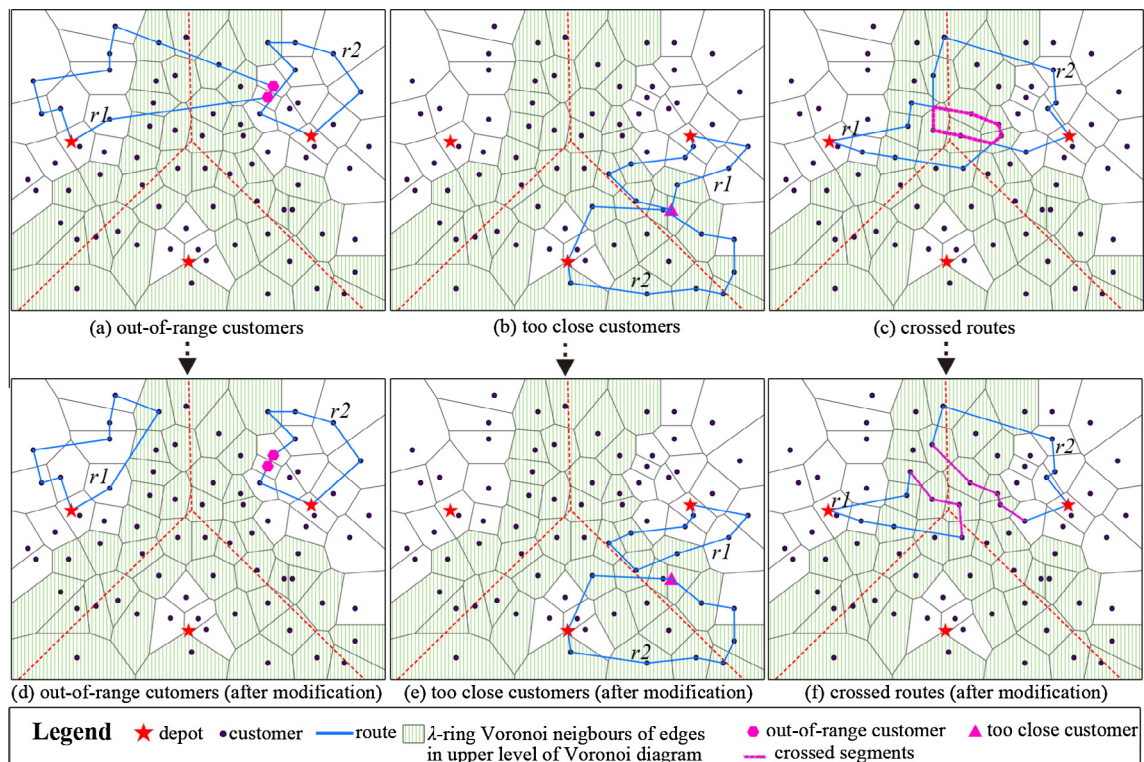


Fig. 3. Inappropriate routes between depots and their modifications.

$$R_i = \exists x \in R_i, x \text{ is out of } VR_d \ \& \ x \notin KVN(\lambda, VE_d) \quad (5)$$

where  $R_i$  is a route  $i$  starting and ending at depot  $d$ ,  $x$  is a customer served by  $R_i$ ,  $VR_d$  is the Voronoi region of depot  $d$ ,  $VE_d$  denotes the Voronoi edges of  $VR_d$ , and  $KVN(\lambda, VR_d)$  denotes the  $\lambda$ -ring Voronoi neighbors of  $VE_d$ . Fig. 3a illustrates this kind of inappropriate route. A modification of this inappropriate route is to eject those points that do not belong to the current depot from their original route and insert them into the nearest route belonging to another depot. Fig. 3d shows the modified result of the inappropriate route.

(ii) Two routes belonging to different depots have edges and customers that are too close to each other, which can be defined as:

$$R_i = \exists x \in R_i, \exists y \in KVN(1, x), d_x \neq d_y \ \& \ dist(x, y) < MinE \quad (6)$$

where  $x, y$  are customers,  $KVN(1, x)$  is the set of 1-ring Voronoi neighbors of  $x$ ,  $d_x$  is the depot serving customer  $x$ ,  $d_y$  is the depot serving customer  $y$ ,  $dist(x, y)$  denotes the distance between  $x$  and  $y$ , and  $MinE$  is a small constant parameter used to filter near neighbors. This value is set to three times the shortest distance between customers. Fig. 3b illustrates this kind of inappropriate route and Fig. 3e shows its modification, for example, moving customer  $x$  to the position before or after customer  $y$  in its nearest route.

(iii) Two routes belonging to different depots cross each other, which can be defined as:

$$R_i = \exists R_j, R_i \text{ is crossed with } R_j, d_i \neq d_j \quad (7)$$

where  $R_i$  denotes a route starting from depot  $d_i$  and  $R_j$  denotes a route starting from depot  $d_j$ . Fig. 3c shows an example of this kind of route. The modification of this inappropriate route is to swap the two crossing route segments and form new routes. Fig. 3f shows the modified result.

During these modifications, the vehicle capacity and route length are not observed strictly as the parameters in MDVRP. Therefore, after these modifications, the solution may contain some over-capacity routes. These routes should be further optimized. To address each over-capacity route, this algorithm finds its customer member who has the minimum added cost and it is moved to a nearby route within  $k$ -ring Voronoi neighbors belonging to the same depot. The cost is calculated as:

$$f(b, i) = (d_{ib} + d_{bj} + d_{ac}) - (d_{ab} + d_{bc} + d_{ij}) \quad (8)$$

where  $a, b, c, i, j$  are customers,  $a$  is the node visited before  $b$  in the route,  $c$  is the node visited after  $b$ , and  $j$  is the node visited after  $i$ .  $b$  and  $j$  should be on different routes. This equation represents the cost to insert node  $b$  between nodes  $i$  and  $j$ . This algorithm moves selected customer members on the over-capacity route to its nearby routes with the minimum added cost. The pseudo-code for the modification of over-capacity routes is provided below:

