A Multi-Depot Two-Echelon Vehicle Routing Problem with Delivery Options Arising in the Last Mile Distribution

A Multi-Depot Two-Echelon Vehicle Routing Problem with Delivery Options Arising in the Last Mile Distribution

Lin Zhou

College of Management, Chongqing University of Technology, Chongqing, 400054, China

Roberto Baldacci[1]

DEI, University of Bologna, Cesena, Italy

Daniele Vigo

DEI, University of Bologna, Cesena, Italy

Xu Wang

College of Mechanical Engineering, Chongqing University, Chongqing, China

Abstract

In this paper, we introduce a new city logistics problem arising in the last mile distribution of e-commerce. The problem involves two levels of routing problems. The first requires a design of the routes for a vehicle fleet located at the depots to transport the customer demands to a subset of the satellites. The second level concerns the routing of a vehicle fleet from the satellites to serve all of the customers. A feature of the problem is that customers may provide different delivery options, allowing them to pick up their packages at intermediate pickup facilities. The objective is to minimize the total distribution cost. To solve the problem, a hybrid multi-population genetic algorithm is proposed. An effective heuristic algorithm is designed to generate initial solutions, and several procedures are designed to better manage the population as well as exploit and explore the solution space. The proposed method is tested on a large family of instances, including a real-world instance; the computational results obtained show the effectiveness of the different components of the algorithm.

Keywords: Routing; City logistics; Two-echelon; Hybrid genetic algorithm

August 1, 2017

1 Introduction

The enormous growth in China's e-commerce market has driven a comparably explosive growth in the logistical demands of online shopping. The total number of packages delivered during 2015 was 20.6 billion, and the number of packages delivered daily will reach two billion in the next five years. Every day, large amounts of goods are transported via long-haul vehicles, capable of carrying large quantities of freight (Gianessi et al., 2015) but

[1]Corresponding author. Tel.: +39 0547 338834; fax: +39 0547 338890. E-mail address: r.baldacci@unibo.it.

inappropriate for delivering packages to individual customers, due to increasing traffic congestion, environment issues, and small truckloads. Thus, the last mile distribution plays an indispensable role in serving the final customers. Moreover, in contrast to traditional city distribution, customers with small daily delivery demands and different availability schedules are widely distributed spatially, which makes last mile distribution a complex issue. As a result, the last mile distribution is regarded as the most expensive, most polluting, and least efficient part of the e-commerce supply chain, and accounts for 13% to 75% of the total supply chain cost (Gevaers et al., 2009). The last mile distribution challenge has become the bottleneck of e-commerce (Wang et al., 2014).

In the last mile distribution, direct package delivery to recipients' homes or workplaces (Home Delivery, HD) is a time-consuming and costly process. Customer pickup (CP) provides another delivery option, by allowing customers to pick up their packages at a pickup facility close to home or work. Providing this service means that, instead of delivering all packages directly to the customers, some packages could be delivered to the pickup facilities; the delivery efficiency is improved. For the customers, picking up their packages at a convenient time, at a preferred place, will satisfy them to the largest extent.

In this paper, we address a new city logistics problem called the Multi-Depot Two-Echelon Vehicle Routing Problem with Delivery Options for the Last Mile Distribution (MD-TEVRP-DO). In the problem, a customer can be served either by HD or CP, and each customer is associated with a depot. Each depot belongs to a different logistics operator. The deliveries to the customers from the depots are managed through shared intermediate capacitated facilities, called satellites. In the two-level distribution network, the first level consists of vehicle routes that start and end at the depots and consolidate customers' orders at a subset of satellites. At the second level, the vehicles have a capacity smaller than those at the first level; they deliver the consolidated orders from the satellites to individual customers and pickup facilities.

To solve the problem, a hybrid multi-population genetic algorithm is proposed. An effective heuristic algorithm is designed to generate initial solutions, and several procedures are designed to better manage the population as well as exploit and explore the solution space.

The rest of the paper is organized as follows. The next section reviews the relevant literature. In Section 2 we formally describe the problem. The hybrid genetic algorithm is detailed in Section 3. Computational results are reported and analyzed in Section 4; concluding remarks are given in Section 5.

1.1 Literature review

As far as the authors know, the MD-TEVRP-DO has never been addressed in the literature, whereas a number of closely related problems have been proposed and studied.

The most closely related problem is the Two-Echelon Vehicle Routing Problem (2E-

VRP). In the 2E-VRP, deliveries to customers from a depot are managed through intermediate capacitated satellites. The first level consists of vehicle routes that start and end at the depot and deliver the customer demands to a subset of satellites. A satellite has a limited capacity and can be serviced by more than one first-level route. The second level consists of vehicle routes that start and end at the same satellite and supply all customers. A homogeneous vehicle fleet is used at each level. The first-level vehicles are located at the depot and supply the satellites only. The second-level vehicles have a capacity smaller than that of the first-level vehicles and supply the customers from the satellites. The unloading of first-level vehicles and loading of second-level vehicles at the satellites imply a handling cost proportional to the quantity loaded/unloaded. The 2E-VRP aims to find two sets of first and second level routes such that each customer is visited exactly once by a second-level route and the total routing and handling cost is minimized. In the 2E-VRP, split deliveries are allowed in the first-level whereas in the MD-TEVRP-DO, a multi-commodity many-to-many problem, split deliveries are not allowed in both the levels. Therefore, the 2E-VRP can be seen as a relaxation of the special case of the MD-TEVRP-DO with one depot, a non-binding working time limit and with HD options only.

Initially proposed by Gonzales Feliu et al. (2007), the 2E-VRP has been considered by several authors (see, for example, Perboli et al. (2011), Jepsen et al. (2013), Baldacci et al. (2013), and Santos et al. (2014)). Crainic et al. (2009) tackled a different version of the 2E-VRP, with time-dependent, synchronized, multi-depot, multi-product, heterogeneous fleets and time windows. Soysal et al. (2015) focused on a time-dependent 2E-VRP with environmental considerations. A 2E-VRP with cross-docking facilities has been considered by Ahmadizar et al. (2015). Most recently, a multi-trip 2E-VRP was discussed by Grangier et al. (2016).

A similar problem is the two-echelon location routing problem (2E-LRP), which can be divided into a single-depot 2E-LRP (Jacobsen and Madsen, 1980; Nguyen et al., 2012a,b) and a multi-depot 2E-LRP (Madsen, 1983; Laporte and Nobert, 1988; Crainic et al., 2011a,c; Schwengerer et al., 2012; Contardo et al., 2012; Govindan et al., 2014).

Concerning other related problems, Gianessi et al. (2015) proposed a multicommodity-ring location routing problem (MRLRP) in which distribution centers are connected by rings and the commodities can be delivered either to the second distribution centers or directly to the customers. Rieck et al. (2014) considered a many-to-many LRP with inter-hub and mixed routes, similar to the MRLRP.

Several other studies share the concern of optimizing delivery options; a few examples follow. In the generalized vehicle routing problem (GVRP) (Ghiani and Improta, 2000; Pop et al., 2012; Bektaş et al., 2011; H et al., 2014), the vertices partitioned into each cluster can act as alternative delivery locations. The ring-star problem (RSP) (Baldacci et al., 2007; Baldacci and Dell'Amico, 2010) allows customers to be connected to a ring through transition points, and the covering tour VRP (CTVRP) (Tricoire et al., 2012;

Naji-Azimi et al., 2012; H et al., 2013; Jozefowiez, 2014; Allahyari et al., 2015; Flores-Garza et al., 2015) satisfies customer demand by visiting or covering the customer along the tour.

The interested reader is referred to the reviews by Prodhon and Prins (2014), Drexl and Schneider (2015), and Cuda et al. (2015) on 2E-VRP and 2E-LRP; Braekers et al. (2016) and Cattaruzza et al. (2015) on VRP; and, specifically, Bektaş et al. (2015) for a systematic view of city logistics. In particular, Cuda et al. (2015) provide a classication and a systematic overview of two-echelon distribution systems with routes present at both levels. Their work considers three classes of problem: the 2E-VRP, the 2E-LRP, and the truck and trailer routing problem.

As realistically sized instances are rarely solved to optimality by pure exact methods, metaheuristics is the methodology of choice in most cases (Crainic and Sgalambro, 2014). Several recent works utilizing metaheuristics can be found in the literature: tabu search (Crainic et al., 2011c); iterated local search (Nguyen et al., 2012a); genetic algorithm (Rieck et al., 2014); adaptive large neighborhood search (Hemmelmayr et al., 2012; Contardo et al., 2012); variable neighborhood search (Schwengerer et al., 2012); greedy randomized adaptive search procedure, complemented by a learning process and path relinking (Nguyen et al., 2012b); and genetic algorithm with local search (Vidal et al., 2012; Ahmadizar et al., 2015).

2 Problem description

The MD-TEVRP-DO in this paper can be formally described as follows.

A mixed graph $G = (N, A, C)$ is given, where the node set N is partitioned as $N = N_D \cup N_S \cup N_P \cup N_C$. Set $N_D = \{1, \ldots, n_d\}$ represents n_d depots, $N_S = \{n_d+1, \ldots, n_d+n_s\}$ represents n_s satellites, $N_P = \{n_d+n_s+1, \ldots, n_d+n_s+n_p\}$ represents n_p pickup facilities, and $N_C = \{n_d + n_s + n_p + 1, \ldots, n_d + n_s + n_p + n_c\}$ represents n_c customers. The arc set A is defined as $A = \{(d, j), (j, d) : d \in N_D, j \in N_S\} \cup \{(i, j) : i, j \in N_S \cup N_P \cup N_C\}$. With each arc $(i, j) \in A$ are associated a *routing cost* r_{ij} and a *travel time* $t_{ij} > 0$.

Associated with the customer set N_C are n_c customer requests. Each customer request $i \in N_C$ has an associated depot node $o_i \in N_D$, demand $q_i > 0$, and service time $s_i > 0$. For each customer $i \in N_C$, let $P_i \subseteq N_P \cup \{i\}$ represent *delivery options* for the customer. The following cases apply to set P_i:

i) $P_i = \{i\}$: customer i requires a delivery directly to the home location;

ii) $P_i \subseteq N_P$, $P_i \neq \emptyset$: customer i will pickup the delivery from a pickup location selected from a set of preferred pickup facilities;

iii) $P_i \subseteq N_P \cup \{i\}$, $i \in P_i$, $P_i \cap N_P \neq \emptyset$: customer i is willing to receive the delivery either at the home location or through a pickup facility.

Set C represents the possible connections between customers and pickup facilities, i.e. $C = \{(i,j) : i \in N_C, j \in P_i\}$. Each arc is associated with a nonnegative *connection cost* c_{ij}. Arc $(i,i) \in C$, $i \in N_C$, indicates that customer i is connected to him/herself, i.e. on a route.

A fleet of m_d^1 identical vehicles of capacity Q_d^1 are located at depot $d \in N_D$ and used to transport goods to satellites. A maximum working time equal to L_d^1 is associated with each first-level vehicle of depot d. A fleet of m_k^2 identical vehicles of capacity Q_k^2 are available at satellite $k \in N_S$ to service the customers. Nevertheless, at most $m^2 \leq \sum_{k \in N_S} m_k^2$ second-level vehicles can be used globally. In addition, associated with each satellite k is the time u_k required to unload a unit of product at that satellite location. A maximum working time equal to L_k^2 is associated with each second-level vehicle of satellite k.

Given a depot $d \in N_D$, we denote with $N_C(d) = \{i \in N_C : o_i = d\}$ the set of customer requests delivered by depot d. A first-level *route* $R^1 = (i_0 = d, i_1, \dots, i_r, i_{r+1} = d)$, with $r \geq 1$, for depot $d \in N_D$, is a simple circuit in G passing through depot d, loading a subset $T(R^1) \subseteq N_C(d)$ of customer requests at the depot and visiting satellites $V(R^1) = \{i_1, \dots, i_r, \} \subseteq N_S$, such that:

i) the total demand of the transportation requests does not exceed the vehicle capacity Q_d^1, i.e., $\sum_{i \in T(R)} q_i \leq Q_d^1$;

ii) the total working time of the route does not exceed the vehicle working time, i.e., $\sum_{h=1}^{r+1} t_{i_{h-1} i_h} + \sum_{h=1}^{r} u_{i_h} \sum_{i \in T_{i_h}(R)} q_i \leq L_d^1$, for $T_{i_h}(R)$ the subset of transportation requests unloaded by the vehicle at satellite i_h, $h = 1, \dots, r$.

