



Article Hybrid Chaotic Discrete Bat Algorithm with Variable Neighborhood Search for Vehicle Routing Problem in Complex Supply Chain

Yuanhang Qi ^{1,2} and Yanguang Cai ^{2,*}

- School of Computer Science, University of Electronic Science and Technology of China, Zhongshan Institute, Zhongshan 528402, China; qiyuanhang@zsc.edu.cn
- ² School of Automation, Guangdong University of Technology, Guangzhou 510006, China
- * Correspondence: caiyg99@163.com

Abstract: Driven by the supply chain, suppliers, manufacturers and warehouses are working more closely together for improving service quality. However, tremendous cost may incur in the supply chain if transportation is not planned properly and efficiently, which frustrates enterprises in the intense market. In this paper, we present a model of vehicle routing problem in complex supply chain (VRPCSC) and propose an intelligent algorithm called hybrid chaotic discrete bat algorithm with variable neighborhood search for minimizing the purchase cost of materials, processing cost, and delivery cost along the path from suppliers, to manufacturers and warehouses in the vehicle routing problem. Based on the principles of bat algorithm, a discrete chaotic initialization strategy (DCIS) and a variable neighborhood search (VNS) are adopted to enhance the convergence capacity. Finally, two sets of experiments are conducted, which show that the proposed algorithm can solve the VRPCSC effectively.

Keywords: supply chain; transportation systems; artificial intelligence; intelligent algorithm; vehicle routing; bat algorithm

1. Introduction

With the development of big data, artificial intelligence, and high-performance computing techniques, the supply chain is reaching a new level. Based on these, more and more complex segments can be considered in the supply chain management to create lower costs and greater benefits. Nowadays, suppliers, manufacturers, and warehouses are working more closely together, driven by the supply chain, and they cooperate with each other to gain a share of the market. The enterprises have to effectively manage the transportation of supply chain, so as to meet the current stringent market demand at the lowest cost [1]. How to improve the efficiency of the supply chain has become a hot topic in supply chain management.

Supply chain cost is mainly composed of material costs, labor cost, transportation cost, equipment cost, and other variable cost [2]. Among them, transportation cost is an important part of supply chain cost. Research of Apte U.M indicated that 30% of product cost is used for vehicle routing [3]. A reasonable vehicle routing in the supply chain plays an important role in reducing supply chain costs [4]. Therefore, reducing transportation cost in the supply chain is one of the effective ways to improve the efficiency of the supply chain [5].

Cross Docking (CD) is one of the effective transportation strategies for supply chain management, which involves the transportation and store management of products: products are collected and transported to cross docks by inbound vehicles, and after integrating and sorting the products, the products are loaded and transported to customers by inbound vehicles [6]. CD can effectively improve the flowability of the products while reducing



Citation: Qi, Y.; Cai, Y. Hybrid Chaotic Discrete Bat Algorithm with Variable Neighborhood Search for Vehicle Routing Problem in Complex Supply Chain. *Appl. Sci.* 2021, *11*, 10101. https://doi.org/10.3390/ app112110101

Academic Editors: Jarosław Wikarek and Paola Pellegrini

Received: 15 August 2021 Accepted: 25 October 2021 Published: 28 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). the storage cost and the storage space [7]. Therefore, since CD was first proposed by Rohrer [8] in 1995, it has attracted significant research attention, and has also been successfully applied in the supply chain. Yu et al. [9] designed nine heuristic algorithms with different strategies to solve CD, whose objective is to minimize total operation time or to maximize the throughput of cross docks. The actual transportation capacity of cross docks is limited, which may cause product transportation delays. Therefore, considering the penalty costs for all the unfulfilled products, Lim et al. [10] propose a over-constrained vehicle routing problem with a time window, operational time, and capacity constraint for CD. Some special industries have their own special requirements for transportation and storage. For example, the food industry has strict cooling requirements and is highly intolerant of intermediate storages inside the terminal. In addressing this, Boysen [11] proposes a model for truck scheduling at zero-inventory cross docking terminals for the food industry. Considering multiple products and different product prices supplied by different suppliers, Ahmadizar et al. [12] established a model whose objective is to minimize the purchasing, transportation, and holding costs. Reverse logistics is an important part of modern logistics. Zuluaga et al. [13] analyzed the process of reverse logistics and proposed a reverse cross docking model. However, in the current research on CD, most researchers only focus on the integrated distribution process without considering the impact of manufacturers' processing.

