sustainability

MDPI

Article

Location-Routing Problem with Simultaneous Home Delivery and Customer's Pickup for City Distribution of Online Shopping Purchases

Lin Zhou [1], Xu Wang [2,3,*], Lin Ni [1,2] and Yun Lin [1,2]

[1] College of Mechanical Engineering, Chongqing University, Chongqing 400030, China; zhoulin1205@126.com or lin.zhou@cqu.edu.cn (L.Z.); nilin71@163.com (L.N.); linyun313@163.com (Y.L.)
[2] Chongqing Key Laboratory of Logistics, Chongqing University, Chongqing 400030, China
[3] State Key Laboratory of Mechanical Transmission, Chongqing University, Chongqing 400030, China
* Correspondence: wx921@163.com; Tel.: +86-23-6510-5109

Academic Editor: Marc A. Rosen
Received: 21 June 2016; Accepted: 18 August 2016; Published: 22 August 2016

Abstract: With the increasing interest in online shopping, the Last Mile delivery is regarded as one of the most expensive and pollutive—and yet the least efficient—stages of the e-commerce supply chain. To address this challenge, a novel location-routing problem with simultaneous home delivery and customer's pickup is proposed. This problem aims to build a more effective Last Mile distribution system by providing two kinds of service options when delivering packages to customers. To solve this specific problem, a hybrid evolution search algorithm by combining genetic algorithm (GA) and local search (LS) is presented. In this approach, a diverse population generation algorithm along with a two-phase solution initialization heuristic is first proposed to give high quality initial population. Then, advantaged solution representation, individual evaluation, crossover and mutation operations are designed to enhance the evolution and search efficiency. Computational experiments based on a large family of instances are conducted, and the results obtained indicate the validity of the proposed model and method.

Keywords: location-routing problem; simultaneous home delivery and customer's pickup; genetic algorithm

1. Introduction

The rapid development of e-commerce has generated growing interest in online shopping. Taking China as an example, according to China's E-commerce Report 2015 [1], in 2015, the e-commerce transactions totaled 2.5 trillion US dollars and online shopping transactions accounted for 80% of the total e-commerce transactions with an increasing rate of 27.2% compared to 2014. The sharply increased online shopping market leads to an explosive growth of city distribution demand. The annual package delivery quantity reached 20 billion in 2015, leading China to rank first in the world. The Last Mile delivery works as the final stretch that delivers consignments to the recipients [2] and plays an indispensable role in promoting e-commerce development. A survey showed that 85% of buyers who received their orders on time would purchase online again compared with those 33% who received delayed orders [3]. Different to traditional city distribution, the customers engaging in online shopping are spatially distributed and have personalized delivery demands, rendering the Last Mile delivery the most expensive and pollutive, as well as the least efficient, stages of the whole e-commerce supply chain, accounting for 13% up to 75% of the total supply chain costs [4]. Thus the Last Mile delivery problem has recently garnered a great deal of attention from the government and all the stakeholders of e-commerce supply chain.

In the Last Mile delivery, the most widespread delivery mode is home delivery (HD), by which couriers send packages directly to a customer´s home or office. HD service provides a face-to-face opportunity with customers, but its low operational efficiency is likely to make it undesirable to handle mass orders, thereby increasing operation costs [5]. Subsequently, another kind of service named customer's pickup (CP) has been widely accepted by both couriers and customers. More precisely, CP service allows customers to retrieve their packages at a facility close to their home or work (we refer to it as pickup point). By providing such an option that allows packages to be delivered to a facility with a large storage area, efficiency will be improved while maintaining personalized service. Customers can pick up their packages at their convenience, thus improving their satisfaction. As the key issue for the Last Mile delivery is to efficiently design the Last Mile delivery system [6]. To better deal with the Last Mile delivery problem we are facing, in this paper, we proposed a novel multi-depot location-routing problem with simultaneous home delivery and customer's pickup (LRPSHC). In this problem, each customer can be serviced either by HD or CP; the purpose is to decide on the location of pickup points and customers' service options, as well as schedule the vehicles in order to minimize the total operation costs. For solving this problem, a hybrid evolution search algorithm by combining genetic algorithm (GA) and local search (LS) is designed. This research allows us to operate the Last Mile delivery in an economical and sustainable way.

The remainder of this paper is structured as follows. The next section provides a comprehensive review of the recent research on Last Mile delivery and the variants of the location-routing problem, as well as the solving methods. Section 3 details the description and formulation of the proposed LRPSHC. In Section 4, the proposed hybrid evolution search algorithm is presented. Section 5 displays the results of the computational experiments and the discussion. Conclusions are given in Section 6. Finally, future research directions are given in Section 7.

2. Literature Review

Realizing that the Last Mile delivery is one of the most expensive stages of the entire e-logistics chain, a cost simulation tool was developed by Gevaers et al. [7] to simulate Last Mile delivery costs. According to their research, changes within Last Mile characteristics significantly influence cost. To more clearly show how certain factors affect the Last Mile delivery cost, customer density and delivery window size were considered by Boyer et al. [8]. Simulation results showed that greater customer density and longer delivery windows benefit the delivery efficiency enormously. Based on three delivery modes—HD, reception box and collection-and-delivery points—Wang et al. [5] undertook a quantitative study of the competitiveness of the three modes through analyzing the delivery cost structure and operation efficiency in different scenarios, which helps identify the most suitable mode for different customer distribution densities. Hayel et al. [9] proposed a queuing model to describe the Last Mile delivery system with HD and CP options. Considering the monetary and congestion effect of the two options, a game-theoretical approach was designed for determining the optimum option a consumer would make. Aized and Srai [4] provided a conceptual approach for modeling the Last Mile delivery system based on hierarchy to support the routing planning.

Another key issue related to our research is the location-routing problem (LRP). Several recent surveys are available which help provide a comprehensive overview of the studies about LRP, such as the work of Lopes et al. [10], and most recently that of Prodhon and Prins [11] and Drexl and Schneider [12]. Among these studies, LRP variants were demonstrated, including open LRP [13], LRP with time windows [14,15], LRP with split demands [16], LRP with subcontracting options [17], LRP with simultaneous delivery and pickup [18–22], LRP with depots connected by ring [23], and more complex variants with multi-period planning [24,25] and inventory management [26–28]. Another key variant of the LRP is two-echelon LRP [20,22,29–31].

When it comes to the solving approaches, both exact algorithms and heuristics were proposed to solve LRP and its variants. Interested readers are encouraged to examine the research of Lopes et al. [10], in which a review of the solving methods was provided. For the recent and relevant approaches, one can find ant colony optimization (ACO) [32], genetic algorithm(GA) [33,34], simulated annealing (SA) [13,35], simulated annealing with variable neighborhood search (SA-VNS) [18], local search (LS) [20], greedy randomized adaptive search with evolutionary local search (GRASP+ELS) [36], granular tabu search (GTS) [37], granular tabu search with variable neighborhood search (GVTNS) [38], multi-start iterated local search with tabu list and path relinking (MS-ILS) [31] and multi-start simulated annealing heuristic (MSA) [21]. The most related methods are the metaheuristics hybrid with GA and LS. Prins et al. [39] presented a memetic algorithm with population management (MAPM). The main idea behind MAPM is that LS is used to improve the solution generated by GA. A similar method was described by Derbel et al. [40]; iterative local search [41] is adopted to improve the solution generated by the GA. Marinakis and Marinaki [42] formulated a bi-level programming for LRP, and a bi-level GA was designed to solve the problem accordingly. In this method, GA is used to locate the facilities in the first level and the VRP in the second level is solved by expanding neighborhood search (ENS). Most recently, Lopes et al. [43] proposed a simple and effective hybrid GA with LS procedure that works as a mutation operator.

By reviewing previous studies, we find that scant research has been conducted on the LRP with simultaneous HD and CP, and thus comes the main contribution of this study: we propose a novel location-routing problem with simultaneous home delivery and customer's pickup for the Last Mile delivery. Since customers have two available services to choose from—HD to be served by vehicles directly or CP to pick up their packages from the closest pickup points—delivery efficiency will be improved and customers' personalized demands addressed.

3. Problem Formulation

3.1. Problem Description

The proposed LRPSHC can be formally stated as follows. Letting $G = (N, A, C)$ represents a weighted and undirected network containing a set of vertices N, a set of edge A and a set of connection C. Where, $N = N_D \cup N_P \cup N_C$, $N_D = \{1, \cdots, n_d\}$ stands for n_d depots, $N_P = \{n_d + 1, \cdots, n_d + n_p\}$ stands for n_p potential pickup points and $N_C = \{n_d + n_p + 1, \cdots, n_d + n_p + n_c\}$ stands for n_c customers should be served. Each customer $i \in N_C$ has a delivery demand q_i and a service time t_i, $t_i = q_i u_H$, where u_H represents the unit demand service time. Once selected, a pickup point $p \in N_P$ is opened with an opening cost U_p containsedcapacity B_p and fixed service time u_P. Set $A = \{(i, j) : i, j \in N\}$ is a set of edges connecting each pair of nodes in N, with each edge $(i, j) \in A$ associated with a routing cost $c_{ij}$and a travelling time t_{ij} which are dependent on the distance d_{ij} between nodes i and j, $i, j \in N$. Set $C = \{(p, c) : p \in N_P, c \in N_C\}$ contains the connections between pickup points and customers, a connection (p, c) occurs only when the distance d_{pc} between the selected pickup point $p \in N_P$ and customer $c \in N_C$ is less than the acceptable distance D_{pc}. For a given $d \in N_D$, a capacity S_d is associated with it. A fleet of identical vehicles n_v with capacity Q_v and maximum working time T_v are available and denoted as V_D. Once used, a fixed cost F_v incurs.