---

1	$R = \Phi$ .
2	scan current solution $s$ , <b>if</b> $D_r > Q    L_r > L$ , $R = R \cup r$ .
3	<b>while</b> $R \neq \Phi$
4	<b>for</b> route $r \in R$
5	<b>while</b> $D_r > Q    L_r > L$ <b>do</b>
6	$b^* = -1$ ; $i^* = -1$ ; $f^* = +\infty$
7	<b>for</b> customer $b$ in the route $r$
8	<b>for</b> customer $i \in KVN(b, k)$
9	<b>if</b> $r_b \neq r_i \ \& \ f(b, i) < f^*$
10	$b^* = b$ ; $i^* = i$ ; $f^* = f(b, i)$ .
11	<b>end if</b>
12	<b>end for</b>
13	<b>end for</b>
14	Insert $b^*$ at the best position $i^*$ .
15	<b>if</b> $D_r \leq Q \ \& \ L_r \leq L$ , <b>break</b> .
16	<b>end while</b>
17	<b>end for</b>
18	<b>end while</b>

---

#### 4.3. Pseudo-code for the BVDH algorithm

The parameters and the pseudo-code of the proposed BVDH algorithm are summarized below:

Parameters:

- $T_0$ : starting temperature for simulated annealing.
- $C$ : cooling ratio

- $I_{\max}$ : maximum iteration number for simulated annealing
- $K$ : Voronoi neighborhood size for local search.
- $\lambda$ : Voronoi neighborhood size for the dynamic allocation strategy.
- $I_1$ : iteration interval to use the dynamic allocation strategy.

Algorithm (BVDH):

---

**Construction phase:**

- 1 Read data and initialize parameters.
- 2 Build the bi-level Voronoi diagrams, including the Voronoi diagram for depots, the Voronoi diagram for customers, and the set of  $k$ -ring Voronoi neighbors for each customer.
- 3 **for** each depot  $d$
- 4      $C_d = \emptyset$
- 5     **for** each **customer**  $c$
- 6         **if**  $c$  is **within**  $VR_d$ ,  $C_d = C_d \cup c$
- 7     **end for**
- 8 Generates initial routes with the depot  $d$  and the corresponding customer set  $C_d$  using the construction algorithm of Fang et al. (2012).
- 9 **end for**

**Improvement phase:**

- 10 Identify the  $\lambda$ -ring Voronoi neighbors along the depot's Voronoi edges and add them to an empty node set  $V_\lambda$ .
- 11 Set  $i = 1$ ,  $t = T_0$ .
- 12 **for**  $i = 1, 2, \dots, I_{\max}$
- 13     **for** each depot  $d$ ,
- 14         Generate a random sequence of its served customers. For each customer, perform a local search operation on the  $k$ -ring Voronoi neighbors served by the same depot using the 1–0 exchange move, 1–1 exchange move, and 2-opt move. Perform acceptable local change operations.
- 15     **end for**
- 16     **for** each customer  $c$  in  $V_\lambda$
- 17         Perform local search operations on the  $k$ -ring Voronoi neighbors served by different depots using 1–0 exchange moves and 1–1 exchange moves. Perform acceptable operations.
- 18     **end for**
- 19 Update the best solution  $S^*$  if there has been an improvement.
- 20 **if**  $i/I_1 == 0$ , execute the dynamic customer allocation strategy.
- 21 Update parameters,  $i = i + 1$ ,  $t = t * c$ .
- 22 **end for**
- 23 Return the best found solution  $S^*$ .

**Output:**

Report the best found solution  $S^*$ .

---

## 5. Computational experiment

A computational experiment was conducted to investigate the performance of the proposed algorithm. In this experiment, parameters of the BVDH algorithm were tested first, and then the results were compared with several state-of-the-art MDVRP metaheuristics listed in Table 1, which include Tabu search (Cordeau et al., 1997), set partitioning (SP, Baldacci and Mingozzi, 2009), multi-RPERT (MP, Salhi and Sari, 1997), hybrid genetic search with adaptive diversity control algorithm (HGSADC, Vidal et al., 2012), parallel improved ant colony optimization (PIACO Yu et al., 2010), and adaptive large neighborhood search (ALNS, Pisinger and Ropke, 2007). This experiment also compared the results of BVDH algorithm with the result of a modified vehicle routing problem with neighborhood ejection (VRPEJ) algorithm known as MDVRPEJ (multi-depot-VRPEJ) that enables the solving of very large-scale MDVRP instances (Groër et al., 2010). In the MDVRPEJ, the initial solutions were modified according to the rule that each customer was assigned to its nearest depot. Then, these solutions were optimized using the operations in the improvement phase of the VRPEJ algorithm. These approaches were tested using twenty-three published benchmark MDVRP instances and seven large-scale real-world MDVRP instances. The twenty-three published benchmark MDVRP datasets collected by Cordeau et al. (1997) were obtained from the Web site at <http://neo.lcc.uma.es/radi-aeb/WebVRP/> (see Table 1). These datasets include seven small-scale MDVRP problems (P1–P7) contributed by Christofides and Eilon (1969), four medium-scale problems (P8–P11) contributed by Gillett and Johnson (1976), and twelve medium-scale problems (P12–P23) contributed by Chao et al. (1993).

**Table 1**

Solution results of the benchmark datasets for the BVDH and other metaheuristics.