If used, a first-level vehicle incurs a fixed cost U_d^1. The cost of a first-level route is the sum of the costs of the traversed arcs plus the fixed cost U_d^1 of the associated depot d.

Each satellite $k \in N_S$ can be visited by more than one first-level route and has a capacity B_k that limits the total customer demand that can be delivered to it by the first-level routes. A second-level *route* is defined by a pair (R^2, C') where $R^2 = (i_0 = k, i_1, \dots, i_r, i_{r+1} = k)$, $r \geq 1$, is a simple circuit in G passing through the satellite $k \in N_S$, visiting nodes $V(R^2) = \{i_1, \dots, i_r\} \subseteq N_C \cup N_P$, and $C' \subseteq C$ are assignments between customers of $N_C \setminus V(R^2)$ and nodes of $V(R^2) \cap N_P$.

We say that a customer i is *assigned* to a route R^2 if it is either *visited* by the simple circuit (i.e., $i \in V(R^2)$) or it is *connected* to a node of the route representing a pickup facility (i.e., a node $j \in V(R^2) \cap N_P$ exists such that $(i,j) \in C'$). Let $V_C(R^2)$ and $V_P(R^2)$ be the subsets of customers and pickup facilities visited by route R, respectively.

Each pickup facility $p \in N_P$ can be visited by more than one second-level route.

The total load of a route is computed as the sum of the demands of the customers assigned to the route. The route is *feasible* if its total load does not exceed the vehicle capacity Q_k^2, i.e. $\sum_{i \in V_C(R)} q_i + \sum_{i \in V_P(R)} \sum_{(j,i) \in C'} q_i \leq Q_k^2$ and if its working time, computed as $\sum_{h=1}^{r+1} t_{i_{h-1} i_h} + \sum_{i \in V_C(R)} s_i + \sum_{i \in V_P(R)} v_i \sum_{(j,i) \in C'} s_i$, is less than or equal to L_k^2,

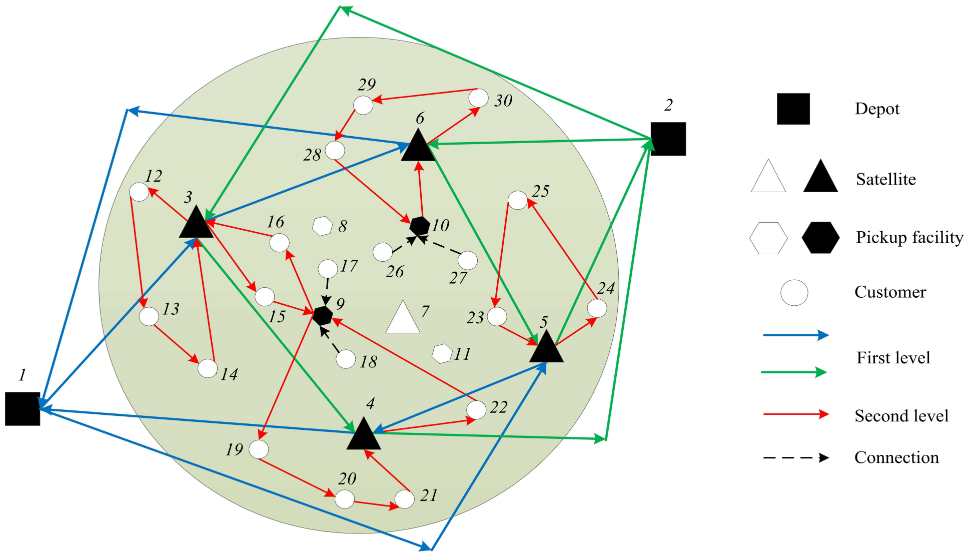


Figure 1: A solution to the MD-TEVRP-DO

where $0 < v_i \leq 1$ is the service time coefficient associated with pickup facility $i \in N_P$.

If used, a second-level vehicle incurs a fixed cost U_k^2. The cost of a second-level route is the sum of the traversed arcs, plus the sum of the assignment costs of the arcs in C', plus the fixed cost U_k^2 of the associated satellite k. The handling cost at satellite $k \in N_S$ is given by H_k times the quantity delivered to satellite k.

The problem seeks to design the vehicle routes of both levels so that each customer request is visited exactly once, the transportation requests delivered to customers from each satellite correspond to the transportation requests received from the depots, and the sum of the routing, connection, and handling costs is minimized.

An example of a last mile two-echelon distribution system is shown in Figure 1. It consists of 2 depots, 5 potential satellites, 4 potential pickup facilities, and 19 customers to be served.

The MD-TEVRP-DO considered in this paper differs from the 2E-VRP studied in the literature in the following ways: (i) it considers more than one depot; (ii) it considers vehicle working time constraints in addition to vehicle capacity constraints; (iii) it considers delivery options; and (iv) due to the presence of the customer requests, it is a multi-commodity many-to-many problem. In contrast, the 2E-VRP is a one-commodity one-to-many problem in which a single commodity must be delivered from the depot to many customers. This means that in a feasible 2E-VRP solution the quantity to be delivered to a customer can be delivered to the satellite serving the customer from different first-level routes, a situation that cannot occur in the MD-TEVRP-DO case.

3 Hybrid approach

The proposed Hybrid Multi-Population Genetic (HMPG) metaheuristic is motivated by the following considerations. First, the ability of local search (LS) to find local optima by exploring different regions of search space (Derbel et al., 2012), and the global search characteristic that can be gained by using genetic algorithm (GA). Second, the use of a multi-population strategy can help to improve the search efficiency and speed up the evolution process.

Our algorithm is based on the Hybrid Genetic Search with Adaptive Diversity Control metaheuristic (HGSADC) proposed by Vidal et al. (2012); it is regarded as the most efficient heuristic in solving vehicle routing problems. Indeed, it has been proved to outperform the current state-of-the-art metaheuristics, such as tabu search, scatter search, variable neighborhood search, fuzzy-logic guided-GA, record-to-record ILP, and adaptive large neighborhood search some of which have been widely used for the 2E-LRP (Crainic et al., 2011c; Schwengerer et al., 2012; Contardo et al., 2012) and the 2E-VRP (Hemmelmayr et al., 2012; Nguyen et al., 2012b; Ahmadizar et al., 2015).

The scheme of the proposed heuristic is displayed in algorithm 1. The method evolves both feasible and infeasible solutions, which are kept in subpopulations. More precisely, an initial population of feasible solutions is generated and further partitioned into n_{sp} subpopulations (denoted by P_i, $i = 1, \ldots, n_{sp}$). A unique subpopulation of infeasible solutions in considered (denoted by PI). The algorithm then successively applies a number of operators to each feasible subpopulation in order to select two parent individuals and combine them, yielding two new individuals (offsprings), which are first enhanced using a local search procedure (education) and then included in the appropriate subpopulation according to their feasibility (see algorithm 2). Further, one of the two new individuals generated is selected to have a mutation operator applied. The resulting solution is included in the appropriate subpopulation according to its feasibility. At each main iteration, the algorithm also applies the algorithm 2 to a pair of parent individuals, one selected from a feasible subpopulation and one from the infeasible subpopulation. In the algorithm, every I_{SHARE} iterations the best feasible solution belonging to subpopulations P_i, $i = 1, \ldots, n_{sp}$. is shared among the different subpopulations.

In the following sections, we further detail the different steps of algorithms 1 and 2.

3.1 Solution representation

The individuals in the HMPG population are represented as a set of the following three chromosomes.

(1) The first chromosome represents first-level routes, with a permutation of depots and satellites. Routes belonging to the same depot are grouped in the chromosome by the

Algorithm 1 HMPG

Initialize population and improve it (*local search procedure*)
Partition the initial population into n_{sp} subpopulations represented by $P_1, P_2, \ldots, P_{n_{sp}}$
Set $PI = \emptyset$ (initial infeasible subpopulation)
while *number of iterations* $< I_{MAX}$ **and** *time* $< T_{MAX}$ **do**
  for all $P \in \{P_1, \ldots, P_{n_{sp}}\}$ **do**
    Select parent solutions $S1$ and $S2$ from P (binary tournament)
    Apply Algorithm 2 to $S1$ and $S2$
  end for
  if the variable criterion is satisfied **then**
    Randomly select P from $\{P_1, \ldots, P_{n_{sp}}\}$
    Select parent solutions $S1$ from P and $S2$ from PI (binary tournament)
    Apply Algorithm 2 to $S1$ and $S2$
  end if
  for all $P \in \{P_1, \ldots, P_{n_{sp}}\} \cup PI$ **do**
    if $|P| = M_{MAX}$ **then**
      select survivors
    end if
  end for
  if *number of iterations* $= I_{SHARE}$ **then**
    share the best solutions within feasible subpopulations
  end if
  for all $P \in \{P_1, \ldots, P_{n_{sp}}\}$ **do**
    if *best solution in* P *not improved for* I_{NMAX} iterations **then**
      diversify P
    end if
  end for
end while
return best feasible solution

Algorithm 2 EVOLVE

Require: Parent solutions $S1$ and $S2$, feasible P and infeasible PI subpopulations
Generate offsprings $C1$ and $C2$ from $S1$ and $S2$ (*crossover*)
Educate offsprings $C1$ and $C2$ with probability p_{ls} (*local search procedure*)
if $C1$ ($C2$) is feasible **then**
    insert $C1$ ($C2$) into feasible subpopulation P
end if
if $C1$ ($C2$) is infeasible **then**
    insert $C1$ ($C2$) into infeasible subpopulation PI
end if
Randomly select $C1$ or $C2$ and create new solution $C3$ (*mutation*)
Improve $C3$ with probability p_{ls} (*local search procedure*)
if $C3$ is feasible **then**
    insert $C3$ into feasible subpopulation P
end if
if $C3$ is infeasible **then**
    insert $C3$ into infeasible subpopulation PI
end if

corresponding depot index. Zeros are used to terminate a route and start a new one.

(2) The second chromosome includes three tiers, with a fixed length of n_c entries. This chromosome represents the following three aspects: (i) satellites and pickup facilities used in the solution (first two tiers); (ii) customer service options (second and third tiers); and (iii) assignments of the customers to the opened satellites (first and third tiers). The third tier consists of the list of customers. In the second tier, entries represent the selected pickup facilities and customers' delivery options. More precisely, if a customer is assigned to a pickup facility (CP option), the index of the pickup facility is reported in the corresponding position of the chromosome; otherwise, the index value is set equal to zero.

(3) The third chromosome represents second-level routes and consists of a permutation of satellites, customers with the HD option, and pickup facilities. Routes belonging to the same satellite are grouped in the chromosome by the corresponding satellite index. Zeros are used to terminate a route and start a new one.

Figure 2 provides the representation of the solution of the instance with $n_d = 2$, $n_s = 5$, $n_p = 4$, and $n_c = 19$ given in Figure 1.