In this research, to ensure acceptability and reliability, the following two real-world situations are considered:

(1) The preference for a customer to choose CP service is based on the acceptable distance to the closest selected pickup point, and the customer will choose CP service only when within the acceptable distance D_{cp} as mentioned before; otherwise, HD service should be provided.

(2) Due to uncertain factors that customers face in real life, HD service may fail for the first delivery with probability p_H with unit second delivery cost μ_{sd}.

An example of the Last Mile distribution system with three depots, six candidate pickup points and 17 customers are given in Figure 1.

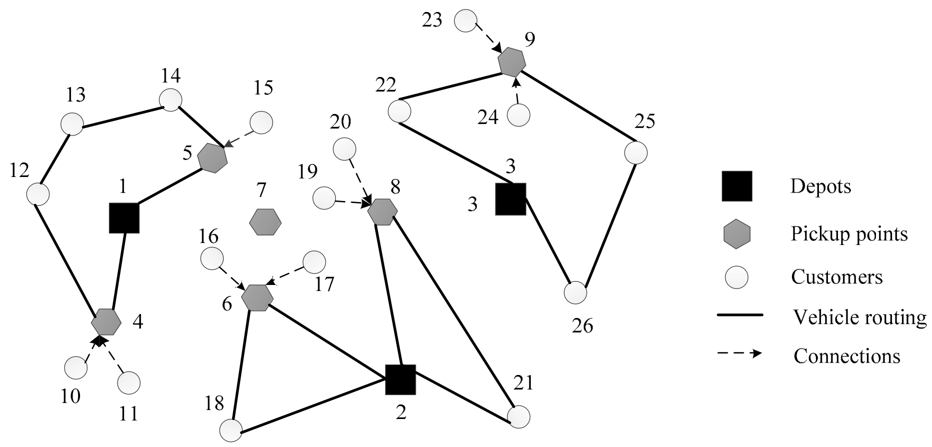

Figure 1. The Last Mile distribution system with simultaneous HD and CP services.

The aim is then to determine the following:

(1) A set of pickup point locations;
(2) Service options for customers;
(3) Assignments between customers and selected pickup points;
(4) Assignments between the selected pickup points and depots;
(5) Vehicles´ scheduling and routing.

Without losing generality, the following assumptions are made:

(1) As online shoppers may be clustered in the same or nearby buildings, all the customers are divided into small groups according to their locations. We then take each small group as a customer with fixed demand;
(2) Each customer can be served by either HD or CP service;
(3) We do not consider the operation of the second delivery caused by uncertain factors in the current scheduling, and take the second delivery cost instead.

3.2. Notions and the Proposed Model

The notions used in the formulation are summarized in Table 1.

Table 1. Notions used in the formulation.

Sets	***Description***
G	Network, $G = (N, A, C)$
N	Node set, $N = N_D \cup N_P \cup N_C$
N_D	Depot set, $N_D = \{1, \cdots, n_d\}$
N_P	Pickup point set, $N_P = \{n_d + 1, \cdots, n_d + n_p\}$
N_C	Customer set, $N_C = \{n_d + n_p + 1, \cdots, n_d + n_p + n_c\}$
A	Arc set, $A = \{(i, j) : i, j \in N\}$
C	Connection set, $C = \{(p, c) : p \in N_P, c \in N_C\}$
V_D	Available vehicle set
Parameters	
n_d	Number of depots
n_p	Number of pickup points
n_c	Number of customers
d_{ij}	Distance of arc $(i, j) \in A$
t_{ij}	Travel time of arc $(i, j) \in A$
c_{ij}	Routing cost of arc $(i, j) \in A$

Table 1. Notions used in the formulation.

Sets	*Description*
q_i	Demand of customer $i \in N_C$
u_H	Unit demand service time for HD service
u_P	Fixed service time at a pickup point
μ_{sd}	Probability of the second delivery
B_p	Capacity of pickup point $p \in N_P$
S_d	Capacity of depot $d \in N_D$
U_p	Opening cost of pickup point $p \in N_P$
Q_v	Vehicle capacity
F_v	Fixed vehicle cost
n_v	Available vehicle number
D_{pc}	Acceptable distance for CP service
Decision variables	
x^d_{ijk}	Equal to 1 if vehicle $k \in V_D$ departs from depot $d \in N_D$ that passes through arc $(i,j) \in A$; 0, otherwise
y_p	Equal to 1 if pickup point $p \in N_P$ is selected to serve customers; 0, otherwise
z_{ip}	Equal to 1 if customer $i \in N_C$ is served by pickup point $p \in N_P$; 0, otherwise
w_{pd}	Equal to 1 if pickup point $p \in N_P$ is served by depot $d \in N_D$; 0, otherwise
φ^d_{ijk}	Freight transported from depot $d \in N_D$by vehicle $k \in V_D$ that passes through arc $(i,j) \in A$

Based on the above statements, we now formulate the model for the LRPSHC as follows:

$$\min \sum_{p \in N_P} U_p y_p + \sum_{d \in N_D} \sum_{k \in V_D} F_v x^d_{djk} + \sum_{d \in N_D} \sum_{k \in V_D} \sum_{i \in N} \sum_{j \in N} c_{ij} x^d_{ijk} + \sum_{d \in N_D} \sum_{k \in V_D} \sum_{i \in N} \sum_{j \in N} q_i p_H \mu_{sd} x^d_{ijk}. \tag{1}$$

subject to

$$\sum_{j \in N} \sum_{k \in V_D} x^d_{ijk} + \sum_{p \in N_P} z_{ip} = 1, \forall i \in N_C, d \in N_D. \tag{2}$$

$$\sum_{d \in N_D} w_{pd} \le \sum_{i \in N_C} z_{ip}, \forall p \in N_P. \tag{3}$$

$$\sum_{d \in N_D} w_{pd} = y_p, \forall p \in N_P. \tag{4}$$

$$\sum_{j \in N} x^d_{ijk} = \sum_{j \in N} x^d_{jik}, \forall i \in N, k \in V_D, d \in N_D. \tag{5}$$

$$\sum_{j \in N} \varphi^d_{ijk} - \sum_{j \in N} \varphi^d_{jik} = q_i, \forall i \in N_C, k \in V_D, d \in N_D. \tag{6}$$

$$\sum_{j \in N} \varphi^d_{ijk} - \sum_{j \in N} \varphi^d_{jik} = \sum_{j \in N_C} q_i z_{jp}, \forall i \in N_P, k \in V_D, d \in N_D. \tag{7}$$

$$\varphi^d_{dik} = \sum_{c \in N_C} \sum_{j \in N} x^d_{cjk} q_c + \sum_{p \in N_P} \sum_{c \in N_C} \sum_{j \in N} x^d_{pjk} q_c z_{cp}, \forall i \in N_C \cup N_P, k \in V_D, d \in N_D. \tag{8}$$

$$\varphi^d_{idk} = 0, \forall i \in N_P \cup N_C, k \in V_D, d \in N_D. \tag{9}$$

$$\varphi^d_{ijk} \le Q_v, \forall i,j \in N, k \in V_D, d \in N_D. \tag{10}$$

$$\sum_{i \in N_C} q_i u_H x^d_{ijk} + \sum_{p \in N_P} u_P x^d_{pjk} + \sum_{i \in N} \sum_{j \in N} t_{ij} x^d_{ijk} \le T_v, \forall k \in V_D, d \in N_D. \tag{11}$$

$$\sum_{i \in N_C} \sum_{j \in N} q_i x^d_{ijk} + \sum_{p \in N_P} \sum_{i \in N_C} q_i z_{ip} w_{pd} \le S_d, \forall d \in N_D. \tag{12}$$

$$\sum_{i \in N_C} q_i z_{ip} \leq B_p, \forall p \in N_P. \tag{13}$$

$$\sum_{d \in N_d} \sum_{j \in N_P \cup N_C} \sum_{k \in V_D} x^d_{djk} \leq n_v, \forall d \in N_D. \tag{14}$$

$$\varphi^d_{ijk} \geq 0, \forall i, j \in N, k \in V_D, d \in N_D. \tag{15}$$

$$x^d_{ijk} = \{0, 1\}, \forall i, j \in N, k \in V_D, d \in N_D. \tag{16}$$

$$y_p = \{0, 1\}, \forall p \in N_P. \tag{17}$$

$$z_{ip} = \{0, 1\}, \forall i \in N_C, p \in N_P. \tag{18}$$

$$w_{pd} = \{0, 1\}, \forall p \in N_P, d \in N_D. \tag{19}$$

The objective function (1) minimizes the sum of the pickup point opening cost, fixed vehicle cost, routing cost and the second delivery cost. Constraints (2) ensure that each customer is served exactly once by either HD or CP service. Constraints (3) impose that once a customer is assigned to a pickup point, this pickup point should be served by a depot. Constraints (4) make sure that once a pickup facility is selected, it should be served by a depot. Constraints (5) are the route continuity constraints that once a vehicle enters a node, it must also leave from it. Constraints (6) and (7) are the flow constraints for demand of customer and pickup point nodes respectively. Constraints (8) calculate the demand delivered by vehicles. Constraints (9) specify that the remaining demand of a vehicle is zero after serving its last customer. Constraints (10) and (11) guarantee that the capacity and working time of vehicles should be satisfied. Constraints (12) and (13) imply that vehicles should not violate the capacity constraints of the depots and pickup points respectively. Constraints (14) require that selected vehicles should not violate available vehicles. Constraints (15)–(19) define all variables.