Pro	N	M	Q	L	Best scores of other metaheuristic								BVDH algorithm				
					SP	Tabu	MP	HGSADC	PIACO	ALNS	MDVRPEJ	BKS	$S_{avg}$	$S_{best}$	$Gap\_S_{avg}$	$Gap\_S_{best}$	$T_{avg}$
P01	50	4	80	$\infty$	576.87	576.87	587.8	576.87	576.86	576.87	578.20	576.86	577.33	576.87	0.08	0.00	3.30
P02	50	4	160	$\infty$	473.53	473.53	476.7	473.53	473.53	473.53	480.26	473.53	475.65	473.53	0.45	0.00	3.02
P03	75	5	140	$\infty$	640.65	641.19	644.5	641.19	641.18	641.19	654.34	640.65	644.86	641.19	0.66	0.08	4.56
P04	75	2	100	$\infty$	999.21	1001.59	1042.3	1001.23	1001.49	1001.04	1006.59	999.21	1005.99	1002.05	0.68	0.28	6.13
P05	100	2	200	$\infty$	752.16	750.03	777.2	750.03	750.26	751.26	751.90	750.03	751.77	750.03	0.23	0.00	6.72
P06	100	3	100	$\infty$	876.50	876.50	888.2	876.50	876.50	876.70	887.30	876.50	881.81	878.76	0.61	0.26	5.89
P07	100	4	100	$\infty$	881.97	885.80	911.8	884.43	885.69	881.97	908.96	881.97	887.77	882.22	0.66	0.03	5.72
P08	249	2	500	310	–	4437.68	4513.3	4372.78	4482.38	4390.80	4405.94	4372.78	4449.57	4426.37	1.34	0.88	17.07
P09	249	3	500	310	–	3900.22	4005.3	3858.66	3912.23	3873.64	3917.55	3858.66	3908.03	3886.43	1.28	0.72	17.12
P10	249	4	500	310	–	3663.02	3824.7	3631.11	3663.00	3650.04	3749.57	3631.11	3676.39	3661.16	1.25	0.83	16.38
P11	249	5	500	310	–	3554.18	3714.3	3546.06	3554.08	3546.06	3644.97	3546.06	3610.81	3554.18	1.83	0.23	16.26
P12	80	2	60	$\infty$	1318.95	1318.95	1326.8	1318.95	1318.95	1318.95	1347.09	1318.95	1320.71	1318.95	0.13	0.00	7.27
P13	80	2	60	200	–	1318.95	1318.9	1318.95	1318.95	1318.95	1329.71	1318.95	1318.95	1318.95	0.00	0.00	6.43
P14	80	2	60	180	–	1360.12	1360.1	1360.12	1365.68	1360.12	1422.65	1360.12	1360.67	1360.12	0.04	0.00	6.82
P15	160	4	60	$\infty$	2505.42	2505.42	2579.3	2505.42	2551.45	2505.42	2569.80	2505.42	2523.80	2505.42	0.73	0.00	10.51
P16	160	4	60	200	–	2572.23	2584.5	2572.23	2572.23	2572.23	2584.53	2572.23	2582.90	2572.23	0.41	0.00	9.82
P17	160	4	60	180	–	2709.09	2720.5	2709.09	2708.99	2709.09	2748.46	2708.99	2721.86	2709.09	0.47	0.00	11.12
P18	240	6	60	$\infty$	–	3702.85	3822.1	3702.85	3781.03	3702.85	3832.10	3702.85	3815.57	3702.85	3.04	0.00	13.61
P19	240	6	60	200	–	3827.06	3863.9	3827.06	3827.06	3827.06	3869.45	3827.06	3834.30	3827.06	0.19	0.00	13.09
P20	240	6	60	180	–	4058.07	4074.8	4058.07	4097.06	4058.07	4134.94	4058.07	4120.77	4058.07	1.55	0.00	12.87
P21	360	9	60	$\infty$	–	5474.84	5788.0	5474.84	5474.74	5474.74	5692.27	5474.74	5600.55	5474.74	2.30	0.00	19.74
P22	360	9	60	200	–	5702.15	5765.9	5702.16	5772.23	5702.15	5738.68	5702.15	5739.11	5702.15	0.65	0.00	18.38
P23	360	9	60	180	–	6095.46	6106.6	6078.75	6125.58	6078.75	6191.11	6078.75	6249.46	6106.60	2.81	0.46	19.76
Found best count					7	13	2	17	9	14	0			12			
Mean gap					0.03	0.21	2.14	0.02	0.57	0.08	1.84				0.95	0.18	
$T_1$ (/s)					875.3	803.4	1116	244.8	–	237	105.59						11.65
CPU (Mflop/s)					1098	23	5.7	3181	–	3181	3859						3859
$T_2$ (/s)					249.05	4.78	1.64	201.79	–	195.36	105.59						11.65

Pro: MDVRP instance name; N: the number of customers; M: the number of depots; Q: the capacity of vehicle; L: the max length of route. SP: Baldacci and Mingozzi (2009) (run on AMD Athlon 64 X2 Dual Core 4200+ processor at 2.6 GHz); Tabu: algorithm of Cordeau et al. (1997) (run on Sun Sparcstation 10); MP: Salhi and Sari (1997) (run on VAX 4000-500); HGSADC: algorithm of Vidal et al. (2012) (the computing times were on Pentium IV 3.0 GHz factors); PIACO: algorithm of Yu et al. (2010); ALNS: algorithm of Pisinger and Ropke (2007) (run on Pentium IV 3 GHz). MDVRPEJ: a modified VRPEJ algorithm (Groër et al., 2010) for the MDVRP (run on Intel(R) Core(TM) i3-2100 @ 3.10 GHz). BKS: the best-known result of these metaheuristics or algorithms ( $=\min(\text{SP}, \text{Tabu}, \text{MP}, \text{HGSADC}, \text{PIACO}, \text{ALNS})$ ).  $S_{avg}$ : the average total route length over 10 runs for BVDH.  $S_{best}$ : the best total route length for BVDH.  $T_{avg}$  is the average computation time of 10 runs, reported in seconds.  $Gap\_S_{avg}$ : the percent gap between  $S_{avg}$  and  $BKS(=(S_{avg} - BKS)/BKS)$ .  $Gap\_S_{best}$ : the percent gap between  $S_{best}$  and  $BKS(=(S_{best} - BKS)/BKS)$ .  $T_1$ : the original computation time for algorithms.  $T_2$ : the computation time on the same PC with BVDH (run on Core i3-2100, 3.10 GHz). CPU Mflop/s: Dongarra's (2011) factor of the CPU. All computation time reported are in seconds.



**Table 2**

Computation results of the seven large-scale MDVRP instances.

Pro	N	M	Q	MDVRPEJ		BVDH algorithm				
				$S_{best}^0$	$T_{avg}^0$	$S_{avg}$	$S_{best}$	$gap_{avg}$	$gap_{best}$	$T_{avg}$
BJ1	2000	2	2000	3,332,412.9	2170.60	3,313,274.5	3,267,542.7	−0.57	−1.95	148.49
BJ2	3000	3	2000	4,062,014.8	4254.49	4,059,966.7	4,017,472.3	−0.05	−1.10	214.77
BJ3	5000	5	2000	5,418,969.9	6823.94	5,421,604.7	5,391,065.7	0.05	−0.51	535.72
BJ4	8000	8	2000	6,618,834.64	12,861.56	6,644,987.0	6,593,490.9	0.40	−0.38	1583.46
BJ5	12,000	12	2000	8,076,173.98	21,154.06	8,066,060.3	7,948,419.5	−0.13	−1.58	3369.99
BJ6	16,000	16	2000	8,734,438.29	30,362.49	8,778,254.6	8,647,432.3	0.50	−1.00	8967.21
BJ7	20,000	20	2000	10,428,568.5	64,936.62	10,337,753.2	10,228,429.3	−0.87	−1.92	16,099.63
Average gap				0.00				−0.10	−1.21	
Average computing time (/s)					20,366.25					4417.00