3.2 Greedy algorithm for generating initial population

To generate the initial population of feasible solutions, we used a Three-Phase Heuristic algorithm, called TPH, which works as follows. In the first phase (*Initial second-level routes*), an initial set of second-level routes is generated; the set is locally optimized

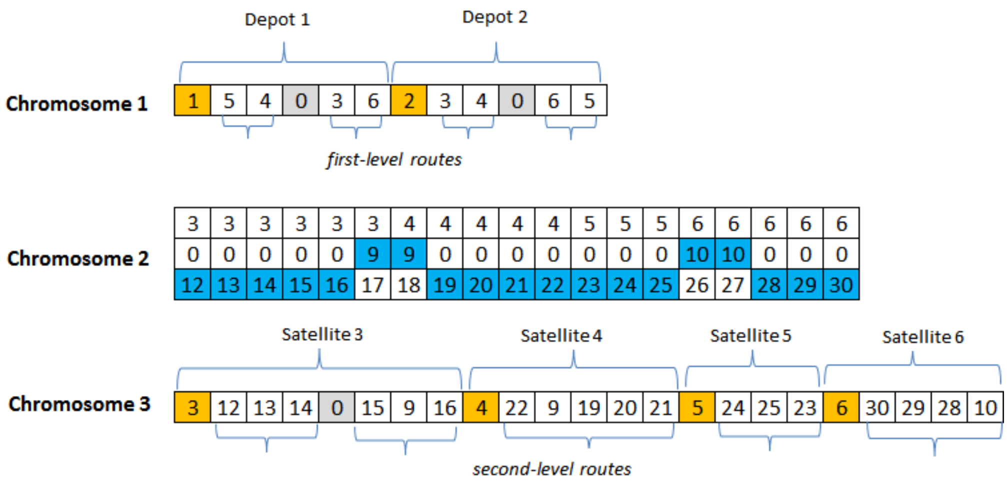


Figure 2: Chromosome encoding for the solution of Figure 1

during the second phase (*second-level route optimization*) by considering delivery options. In the third and last phase (*first-level routes*), first-level routes are built based on the second-level routes generated. Below, the three phases are described in more detail.

Initial second-level routes. This phase is based on the greedy algorithm proposed by Yu and Lin (2015) for building second-level routes without considering delivery options. Below are the main steps of the algorithm.

Step 1 Compute the number $s(k)$, $k \in N_S$, of unassigned customers whose closest satellite is k. Choose the satellite k^* with the biggest $s(k)$ value and let UC be the set of unassigned customers;

Step 2 Assign customers in UC to satellite k^* in increasing order of distance between the customers and the satellite. Stop when assigning the next customer to the satellite violates the satellite's capacity constraint;

Step 3 Construct a giant TSP tour using the algorithm proposed by Lin and Kernighan (1973);

Step 4 Construct second-level routes for satellite k^* by splitting the TSP route generated in Step 3 into several routes using the method proposed by Prins (2004).

Step 5 If not all customers have been assigned, return to Step 1.

second-level route optimization. This phase locally optimizes each initial second-level route by considering CP services. The procedure is based on a similar procedure proposed by Baldacci and Dell'Amico (2010) for the m-Ring-Star problem. The procedure receives a single feasible second-level route as input and tries to re-optimize the routing by allowing changes in the customer delivery options. The reader is referred to Baldacci and Dell'Amico (2010) for details about the procedure.

first-level routes. Similar to the first phase, first-level routes are generated, starting from the set of second-level routes, as follows. The satellites in the solution are first assigned to the different depots. Then, for each depot, a giant TSP tour is built using the algorithm proposed by Lin and Kernighan (1973). First-level routes for each depot are then obtained by splitting each TSP tour, using the method proposed by Prins (2004).

3.3 Evaluation and selection of individuals

Because HMPG considers both feasible and infeasible solutions, we use the following fitness function $F_P(S)$, where S is a generic individual, to penalize constraint violations:

$$\begin{aligned} F_P(S) = f(S) + \alpha_1 \cdot P_{caS}(S) + \alpha_2 \cdot P_{nvS}(S) + \alpha_3 \cdot P_{vwt1}(S) \\ + \alpha_4 \cdot P_{vca1}(S) + \alpha_5 \cdot P_{vwt2}(S) + \alpha_6 \cdot P_{vca2}(S). \end{aligned} \tag{1}$$

In the function, $f(S)$ is the objective value of solution S, $P_{caS}(S)$ is the satellite capacity violation, $P_{nvS}(S)$ is the total vehicle number violation, $P_{vwt1}(S)$ and $P_{vca1}(S)$ are the working time and capacity violations for vehicles of the first-level routes, and $P_{vwt2}(S)$ and $P_{vca2}(S)$ are working time and capacity violations for vehicles of the second-level routes. The corresponding penalty factors are α_1, α_2, α_3, α_4, α_5 and α_6.

For population evaluation, we adopted the biased fitness measures proposed by Vidal et al. (2012), who proposed a mechanism that addresses both the value of the fitness function and the diversity of the population. Therefore, the evaluation function accounts for the cost of an individual and its contribution to the populations diversity.

In our case, the diversity contribution $\nabla(S)$ is introduced, which provides the average distance of an individual S from its n_{close} closest neighbors, grouped in set N_{close}, computed according to the following expression

$$\nabla(S) = \frac{1}{n_{close}} \sum_{S_2 \subseteq N_{close}} \delta^H(S, S_2). \tag{2}$$

In the expression above, function $\delta^H(S_1, S_2)$ is the Hamming distance between individuals S_1 and S_2. The function is based on the differences between the used satellites, the used pickup facilities, and vehicle routes of two individuals S_1 and S_2, and is computed as

$$\begin{aligned} \delta^H(S_1, S_2) = \beta_1 \frac{1}{n_s} \sum_{i=1}^{n_s} 1((\phi_i(S_1) \neq \phi_i(S_2))) + \beta_2 \frac{1}{n_p} \sum_{j=1}^{n_p} 1((\varphi_j(S_1) \neq \varphi_j(S_2))) + \\ \beta_3 \frac{1}{n_{arc1}} \sum_{k=1}^{n_{arc1}} 1((\psi_{1k}(S_1) \neq \psi_{1k}(S_2))) + \beta_4 \frac{1}{n_{arc2}} \sum_{m=1}^{n_{arc2}} 1((\psi_{2m}(S_1) \neq \psi_{2m}(S_2))), \end{aligned} \tag{3}$$

where $1(cond)$ is a valuation function that returns one if the condition *cond* is true, zero otherwise. Coefficients β_1, β_2, β_3 and β_4 are such that $\beta_1 + \beta_2 + \beta_3 + \beta_4 = 1$; $\phi_i(S)$

and $\varphi_j(S)$ indicate that satellite i and pickup facility j are open or used, $\psi_{1k}(S)$, and $\psi_{2m}(S)$ denote the arcs in the first-level and second-level routes; n_c stands for the number of customers, and n_{arc1} and n_{arc2} are the number of arcs of the two individuals of the first-level and second-level routes, respectively.

Let $fit(S)$ be the rank of an individual S in a subpopulation of size n_{close} with penalized cost $F_P(S)$; the fitness function $F_E(S)$ is computed as follows:

$$F_E(S) = fit(S) - \nabla(S). \tag{4}$$

3.4 Parent selection and crossover

To diversify the search, a crossover operator recombines the gene-codes of two parents and produces new offsprings. As satellites and pickup facilities vary in solutions, and each of them can be visited more than once, we propose a two-tier mapping 2-point crossover (TM2PX) operator.

Parent selection is performed through a binary tournament, which twice randomly (with uniform probability) picks two individuals from the selected population and keeps the one with the best fitness function.

The TM2PX is performed by exchanging the middle parts of the second tier of chromosome two (customers' delivery options) of the two selected parents between two determined crossover positions. Once these changes are made in chromosome two, chromosome three is modified accordingly. The use of TM2PX creates new solutions exploring new pickup facilities, customers' delivery options, and routes.

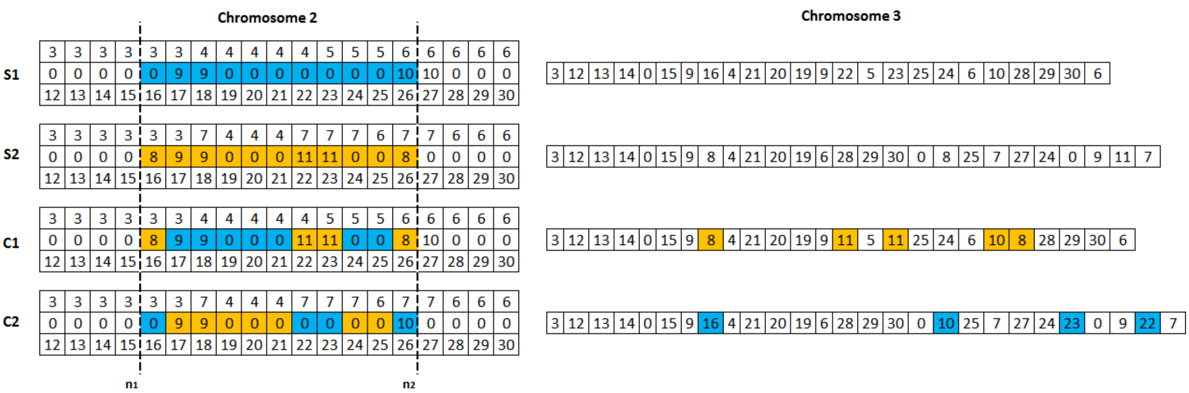


Figure 3: Two-tier mapping 2-point crossover

Figure 3 shows an example of a crossover operation between individuals $S1$ and $S2$, with crossover positions n_1 and n_2, to generate offsprings $C1$ and $C2$. In the figure, the different colors highlight the changes in the chromosomes.

The pseudo-code for replacing the middle part of parent individual $S2$ by the corresponding part of parent individual $S1$ to generate offspring $C2$ is reported in algorithm 3.

In the algorithm, $V2(.)$ and $V3(.)$ represent second and third chromosomes, and the genes in position n of the second and third tiers of $V2(.)$ are $g_{2n}(.)$ and $g_{3n}(.)$, respectively. The number of customers served by pickup facility p is n_p; the index of the satellite that serves pickup facility p is s_p.

3.5 Mutation operation

Mutation is performed by varying the genes of a chromosome to prevent the algorithm from becoming trapped in local optima, thus improving the diversity of solutions. In this paper, we use a gene-based multi-dimension mutation (MDM) operator. The mutation, performed on chromosome three of an individual, begins by defining a mutation position $n \in [0, L_S]$, where L_S is the length of the genes in the chromosome. Let g_n be the gene selected. The following cases are considered.

Case 1: $g_n \in N_S$. One of the following operators is chosen randomly (with uniform probability). Replace g_n by a new one, close g_n, open a new satellite and exchange g_n with the new satellite.

Case 2: $g_n \in N_P$. Replace g_n by the closest unused pickup facility.

Case 3: $g_n \in N_C$. If g_n is associated with the CP option, release CP and let it be served by HD; otherwise, let g_n be served by the closest pickup facility.

Case 4: $g_n = 0$. Randomly insert g_n between 1 and L_S in chromosome three.

3.6 Local search procedure

LS is a metaheuristic which performs well at generating an improved local solution by exploring the neighborhoods. One of the main criteria in designing LS is the choice of the neighborhood structure. To achieve better local optima, two types of neighborhood structures are implemented. More specifically, we extended the routing neighborhood structure proposed by Prins (2004) to address two-echelon routing, and we introduced neighborhood structures to modify the customers delivery options.

The two types of neighborhood structures are examined in random order (using a uniform distribution) and a neighborhood structure is terminated at the first improving move. Within each neighborhood structure, the moves are executed in sequence as described below. The LS procedure terminates whenever all the moves are applied without any improvement.

3.6.1 Routing improvement

Let $T(u)$ be the route visiting node u, and let (u, v) identify the partial route from u to v. Define the neighborhood of vertex u as the $h_1 n_c$ closest vertices, where $h_1 \in [0, 1]$ is a *granular threshold* restricting the search to nearby vertices (Toth and Vigo, 2003).