4. Hybrid Approach

As LRP is an NP-hard optimization problem [40], the multi-pickup point and simultaneous HD and CP services included in the considered problem have further increased the complexity of the solution. In this section, a hybrid evolution search algorithm (HGALS) that combines GA and local LS is presented, which is motivated by the excellent global search ability obtainable by using GA and the ability in finding local optima by taking LS.

In this algorithm, GA-based operations are intended to guide the global evolution while LS is taken to find better local solutions both for the initial population and the new generated individuals during the evolution process. LS is also used to work as a post-optimization procedure in order to improve the best solutions in each iteration. The solution improved by LS is accepted with probability 1 when the solution is feasible; otherwise, the solution with probability p_{ac} is accepted. Additionally, a hybrid approach along with a two-phase solution generation heuristic is proposed to obtain high quality initial population; specific crossover and mutation operators are also proposed followed by a biased fitness function to evaluate the individuals. A high level overview of the presented HGALS is depicted below.

Step 1: Parameter setting: population size M_0, maximum population M, crossover probability, mutation probability p_m, infeasible solution accept probability p_{ac}, solution improve probability p_{ls}, terminate condition

Step 2: Taking solution generation and population initialization methods to obtain initial population P_0

Step 3: Improve the population by LS

Step 4: Select individual X_1 and X_2 from the population following probability p_c, conduct crossover operation to generate offspring Y_1 and Y_2, apply LS to improve Y_1 and Y_2 with probability p_{ls}, Y_1 and Y_2 are improved to be $Y_1{}'$ and $Y_2{}'$, accept $Y_1{}'$ and $Y_2{}'$ by probability 1 or p_{ac}

Step 5: Select individual X_3 following probability p_m to make new solution Y_3 by mutation operation, apply LS to improve Y_3 with probability p_{ls}, Y_3 is improved to be $Y_3{}'$, accept $Y_3{}'$ by probability 1 or p_{ac}

Step 6: Improve the current best solution Y_4 to be $Y_4{}'$ by LS, accept $Y_4{}'$ by probability 1 or p_{ac}

Step 7: If the population reaches M, evaluation individuals and select M_0 solutions to make population P_0, turn to step 3; otherwise, turn to step 8

Step 8: If the terminate condition is satisfied, turn to step 9; if not, turn to step 4

Step 9: Output the best solution.

The general structure of the presented HGALS is given in Figure 2.

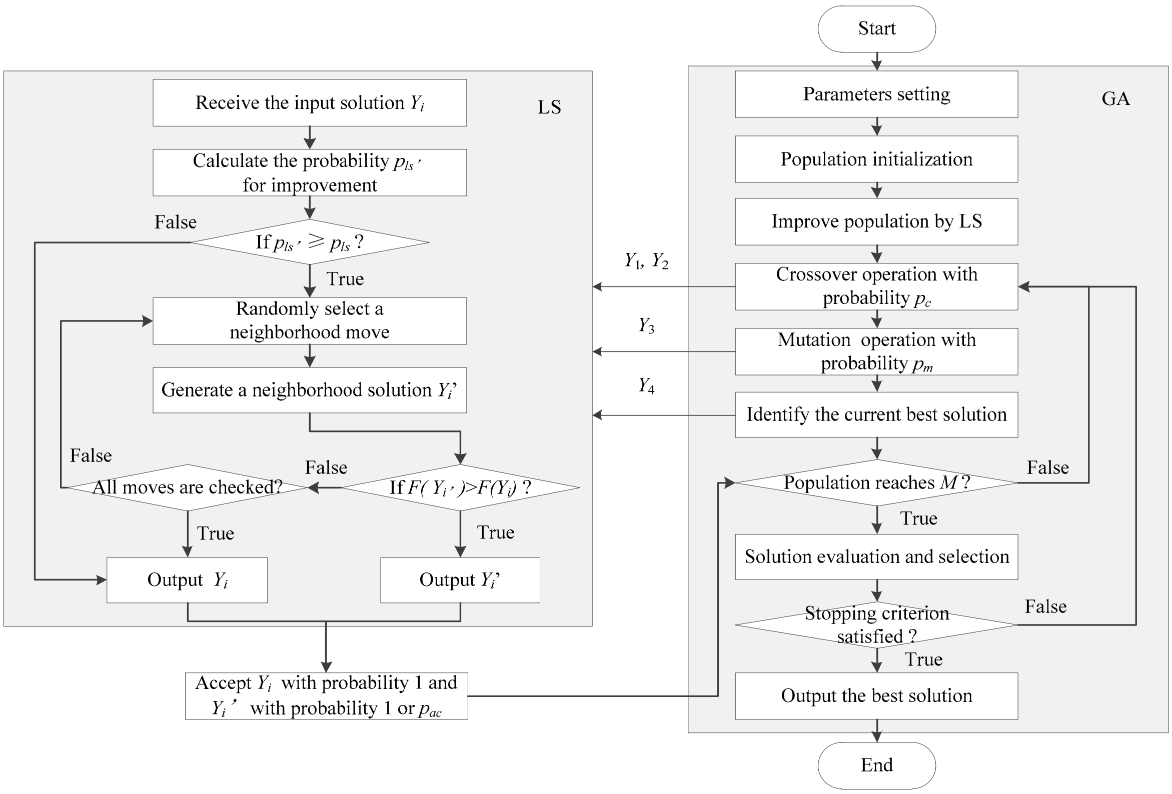

Figure 2. The general structure of HGALS.

4.1. Solution Representation

For a GA-based heuristic, chromosome encoding strategy affects the performance of genetic operations [40]. Thus, the first and most crucial step in designing GA is to define an appropriate encoding scheme. According to the characteristics of the considered problem, a mixed encoding scheme is designed to represent a solution in which each chromosome includes two sections.

The first section is composed of three tiers of gene string with a fixed length of n_c. The third tier is customer tier, which stores the customers. The second tier is used to index the service options information: if the value of a gene belongs to $\{n_d + 1, \ldots, n_d + n_p\}$, it means that the corresponding customer in the third tier is served by CP and served by the pickup point. Otherwise, if the value of the gene is 0, it means that the customer is served by HD. The first tier is for depots and indicates the assignments between the depots and pickup points as well as depots and the customers with HD service.

The second section is to deal with the vehicle routing with a permutation of n_d depots, $n_p{}'$ pickup points selected from N_P, $n_c{}'$ customers with HD service and a set of dummy zeros. Routes are started and ended at the same depots to serve the selected pickup points and customers with HD service one by one, zeros are used to terminate a route and start a new one in depots.

Figure 3 provides a solution representation of the sample given in Figure 1.

1	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	Depot
4	4	0	0	0	5	6	6	0	8	8	0	0	9	9	0	0	Pickup points
10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	Customer

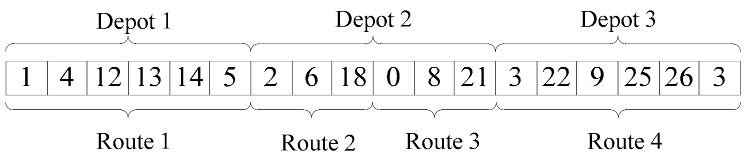

Figure 3. Solution representation.

4.2. Solution and Population Initialization

As the quality of the initial solutions affects the final solution to a large extent, to obtain high quality initial solutions, we designed a two-phase heuristic named "Routing planning after the pickup points selection and services determination". The procedure of the two-phase heuristic is depicted as follows:

Phase 1: Pickup points selection and service determination.

Step 1: Randomly select a pickup point combination set $N_{sp} \in N_{SP}$.

Step 2: Sort all the customers according to distance to the pickup points in N_{sp}, let n_p^c be the number of the customers whose closest pickup point is p, and the corresponding customer set be N_p^C.

Step 3: Select the pickup point p with the largest n_p^c.

Step 4: If within the pickup point's capacity, let the service option of its closest customer in N_p^C be CP. Delete the customer in N_C and N_p^C. If p is in N_{sp}, delete it from N_{sp} and put it into N_C.

Step 5: If $N_p^C = \varnothing$, turn to step 2; otherwise, turn to step 3.

Step 6: If $N_{sp} \neq \varnothing$, turn to step 2; otherwise, turn to step 7.

Phase 2: Route planning

Step 7: Let N_d^c be the node set that store the customers with HD service and selected pickup points in N_C and whose closest depot is depot d. Rank all the nodes in N_C with the increasing distance to the depots and put them into the corresponding sets.

Step 8: For each depot $d \in N_D$ and the nodes in N_d^c, the Traveling Salesman Problem (TSP) can be applied; the efficient algorithm proposed by Lin and Kernighan [44] can be used for generating the TSP tour.

Step 9: Split the TSP tour by the method proposed by Prins [45] without violating the vehicle capacity and working time constraints.

Step 10: Output the solution.