Pro: MDVRP instance name; N: the number of customers; M: the number of depots; Q: the capacity of vehicle; MDVRPEJ: a modified VRPEJ algorithm (Groër et al., 2010) for the MDVRP (run on Intel(R) Core(TM) i3-2100 @ 3.10 GHz).  $S_{best}^0$ : the best total route length over 10 runs of the MDVRPEJ.  $T_{best}^0$ : the average computation time of the MDVRPEJ, reported in seconds.  $S_{avg}$ : the average total route length over 10 runs of the BVDH;  $S_{best}$ : the best total route length of the BVDH.  $T_{avg}$  is the average computation time of 10 runs of the MDVRP, reported in seconds.  $gap_{avg}$ : the deviation of the  $S_{avg}$  to  $S_{best}^0$  ( $= (S_{avg} - S_{best}^0) * 100 / S_{best}^0$ ).  $gap_{best}$ : the deviation of the  $S_{best}$  to  $S_{best}^0$  ( $= (S_{best} - S_{best}^0) * 100 / S_{best}^0$ ).

In these instances, the number of customers ranges from 50 to 360 and the number of depots varies from two to nine. In each instance, customers are geographically distributed in a Euclidean plane, and depots are located within the space of all customers. The seven large-scale real-world MDVRP instances (Table 2) were generated from shopping malls and stores in Beijing, China and each point represents a customer with a random demand between 1 and 100. This dataset has 2000–20,000 customers and 2–20 depots. The capacity of each vehicle is 2000 units of demand. Each vehicle can serve about 40 customers on a route.

### 5.1. Experimental settings and parameter testing

The BVDH algorithm described in Section 4 was implemented using C++ programming in Microsoft VC2008 and executed on a personal computer with a dual-core CPU, the Intel(R) Core(TM) i3-2100 @ 3.10 GHz, and Windows 7 64-bit operating system (only one core was used in the BVDH algorithm).

There are five main parameters in the algorithm: the number of  $k$ -ring Voronoi neighbors for local search, the number of  $\lambda$ -ring Voronoi neighbors along depot edges for customer allocation among the depots, the start temperature  $T_0$ , the cooling ratio  $c$ , and the maximum number of iterations  $I_{max}$  for the SA algorithm. This section describes the testing and setting of these parameters.

The first parameter  $k$  defines the exploration ability of a local search. A larger value of  $k$  means that the algorithm needs to explore more local search candidates and can obtain a higher-quality solution, but the BVDH algorithm will take more computation time. This paper tested the parameter  $k$  using the same approach as in Fang et al. (2012), and selected a value of  $k = 2$  for the benchmark instances used in this experiment.

The second parameter  $\lambda$  determines the search space of neighbors to reallocate customers among the depots and it also affects the solution quality and computation time. This parameter depends on the number of customers in the benchmark instances. Coy et al. (2001, p. 96) proposed “a procedure based on statistical design of experiments that systematically selects high-quality parameters values” and demonstrated that this approach “worked well on both the capacity-constrained and route-length-constrained problems”. This parameter  $\lambda$  in the MDVRP problem therefore was selected based on the Coy’s approach. The selected  $\lambda$  parameters in this study avoid the problem of choosing the most proper parameter setting for each instance, but maintain fairness for comparisons of different algorithms (Silberholz and Golden, 2010). Specifically, the BVDH algorithm was tested on benchmark instances P06 and BJ3 to determine this parameter  $\lambda$  for small- and medium-sized benchmark MDVRP instances and for large-scale real-world MDVRP instances. For the small- and medium-sized benchmark MDVRP instances, the value  $\lambda = 1$  was used for the twenty-three published benchmark datasets due to its least total route length and its acceptable computation time (6.1 s). To improve the computation time, the proposed algorithm needs a smaller neighbor search space when reallocating customers between the depots to obtain a higher-quality solution. For the seven large-scale real-world MDVRP instances,  $\lambda = 4$  was used because the BVDH algorithm can find a good solution within a relatively short computation time.

The other three parameters of  $T_0$ ,  $c$  and  $I_{max}$  are related to the SA algorithm. When  $T_0$  has a higher value, the SA usually is easier to accept the worst-case solution (Osman, 1993). The cooling ratio  $c$  controls the speed of temperature drop. A lower  $c$  means a quicker convergence for SA, but the algorithm may generate a worse final solution. The parameters of  $T_0$  and  $c$  were tested ten times with the benchmark instances described in the following section. The SA algorithm was found to generate a good solution when  $T \in [0.2\bar{e}, 2\bar{e}]$  and  $c \in [0.9, 0.999]$ , where  $\bar{e}$  denotes the average distance of customers to their 1-ring Voronoi neighbors. The value of  $T_0$  was set to the average distance to all Voronoi neighbors for each small- and medium-sized benchmark MDVRP instance to start the simulated annealing at the same level in Section 5.2, and was set to the average distance to all Voronoi neighbors in the lower-level Voronoi diagram for the seven large-scale real-world MDVRP instances in Section 5.3. Based on the results of ten tests, the value of  $c$  was set to the average value. Therefore,  $c = 0.9965$  was selected

for small- and medium-sized benchmark MDVRP instances in Section 5.2, and  $c = 0.992$  was selected for the seven large-scale real-world MDVRP instances in Section 5.3.

The final parameter  $I_{\max}$  affects the ending temperature of the SA, and the value of  $I_{\max}$  is highly dependent on the customer number in the benchmark instances. This study set the maximum number of iterations  $I_{\max}$  at the maximum value when the solution equality (i.e. total route length) becomes stable among the results of ten experimental tests, which resulted in  $I_{\max} = 2000$  for small- and medium-sized benchmark MDVRP instances in Section 5.2 and  $I_{\max} = 1000$  for the seven large-scale real-world MDVRP instances in Section 5.3.

## 5.2. Results of the twenty-three published benchmark problems

The BVDH algorithm was tested on the benchmark dataset with the following parameters:  $k = 2$ ,  $c = 0.9965$ ,  $I_{\max} = 2000$ ,  $\lambda = 1$  and  $I_1 = 50$  for small- and medium-sized benchmark instances.