Algorithm 3 TM2PX

Require: Two parent individuals $S1$ and $S2$
Ensure: Modify chromosomes $V2(S2)$ and $V3(S2)$ from $S1$
Generate two random integer numbers $n_1 < n_2$ in $(0, n_c]$
for $(n = n_1, \cdots, n_2)$ **do**
$p_1 \leftarrow g_{2n}(S1)$, $p_2 \leftarrow g_{2n}(S2)$, $i \leftarrow g_{3n}(S1)$; define n_{p_1}, n_{p_2}, s_{p_1} and s_{p_2}
Replace p_2 in $V2(S2)$ by p_1
if $p_1 = 0$ **then**
if $p_2 \neq 0$ **then**
if $n_{p_2} = 1$ **then**
Replace p_2 by i in $V3(S2)$
else
Insert i after p_2 in $V3(S2)$
end if
end if
else
if $p_2 = 0$ **then**
if p_1 is already visited by s_{p_2} **then**
Delete i in $V3(S2)$
else
Replace i by p_1 in $V3(S2)$
end if
else
if $p_1 \neq p_2$ **then**
if p_1 is already visited by s_{p_2} **then**
if $n_{p_2} = 1$ **then**
Delete p_2 in $V3(S2)$
end if
else
if $n_{p_2} = 1$ **then**
Replace p_2 by p_1 in $V3(S2)$
else
Insert p_1 after p_2 in $V3(S2)$
end if
end if
end if
end if
end if
end for

Further, let x and y be the successors of u in $T(u)$ and v in $T(v)$, respectively. The following moves are considered.

(RM1) If u is a customer or pickup facility node, remove u and insert it after v;

(RM2) If u is a customer or pickup facility and x is a customer or pickup facility or dummy zero node, remove them, then insert (u, x) after v;

(RM3) If u is a customer or pickup facility and x is a customer or pickup facility or dummy zero node, remove them, then insert (x, u) after v;

(RM4) If u and v are customer or pickup facility nodes, swap u and v;

(RM5) If u and v are customer or pickup facility nodes, and x is a customer or pickup facility or dummy zero node, swap (u, x) and v;

(RM6) If u and v are customer or pickup facility nodes, and x and y are customer or pickup facility or dummy zero nodes , swap (u, x) and (v, y);

(RM7) If $T(u) = T(v)$, replace (u, x) and (v, y) by (u, v) and (x, y);

(RM8) If $T(u) \neq T(v)$, replace (u, x) and (v, y) by (u, v) and (x, y);

(RM9) If $T(u) \neq T(v)$, replace (u, x) and (v, y) by (u, y) and (x, v).

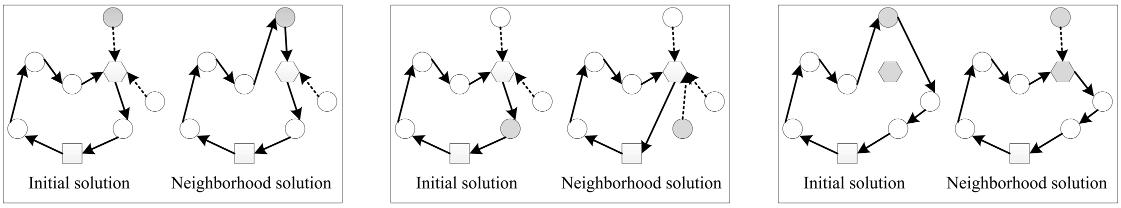
3.6.2 Delivery option improvement

The following three moves have been implemented.

(DM1) *Remove a CP service.* For a selected pickup facility, a customer with initial CP service is removed from the facility and inserted into a new position of minimum extra cost (see Figure 4-(a)). The removed customer can be inserted along a route or served by another pickup facility. This move is applied to all the open pickup facilities and the customers they serve.

(DM2) *Provide a CP service.* This move is conducted by providing a new CP service for the customers close to a pickup facility. More precisely, for an open pickup facility p, customers with the HD option or served by other pickup facilities are assigned to p (see Figure 4-(b)). This move is performed on the $h_2 n_c$ closest customers, with a *granularity threshold* $h_2 \in [0, 1]$.

(DM3) *Open a new pickup facility.* This move aims to improve the solution by opening a new pickup facility to provide CP service(see Figure 4-(c)).



(a) DM1 (b) DM2 (c) DM3

Figure 4: Delivery option moves

3.7 Population management

3.7.1 Population initialization

To obtain a high-quality feasible population initially, we use a *diverse population generation algorithm* (DPGA) based on algorithm TPH described in Section 3.2. Using DPGA ensures that the initial solutions vary in terms of open satellites and pickup facilities as well as routes. As DPGA iteratively executes TPH it randomly (with uniform probability) generates a subset of the satellite set, to be used during Step 1 of TPH.

3.7.2 Multi-population evolution strategy

The metaheuristic literature indicates that allowing a controlled exploration of infeasible solutions may enhance the performance of the search (Ahmadizar et al., 2015), and the GA-based heuristic proposed by Vidal et al. (2012) showed the validity of considering the infeasible subpopulation.

The multi-population evolution strategy (MPES) considered in this paper is composed of n_{sp} feasible subpopulations plus one infeasible subpopulation, which are independently managed to contain between μ and $\mu + \lambda$ individuals, where μ represents the minimum subpopulation size and λ the generation size. The infeasible subpopulation is initially set to be empty, and solutions are obtained during the evolution process. Each incoming individual is directly put into the appropriate subpopulation according to its feasibility. Once each subpopulation reaches the maximum size $\mu + \lambda$, a survivor selection will be conducted to discard λ individuals. The current best solutions are shared between feasible subpopulations every I_{MSHARE} iterations. The scheme of the proposed evolution strategy is depicted in Figure 5.

The crossover between feasible and infeasible subpopulations is applied based on the following *variable criterion mechanism*. Let n_{fe} and n_{infe} be the numbers of the current feasible and infeasible individuals, respectively; κ_1 and κ_2 are decision coefficients, and p_n is a random number between $[0, n_{fe} + n_{infe}]$. Whenever the condition $\kappa_1 \cdot n_{fe} \le p_n < n_{fe} + \kappa_2 \cdot n_{infe}$ is satisfied, the crossover operator is applied.

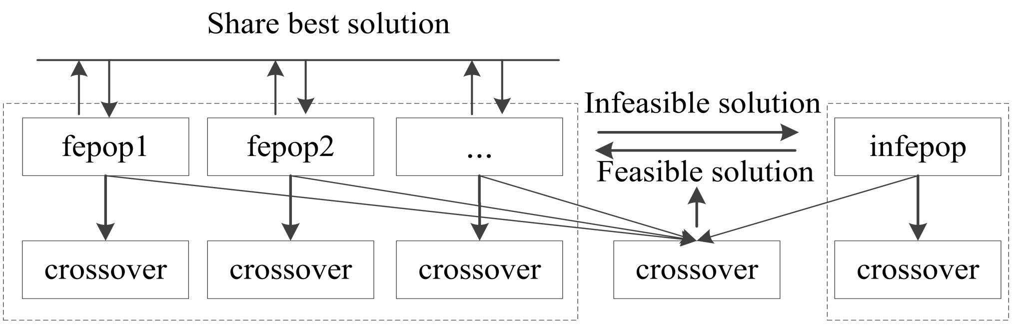


Figure 5: Multi-population evolution

3.7.3 Population diversification

To diversify the search, a *Routing global perturbation mechanism* (RGPM) is implemented whenever the best solution does not improve after I_{NMAX} iterations. The mechanism eliminates all but the best $\mu/2$ individuals, and $\mu/2$ new individuals are generated using the RM1-RM6 moves without considering the *granular threshold*.

3.7.4 Stopping criterion

To comprehensively address the behavior of the HMPG, the termination condition combines the T_{MAX} and I_{MAX} parameters. Depending on the instance dimension, three different stopping criterions are used: (I_{MAX}, T_{MAX})-(10^4, 10 min), (2.10^4, 15 min), and (3.10^4, 20min); the parameter I_{NMAX} is set equal to 400, 600 and 800, respectively.

4 Computational experiments

This section reports the computational experiments that were conducted to evaluate the proposed algorithm based on both real-world and randomly generated instances. The algorithm was coded in C language, and compiled with Visual Studio 2013. All experiments were conducted on an Intel 4790, 3.60 GHz (8 Core) processor with 32 GB of memory running under the Windows 7 operating system.

4.1 Parameter setting

Metaheuristic algorithms generally rely on a set of correlated parameters and configuration choices for their key operators (Vidal et al., 2012). To test the impact of the parameters on HMPG and decide which were suitable, a set of representative instances was selected from the randomly generated instances described in Section 4.3.

The algorithm was run for each instance, and every parameter was evaluated by holding other parameters constant. The validity of using an infeasible subpopulation was first tested, followed by deciding on the number of feasible subpopulations. Table 1 provides a summary of the parameters considered, together with the range of values we estimated to be appropriate. In particular, to decide the initial feasible population size, we observed that too small a size cannot sample a sufficient amount of the search space, thus exhibiting poor performance. However, too large a population size may lead to long computing time. The results showed that the best population size is related to the number of candidate satellites. Hence, variable population size was adopted; that is, the initial population size was selected according to the number of satellites. We found that the initial population size $n_{fe} = 4n_s$ is ideal, simultaneously taking both the solution space and population evolution efficiency into account. The number of feasible subpopulations n_{sp} was set equal to 2.

Table 1: Parameter setting

Parameter	Range	Final parameter value
n_{fe}	[1,200]	$n_{fe} = 4n_s$ or $n_{fe} = n_{fe}(n_{fe} < 4n_s)$
n_{sp}	[1,4]	2
μ	[1,200]	n_{fe}/n_{sp}
λ	[1,200]	μ
h_1	[0,1]	0.2
h_2	[0,1]	0.1
α	[0,1]	0.6
κ_1, κ_2	[0,1]	$\kappa_1 = 0.55$, $\kappa_1 = 0.45$
α_1, α_2, α_3, α_4, α_5, α_6	[0,1000]	$\alpha_1 = \alpha_2 = 1000$, $\alpha_3 = \alpha_4 = \alpha_5 = \alpha_6 = 10$
β_1, β_2, β_3, β_4	[0,1]	$\beta_1 = 0.4$, $\beta_2 = 0.3$, $\beta_3 = 0.15$, $\beta_4 = 0.15$
p_{ls}	[0,1]	0.5
I_{MSHARE}	[0,100]	40

4.2 Real-world instance

We first applied our approach to a real-world instance, designing a joint city distribution system for the last mile distribution for online shopping. To deal with the challenges caused by increasing delivery cost and personalized services, two logistics operators in Chongqing city formed an alliance to provide a last mile distribution service. Each operator has its own customers, served through a set of satellites by a fleet of vehicles. A joint distribution system providing HD and CP services simultaneously is required; the packages are transported by first-level routes from the depots of the two logistics operators to the shared set of satellites, and then delivered by second-level routes to the final customers using shared pickup facilities. The instance is based in Shangpingba District, and the instance data have been provided by the two logistics operators. In this instance, two depots, 12 satellites, 40 pickup facilities, and a total of 164 customers are included. Figure 6 shows the layout of the instance.