Based on the solution generation heuristic, the diverse population generation procedure is designed as follows:

Step 1: Randomly generate a non-redundant pickup point combination set N_{SP} with n_{sp} combinations, letting $P_0 = \varnothing$.

Step 2: Conduct the solution generation algorithm to generate solution S.

Step 3: If the selected pickup point in solution S already exists in P_0, turn to step 1; otherwise, put S into P_0.

Step 4: If the initial population size M_0 is reached, turn to step 5; otherwise, turn to step 1.

Step 5: Terminate and output the initial population.

As can be seen by taking the solution and population initialization methods, solutions are different for selected pickup points and service options, each of which stands for a unique solution space, which will cover the whole solution space to the largest extent.

4.3. Fitness Function

To compare two individual factors, the evaluation function F is defined as the sum of two components which are the objective function value and penalty cost, the evaluation function is formulated as follows:

$$F = F_0 + \alpha_1 \cdot \Delta S_D + \alpha_2 \cdot \Delta B_p + \alpha_3 \cdot \Delta n_v + \alpha_4 \cdot \Delta Q_v + \alpha_5 \cdot \Delta T_v \tag{20}$$

where, F_0 is the objective value of a solution; ΔS_D, ΔB_P, Δn_v, ΔQ_v and ΔT_v stand for the amount that are beyond the constraints of the depot capacity, pickup point capacity, available vehicles, vehicle capacity and working time respectively; α_1, α_2, α_3, α_4 and α_5 are the corresponding constant parameters that reflect the degree of the penalty.

4.4. Individual Evolution and Selection

In population-based heuristics, one of the greatest advantages is search parallel in wide solution space, rendering population diversity an important factor that affects the performance. Current widely used objective function-based individual evolution may shrink the search space and lead to premature convergence. In this study, we revise the biased fitness function proposed by Vidal et al. [46]. In this method, the objective value of a solution and its contribution to population diversity are jointly considered.

For an individual I, letting its n_{close} closest neighbors be stored in set N_{close}, the diversity contribution of individual I to the population in N_{close} is defined as the average distance to the individuals in N_{close} computed by Formula (21). A normalized Hamming distance based on pickup points' status, customers' service options and the assignments of two individuals is adopted to show the difference between solutions. The distance is computed according to Formula (22).

$$\Delta(I) = \frac{1}{n_{close}} \sum_{I_1 \in N_{close}} D^H(I, I_1) \tag{21}$$

$$\begin{aligned} D^H(I, I_1) = \quad & \gamma_1 \tfrac{1}{n_c} \sum_{i \le n_c} \left(1(sc_i(I) \neq sc_i(I_1))\right) + \gamma_2 \tfrac{1}{n_p} \sum_{i \le n_s} \left(1(sp_i(I) \neq sp_i(I_1))\right) \\ & + \gamma_3 \tfrac{1}{n_c} \sum_{i \le n_c} \left(1(pc_i(I) \neq pc_i(I_1))\right) \end{aligned} \tag{22}$$

where, $sc_i(I)$ stands for the assignment of customers with HD service, $sp_i(I)$ stands for the assignment of pickup facilities and $pc_i(I)$ stands for the assignment of customers with CP service. γ_1, γ_2 and γ_3 are the coefficients of the three factors, $\gamma_1 + \gamma_2 + \gamma_3 = 1$.

Let $r(I)$ be the rank of individual I in a subpopulation with respect to F and the biased fitness function is given by Formula (23).

$$F_E = r(I) - \Delta(I) \tag{23}$$

It can be seen that, for each subpopulation with n_{close} customers, when solutions are selected by elitist strategy, excellent solutions can be protected as well as the diversity achieved.

4.5. Crossover Operation

Crossover operation aims to generate offspring by mimicking mating between parents and exchanging gene information in the hope that children will obtain the best parts of their parents, as well as diversify the search. Due to the fact that the genes and length of chromosomes vary in solutions, a modified two-point crossover is introduced, which is depicted in Figure 4. This crossover operation is conducted by exchanging the middle part of the second tier of the first section of the selected two parents between the two selected crossover positions. The process is described as follows:

Step 1: Randomly select a pair of parents $P1$ and $P2$ by probability p_c from the population.

Step 2: Uniformly select two positions from [1, n_c], letting the smaller one be $cp1$ and the other $cp2$.

Step 3: For each gene in position [$cp1$, $cp2$], exchange the middle part of the second tier of the first part of the two chromosomes.

Step 4: Adjust the second sections of the two chromosomes according to the mapping relationship.

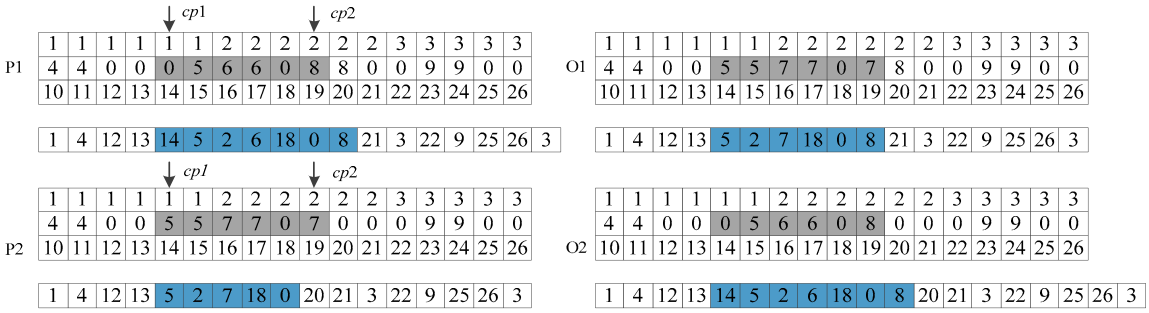

Figure 4. Crossover operation.

By taking this crossover operation, new solutions are generated by exploring new statuses of pickup points, customers' service options as well as the routes.

4.6. Mutation Operation

Mutation operation takes place immediately after the crossover operation, which may help GA jump out of the local optima by finding uncovered solution spaces [47]. In our research, we take gene-based mutation, which is conducted on the second section of a chromosome, in the following process:

Step 1: Select a solution $P3$ according to probability p_m.

Step 2: Randomly select a mutation position $cp3$ from 1 to $L(P3)$, where $L(P3)$ is the length of the second section of the chromosome.

Step 3: Identify the type of the selected gene; conduct the following two cases equally.

Case 1: If it is a pickup point, each of the three operators is selected randomly:

Operator 1: Close this pickup point.

Operator 2: Exchange the assigned customers with the closest pickup point if it is open.

Operator 3: Close it and open a randomly selected one which is not in the current solution.

Case 2: If the gene is a customer or zero, remove it and insert it into a randomly selected position between 1 and $L(P3)$.

It can be seen that, in this crossover, the operation can be performed on all search spaces by changing the status of pickup points, customers' service options as well as the routes.

4.7. Local Search

Local search aims to improve the new generated solutions during the evolution process within a few modifications. As the neighborhood defines the way to produce a new solution, the most important issue when designing a local search is to identify effective neighborhood structure [40]. For the customers' service options and vehicle routing in our problem, we design two kinds of neighborhood structures for service option and vehicle routing respectively.

For the service option neighborhood structure, two moves are included: they are HD to CP and CP to HD move, namely N1 and N2. More precisely, for N1, we randomly select a customer served by HD and identify its closest pickup point. If the pickup point is already selected, let the customer be served by it; otherwise, open the pickup point and serve the customer. For N2, we randomly select a pickup point and release one randomly selected customer and let him or her be served by HD.

When it comes to routing neighborhood structure, we take the nine route neighborhood moves proposed by Prins [45] which can comprehensively search the routing space.

Let $T(u)$ stand for the trip visiting vertex u, x and y which are the successor of u and v in their respective trips. Define the neighborhood of vertex u as the hn_c closest vertices, where $h \in [0,1]$ is a granular threshold restricting the search to nearby vertices [48]. The following moves are concluded:

N3: If u is a customer or pickup point, remove u then insert it after v;

N4: If u is a customer or pickup point and x is a customer or dummy zero, remove them then insert (u, x) after v;

N5: If u is a customer or pickup point and x is a customer or dummy zero, remove them then insert (x, u) after v;

N6: If u and v are customers or pickup points, swap u and v;

N7: If u and v are customers or pickup points, and x is a customer or dummy zero, swap (u, x) and v;

N8: If u and v are customers, x and y are customers of dummy zeros, swap (u, x) and (v, y);

N9: If $T(u) = T(v)$, replace (u, x) and (v, y) by (u, v) and (x, y);

N10: If $T(u) = T(v)$, replace (u, x) and (v, y) by (u, v) and (x, y);

N11: If $T(u) \neq T(v)$, replace (u, x) and (v, y) by (u, y) and (x, v).

N3 to N5 correspond to insertions, whereas N6 to N8 are swaps. The aforementioned six moves can be worked on both the intra-route and inter-route. N9 is an intra-route 2-opt, while N10 and N11 are inter-route 2-opt. It can be seen that, compared to the initial moves in Prins [45], zeros are also used to formulate new neighborhood solutions, which may help to explore the search space.

When applying LS procedure, the above neighborhood moves are selected randomly to generate a new solution. The procedure terminates when either an improved solution is obtained or all neighborhood moves are checked without finding a better solution. The structure of LS can be seen in Figure 2.