Table 1 lists the solutions generated by the BVDH algorithm and other state-of-the-art algorithms for the twenty-three published benchmark datasets. We ran the BVDH algorithm ten times to test its stability in solving the MDVRP. The total route length and the computation time in each run were recorded to calculate the  $S_{avg}$ ,  $S_{best}$  and  $T_{avg}$  in Table 1. The gaps reported in this table indicate the extent to which each solution is poorer than the best known solution (Silberholz and Golden, 2010). The BVDH algorithm found the same best known solution in 12 instances. The solutions for the remaining eleven instances (P01, P03–04, P06–11, P17 and P23) have a maximum gap value less than 1.23%. The proposed algorithm has a smaller average gap value (0.18%) than several state-of-the-art MDVRP metaheuristics such as Tabu (0.21%), MP (2.14%), PIACO (0.57%), and MDVRPEJ (1.84%). Only SP, HGSADC, and ALNS algorithms have a smaller average gap value (0.03%, 0.02%, and 0.02% respectively) than the proposed BVDH algorithm. Therefore, the proposed BVDH is an effective algorithm that can generate a high quality solution for these small-scale and medium-scale MDVRP instances.

To compare the computation time of these algorithms, the published computation time of different machines (CPUs) were converted to an equivalent time on the personal computer employed in this study using Dongarra's (2011) factor. Dongarra's factor is commonly used to convert the CPU time of different machines to support comparison of computing speeds among different algorithms. The conversion equation is:

$$T_2 = T_1 * \text{Dongarra}_1 / \text{Dongarra}_2, \quad (9)$$

where  $T_1$  and  $T_2$  are the computation time of an algorithm on machines with CPU1 and CPU2 respectively, and  $\text{Dongarra}_1$  and  $\text{Dongarra}_2$  are the Dongarra factors for CPU1 and CPU2 respectively. The value of  $T_2$  in Table 1 represents the converted computation time of each algorithm. As indicated in Table 1, Tabu and MP have a relatively small amount of computation time (average 4.78 s and 1.64 s). The BVDH algorithm takes much less computation time (average 11.65 s) than SP (average 249.05 s), HGSADC (average 201.79 s), ALNS (average 195.36 s), and MDVRPEJ (average 105.59 s) algorithms. The proposed algorithm therefore performs well for solving small-scale and medium-scale MDVRP instances with respect to computation time.

## 5.3. Results of the seven real-world large-scale MDVRPs

The BVDH algorithm was tested with the benchmark dataset using the following parameters:  $k = 2$ ,  $c = 0.992$ ,  $I_{\max} = 1000$ ,  $\lambda = 4$  and  $I_1 = 50$  for the seven real-world large-scale MDVRPs. Since the performance of some state-of-the-art algorithms is not publically available, we report only the results of the MDVRPEJ heuristic and the BVDH algorithm here.

Table 2 shows the solutions of the MDVRPEJ heuristic and the BVDH algorithm for the seven real-world large-scale MDVRP instances. In terms of the solution quality, the BVDH algorithm outperformed the MDVRPEJ. For example, the best solution  $S_{best}$  of the BVDH is better than that of the best solution  $S_{best}^0$  of the MDVRPEJ by 1.21% (Table 2). In terms of the computation time, the BVDH has an average time of 4417 s, which is only 21.69% of the MDVRPEJ (20,366.25 s). The average gap between the average solution  $S_{avg}$  and the best solution  $S_{best}$  for the seven large-scale real-world MDVRP instances is 1.11% (i.e.,  $-0.10\% - (-1.21\%)$ ). The maximum gap is 1.50% for instance BJ6 (i.e.,  $0.50\% - (-1.00\%)$ ), and the minimum gap is 0.56% for instance BJ3 (i.e.,  $0.05\% - (-0.51\%)$ ). These results suggest that the BVDH algorithm is robust in solving large-scale MDVRP instances up to 20,000 customers. The average computation time to derive the best result is 16,099.63 s. This study has demonstrated that the BVDH algorithm is able to find a high-quality solution for real-world large-scale MDVRP instances with a reasonable amount of computation time.

## 6. Discussion

### 6.1. The Voronoi distance of MDVRP solutions

To demonstrate the usefulness of the proposed bi-level Voronoi diagram in generating high-quality solutions, this study compared the Voronoi distances of published solutions of the 23 benchmark MDVRP instances generated by the Tabu heuristic (Cordeau et al., 1997) (<http://neo.lcc.uma.es/radi-aeb/WebVRP/>) and the proposed BVDH algorithm. Table 1 shows that the proposed BVDH algorithm can generate solutions of better quality than the Tabu search algorithm. Table 3 gives the

**Table 3**

Distribution of customers' Voronoi distances in the solutions of the twenty-three benchmark datasets.

Pro	Tabu				BVDH					
	vd=	1	2	3	>=4	vd=	1	2	3	>=4
P01		39	0	0	0		39	0	0	0
P02		43	2	0	0		44	1	0	0
P03		63	1	0	0		63	1	0	0
P04		77	6	2	0		77	8	0	0
P05		91	1	0	0		91	1	0	0
P06		80	3	1	0		80	4	0	0
P07		78	5	1	0		79	5	0	0
P08		189	27	6	2		202	17	5	0
P09		204	16	2	1		201	20	2	0
P10		209	12	2	0		209	12	2	0
P11		212	9	2	0		212	9	2	0
P12		66	6	0	0		66	6	0	0
P13		66	6	0	0		66	6	0	0
P14		64	7	0	1		64	7	0	1
P15		136	8	0	0		136	8	0	0
P16		136	6	2	0		135	8	1	0
P17		128	12	0	4		128	12	0	4
P18		203	14	0	0		203	14	0	0
P19		206	7	3	0		206	7	3	0
P20		192	17	0	7		192	17	0	7
P21		306	20	0	0		306	20	0	0
P22		312	8	4	0		312	8	4	0
P23		288	27	0	9		288	29	0	7
Sum		3388	220	25	24		3400	218	20	19
Percentage		92.64	6.02	0.68	0.66		92.97	5.96	0.55	0.52

**Table 4**

Distribution of customers' Voronoi distances in the solutions of the seven large-scale MDVRP instances.