The data of the instance were defined by the two logistics operators as follows: $t_{ij} = d_{ij}/speed$, $r_{ij} = d_{ij}u_d$, $c_{ij} = d_{ij}ucp$, where (i) d_{ij} is the distance between two nodes, which is approximately 1.2 times the straight-line distance; (ii) the speeds for vehicles traveling in the first and second levels are 40 km/h and 30 km/h, respectively; (iii) $ucp = 1.5$ is the unit cost coefficient associated with CP service; and (iv) u_d is the unit distance traveling

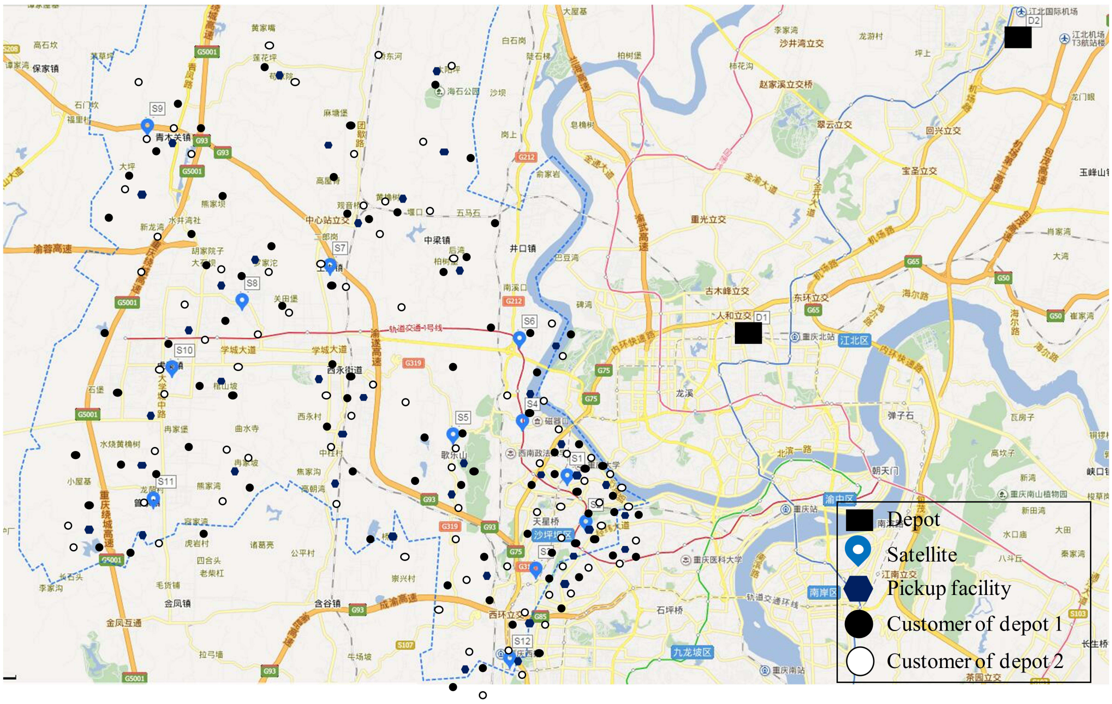


Figure 6: Map layout of the real-world instance

cost, equal to 1.5 for the first-level vehicles and 1.1 for the second-level vehicles. Other parameters are: $s_i = 1.2$, $v_i = 0.2$, $L_d^1 = L_k^2 = 180$ minutes, $Q_d^1 = 2000$, and $Q_k^2 = 500$. Information about the depots and satellites is reported in Table 2.

Table 2: Depots and satellites information

No.	Type	m_d^1/m_k^2	u_k	H_k	B_k	x-cor	y-cor	No.	Type	m_d^1/m_k^2	u_k	H_k	B_k	x-cor	y-cor
1	D	3	-	-	-	37.1	23.24	8	S	7	0.45	0.032	600	18.9	12.32
2	D	3	-	-	-	27.02	12.6	9	S	8	0.45	0.032	700	18.9	12.32
3	S	10	0.4	0.03	1000	20.72	7.28	10	S	5	0.3	0.034	800	8.68	13.58
4	S	4	0.45	0.035	600	21.28	5.6	11	S	6	0.3	0.034	600	5.32	20.72
5	S	8	0.35	0.032	800	19.46	3.78	12	S	4	0.35	0.032	500	6.16	11.2
6	S	5	0.5	0.034	600	19.04	9.24	13	S	4	0.35	0.038	600	5.46	6.3
7	S	5	0.45	0.036	600	16.52	8.82	14	S	6	0.4	0.04	500	18.62	0.7

It is worth mentioning that for online shopping, customers locations present spatial aggregation features. More precisely, several customers associated with a specific area, such as a workshop or a shopping mall, may be grouped into a single customer. In this case, the data associated with the aggregated customer are computed using a preprocessing step that takes into account the data of the different customers that have been aggregated together.

4.2.1 Results obtained

In Table 3, we report the results obtained by the heuristic algorithm for four scenarios: (i) joint distribution with delivery options (JD-DO), (ii) joint distribution without delivery options (JD), (iii) independent distribution with delivery options (D1-DO+D2-DO), and (iv) independent distribution without delivery options (D1+D2).

Table 3 and Figures 7-(a)-7-(f) report the following details: total solution cost (tc), first-level cost ($1^{st}c$), second-level cost ($2^{nd}c$), connection cost (cc), number of selected satellites (nss), number of selected pickup facilities (nsp), number of first-level routes ($1^{st}rs$), number of second-level routes ($2^{nd}rs$) and number of connections ($ncon$).

In Table 3, we can see that joint distribution has the advantage of reducing the cost by comparing JD-DO and D1-DO+D2-DO, as well as JD and D1+D2, with the percentages of reductions equal to 7.1 and 7.6, respectively. Moreover, allowing delivery options results in a cost reduction of about 9%. A comparison of JD-DO and JD in the table illustrates one way that delivery options help reduce cost. When CP service is provided, the first-level cost remains stable, while the number of vehicles at the second-level is greatly reduced (from 37 to 20); the increased connection cost (2506.6) is more than offset by the reduced vehicle cost.

Table 3: Results of the real-world instance

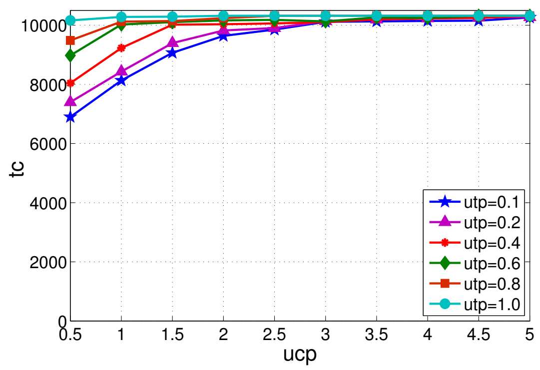
Item	JD-DO	JD	D1-DO+D2-DO	D1+D2
tc	9395.0	10317.1	10113.5 (5452.5+4661.0)	11166.8 (6127.3+5039.5)
$1^{st}c$	2633.0	2689.8	2649.1 (1589.8+1059.3)	2707.6 (1595.1+1112.5)
$2^{nd}c$	6762.0	7627.3	7464.4 (3862.7+3601.7)	8459.2 (4532.2+3927.0)
cc	2506.6	0	1089.7 (676.3+413.4)	0
nss	6	6	6 (3+3)	7 (4+3)
nsp	27	0	18 (11+7)	0
$1^{st}rs$	3	3	3 (2+1)	3 (2+1)
$2^{nd}rs$	20	37	30 (15+15)	39 (21+18)
$ncon$	81	0	31 (20+11)	0

4.2.2 Sensitivity analysis of delivery options

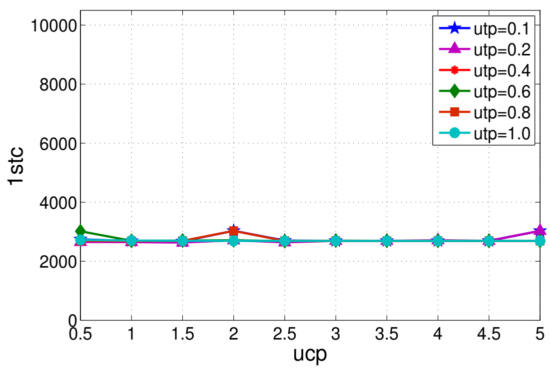
In this section, we consider the impact of delivery options on the solutions by performing sensitivity analyses on CP parameters. The experiments were implemented by varying ucp (the unit cost coefficient associated with CP service, see Section 4.2) from 0.5 to 5 with steps of 0.5, and values v_i (the service time coefficient associated with pickup facility $i \in N_P$), denoted by utp in the figures, from 0.1 to 1 with steps of 0.2. The results are presented in Figure 7.

The results show that when ucp ranges from 0.5 to about 2.5 and utp ranges from 0.1 to about 0.6, the total costs tc increase consistently. Moreover, when ucp ranges from 2.5 to 5.0, tc remains unchanged; as shown by $ncon$ values, these are scenarios with no CP services. The first-level cost $1^{st}c$ is marginally affected by the two parameters, whereas the second-level cost $2^{nd}c$ increases similarly to cost tc.

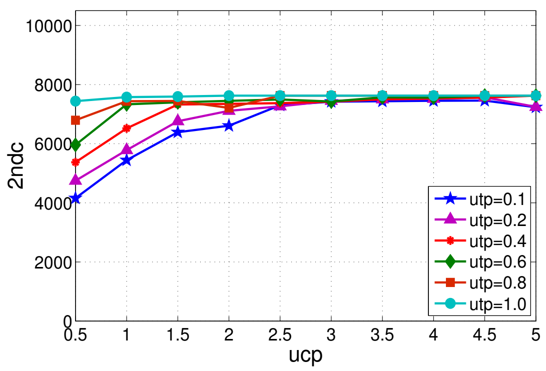
The figures show how CP parameters affect the results: the number of open pickup facilities (nsp) and connections ($ncon$) are related to ucp and utp (see subfigures (e) and (f) in Figure 7). With small ucp and utp values, more pickup facilities are open; as the



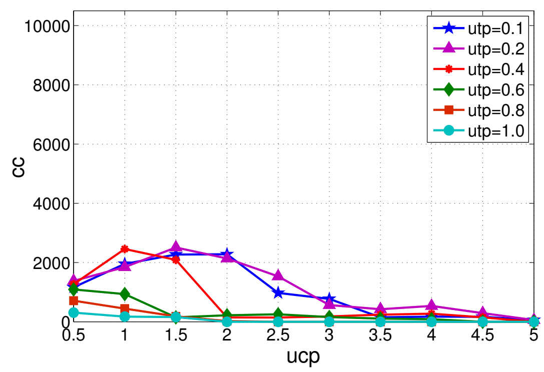
(a) tc



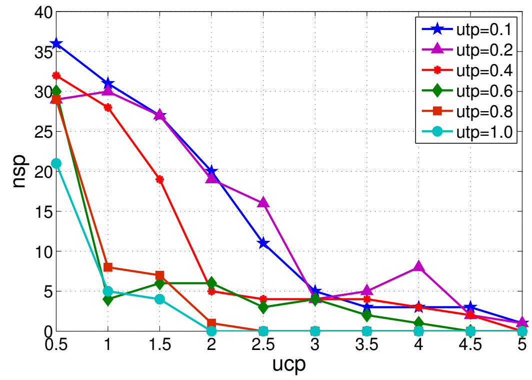
(b) $1^{st}c$



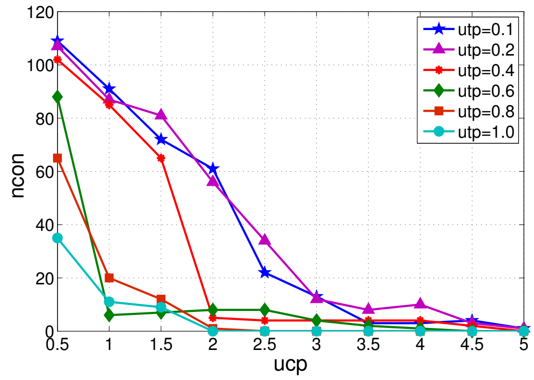
(c) $2^{nd}c$



(d) cc



(e) nsp



(f) $ncon$

Figure 7: Sensitivity analyses of the CP parameters

values increase, the numbers of open pickup facilities and connections decrease sharply.

The figures also show that for almost all scenarios, with the increasing of ucp and utp values, the number of connections $ncon$ decreases, while the connection cost cc first increases and then decreases. This indicates that a trade-off exists among $ncon$, ucp and utp.

4.3 Randomly generated instances

To our knowledge, the MD-TEVRP-DO has never been addressed in the literature. Therefore, we generated a set of MD-TEVRP-DO instances based on the instance generator for the 2E-LRP proposed by Crainic et al. (2011b). The aim of the instance generator is to create a schematic representation of a multi-level urban area. Customers and facilities are located within concentric circle of increasing radius (see Figure 8). The locations of depots, satellites, pickup facilities, and customers are generated in the appropriate urban areas according to the following criteria: depots are randomly located within Area 1; satellites are randomly located within Area 2 or Area 3, according to a parameter β; $\beta\%$ of the total number of satellites are within Area 2 and $(1 - \beta\%)$ within Area 3; pickup facilities are randomly generated within Area 3; customers are randomly generated according to the generated pickup facilities specifically, customers are generated within Area 4. We denote with R_1, R_2, R_3 and R_4 the radii of Areas 1, 2, 3 and 4, respectively.