The manufacturing process, as an important factor, is necessary to consider in supply chain management. Therefore, some supply chain researches considered the production, inventory, and transportation scheduling in the supply chain, which is called the production routing problem (PRP). Based on a supply chain with a single product, multiple factories, and multiple customers, Liu [14] establishes a model for the integrated PRP which considers the production capacity, inventory capacity, material supply contract, inventory holding cost, and production cost. Qiu et al. [15] propose a model for a pollution production routing problem (PPRP) with carbon cap-and-trade whose objective is to reduce emission levels of carbon dioxide. Based on the PPRP, considering the time window constraint, aiming to minimize fuel costs and operating costs, Kumar et al. [16] propose a multiobjective PPRP model with a time window. Considering the impact of carbon emissions in the transportation process and integrating the production, inventory, transportation, and distribution, Darvish et al. [17] established a three-echelon supply chain vehicle scheduling model and adopted accurate algorithms to solve it. However, the above papers only consider the transportation after the production process, and does not consider the transportation before the production process, that is, the process of suppliers providing materials to the factory. In addition, complex products in real life are often manufactured from multiple materials, and different materials may be provided by different suppliers, even different suppliers supply different materials with different costs.

Base on above, this paper takes into consideration the fact that complex products are made from multiple materials on the basis of the three-echelon supply chain, taking into account factors such as maximum material supply quantity, vehicle capacity, material costs, processing cost and distribution costs, and proposes a model of the vehicle routing problem in complex supply chain (VRPCSC) with an aim to minimize the purchasing, production, and transportation costs. The VRPCSC satisfies the transport constraints, which is an NP-hard problem. More and more scholars have been applying meta heuristic algorithms to deal with this kind of problem [18,19]. Based on our previous studies [20,21], the Discrete Bat Algorithm (DBA) can solve routing problems effectively. Because there are two-part routings in the VRPCSC (one is between supplier and manufacturers, the other is between manufacturers and the warehouse), this paper proposes a hybrid discrete bat algorithm optimized by the local search. In addition, this paper adopts a discrete chaotic initialization strategy (DCIS) [22] to strengthen the prophase convergence rate and overall convergence effect of the algorithm.

The contributions of this paper are as follows: (1) Based on a three-echelon supply chain included suppliers, manufacturers, and warehouses, considering the fact that com-

plex products with multiple materials are process by-manufacturers, a model of VRPCSC is proposed; (2) For the complex supply chain model, a hybrid chaotic discrete bat algorithm with a variable neighborhood search (denoded as HCDBAVNS) is presented to solve it.

The remainder of this paper is organized as follows: Section 2 presents the model of VRPCSC to describe the vehicle routing among suppliers, manufacturers, and warehouses. Section 3 proposes the HCDBAVNS algorithm and describes the implementation in detail. 20 test instances are chosen as the study case to demonstrate the proposed algorithm and its optimization strategies in Section 4. Some conclusions and future work are finally addressed in Section 5.

2. Problem Model

In this subsection, VRPCSC is described in detail. Based on the description, the mathematical model of VRPCSC is proposed.

2.1. Problem Description

In VRPCSC, the warehouses purchase materials from suppliers according to their respective demand for products. The suppliers deliver materials to manufacturers by vehicle. The manufacturers make materials into products, and then deliver the products to the warehouses by vehicle. The VRPCSC distribution process is shown in Figure 1, which consists of four parts, i.e., material procurement, material delivery, material processing, and product delivery. VRPCSC is based on the perspective of third-party logistics companies. Third-party logistics companies choose different suppliers and manufacturers to satisfy the demands of clients(warehouses), with the goal being to minimize the total costs. This study employs a parametric approach for modeling based on previous studies. The model is based on the following hypothesis:



Figure 1. The distribution schematic diagram of VRPCSC.