Pro	BVDH				
	vd=	1	2	3	>=4
BJ1		1759	158	20	10
BJ2		2663	213	34	11
BJ3		4430	356	64	15
BJ4		7022	632	106	26
BJ5		10,420	1038	171	47
BJ6		13,888	1378	236	88
BJ7		16,772	2228	348	102
Sum		56,954	6003	979	299
Percentage		88.67	9.35	1.52	0.47

results of customers' Voronoi distances for the 23 benchmark datasets using both algorithms. Of the edges generated by the proposed BVDH algorithm, 92.97% have a Voronoi distance of one. This percentage is 0.33% higher than that generated by the Tabu search algorithm. Considering the percentage of edges within a maximum Voronoi distance of two, the BVDH algorithm (92.97% + 5.96% = 98.93%) is also better than the Tabu search algorithm (92.64% + 6.02% = 98.66%). The high percentage of short Voronoi distances in the solutions is an indication of better quality solutions generated by the proposed BVDH algorithm. The high percentage of route segments with short Voronoi distances demonstrates the usefulness of the proposed bi-level Voronoi diagram in generating high-quality solutions for small- and medium-size benchmark instances.

For the seven large-scale MDVRP instances shown in Table 4, the proposed BVDH algorithm can generate high-quality solutions that have 98.02% (i.e., 88.67% + 9.35%) of their edges within a maximum Voronoi distance of two. This high percentage of edges with short Voronoi distances demonstrates that the proposed BVDH algorithm can generate high-quality solutions for real-world large-scale MDVRP instances up to 20,000 customers.

## 6.2. Tradeoff between solution quality and computation time

To demonstrate the capability of balancing solution quality and computation time, this study investigated the total route length and the computation time of instance BJ5 generated by the proposed BVDH algorithm under different  $k$  and  $\lambda$  combinations. Fig. 4 illustrates the computation performance of these solutions. In terms of total route length, Fig. 4a shows a general tendency for the average total route length to decrease with increasing  $k$ . When  $\lambda = 4$ , the proposed algorithm

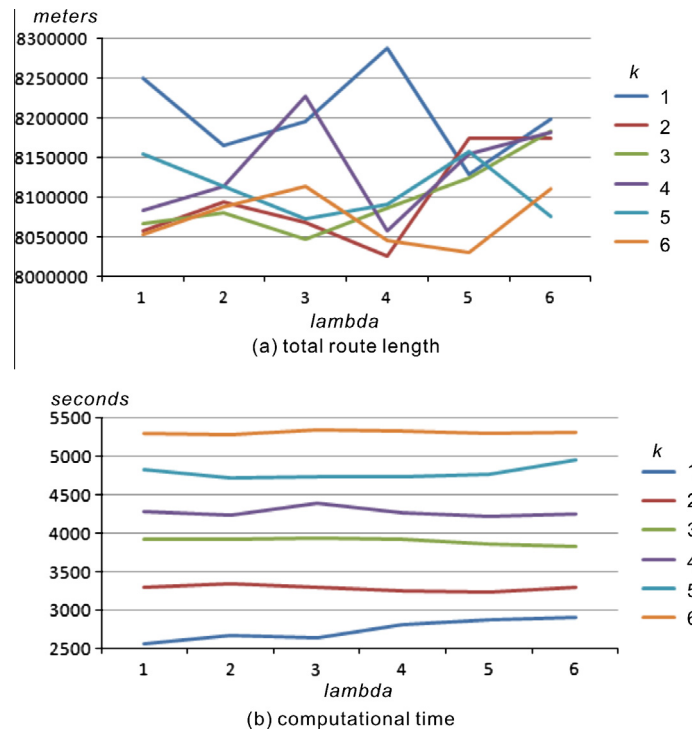


Fig. 4. Computational performance under different  $k$  and  $\lambda$  combinations.

generated high-quality solutions for most  $k$  values. This outcome confirms the rationale of choosing  $\lambda = 4$  for the real-world large-scale MDVRP instances. In Fig. 4b, the computation time is obviously affected by the parameter  $k$ . The proposed BVDH algorithm needs more computation time with increasing  $k$  values. Therefore, the proposed BVDH algorithm includes an easy approach to implement the tradeoff between solution quality and computation time by using the parameter  $k$ .

## 7. Conclusions

This paper proposes a bi-level Voronoi diagram metaheuristic for large-scale real-world MDVRPs. The bi-level Voronoi diagram consists of two diagrams: the upper-level diagram derived from the depots, and the lower-level Voronoi diagram derived from the customers. The upper-level Voronoi diagram helps identify the border and non-border customers of each depot, while the lower-level diagram helps establish a set of  $k$ -ring neighbors. Both diagrams are used to limit the local search space for customer reallocation between the depots and customer rearrangement among the routes of each depot. This metaheuristic reduces the neighborhood search space in the simulated annealing algorithm. Computational experiments indicate that the proposed metaheuristic can solve large-scale real-world MDVRPs more efficiently than many other state-of-the-art algorithms. Moreover, the proposed algorithm performs well with several small- and medium-scale benchmark MDVRP instances. We therefore have demonstrated that the proposed algorithm is efficient and effective to solve MDVRPs.

The contributions of this paper to the MDVRP literature can be summarized as follows: (1) Although several spatial neighbor-reduction strategies such as granular neighborhood search (Toth and Vigo, 2003),  $k$ th nearest neighbor (Li et al., 2005), and the set of  $k$ -ring Voronoi neighbors (Fang et al., 2012), have been used to reduce the search effort for vehicle routing in MDVRPs, few studies have demonstrated the benefits of using Voronoi diagrams to generate high-quality MDVRP solutions while balancing solution quality and computation time. This paper demonstrates the efficiency of a bi-level Voronoi diagram in limiting the local search space for customer reallocation between the depots and customer rearrangement between the routes served by each depot. (2) Although there have been many studies of MDVRP (e.g., Chao et al., 1993; Renaud et al., 1996; Cordeau et al., 1997; Thangiah and Salhi, 2001; Dondo and Cerdá, 2007; Ho et al., 2008; Baldacci and Mingozzi, 2009, and Berman and Hanshar, 2009), very few approaches have attempted to solve a large-scale MDVRP with thousands of spatially distributed customers. Our proposed metaheuristic can efficiently solve large-scale MDVRP instances up to 20,000 customers.