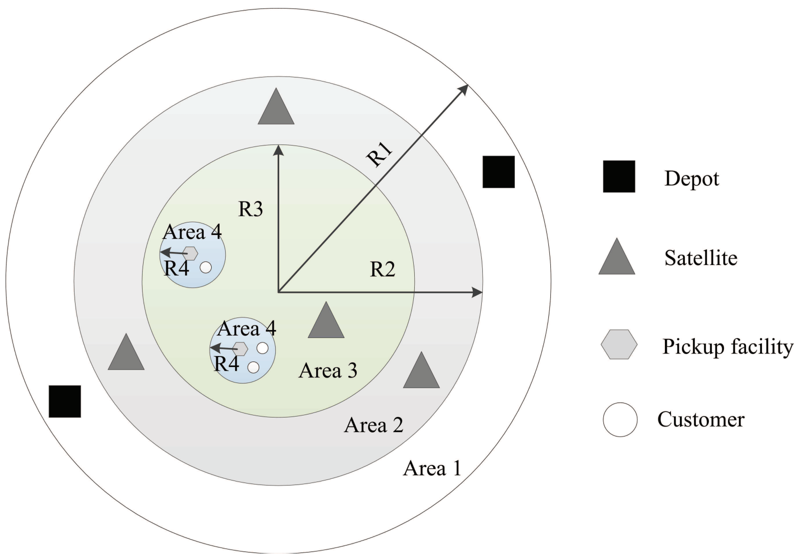


Figure 8: Schematic representation of a multi-level urban area used to generate the instances

To ensure the validity of the generated instances, we take *coordinate granularity* (defined as the minimum coordinate difference between two generated nodes) into consideration. The granularities of depots, satellites, pickup facilities, and customers are λ_1, λ_2, λ_3, and λ_4, respectively. When locations are generated, *coordinate granularity* should be satisfied. The values of the parameters used for generating the instances are reported in Table 4. The other parameters have been set equal to the ones used for the real-world instance. The following naming scheme is adopted for the different instances: I<depots>-<satellites>-<pickup facilities>-<customers>. Our instances are publicly available at http://www.vrp-rep.org (VRP-REP:2017-0026, Mendoza et al. (2014)).

Table 4: Randomly generated instances

Parameter	Range of value
β	0.5
$R1$	30 km
$R2$	20 km
$R3$	15 km
$R4$	1 km
λ_1	2 km
λ_2	1 km
λ_3	0.3 km
λ_4	0.02 km
B_k	$B_k \in [400, 500, 600, 700, 800]$
u_k	$u_k \in [0.05, 0.06, 0.08, 0.09, 0.1]$
H_k	$H_k \in [0.1, 0.15, 0.2, 0.25, 0.3]$
U_d^1	$U_d^1 \in [280, 300, 320]$
U_k^2	$U_k^2 \in [190, 200, 210, 220, 230]$
q_i	$q_i \in [5, 60]$
m_d^1	$\lceil 1.5 \cdot \sum_{i \in N_C} q_i / Q_d^1 \rceil, o_i = d$
m_k^2	$m_k^2 \in [4, 5, 6, 7, 8]$
m^2	$0.8 \cdot \sum_{k \in N_S} m_k^2$

4.3.1 Analysis of algorithmic components

To analyze the impact on the performance of the various algorithmic components of the proposed metaheuristic, experiments were performed on HMPG algorithm by removing each of them in turn.

The "No-LS" version was obtained by running the algorithm without LS (see Section 3.6) in steps 2 and 10 of Algorithm 2. In the "No-MPES" version, only one population was considered, and infeasible solutions are mixed with feasible solutions instead of taking infeasible solutions into account independently. For the "No-TPH" version, we did not consider the second phase of the algorithm TPH (see Section 3.2) and we di not applied the LS procedure to the initial population (see step 1 of Algorithm 1). Moreover, we also disabled crossover operation TM2PX ("No-TM2PX" version), mutation operation ("No-MUM" version), and global perturbation mechanism RGPM ("No-RGPM" version).

In the following experiments, all the algorithm versions were run ten times for each instance. The results are reported in Tables 5-10. The running times are in minutes and the numbers in boldface indicate the best solutions found.

In addition to the notation already introduced, Tables 5-10 show the following columns: average percentage deviation of the best solution cost ($\%B$), average percentage deviation of the solution cost ($\%A$) and average percentage deviation of the running time ($\%T$). Moreover, Tables 6, 8 and 10 show the following details about algorithm HMPG: best solution cost (B), average solution cost A) and average running time (T).

From the results we can see that that each algorithm component plays an important role in the good performance of the HMPG algorithm. The most crucial component is TPH which helps to improve the solutions by an average of 20%, followed (to a lesser

Table 5: Sensitivity analysis on the HMPG components for the instances with 1 depot

Name	No-LS			No-MPES			No-TPH		
	%B	%A	%T	%B	%A	%T	%B	%A	%T
I1-4-10-50	**0.0**	+0.1	-24.3	+0.2	+1.3	-51.2	+22.9	+21.0	0.0
I1-4-10-100	+1.0	+3.5	-19.3	+1.6	+2.3	-51.2	+22.9	+21.0	0.0
I1-4-20-50	+0.5	+2.6	-26.3	+2.3	+4.6	-59.6	+22.1	+25.7	0.0
I1-4-20-100	**-0.6**	+0.9	-27.9	+2.4	+2.9	0.0	+12.5	+16.3	0.0
I1-8-10-100	+0.4	+4.8	-29.7	+8.1	+10	0.0	+14.1	+15.8	0.0
I1-8-10-150	+4.3	+4.9	0.0	+4.1	+5.6	0.0	+9.1	+12.5	0.0
I1-8-20-100	+5.1	+2.3	0.0	+6.3	+5.9	0.0	+29.8	+24.2	0.0
I1-8-20-150	**-0.8**	+1.1	0.0	+1.0	+2.1	0.0	+13.2	+13.5	0.0
I1-12-20-150	+0.6	+1.2	0.0	+0.4	+5.3	0.0	+16.3	+16.8	0.0
I1-12-20-200	+0.5	+1.3	0.0	+4.3	+5.9	0.0	+9.4	+9.2	0.0
I1-12-30-150	+2.1	+2.2	0.0	+0.8	+4.7	0.0	+18.9	+20.7	0.0
I1-12-30-200	+0.8	+2.7	0.0	+5.5	+6.4	0.0	+19.2	+19.2	0.0

Table 6: Sensitivity analysis on the HMPG components for the instances with 1 depot

Name	No-TM2PX			No-MUM			No-RGPM			HMPG		
	%B	%A	%T	%B	%A	%T	%B	%A	%T	B	A	T
I1-4-10-50	+2.0	+3.3	-51.7	+1.1	+2.1	-16.0	**0.0**	+6.5	0.0	**3074.7**	**3177.1**	10.0
I1-4-10-100	+5.6	+5.8	-20.8	+4.5	+4.7	-10.0	+5.6	+5.4	0.0	**5716.8**	**5726.1**	10.0
I1-4-20-50	+2.0	+3.2	-41.9	+2.3	+4.1	-10.7	+1.3	+3.4	0.0	**3164.0**	**3164.3**	10.0
I1-4-20-100	+9.1	+10.0	-31.0	+8.0	+9.8	-10.0	+10.8	+11.4	0.0	6314.6	**6336.9**	10.0
I1-8-10-100	+4.6	+7.2	-26.7	+5.8	+12.6	0.0	+1.9	+5.4	0.0	**5747.4**	**5757.4**	15.0
I1-8-10-150	+2.8	+3.8	0.0	+2.6	+2.4	0.0	+3.0	+4.5	0.0	**8630.7**	**8699.9**	15.0
I1-8-20-100	+5.3	+1.6	0.0	+8.7	+4.8	0.0	+7.1	+5.3	0.0	**5682.4**	**5939.9**	15.0
I1-8-20-150	+1.0	+1.0	0.0	+0.8	+1.9	0.0	+1.2	+1.8	0.0	9823.3	**9835.7**	15.0
I1-12-20-150	+2.2	+5.7	0.0	+2.2	+1.8	0.0	+2.3	+1.9	0.0	**8201.4**	**8238.9**	20.0
I1-12-20-200	+1.5	+1.3	0.0	+1.5	+2.2	0.0	+1.8	+2.3	0.0	**11811.1**	**11843.3**	20.0
I1-12-30-150	+1.5	+2.0	0.0	+2.3	+2.5	0.0	+2.7	+2.9	0.0	**8564.2**	**8571.8**	20.0
I1-12-30-200	+2.0	+1.9	0.0	+2.5	+1.9	0.0	+2.9	+2.5	0.0	**11087.5**	**11175.3**	20.0

Table 7: Sensitivity analysis on the HMPG components for the instances with 2 depots

Name	No-LS			No-MPES			No-TPH		
	%B	%A	%T	%B	%A	%T	%B	%A	%T
I2-4-10-50	+0.3	+0.9	-47.9	**0.0**	+1.6	-74.7	+11.9	+14.0	0.0
I2-4-10-100	+0.9	+2.9	-40.0	+6.8	+6.4	-40.0	+13.6	+14.3	0.0
I2-4-20-50	+1.3	+2.1	-40.0	+3.6	+3.0	-62.9	+24.4	+27.0	0.0
I2-4-20-100	+1.7	+1.4	-40.0	+5.5	+4.8	-38.7	+29.6	+29.6	0.0
I2-8-10-100	**-0.2**	+4.6	-14.6	+2.3	+4.5	0.0	+9.4	+13.4	0.0
I2-8-10-150	+1.0	+3.8	-6.3	+0.8	+1.5	0.0	+6.2	+5.9	0.0
I2-8-20-100	+4.1	+5.6	-35.3	+2.2	+8.6	0.0	+25.3	+24.8	0.0
I2-8-20-150	+1.7	+1.7	0.0	+6.2	+5.9	0.0	+13.3	+14.9	0.0
I2-12-20-150	+2.7	+4.3	0.0	+1.1	+2.1	0.0	+13.6	+14.0	0.0
I2-12-20-200	+1.4	+2.1	0.0	+1.4	+2.6	0.0	+12.9	+12.7	0.0
I2-12-30-150	+0.9	+2.5	0.0	+1.1	+2.0	0.0	+21.0	+21.6	0.0
I2-12-30-200	+0.6	+1.5	0.0	+1.3	+2.2	0.0	+16.6	+16.5	0.0

extent) by TM2PX, MUM, MPES, RGPM, and LS. The HMPG can always get the best average solutions and almost all the best solutions.

In terms of running times, HMPG takes longer compared to the versions without LS, TM2PX, MUM, or MPES, due to more evolution and search operations.