4.8. Computational Complexity Analysis

Letting the scale of the considered problem be $n_d \cdot n_p \cdot n_c$, I_{max} is the maximum iteration. In the presented algorithm, the computational complexity for population initialization is $O(n_{sp}({n_p}^2 + n_p n_c + n_d n_c + {n_c}^2))$. In each iteration, the computational complexity for decoding, crossover operation, mutation, evaluation and local search are $O(2n_s + n_c)$, $O(2n_p(2n_s + n_c))$, $O(2n_s + n_c)$, $O(M(n_s n_c + n_s n_p + n_p n_c))$ and $O(n_p \cdot (2n_s + n_c)^2)$, respectively. The computational complexity of the whole algorithm can be calculated as:

$$\begin{aligned} O(n_{sp}, M, n_d, n_p, n_c, I_{\max}) &= O(n_{sp}({n_p}^2 + n_p n_c + n_d n_c + {n_c}^2)) + O(I_{\max}((M+4)(2n_s + n_c) \\ &\quad +2n_p(2n_s + n_c) + M(n_s n_c + n_s n_p + n_p n_c) + 4n_p \cdot (2n_s + n_c)^2) \\ &\approx O(4I_{\max} n_p (2n_s + n_c)^2) \end{aligned} \tag{24}$$

As can be seen from Formula (24), the presented method belongs to polynomial algorithm and can be solved in polynomial time. More specifically, the computational complexity of the algorithm increases with the iterations and is proportional to n_p and $(2n_s+n_c)^2$.

5. Computational Experiments

In this section, we perform numerical experiments to evaluate the overall performance of the proposed model and method. We first provide the implementation details of HGALS for the parameters setting. As the proposed problem belongs to a variant of LRP, the performance of the HGALS in solving LRP is then evaluated based on benchmark instances. Subsequently, a real-world instance is taken to analyze the effects of simultaneous HD and CP on the Last Mile distribution system. Finally, the validity of the presented method in solving the proposed problem is verified based on generated instances. All algorithms are coded in C language and compiled on Visual Studio 2013, implemented on a PC with an Intel Core Duo CPU at 1.74 GHz and 4GB memory under Window XP system.

5.1. Parameters Setting

To find suitable parameters, extensive experiments are conducted based on the generated instances depicted in Section 5.4. For GA-based heuristics, the most important parameter is the population size; with bigger M_0, excellent solutions will be improved with much smaller probability, and it will take longer to find better solutions. Conversely, the population cannot sample the solution space which may lead to premature convergence. During the tests, since we found that the suitable initial population is closely related to n_p and variable population strategy helps a lot, it was finally decided as $M_0 = 4n_p$ and $M = 2M_0$. According to the decided M_0 and M, p_c, p_m, p_{ls} and p_{ac} are selected to be 0.45, 0.25, 1.0 and 0.25 respectively. Other parameters: $h = 0.2$, $\alpha_1 = 1000$, $\alpha_2 = 200$, $\alpha_3 = 100$, $\alpha_4 = \alpha_5 = 10$, $\gamma_1 = \gamma_3 = 0.3$, $\gamma_2 = 0.4$.

5.2. Tests Based on Benchmark Instance

To show that the presented method is suitable to solve LRP, the benchmark instances proposed by Prins et al. [49] are used, which can be found at http://prodhonc.free.fr/Instances/instances_us.htm. The designed operators for pickup points are performed on the depots both for the solution initialization and mutation. For the terminate condition, we take the maximum iterations without improving the best solution. The specific value depends on instance size: it is 200 for small-sized instances ($n \leq 50$), 400 for mid-sized instances ($50 < n \leq 100$) and 600 for large-sized instances ($n > 100$). During the tests, each instance was independently run 10 times; the results obtained by the presented heuristic are provided in Table 2. For the notions used in the table, LB stands for the best known lower bound, BKR stands for the best known result, CPU stands for the average running time in seconds and Gap/BKR stands for the gap to BKR.

From Table 2, we can see that all the instances are solved with an overall average runtime of 277.2 s. Average gap to the best-known solution is 0.47%. For all instances containing 50 or less customers, the presented method is successful at finding almost all the optimal solutions. Moreover, for the instances with 100 and 200 customers, the average gap to the best-known solution is not changed dramatically.

Table 2. Results for the benchmark instances.

	Instance	LB	BKR	CPU	Average		Best	
					Result	Gap/BKR	Result	Gap/BKR
1	20-5-1	54,793	54,793	1.8	54,836	0.08	**54,793**	**0**
2	20-5-1b	39,104	39,104	2.2	39,104	0	**39,104**	**0**
3	20-5-2a	48,908	48,908	1.5	48,908	0	**48,908**	**0**
4	20-5-2b	37,542	37,542	2.6	37,542	0	**37,542**	**0**
5	50-5-1	90,111	90,111	12.9	90,130	0.03	**90,111**	**0**
6	50-5-1b	67,340	63,242	17.8	63,242	0	**63,242**	**0**
7	50-5-2	88,298	88,298	24.3	88,643	0.40	88,643	0.39
8	50-5-2	67,340	67,340	21.4	67,340	0	**67,340**	**0**
9	50-5-2bis	84,055	84,055	20.8	84,139	0.10	**84,055**	**0**
10	50-5-2bbis	51,822	51,822	16.2	51,958	0.26	**51,822**	**0**
11	50-5-3	86,203	86,203	19.3	86,456	0.29	86,456	0.29
12	50-5-3b	61,830	61,830	18.1	61,830	0	**61,830**	**0**
	Avg			13.2		0.10		0.06
13	100-5-1	275,993	274,814	92.1	277,135	0.84	277,035	0.81
14	100-5-1b	214,392	213,615	122.4	2,145,034	0.42	214,313	0.33
15	100-5-2	194,598	193,671	91.8	194,366	0.36	194,124	0.23
16	100-5-2b	157,173	157,095	132.5	157,560	0.3	**157,095**	**0**
17	100-5-3	200,246	200,079	124.2	201,844	0.88	201,628	0.77
18	100-5-3b	152,586	152,441	107.7	154,484	1.34	152,992	0.36
	Avg			111.8		0.69		0.42
19	100-10-1	290,429	287,695	113.3	291,339	1.27	290,243	0.89

Table 2. *Cont.*

Instance		LB	BKR	CPU	Average		Best	
					Result	Gap/BKR	Result	Gap/BKR
20	100-10-1b	234,641	230,989	127.5	235,057	1.76	233,512	1.09
21	100-10-2	244,265	243,590	95.6	249,180	2.29	245,123	0.63
22	100-10-2b	203,988	203,988	130.9	205,511	0.74	204,667	0.33
23	100-10-3	253,344	250,882	133.5	257,059	2.46	253,865	1.19
24	100-10-3b	204,597	204,317	122.7	205,449	0.55	205,232	0.48
Avg				120.6		1.51		0.76
25	200-10-1	479,425	475,294	877.1	484,748	1.99	479,926	0.97
26	200-10-1b	378,773	377,043	922.3	382,595	1.47	380,613	0.94
27	200-10-2	450,468	449,006	823.6	451,835	0.63	450,312	0.29
28	200-10-2b	374,435	374,280	789.4	380,667	1.70	375,674	0.37
29	200-10-3	472,898	469,433	865.7	476,230	1.45	473,875	0.94
30	200-10-3b	364,178	362,653	900.5	364,762	0.58	364,201	0.43
Avg				863.1		1.30		0.66
Global Avg				277.2		0.77		0.47

Note: The best known solution values attained are marked in bold.

Moreover, considering the most recent and effective methods in the literature, the performance comparison is provided in Table 3 based on the CPU and Best Gap/BKR. The compared algorithms are GRASP in Prins et al. [49], MAPM in Prins et al. [39], LRGTS in Prins et al. [50], GRASP+ELS in Duhamel et al. [36], SALRP in Vincent et al. [35], ALNS in reference [51], MACO in [32], GRASP+ILP in Contardo et al. [52], GVTNS in Escobar et al. [38] and Hybrid GA in Lopes et al. [43].

Table 3. Comparison results of the methods.

Algorithm	Performance		Algorithm	Performance	
	CPU	Best Gap/BKR		CPU	Best Gap/BKR
GRASP	96.5	3.57	MACO	176.4	0.40
MAPM	76.7	1.35	GRASP+ILP	1163.0	0.12
LRGTS	17.5	0.70	GVTNS	91.2	0.37
GRASP+ELS	258.2	1.11	Hybrid GA	199.1	0.32
SALRP	422.4	0.46	HGALS	277.2	0.47
ALNS	4221.0	0.27			

Among those compared algorithms, GRASP+ILP is on average the most effective on this set of benchmark instances followed by ALNS, GVTNS, MACO and SALRP. Our algorithm obtains similar results with SALRP and performs much better than GRASP, MAPM, LRGTS and GRASP+ELS. As CPU times depend on many factors (such as computers and programming languages used) [43], the proposed method seems faster than SALRP, ALNS and GRASP+ILP.

The computational results show that our algorithm can effectively solve the instances and is applicable to LRP.