Our future research plans to extend the proposed bi-level Voronoi diagram metaheuristic and algorithm with the following approaches: (1) design an adaptive approach to determine the ring parameter  $k$  based on different spatial patterns of depots and customers; (2) develop a parallel computing approach using Voronoi spatial decomposition to improve the

efficiency of solving MDVRPs with large-scale real-world customer base; and (3) extend the Voronoi-based spatial neighborhood metaheuristic to solve some important variants of MDVRP such as MDVRP with time windows and MDVRP with pickup and delivery. Future improvements in these aspects will enable applications in transportation, logistics and supply-chain management to optimize large-scale vehicle routing problems in a more realistic context.

## Acknowledgements

This research was supported in part by the National Science Foundation of China (Grants #41231171, #41301511, #41371420, #41071285), Shenzhen Scientific Research and Development Funding Program (#ZDSY20121019111146499), Shenzhen Dedicated Funding of Strategic Emerging Industry Development Program (#JCYJ20121019111128765) and the Funding for Excellent Talents in Wuhan University. The authors also would like to thank the anonymous reviewers and Professor Wayne Talley for their valuable comments and suggestions.

## References

- Aurenhammer, F., 1991. Voronoi diagrams – a survey of a fundamental geometric data structure. *ACM Computing Surveys* 23 (3), 345–405.
- Bae, S.-T., Hwang, H.S., Cho, G.-S., Goan, M.J., 2007. Integrated GA-VRP solver for multi-depot system. *Computers & Industrial Engineering* 53 (2), 233–240.
- Baldacci, R., Mingozzi, A., 2009. A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming* 120 (2), 347–380.
- Berman, B.O., Hanshar, F., 2009. Using genetic algorithms for multi-depot vehicle routing. In: Pereira, F., Tavares, J. (Eds.), *Bio-inspired Algorithms for the Vehicle Routing Problem*, vol. 161. Springer, Berlin/Heidelberg, pp. 77–99.
- Chao, I., Golden, B., Wasil, E., 1993. A new heuristic for the multi-depot vehicle routing problem that improves upon best known solutions. *American Journal of Mathematical and Management Sciences* 13 (3–4), 371–406.
- Chen, J., Li, C., Li, Z., Gold, C., 2000. A Voronoi-based 9-intersection model for spatial relations. *International Journal of Geographical Information Science* 15 (3), 201–220.
- Christofides, N., Eilon, S., 1969. An algorithm for the vehicle-dispatching problem. *The Operational Research Society* 20 (3), 309–318.
- Cordeau, J.F., Gendreau, M., Laporte, G., 1997. A Tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks* 30 (2), 105–119.
- Coy, S., Golden, B., Runger, G.C., Wasil, E.A., 2001. Using experiment design to find effective parameter setting for heuristic. *Journal of Heuristics* 7 (1), 77–97.
- Crevier, B., Cordeau, J., Laporte, G., 2007. The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research* 176 (2), 756–773.
- Dantzig, G., Ramser, J., 1959. The truck dispatching problem. *Management Science* 6 (1), 80–91.
- Dondo, R., Cerdá, J., 2007. A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *European Journal of Operational Research* 176 (3), 1478–1507.
- Dongarra, J.J., 2011. Performance of Various Computers Using standard Linear Equation Software. <<http://www.netlib.org/benchmark/performance.pdf>> (accessed 15.06.12).
- Duczmal, L.H., Moreira, G.J.P., Burgarelli, D., Takahashi, R.H.C., Magalhães, F.C.O., Bodevan, E.C., 2011. Voronoi distance based prospective space-time scans for point data sets: a dengue fever cluster analysis in a southeast Brazilian town. *International Journal of Health Geographics* 10 (1), 29–43.
- Erdogan, S., Miller-Hooks, E., 2012. A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review* 48 (1), 100–114.
- Ergun, O., Orlin, J.B., Steele-Feldman, A., 2006. Creating very large scale neighbourhoods out of smaller ones by compounding moves. *Journal of Heuristics* 12 (1–2), 115–140.
- Fang, Z., Tu, W., Li, Q., Shaw, S.-L., Chen, S., Chen, B.Y., 2013. A Voronoi neighborhood-based search heuristic for distance/capacity constrained very-large vehicle routing problems. *International Journal of Geographical Information Science* 27 (4), 741–764.
- Fleszar, K., Osman, I.H., Hindi, K.S., 2009. A variable neighbourhood search algorithm for the open vehicle routing problem. *European Journal of Operational Research* 195 (3), 803–809.
- Fortune, S., 1987. A sweepline algorithm for Voronoi diagrams. *Algorithmica* 2, 153–174.
- Gendreau, M., Potvin, J.Y., Bräumlays, O., Hasle, G., Løkketangen, A., 2008. Metaheuristics for the vehicle routing problem and its extensions: a categorized bibliography. In: Golden, B., Raghavan, S., Wasil, E. (Eds.), *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer Science, New York, pp. 143–169.
- Gillett, B.E., Johnson, J.G., 1976. Multi-terminal vehicle-dispatch algorithm. *Omega* 4 (6), 711–718.
- Giosa, I., Tansini, I., Viera, I., 2002. New assignment algorithms for the multi-depot vehicle routing problem. *Journal of the Operational Research Society* 53 (9), 977–984.
- Groër, C., Golden, B., Wasil, E.A., 2010. A library of local search heuristics for the vehicle routing problem. *Mathematical Programming Computation* 2 (2), 79–101.
- Hansen, P., Mladenovic, N., 2001. Variable neighborhood search: principles and applications. *European Journal of Operational Research* 130 (3), 449–467.
- Hemmelmayr, V.C., Doerner, K.F., Hartl, R.F., 2009. A variable neighborhood search heuristic for periodic routing problems. *European Journal of Operational Research* 195 (3), 791–802.
- Ho, W., Ho, G.T.S., Ji, P., Lau, H.C.W., 2008. A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Engineering Applications of Artificial Intelligence* 21 (4), 548–557.
- Imran, A., Salhi, S., Wassan, N.A., 2009. A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem. *European Journal of Operational Research* 197 (2), 509–518.
- Kirkpatrick, S., Gelatt, J.C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. *Science* 220 (4598), 671–680.
- Kuo, Y., Wang, C.-C., 2012. A variable neighborhood search for the multi-depot vehicle routing problem with loading cost. *Expert Systems with Applications* 39 (8), 6949–6954.
- Kytöjoki, J., Nuortio, T., Bräysy, O., Gendreau, M., 2007. An efficient variable neighborhood search heuristic for very large scale vehicle routing problems. *Computers & Operations Research* 34 (9), 2743–2757.
- Laporte, G., 2009. Fifty years of vehicle routing. *Transportation Science* 43 (4), 408–416.
- Laporte, G., Nobert, Y., Arpin, D., 1984. Optimal solutions to capacitated multipoint vehicle routing problems. *Congressus Numerantium* 44, 283–292.
- Laporte, G., Nobert, Y., Taillefer, S., 1988. Solving a family of multi-depot vehicle routing and location-routing problems. *Transportation Science* 22 (3), 161–172.
- Lee, Y.H., Kim, J.I., Kang, K.H., Kim, K.H., 2008. A heuristic for vehicle fleet mix problem using Tabu search and set partitioning. *Journal of the Operational Research Society* 59 (6), 833–841.
- Li, F., Golden, B., Wasil, E., 2005. Very large-scale vehicle routing: new test problems, algorithms, and results. *Computers & Operations Research* 32 (5), 1165–1179.
- Lim, A., Wang, F., 2005. Multi-depot vehicle routing problem: a one-stage approach. *IEEE Transactions on Automation Science and Engineering* 2 (4), 397–402.