Table 8: Sensitivity analysis on the HMPG components for the instances with 2 depots

Name	No-TM2PX			No-MUM			No-RGPM			HMPG		
	%B	%A	%T	%B	%A	%T	%B	%A	%T	B	A	T
I2-4-10-50	+1.1	+2.4	0.0	+1.6	+3.4	0.0	+0.6	+1.6	-51.2	**2764.7**	**2853.8**	10.0
I2-4-10-100	+9.7	+9.0	0.0	+10.7	+10.4	-11.9	+2.4	+2.8	0.0	**6449.9**	**6477.6**	10.0
I2-4-20-50	+4.9	+4.5	0.0	+4.9	+5.4	0.0	+2.9	+3.0	0.0	**3069.6**	**3092.3**	10.0
I2-4-20-100	+9.5	+8.4	0.0	+9.9	+8.3	-11.5	+8.7	+8.3	0.0	**6157.5**	**6235.3**	10.0
I2-8-10-100	-0.1	+3.5	0.0	+1.6	+3.9	-14.5	+4.7	+6.6	0.0	6757.5	**6635.4**	15.0
I2-8-10-150	+0.8	+0.6	0.0	+0.9	+0.7	0.0	+0.9	+4.7	0.0	**11059.5**	**11094.6**	15.0
I2-8-20-100	+6.7	+5.7	0.0	+6.8	+7.8	0.0	+6.5	+5.9	0.0	**5542.7**	**5573.9**	15.0
I2-8-20-150	+2.7	+2.1	0.0	+2.3	+1.8	0.0	+2.7	+2.1	0.0	**9673.4**	**9740.0**	15.0
I2-12-20-150	+4.6	+4.4	0.0	+3.0	+6.3	0.0	+2.7	+3.0	0.0	**8921.8**	**8996.0**	20.0
I2-12-20-200	+1.3	+0.1	0.0	+2.1	+3.0	0.0	+1.1	+1.8	0.0	**11706.8**	**11724.5**	20.0
I2-12-30-150	+2.9	+3.8	0.0	+2.2	+2.9	0.0	+2.3	+4.6	0.0	**7788.3**	**7800.6**	20.0
I2-12-30-200	+0.6	+1.5	0.0	+0.9	+1.5	0.0	+1.2	+3.9	0.0	**11831.9**	**12005.4**	20.0

Table 9: Sensitivity analysis on the HMPG components for the instances with 3 depots

Name	No-LS			No-MPES			No-TPH		
	%B	%A	%T	%B	%A	%T	%B	%A	%T
I3-4-10-50	+0.7	+1.1	-10.3	+1.4	+3.8	0.0	+24.8	+22.5	0.0
I3-4-10-100	+0.2	+2.6	-28.8	+1.4	+1.2	0.0	+13.2	+21.1	0.0
I3-4-20-50	+0.9	+4.5	-31.8	+5.6	+5.7	0.0	+23.6	+26.2	0.0
I3-4-20-100	**-0.3**	+1.3	0.0	+5.6	+5.3	0.0	+15.0	+15.2	0.0
I3-8-10-100	+0.1	+0.5	0.0	+6.9	+2.9	0.0	+17.9	+14.5	0.0
I3-8-10-150	+0.4	+0.7	0.0	+6.1	+6.2	0.0	+7.8	+8.8	0.0
I3-8-20-100	+1.2	+2.2	0.0	+0.7	+3.4	0.0	+16.1	+17.5	0.0
I3-8-20-150	+1.8	+2.0	0.0	+0.3	+0.2	0.0	+13.6	+13.7	0.0
I3-12-20-150	+2.2	+3.2	0.0	+1.9	+1.8	0.0	+17.4	+18.7	0.0
I3-12-20-200	+1.4	+2.8	0.0	+8.9	+9.2	0.0	+13.2	+15.0	0.0
I3-12-30-150	+1.4	+1.0	0.0	+2.4	+1.1	0.0	+23.5	+21.6	0.0
I3-12-30-200	+2.4	+2.9	0.0	+4.7	+9.8	0.0	+30.0	+28.4	0.0

Table 10: Sensitivity analysis on the HMPG components for the instances with 3 depots

Name	No-TM2PX			No-MUM			No-RGPM			HMPG		
	%B	%A	%T	%B	%A	%T	%B	%A	%T	B	A	T
I3-4-10-50	+2.2	+3.5	-32.6	+1.2	+3.2	0.0	**0.0**	+1.9	0.0	**4028.6**	**4338.7**	10.0
I3-4-10-100	+2.3	+3.0	-32.4	+4.1	+4.2	-4.5	+3.0	+3.0	0.0	**7351.5**	**7396.7**	10.0
I3-4-20-50	+7.1	+7.9	0.0	+8.0	+7.6	0.0	+7.2	+7.9	0.0	**3538.6**	**3552.8**	10.0
I3-4-20-100	+6.1	+5.2	0.0	+6.0	+5.0	-11.0	+6.3	+5.2	0.0	6941.1	**7016.5**	10.0
I3-8-10-100	+6.5	+2.6	0.0	+6.5	+2.8	0.0	+6.9	+2.8	0.0	**6006.4**	**6263.1**	15.0
I3-8-10-150	+0.9	+2.1	0.0	+2.3	+2.4	0.0	+2.5	+2.6	0.0	**9040.7**	**9045.9**	15.0
I3-8-20-100	+1.1	+1.3	0.0	+0.2	+3.3	0.0	+1.3	+2.9	0.0	**5803.4**	**5818.4**	15.0
I3-8-20-150	+2.6	+2.8	0.0	+2.7	+2.6	0.0	+2.9	+3.0	0.0	**9233.2**	**9259.4**	15.0
I3-12-20-150	+4.1	+4.9	0.0	+3.1	+3.5	0.0	+5.7	+5.5	0.0	**8276.9**	**8305.3**	20.0
I3-12-20-200	+3.4	+4.5	0.0	+5.5	+5.3	0.0	+6.0	+6.1	0.0	**14771.5**	**14833.2**	20.0
I3-12-30-150	+2.2	+0.6	0.0	+1.4	+2.3	0.0	+1.9	+3.9	0.0	**8382.6**	**8528.2**	20.0
I3-12-30-200	+5.7	+4.6	0.0	+6.4	+6.4	0.0	+4.6	+6.3	0.0	**12387.6**	**12593.9**	20.0

4.3.2 Effectiveness of the new features of HMPG

As mentioned in Section 3, algorithm HMPG is based on the HGSADC metaheuristic proposed by Vidal et al. (2012). Its implementation mainly differs from the solution framework used for HGSADC as follows:

i) it considers more than one feasible subpopulation and shares the best solutions among the feasible subpopulations;

Table 11: Performance comparison of HMPG-1 with HMPG

Name	HMPG-1					HMPG				
	Middle condition		Termination condition			Middle condition		Terminate condition		
	B	A	B	A	T	B	A	B	A	T
I1-4-10-50	3321.3	3493.7	3321.3	3493.7	**3.5**	**3129.4**	**3301.9**	**3074.7**	**3177.1**	10.0
I1-4-10-100	5756.2	5759.9	5756.2	5759.9	**7.8**	**5736.3**	**5750.5**	**5716.8**	**5726.1**	10.0
I1-4-20-50	3389.4	3391.4	3385.5	3389.1	**6.1**	**3158.3**	**3262.7**	**3164.0**	**3164.3**	10.0
I1-4-20-100	6498.4	6562.9	6371.7	6435.0	**8.2**	**6324.4**	**6348.1**	**6314.6**	**6336.9**	10.0
I1-8-10-100	6223.9	6225.6	6223.9	6225.6	15.0	**5747.1**	**5991.9**	**5747.1**	**5757.4**	15.0
I1-8-10-150	8885.4	8885.4	8885.4	8885.4	15.0	**8885.4**	**8971.2**	**8630.7**	**8699.9**	15.0
I1-8-20-100	6059.4	6072.3	6051.2	6066.4	15.0	**5746.1**	**6005.3**	**5682.4**	**5939.9**	15.0
I1-8-20-150	9853.7	9860.0	9853.7	9857.3	15.0	**9852.1**	**9853.4**	**9823.3**	**9835.7**	15.0
I1-12-20-150	8259.4	8266.9	8245.9	8254.7	20.0	**8249.7**	**8253.0**	**8201.4**	**8238.9**	20.0
I1-12-20-200	12055.1	12057.3	12049.5	12050.8	20.0	**11836.2**	**12047.4**	**11811.1**	**11843.3**	20.0
I1-12-30-150	8659.1	8690.8	8659.1	8663.3	20.0	**8448.5**	**8553.5**	**8564.2**	**8571.8**	20.0
I1-12-30-200	11675.6	11687.1	11340.8	11553.1	20.0	**11323.6**	**11474.7**	**11087.5**	**11175.3**	20.0
I2-4-10-50	**2823.5**	**2846.8**	2823.5	2846.8	**4.1**	2824	2914.5	**2764.7**	**2853.8**	10.0
I2-4-10-100	6915.5	6917.1	6913.9	6916.6	**6.2**	**6806.7**	**6891.6**	**6449.9**	**6477.6**	10.0
I2-4-20-50	3173.8	3179.9	3173.8	3179.9	**5.3**	**3111.1**	**3153.2**	**3069.6**	**3092.3**	10.0
I2-4-20-100	7142.5	7170.1	7137.3	7168.4	**9.3**	**6930.4**	**7073.6**	**6157.5**	**6235.3**	10.0
I2-8-10-100	6929.0	6929.6	6929.0	6929.6	**8.8**	**6914.8**	**6939.9**	**6757.5**	**6635.4**	15.0
I2-8-10-150	11159.3	11162.4	11159.3	11159.9	15.0	**11157.3**	**11161.7**	**11059.5**	**11094.6**	15.0
I2-8-20-100	6059.4	6072.3	6051.2	6066.4	15.0	**5609.5**	**5835.5**	**5542.7**	**5573.9**	15.0
I2-8-20-150	10136.8	10238.1	10136.8	10162.8	15.0	**9990.5**	**9889.9**	**9673.4**	**9740.0**	15.0
I2-12-20-150	9223.9	9343.2	9095.6	9143.1	20.0	**8982.8**	**9001.5**	**8921.8**	**8996.0**	20.0
I2-12-20-200	12248.7	12290.7	12248.1	12274.5	20.0	**12037.5**	**12196.4**	**11706.8**	**11724.5**	20.0
I2-12-30-150	7886.3	7911.6	7882.0	7904.9	20.0	**7878.6**	**7892.2**	**7788.3**	**7800.6**	20.0
I2-12-30-200	12115.4	12278.6	11911.1	12233.9	20.0	**12002.4**	**12091.9**	**11831.9**	**12005.4**	20.0
I3-4-10-50	4447.2	4477.7	4447.2	4477.7	**4.3**	**4320.6**	**4431.1**	**4028.6**	**4338.7**	10.0
I3-4-10-100	7496.4	7496.5	7496.3	7496.5	**6.1**	**7489.2**	**7494.8**	**7351.5**	**7396.7**	10.0
I3-4-20-50	3677.1	3699.0	3673.1	3696.5	**6.8**	**3552.6**	**3607.4**	**3538.6**	**3552.8**	10.0
I3-4-20-100	7381.2	7381.2	7381.2	7381.2	**8.8**	**7138.4**	**7322.6**	**6941.1**	**7016.5**	10.0
I3-8-10-100	6458.5	6459.2	6443.5	6448.8	15.0	**6449.6**	**6456.0**	**6006.4**	**6263.1**	15.0
I3-8-10-150	9051.3	9056.9	9051.3	9056.9	15.0	**9041.2**	**9054.9**	**9040.7**	**9045.9**	15.0
I3-8-20-100	5863.3	5884.4	5858.5	5869.5	15.0	**5834.1**	**5883.8**	**5803.4**	**5818.4**	15.0
I3-8-20-150	9301.1	9302.2	9301.1	9302.2	15.0	**9275.5**	**9293.7**	**9233.2**	**9259.4**	15.0
I3-12-20-150	8344.0	8358.4	8335.9	8347.2	20.0	**8299.6**	**8340.3**	**8276.9**	**8305.3**	20.0
I3-12-20-200	15192.7	15265.0	15086.7	15212.0	20.0	**14842.0**	**15017.0**	**14771.5**	**14833.2**	20.0
I3-12-30-150	8500.9	8596.5	8448.5	8542.1	20.0	**8410.0**	**8553.5**	**8382.6**	**8528.2**	20.0
I3-12-30-200	13621.5	13862.1	13621.5	13701.2	20.0	**13108.8**	**13635.5**	**12387.6**	**12593.9**	20.0

ii) it generates the initial population using a constructive heuristic (see Section 3.2) instead of randomly generating it;

iii) it uses an evaluation function that, in addition to node assignments, also considers the differences between the arcs traveled;

iv) it diversifies the population by applying a *Routing global perturbation mechanism* (see Section 3.7.3) instead of randomly generating new individuals.