5.3. Real-World Instance

This section presents the test based on a real-world Last Mile distribution system of Shapingba District in Chongqing city. The network consists of three depots (numbered from 1 to 3), 47 candidate pickup points (with number from 4 to 50) and 136 customers (with numbers from 51 to 186). The locations of these nodes are shown in Figure 5. Other main parameters are given as follows: $D_{pc} = 600$ m, $u_H = 1.5$ min, $u_P = 10$ min, $p_H = 10\%$, $\mu_{sd} = 1.5$. For the algorithm terminate conditions, to get a better solution, we take the maximum iterations I_{max} and T_{max} into consideration simultaneously; three terminate conditions are considered according to the instance scales: $(I_{max}, T_{max}) = (1 \times 10^4, 600\text{ s}), (2 \times 10^4, 900\text{ s}), (3 \times 10^4, 1200\text{ s})$. After running the algorithm, the obtained pickup point locations and customer assignments are reported in Table 4, and the vehicle routing information is given in Table 5.

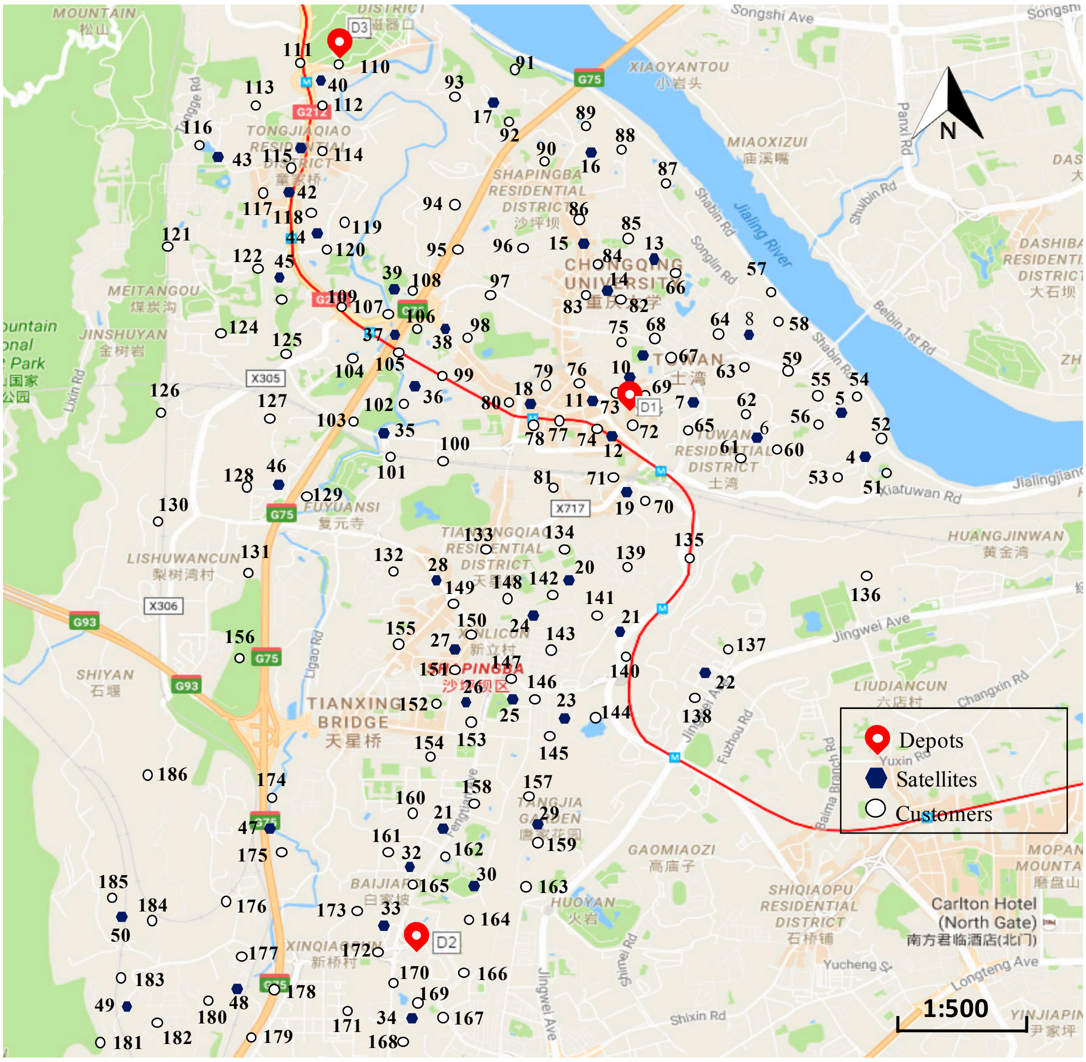

Figure 5. Network of the instance.

Table 4. Results of the instance—pickup point locations and customer assignments.

Pickup Point	Customer	Pickup Point	Customer	Pickup Point	Customer
5	54, 55, 56	6	60, 61, 62, 65	8	58, 59, 63, 64
10	69, 75, 83	11	73, 74, 76,77	12	71,72
15	85,86	17	91, 92, 93	22	137, 138
23	144, 145	24	141, 142, 148	25	146,147
26	151, 152, 153, 154	28	132, 149	29	157, 159
31	158, 160, 161, 162, 165	33	172, 173	34	167, 168, 170
35	101, 102, 103	37	104, 105, 106, 107	38	97, 98
39	95, 108	40	110, 111, 112	41	114, 115
44	119, 120	45	122, 123	46	128, 129
47	174, 175	48	177, 178, 179, 180	49	181, 812, 183

Table 5. Results of the instance—vehicle routing.

No.	Routing	No.	Routing
1	1-11-12-10-67-68-1	7	2-156-26-25-143-140-136-22-23-2
2	1-5-51-53-6-1	8	2-31-29-163-164-166-169-34-171-2
3	1-8-57-66-87-84-82-1	9	2-47-176-186-185-184-49-48-2
4	1-70-135-139-24-150-28-133-134-81-1	10	3-40-113-116-121-124-45-117-41-3
5	1-15-88-89-90-96-38-37-99-79-1	11	3-118-44-99-125-109-39-94-17-3
6	1-80-35-127-126-130-156-131-46-100-78-1		

To show the advantages of the proposed model, we now report the comparison of the proposed model with the scenario of only HD service, which can be seen in Table 6.

Table 6. Comparison of the two scenarios.

Item	Simultaneous HD and CP	Only HD	Difference
Vehicle number	11	26	−15
Pickup point number	30	-	+30
Fixed vehicle cost	2200	5200	−3000
Routing cost	156	233	−77
Pickup point opening cost	2370	-	+2370
Second delivery cost	186	529	−343
Total cost	4912	5962	−1050

From the comparison, it can be clearly seen that although added pickup points cost by providing CP service, the total cost is reduced by 1050 and accounts for 17.6%. When analyzing the reason, we find that the vehicles are largely reduced by providing HD and CP services simultaneously, which saved the fixed vehicle cost and routing cost accordingly. Besides, the increased customers served by CP reduced the second delivery cost. These two factors work together, reducing the cost to a large extent.

In this research, we take the routing cost c_{ij} by d_{ij} multiplying the unit routing cost. From the comparison of the total routing cost of the two scenarios, we can deduce that the travelling distance is reduced by 33%. As the carbon emission is closely related to the travelling distance, the proposed model can help to reduce the carbon emission effectively.

As CP service provides an important role in reducing the operation cost, to have more insights, we conduct a sensitive analysis to show how the acceptable distance D_{pc} influences the cost. When D_{pc} ranges from 100 to 800 with step 100, the results are reported in Table 7 and Figure 6. It is worth mentioning that the scenario when D_{pc} = 0 is with only HD service.

Table 7. Results of sensitive analysis of D_{pc} on cost.

Item	D_{pc}							
	100	200	300	400	500	600	700	800
Vehicle number	26	21	13	11	11	11	11	11
Pickup point number	-	16	36	33	30	30	30	30
Fixed vehicle cost	5200	4000	2600	2200	2200	2200	2200	2200
Routing cost	233	225	109	157	152	156	156	156
Pickup point opening cost	-	1120	2640	2480	2400	2370	2370	2370
Second delivery cost	529	417	140	188	209	186	186	186
Total cost	5962	5762	5489	5025	4961	4912	4912	4912

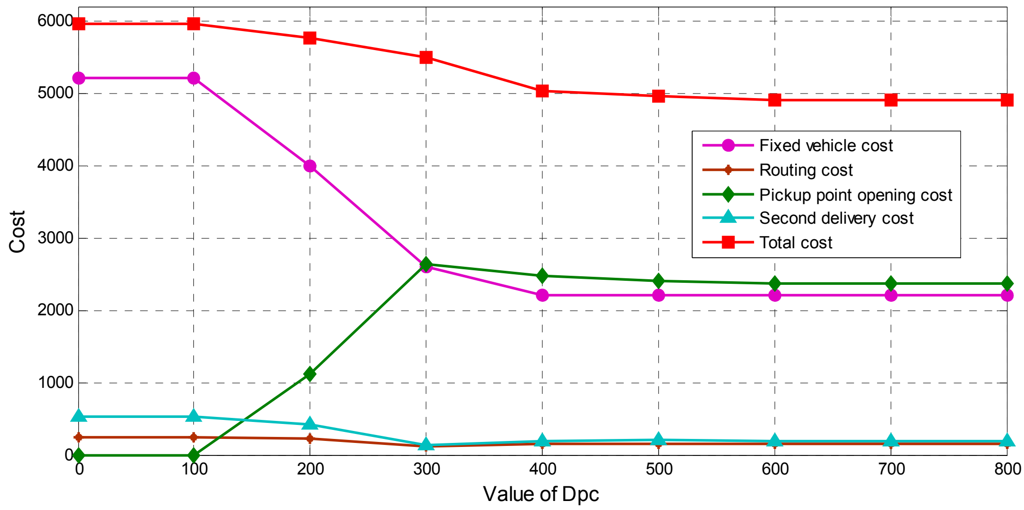

Figure 6. Sensitive analysis of D_{pc} on cost.