- Lim, A., Zhu, W., 2006. A fast and effective insertion algorithm for multi-depot vehicle routing problem with fixed distribution of vehicles and a new simulated annealing approach. In: Ali, M., Dapoigny, R. (Eds.), *Advances in Applied Artificial Intelligence*, vol. 4031. Springer, Berlin/Heidelberg, pp. 282–291.
- Liu, S., Huang, W., Ma, H., 2009. An effective genetic algorithm for the fleet size and mix vehicle routing problems. *Transportation Research Part E: Logistics and Transportation Review* 45 (3), 434–445.
- Liu, T., Jiang, Z., Liu, R., Liu, S., 2011. A review of the multi-depot vehicle routing problem. *Energy Procedia* 13, 3381–3389.
- Marinakos, Y., Marinaki, M., Dounias, G., 2010. A hybrid particle swarm optimization algorithm for the vehicle routing problem. *Engineering Applications of Artificial Intelligence* 23 (4), 463–472.
- Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E., 1953. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* 21, 1087–1092.
- Milthers, N.P.M., 2009. Solving VRP using Voronoi Diagrams and Adaptive Large Neighborhood Search. Master Thesis. Department of Computer Science, University of Copenhagen.
- Mladenović, N., Hansen, P., 1997. Variable neighborhood search. *Computers & Operations Research* 24 (11), 1097–1100.
- Okabe, A., Boots, B., Sugihara, K., Chiu, S.N., 2000. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. John Wiley & Sons, Chichester, England.
- Ombuki-Berman, B., Hanshar, F.T., 2009. Using genetic algorithms for multi-depot vehicle routing. In: Pereira, F.B., Tavares, J. (Eds.), *Bio-inspired Algorithms for the Vehicle Routing Problem*, vol. 161. Springer-Verlag, Berlin Heidelberg, pp. 77–99.
- Osman, I.H., 1993. Metastrategy simulated annealing and Tabu search algorithms for the vehicle routing problem. *Annals of Operations Research* 41 (4), 421–451.
- Ouyang, Y., 2007. Design of vehicle routing zones for large-scale distribution systems. *Transportation Research Part B: Methodological* 41 (10), 1079–1093.
- Pisinger, D., Ropke, S., 2007. A general heuristic for vehicle routing problems. *Computers & Operations Research* 34 (8), 2403–2435.
- Polacek, M., Hartl, R.F., Doerner, K., Reimann, M., 2004. A variable neighborhood search for the multi depot vehicle routing problem with time windows. *Journal of Heuristics* 10 (6), 613–627.
- Potvin, J.Y., 2009. State-of the art review – evolutionary algorithms for vehicle routing. *INFORMS Journal on Computing* 21 (4), 518–548.
- Qi, M., Lin, W.-H., Li, N., Miao, L., 2012. A spatiotemporal partitioning approach for large-scale vehicle routing problems with time windows. *Transportation Research Part E: Logistics and Transportation Review* 48 (1), 248–257.
- Renaud, J., Laporte, G., Boctor, F.F., 1996. A Tabu search heuristic for the multi-depot vehicle routing problem. *Computers & Operations Research* 23 (3), 229–235.
- Ropke, S., Pisinger, D., 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science* 40 (4), 455–472.
- Salhi, S., Sari, M., 1997. A multi-level composite heuristic for the multi-depot vehicle fleet mix problem. *European Journal of Operational Research* 103 (1), 95–112.
- Sheu, J.-B., Talley, W., 2011. Green supply chain management: trends, challenges, and solutions. *Transportation Research Part E: Logistics and Transportation Review* 47 (6), 791–792.
- Silberholz, J., Golden, B., 2010. Comparison of metaheuristics. In: Gendreau, M., Potvin, J.Y. (Eds.), *Handbook of Metaheuristics*. Springer, Berlin, pp. 625–640.
- Thangiah, S.R., Salhi, S., 2001. Genetic clustering: an adaptive heuristic for the multidepot vehicle routing problem. *Artificial Intelligence* 15 (4), 361–383.
- Tillman, F.A., 1969. The multiple terminal delivery problem with probabilistic demands. *Transportation Science* 3 (3), 192–204.
- Toth, P., Vigo, D., 2002. *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, Philadelphia.
- Toth, P., Vigo, D., 2003. The granular Tabu search and its application to the vehicle-routing problem. *INFORMS Journal on Computing* 15 (4), 333–346.
- Vidal, T., Crainic, T., Gendreau, M., Lahrichi, N., Rei, W., 2012. A hybrid genetic algorithm for multi-depot and periodic vehicle routing problems. *Operations Research* 60 (3), 611–624.
- Wu, T., Low, C., Bai, J., 2002. Heuristic solutions to multi-depot location-routing problems. *Computers & Operations Research* 29 (10), 1393–1415.
- Yu, B., Yang, Z.-Z., 2011. An ant colony optimization model: the period vehicle routing problem with time windows. *Transportation Research Part E: Logistics and Transportation Review* 47 (2), 166–181.
- Yu, B., Yang, Z.-Z., Xie, J.X., 2010. A parallel improved ant colony optimization for multi-depot vehicle routing problem. *Journal of the Operational Research Society* 62 (1), 183–188.
- Zachariadis, E.E., Kiranoudis, C.T., 2010. A strategy for reducing the computational complexity of local search-based methods for the vehicle routing problem. *Computers & Operations Research* 37 (12), 2089–2105.