In this section, in order to attest the effectiveness of the new features we included in HMPG, we consider a HMPG variant, called HMPG-1, which is strictly based on the solution framework proposed for HGSADC.

To better compare the performance of the two heuristics, instead of merely taking the results obtained by the termination condition, we also take the results obtained by the middle condition (defined as half of the maximum running time). The comparison based on the instances is provided in Table 11.

From the table, we can see that variant HMPG-1 was not able to obtain the same best solutions within the given running times and number of iterations. The proposed HMPG outperforms its HMPG-1 variant in all instances, both at the middle and the termination conditions, with higher running times for only a few instances.

5 Conclusions

In this paper, we considered a new city logistics problem arising in the last mile distribution of e-commerce. A feature of the problem is that customers may provide different delivery options, allowing them to pick up their packages at intermediate pickup facilities.

The problem is complex and highly constrained as it involves a number of different, interconnected decisions (service options, facility locations, and two levels of vehicle routes). To solve the problem, a hybrid multi-population genetic algorithm was proposed.

The proposed method was first tested on a real-world instance, involving two depots, 12 satellites, 40 pickup facilities and 164 customers, which called for the design of a joint city distribution system involving two different transportation companies for the last mile distribution for online shopping. The results show that the distribution system can be largely optimized by providing both joint distribution and delivery options, allowing a final cost reduction of about 16% with respect to the scenario with independent distribution system and without delivery options.

In this paper, we also considered randomly generated instances, involving up to three depots, 12 satellites, 30 pickup facilities and 200 customers. The computational results obtained on the randomly generated instances demonstrate the effectiveness of the different components of the algorithm.

Future work involves the development and implementation of lower bounds and methods for solving the problem to optimality.

Acknowledgements

This study is supported by National Science and Technology Support Program of China under Grant No. 2015BAH46F01, and it is also part of the project supported by China Scholarship Council (CSC). The authors thank the anonymous referees for making several suggestions that improved the presentation of the paper.

References

Fardin Ahmadizar, Mehdi Zeynivand, and Jamal Arkat. Two-level vehicle routing with cross-docking in a three-echelon supply chain: A genetic algorithm approach. *Applied Mathematical Modelling*, 39(22):7065–7081, 2015.

Somayeh Allahyari, Majid Salari, and Daniele Vigo. A hybrid metaheuristic algorithm for the multi-depot covering tour vehicle routing problem. *European Journal of Operational Research*, 242(3):756–768, 2015.

Roberto Baldacci and Mauro Dell'Amico. Heuristic algorithms for the multi-depot ring-star problem. *European Journal of Operational Research*, 203(1):270–281, 2010.

Roberto Baldacci, Mauro Dell'Amico, and J Salazar González. The capacitated m-ring-star problem. *Operations Research*, 55(6):1147–1162, 2007.

Roberto Baldacci, Aristide Mingozzi, Roberto Roberti, and Roberto Wolfler Calvo. An exact algorithm for the two-echelon capacitated vehicle routing problem. *Operations Research*, 61(2): 298–314, 2013.

Tolga Bektaş, Güneş Erdoğan, and Stefan Røpke. Formulations and branch-and-cut algorithms for the generalized vehicle routing problem. *Transportation Science*, 45(3):299–316, 2011.

Tolga Bektaş, Teodor Gabriel Crainic, and Tom Van Woensel. From managing urban freight to smart city logistics networks. 2015.

Kris Braekers, Katrien Ramaekers, and Inneke Van Nieuwenhuyse. The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 2016.

Diego Cattaruzza, Nabil Absi, Dominique Absi, and Jess Gonzlez-Feliu. ehicle routing problems for city logistics. *Euro Journal on Transportation & Logistics*, pages 1–29, 2015.

Claudio Contardo, Vera Hemmelmayr, and Teodor Gabriel Crainic. Lower and upper bounds for the two-echelon capacitated location-routing problem. *Computers & Operations research*, 39 (12):3185–3199, 2012.

Teodor Crainic, Roberto Tadei, and Antonio Sforza. *Location-routing models for two-echelon freight distribution system design*. CIRRELT, 2011a.

Teodor Gabriel Crainic and Antonino Sgalambro. Service network design models for two-tier city logistics. *Optimization Letters*, 8(4):1375–1387, 2014.

Teodor Gabriel Crainic, Nicoletta Ricciardi, and Giovanni Storchi. Models for evaluating and planning city logistics systems. *Transportation Science*, 43(4):432–454, 2009.

Teodor Gabriel Crainic, Simona Mancini, Guido Perboli, and Roberto Tadei. *Multi-start heuristics for the two-echelon vehicle routing problem*. Springer, 2011b.

Teodor Gabriel Crainic, Antonio Sforza, and Claudio Sterle. *Tabu search heuristic for a two-echelon location-routing problem*. CIRRELT, 2011c.

Rosario Cuda, Gianfranco Guastaroba, and Maria Grazia Speranza. A survey on two-echelon routing problems. *Computers & Operations Research*, 55:185–199, 2015.

Houda Derbel, Bassem Jarboui, Saïd Hanafi, and Habib Chabchoub. Genetic algorithm with iterated local search for solving a location-routing problem. *Expert Systems with Applications*, 39(3):2865–2871, 2012.

Michael Drexl and Michael Schneider. A survey of variants and extensions of the location-routing problem. *European Journal of Operational Research*, 241(2):283–308, 2015.

David A. Flores-Garza, M. Anglica Salazar-Aguilar, Sandra Ulrich Ngueveu, and Gilbert Laporte. The multi-vehicle cumulative covering tour problem. *Annals of Operations Research*, pages 1–20, 2015.

Roel Gevaers, Eddy Van de Voorde, and Thierry Vanelslander. Characteristics of innovations in last-mile logistics-using best practices, case studies and making the link with green and sustainable logistics. In *European Transport Conference, 2009*, 2009.

Gianpaolo Ghiani and Gennaro Improta. An efficient transformation of the generalized vehicle routing problem. *European Journal of Operational Research*, 122(1):11–17, 2000.

Paolo Gianessi, Laurent Alfandari, Lucas Létocart, and Roberto Wolfler Calvo. The multicommodity-ring location routing problem. *Transportation Science*, 50(2):541 – 558, 2015.

Jesús Gonzales Feliu, Guido Perboli, Roberto Tadei, and Daniele Vigo. The two-echelon capacitated vehicle routing problem. *Technical report DEIS OR.INGCE 2007/2(R), Department of Electronics, Computer Science, and Systems,University of Bologna, Bologna, Italy*, 2007.

Kannan Govindan, Ahmad Jafarian, Roohollah Khodaverdi, and Kannan Devika. Two-echelon multiple-vehicle location–routing problem with time windows for optimization of sustainable supply chain network of perishable food. *International Journal of Production Economics*, 152: 9–28, 2014.

Philippe Grangier, Michel Gendreau, Fabien Lehud, and Louis Martin Rousseau. An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. *European Journal of Operational Research*, 254(1):80–91, 2016.

Minh Hong H, Nathalie Bostel, Andr Langevin, and Louis Martin Rousseau. An exact algorithm and a metaheuristic for the multi-vehicle covering tour problem with a constraint on the number of vertices. *European Journal of Operational Research*, 226(2):211–220, 2013.

Minh Hong H, Nathalie Bostel, Andr Langevin, and Louis Martin Rousseau. An exact algorithm and a metaheuristic for the generalized vehicle routing problem with flexible fleet size. *Computers & Operations Research*, 43(43):9–19, 2014.

Vera C Hemmelmayr, Jean-François Cordeau, and Teodor Gabriel Crainic. An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Computers & Operations research*, 39(12):3215–3228, 2012.

S Kruse Jacobsen and Oli BG Madsen. A comparative study of heuristics for a two-level routing-location problem. *European Journal of Operational Research*, 5(6):378–387, 1980.

Mads Jepsen, Simon Spoorendonk, and Stefan Ropke. A branch-and-cut algorithm for the symmetric two-echelon capacitated vehicle routing problem. *Transportation Science*, 47(1):23–37, 2013.

Nicolas Jozefowiez. A branch-and-price algorithm for the multivehicle covering tour problem. *Networks*, 64(3):160–168, 2014.

Gilbert Laporte and Yves Nobert. *A vehicle flow model for the optimal design of a two-echelon distribution system*. Springer, 1988.

Shen Lin and Brian W Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21(2):498–516, 1973.

Oli BG Madsen. Methods for solving combined two level location-routing problems of realistic dimensions. *European Journal of Operational Research*, 12(3):295–301, 1983.

J.E. Mendoza, C. Guéret, M. Hoskins, H. Lobit, V. Pillac, T. Vidal, and D. Vigo. VRP-REP: the vehicle routing community repository. third meeting of the EURO Working Group on Vehicle Routing and Logistics Optimization (VeRoLog). Oslo, Norway, 2014.

Zahra Naji-Azimi, Jacques Renaud, Angel Ruiz, and Majid Salari. A covering tour approach to the location of satellite distribution centers to supply humanitarian aid. *European Journal of Operational Research*, 222(3):596C605, 2012.

Viet-Phuong Nguyen, Christian Prins, and Caroline Prodhon. A multi-start iterated local search with tabu list and path relinking for the two-echelon location-routing problem. *Engineering Applications of Artificial Intelligence*, 25(1):56–71, 2012a.

Viet-Phuong Nguyen, Christian Prins, and Caroline Prodhon. Solving the two-echelon location routing problem by a grasp reinforced by a learning process and path relinking. *European Journal of Operational Research*, 216(1):113–126, 2012b.

Guido Perboli, Roberto Tadei, and Daniele Vigo. The two-echelon capacitated vehicle routing problem: Models and math-based heuristics. *Transportation Science*, 45(3):364–380, 2011.

Petrică C. Pop, Imdat Kara, and Andrei Horvat Marc. New mathematical models of the generalized vehicle routing problem and extensions. *Applied Mathematical Modelling*, 36(1):97–107, 2012.

Christian Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12):1985–2002, 2004.

Caroline Prodhon and Christian Prins. A survey of recent research on location-routing problems. *European Journal of Operational Research*, 238(1):1–17, 2014.

Julia Rieck, Carsten Ehrenberg, and Jrgen Zimmermann. Many-to-many location-routing with inter-hub transport and multi-commodity pickup-and-delivery. *European Journal of Operational Research*, 236(3):863–878, 2014.

Fernando Afonso Santos, Geraldo Robson Mateus, and Alexandre Salles da Cunha. A branch-and-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem. *Transportation Science*, 49(2):355–368, 2014.

Martin Schwengerer, Sandro Pirkwieser, and Günther R Raidl. A variable neighborhood search approach for the two-echelon location-routing problem. In *Evolutionary Computation in Combinatorial Optimization*, pages 13–24. Springer, 2012.

Mehmet Soysal, Jacqueline M Bloemhof-Ruwaard, and Tolga Bektaş. The time-dependent two-echelon capacitated vehicle routing problem with environmental considerations. *International Journal of Production Economics*, 164:366–378, 2015.

Paolo Toth and Daniele Vigo. The granular tabu search and its application to the vehicle-routing problem. *Informs Journal on Computing*, 15(4):333–346, 2003.

Fabien Tricoire, Alexandra Graf, and Walter J. Gutjahr. The bi-objective stochastic covering tour problem. *Computers & Operations Research*, 39(7):1582–1592, 2012.

Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, Nadia Lahrichi, and Walter Rei. A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research*, 60(3):611–624, 2012.

Xuping Wang, Linmin Zhan, Junhu Ruan, and Jun Zhang. How to choose last mile delivery modes for e-fulfillment. *Mathematical Problems in Engineering*, 2014, 2014.

Vincent F. Yu and Shin-Yu Lin. A simulated annealing heuristic for the open location-routing problem. *Computers & OperationsResearch*, 62:184–196, October 2015.