From the sensitive analysis, we can see that when $D_{pc} \leq 100$ m, the results obtained are the same with the only-HD scenario, which indicates that when $D_{pc} \leq 100$ m, CP service shows no advantages when compared to HD and there is no need to provide CP service. When $100 < D_{pc} \leq 600$ m, there comes a continuous fall of the total cost, which means that with the distance increasing, better solutions are obtained due to the provided CP service. After that, the results stay unchanged which is constrained by the closest customers and the capacity of pickup points. From a cost optimization perspective, $D_{pc} = 600$ m is the most acceptable distance when designing the Last Mile distribution system in this instance.

Thus, we can draw the conclusion that, within appropriate acceptable distance D_{pc}, simultaneous HD and CP are effective at reducing the cost within a wide range of D_{pc}, and a suitable distance exists for each Last Mile distribution system.

5.4. Algorithm Components Performance Analysis

In this section, a computational study is carried out to compare our approach with standard GA&LS (SGALS) to show the effectiveness of the designed algorithm components in solving this problem. In the SGALS, random solution generation, objective function-based individual evaluation and classical single point crossover and mutation on the second tier of the chromosome are considered.

We first generate a set of different scales of instances based on the real-world instance parameters. For the instances, the number of depots, pickup points and customers range from 2 to 12, 10 to 30 and 50 to 200, respectively. The instances are presented as "I<depot number>-<pickup point number>-<customer number>". The instances are generated as follows: depots and pickup points are randomly decided within the area with a radius of 15 km to reflect the relationship between the customers and pickup points; customers are randomly assigned within a radius of 1 km of selected pickup points; the capacity of depots as well as the vehicles are assigned with the requirement that all customers could be served; the capacity of pickup points are random value in [100, 120, 150, 180, 200] and the opening costs are [60, 80, 100, 120, 150] accordingly; customers' demands are random value in [15, 20, 25, 30, 35]. Other parameters are the same to the real-world instance in Section 5.3. The two algorithms are run ten times independently for each instance; the results are reported in Table 8.

Table 8. Comparison results of the two algorithms.

Instance	SGALS					HGALS				
	CPU	Average		Best		CPU	Average		Best	
		Result	Gap	Result	Gap		Result	Gap	Result	Gap
I2-10-50	272.9	1833.2	1.69	1806.2	0.19	143.2	1804.5	0.99	**1802.7**	**0**
I2-15-60	533.7	2473.2	4.56	2382.2	0.71	413.8	2388.6	0.98	**2365.3**	**0**
I2-18-80	600.1	3315.7	6.37	**3117.1**	**0**	435.6	3202.7	2.75	**3117.1**	**0**
I2-20-100	900	4404	14.46	4098.6	6.5	795.1	3958.6	2.88	**3847.8**	**0**
I2-22-120	862.3	4471.4	6.81	4220.7	0.82	852.4	4276.1	2.15	**4186.3**	**0**
I2-25-150	900	4967.8	6.13	4787.1	2.27	900	4769.3	1.89	**4680.8**	**0**
I2-28-180	1183	7166	9.21	7074.4	7.82	1200	6710.6	2.28	**6561.3**	**0**
I2-30-200	1200	6748.7	6.74	6543	3.49	1200	6426.9	1.65	**6322.6**	**0**
Agv	807		7.0		2.73	742		1.83		0

Note: The best known solution values attained are marked in bold.

From Table 8 we can observe that our approach can always obtain the best solutions and run faster than SGALS. The results clearly indicate that the proposed components which constitute the algorithm are necessary and useful to enhance the performance both in the search ability and the stability when comparing the results obtained by SGALS. This can be seen by the improvement of 1.83% in Best Gap and 4.27% (7%–2.73%) in Average Gap. We can also draw the conclusion that the LRPSHC is very complex since the SGASA can obtain the same best solution as HGALS only for instance I2-18-80.

6. Conclusions

This paper proposed a novel location-routing problem for the Last Mile distribution for online shopping, in which two kinds of service options, home delivery and customer's pickup, are simultaneously provided. For solving this specific problem, a hybrid heuristic combining GA and LS is presented together with several modified algorithm components.

Computational tests and comparative analysis with other published approaches based on benchmark instances are first conducted to evaluate the efficiency of the algorithm; results show that the presented algorithm framework is suitable to solve the LRP. Then, a real-world instance is adopted to verify the proposed model. By comparing the HD-only service scenario, the proposed model shows its great advantages in reducing the operation cost and vehicles, which proved that the model can help to design an economical and environmental Last Mile distribution system. A sensitive analysis is also made to give more insights into the effects of the acceptable distance for a customer to choose CP service based on the cost. Results show that the cost could be optimized by a wide scope of customer's acceptable distances and an optimal distance for a certain distribution system. Finally, based on the generated instances, the designed algorithm components are proved to be effective in improving the search ability and stability of the algorithm when comparing with standard GA&LS.

7. Future Research Directions

In addition to delivering packages from the depots to customers, collecting packages from customers is another challenge for the Last Mile operation. Future research will focus on location-routing problem with simultaneous home delivery and customer's pickup, as well as simultaneous delivery and collection, which may provide an even more efficient Last Mile solution. As LRPSHC is a new variant of LRP, an exact algorithm would be needed to solve this problem. Furthermore, it is clear that LRPSHC can be applied in many retail industry cases and it is also much more complex than classical LRP. More advanced metaheuristics are urgently needed in order to solve large-scale real-world instances effectively.

Acknowledgments: We appreciate the anonymous referees and the editors for their remarkable comments. This study is supported by National Science and Technology Support Program of China under Grant No. 2015BAH46F01, the Chongqing City Key Science Program Project under Grant No. cstc2014yykfA40006, cstc2015yykfC60002.

Author Contributions: Lin Zhou and Xu Wang proposed this research and created its framework; Lin Ni contributed to amalgamating the research and gave many useful suggestions; Yun Lin helped to collect data and analysis; Lin Zhou completed the paper. All authors have read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. China's E-Commerce Report 2015. Available online: http://finance.ifeng.com/a/20160629/14541580_0.shtml (accessed on 29 June 2016). (In Chinese)
2. Gevaers, R.; Van de Voorde, E.; Vanelslander, T. Characteristics of innovations in last-mile logistics—Using best practices, case studies and making the link with green and sustainable logistics. In Proceedings of the European Transport Conference, Noordwijkerhout, The Netherlands, 5–7 October 2009.
3. Esper, T.L.; Jensen, T.D.; Turnipseed, F.L.; Burton, S. The last mile: An examination of effects of online retail delivery strategies on consumers. *J. Bus. Logist.* **2003**, *24*, 177–203. [CrossRef]
4. Aized, T.; Srai, J.S. Hierarchical modelling of Last Mile logistic distribution system. *Int. J. Adv. Manuf. Technol.* **2014**, *70*, 1053–1061. [CrossRef]
5. Wang, X.; Zhan, L.; Ruan, J.; Zhang, J. How to choose "Last Mile" delivery modes for E-fulfillment. *Math. Probl. Eng.* **2014**, *2014*, 417129. [CrossRef]
6. Agatz, N.A.; Fleischmann, M.; Van Nunen, J.A. E-fulfillment and multi-channel distribution—A review. *Eur. J. Oper. Res.* **2008**, *187*, 339–356. [CrossRef]
7. Gevaers, R.; Van de Voorde, E.; Vanelslander, T. Cost Modelling and Simulation of Last-mile Characteristics in an Innovative B2C Supply Chain Environment with Implications on Urban Areas and Cities. *Procedia Soc. Behav. Sci.* **2014**, *125*, 398–411. [CrossRef]
8. Boyer, K.K.; Prud'homme, A.M.; Chung, W. The last mile challenge: Evaluating the effects of customer density and delivery window patterns. *J. Bus. Logist.* **2009**, *30*, 185–201. [CrossRef]
9. Hayel, Y.; Quadri, D.; Jiménez, T.; Brotcorne, L. Decentralized optimization of last-mile delivery services with non-cooperative bounded rational customers. *Ann. Oper. Res.* **2016**, *239*, 451–469. [CrossRef]
10. Lopes, R.B.; Ferreira, C.; Santos, B.S.; Barreto, S. A taxonomical analysis, current methods and objectives on location-routing problems. *Int. Trans. Oper. Res.* **2013**, *20*, 795–822. [CrossRef]
11. Prodhon, C.; Prins, C. A survey of recent research on location-routing problems. *Eur. J. Oper. Res.* **2014**, *238*, 1–17. [CrossRef]
12. Drexl, M.; Schneider, M. A survey of variants and extensions of the location-routing problem. *Eur. J. Oper. Res.* **2015**, *241*, 283–308. [CrossRef]
13. Vincent, F.Y.; Lin, S.-Y. A simulated annealing heuristic for the open location-routing problem. *Comput. Oper. Res.* **2015**, *62*, 184–196.
14. Ponboon, S.; Qureshi, A.G.; Taniguchi, E. Branch-and-price algorithm for the location-routing problem with time windows. *Transp. Res. E Logist. Transp. Rev.* **2016**, *86*, 1–19. [CrossRef]
15. Govindan, K.; Jafarian, A.; Khodaverdi, R.; Devika, K. Two-echelon multiple-vehicle location–routing problem with time windows for optimization of sustainable supply chain network of perishable food. *Int. J. Prod. Econ.* **2014**, *152*, 9–28. [CrossRef]
16. Wang, H.; Du, L.; Ma, S. Multi-objective open location-routing model with split delivery for optimized relief distribution in post-earthquake. *Transp. Res. E Logist. Transp. Rev.* **2014**, *69*, 160–179. [CrossRef]
17. Stenger, A.; Schneider, M.; Schwind, M.; Vigo, D. Location routing for small package shippers with subcontracting options. *Int. J. Prod. Econ.* **2012**, *140*, 702–712. [CrossRef]
18. Karaoglan, I.; Altiparmak, F.; Kara, I.; Dengiz, B. The location-routing problem with simultaneous pickup and delivery: Formulations and a heuristic approach. *Omega* **2012**, *40*, 465–477. [CrossRef]
19. Rieck, J.; Ehrenberg, C.; Zimmermann, J. Many-to-many location-routing with inter-hub transport and multi-commodity pickup-and-delivery. *Eur. J. Oper. Res.* **2014**, *236*, 863–878. [CrossRef]
20. Rahmani, Y.; Cherif-Khettaf, W.R.; Oulamara, A. A local search approach for the two–echelon multi-products location–routing problem with pickup and delivery. *IFAC-PapersOnLine* **2015**, *48*, 193–199. [CrossRef]

21. Vincent, F.Y.; Lin, S.-W. Multi-start simulated annealing heuristic for the location routing problem with simultaneous pickup and delivery. *Appl. Soft Comput.* **2014**, *24*, 284–290.
22. Rahmani, Y.; Ramdane Cherif-Khettaf, W.; Oulamara, A. The two-echelon multi-products location-routing problem with pickup and delivery: Formulation and heuristic approaches. *Int. J. Prod. Res.* **2015**, *54*, 1–21. [CrossRef]
23. Gianessi, P.; Alfandari, L.; Létocart, L.; Wolfler Calvo, R. The multicommodity-ring location routing problem. *Transp. Sci.* **2015**, *50*, 541–558. [CrossRef]
24. Prodhon, C.; Prins, C. A memetic algorithm with population management (MA | PM) for the periodic location-routing problem. In *Hybrid Metaheuristics*; Springer: Berlin, Germany, 2008; pp. 43–57.
25. Klibi, W.; Lasalle, F.; Martel, A.; Ichoua, S. The stochastic multiperiod location transportation problem. *Transp. Sci.* **2010**, *44*, 221–237. [CrossRef]
26. Guerrero, W.J.; Prodhon, C.; Velasco, N.; Amaya, C.A. Hybrid heuristic for the inventory location-routing problem with deterministic demand. *Int. J. Prod. Econ.* **2013**, *146*, 359–370. [CrossRef]
27. Rath, S.; Gutjahr, W.J. A math-heuristic for the warehouse location–routing problem in disaster relief. *Comput. Oper. Res.* **2014**, *42*, 25–39. [CrossRef]
28. Tang, J.; Ji, S.; Jiang, L. The Design of a sustainable location-routing-inventory model considering consumer environmental behavior. *Sustainability* **2016**, *8*, 211. [CrossRef]
29. Contardo, C.; Hemmelmayr, V.; Crainic, T.G. Lower and upper bounds for the two-echelon capacitated location-routing problem. *Comput. Oper. Res.* **2012**, *39*, 3185–3199. [CrossRef] [PubMed]
30. Nguyen, V.-P.; Prins, C.; Prodhon, C. Solving the two-echelon location routing problem by a GRASP reinforced by a learning process and path relinking. *Eur. J. Oper. Res.* **2012**, *216*, 113–126. [CrossRef]
31. Nguyen, V.-P.; Prins, C.; Prodhon, C. A multi-start iterated local search with tabu list and path relinking for the two-echelon location-routing problem. *Eng. Appl. Artif. Intell.* **2012**, *25*, 56–71. [CrossRef]
32. Ting, C.-J.; Chen, C.-H. A multiple ant colony optimization algorithm for the capacitated location routing problem. *Int. J. Prod. Econ.* **2013**, *141*, 34–44. [CrossRef]
33. Ardjmand, E.; Weckman, G.; Park, N.; Taherkhani, P.; Singh, M. Applying genetic algorithm to a new location and routing model of hazardous materials. *Int. J. Prod. Res.* **2015**, *53*, 916–928. [CrossRef]
34. Ardjmand, E.; Young, W.A.; Weckman, G.R.; Bajgiran, O.S.; Aminipour, B.; Park, N. Applying genetic algorithm to a new bi-objective stochastic model for transportation, location, and allocation of hazardous materials. *Expert Syst. Appl.* **2016**, *51*, 49–58. [CrossRef]
35. Vincent, F.Y.; Lin, S.-W.; Lee, W.; Ting, C.-J. A simulated annealing heuristic for the capacitated location routing problem. *Comput. Ind. Eng.* **2010**, *58*, 288–299.
36. Duhamel, C.; Lacomme, P.; Prins, C.; Prodhon, C. A GRASP×ELS approach for the capacitated location-routing problem. *Comput. Oper. Res.* **2010**, *37*, 1912–1923. [CrossRef]
37. Escobar, J.W.; Linfati, R.; Toth, P. A two-phase hybrid heuristic algorithm for the capacitated location-routing problem. *Comput. Oper. Res.* **2013**, *40*, 70–79. [CrossRef]
38. Escobar, J.W.; Linfati, R.; Baldoquin, M.G.; Toth, P. A Granular Variable Tabu Neighborhood Search for the capacitated location-routing problem. *Transp. Res. B Methodol.* **2014**, *67*, 344–356. [CrossRef]
39. Prins, C.; Prodhon, C.; Calvo, R.W. A memetic algorithm with population management (MA | PM) for the capacitated location-routing problem. In *Evolutionary Computation in Combinatorial Optimization*, Proceedings of the 6th European Conference, EvoCOP 2006, Budapest, Hungary, 10–12 April 2006; Springer: Berlin, Germany, 2006; pp. 183–194.
40. Derbel, H.; Jarboui, B.; Hanafi, S.; Chabchoub, H. Genetic algorithm with iterated local search for solving a location-routing problem. *Expert. Syst. Appl.* **2012**, *39*, 2865–2871. [CrossRef]
41. Montoya-Torres, J. R.; López Franco, J.; Nieto Isaza, S.; Felizzola Jiménez, H.; Herazo-Padilla, N. A literature review on the vehicle routing problem with multiple depots. *Comput. Ind. Eng.* **2015**, *79*, 115–129. [CrossRef]
42. Marinakis, Y.; Marinaki, M. A bilevel genetic algorithm for a real life location routing problem. *Int. J. Logist. Res. Appl.* **2008**, *11*, 49–65. [CrossRef]
43. Lopes, R.B.; Ferreira, C.; Santos, B.S. A simple and effective evolutionary algorithm for the capacitated location–routing problem. *Comput. Oper. Res.* **2016**, *70*, 155–162. [CrossRef]
44. Lin, S.; Kernighan, B.W. An effective heuristic algorithm for the traveling-salesman problem. *Oper. Res.* **1973**, *21*, 498–516. [CrossRef]

45. Prins, C. A simple and effective evolutionary algorithm for the vehicle routing problem. *Comput. Oper. Res.* **2004**, *31*, 1985–2002. [CrossRef]
46. Vidal, T.; Crainic, T.G.; Gendreau, M.; Lahrichi, N.; Rei, W. A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Oper. Res.* **2012**, *60*, 611–624. [CrossRef]
47. Ahmadizar, F.; Zeynivand, M.; Arkat, J. Two-level vehicle routing with cross-docking in a three-echelon supply chain: A genetic algorithm approach. *Appl. Math. Model.* **2015**, *39*, 7065–7081. [CrossRef]
48. Toth, P.; Vigo, D. The granular tabu search and its application to the vehicle-routing problem. *Inf. J. Comput.* **2003**, *15*, 333–346. [CrossRef]
49. Prins, C.; Prodhon, C.; Calvo, R.W. Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking. *4OR* **2006**, *4*, 221–238. [CrossRef]
50. Prins, C.; Prodhon, C.; Ruiz, A.; Soriano, P.; Wolfler Calvo, R. Solving the capacitated location-routing problem by a cooperative Lagrangean relaxation-granular tabu search heuristic. *Transp. Sci.* **2007**, *41*, 470–483. [CrossRef]
51. Hemmelmayr, V.C.; Cordeau, J.-F.; Crainic, T.G. An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Comput. Oper. Res.* **2012**, *39*, 3215–3228. [CrossRef] [PubMed]
52. Contardo, C.; Cordeau, J.-F.; Gendron, B. An exact algorithm based on cut-and-column generation for the capacitated location-routing problem. *INFORMS J. Comput.* **2013**, *26*, 88–102. [CrossRef]