(1) Each supplier owns all varieties of materials needed for the products, but the maximum supply quantity and cost of each material vary from one supplier to another; (2) One supplier can provide a variety of materials to multiple manufacturers at the same time, but the total quantity of each material supplied cannot exceed its maximum supply quantity; (3) A certain material of the products needed by warehouses can only be provided by one supplier; (4) The supplier can deliver materials to one manufacturer at a time, the vehicle must return to the original supplier immediately after the delivery. In other word, there is a one round-trip path between a supplier and a manufacturer, but the material delivery cost varies from one supplier to another; (5) All materials of the products needed by one warehouse must be manufactured into products by just one manufacturer,

but different materials can be supplied from different suppliers and there is no competition between suppliers; (6) The products needed by warehouses can be processed by only one manufacturer, and delivered only by the vehicle of that manufacturer; (7) Different manufacturers have different processing costs, and they also have a different number of vehicles, vehicle models, and product delivery costs; (8) There is a linear relation between the total weight of the various materials before processing and the weight of products after processing; (9) The manufacturer's vehicle starts from the manufacturer's place, delivers the products to warehouses and returns to the original manufacturer's place after delivery; (10) The manufacturer's vehicle can deliver products only once. If it is used for delivery, the total weight of the products it delivers cannot exceed the maximum loading capacity of the vehicle.

2.2. Mathematical Model

In order to establish the mathematical model of VRPCSC more accurately, the mathematical notations are defined as follows:

(1) Sets

SS: Set of suppliers.

SM: Set of manufacturers.

SP: Set of warehouses.

SL: Set of material types.

SK: Set of manufacturer vehicles.

(2) Parameters

s: Element in set *SS*, $s \in SS$.

m: Element in set *SM*, $m \in SM$.

p: Element in set *SP*, $p \in SP$.

l: Element in set *SL*, $l \in SL$.

k: Element in set *SK*, $k \in SK$.

S: Number of elements in the *SS*.

M: Number of elements in the *SM*.

P: Number of elements in the *SP*.

L: Number of elements in the *SL*.

K: Number of elements in the *SK*.

*e*_{*sl*}: Maximum supply quantity of material *l* of supplier *s*.

 c_{sl} : Cost per unit weight of material *l* of supplier *s*.

b_s: Delivery cost per unit distance of materials delivered by vehicles of supplier *s*.

*n*_{*m*}: Number of vehicles owned by manufacturer *m*.

 c'_m : Processing cost per unit weight of products made from materials by manufacturer *m*.

 b'_{mk} : Delivery cost per unit distance of products delivered by vehicle k of manufacturer m.

 u_{mk} : Maximum loading capacity of vehicle k of manufacturer m.

 d_{sm} : Distance from supplier *s* to manufacturer *m*.

 d'_{mp} : Distance from manufacturer *m* to warehouse *p*.

 $d_{pp'}^{\prime\prime}$: Distance from warehouse *p* to warehouse *p'*.

o_p: Weight of products needed by warehouse *p*.

 o'_{pl} : Weight of material *l* corresponding to the products needed by warehouse *p*.

(3) Variables

 y_{mkp} : If warehouse *p* is the first warehouse distributed by vehicle *k* of manufacturer *m*, $y_{mkp} = 1$; otherwise, $y_{mkp} = 0$.

 $y'_{mkpp'}$: If vehicle *k* of manufacturer *m* directly delivers products to warehouse *p*' after delivery to warehouse *p*, $y'_{mkvv'} = 1$; otherwise, $y'_{mkvv'} = 0$.

 y''_{mkp} : If warehouse *p* is the last warehouse distributed by vehicle *k* of manufacturer *m*, $y''_{mkp} = 1$; otherwise, $y''_{mkp} = 0$.

 g_{smkpl} : If the products delivered by vehicle *k* of manufacturer *m* to warehouse *p* are made from material *l* of supplier *s*, $g_{smkpl} = 1$; otherwise, $g_{smkpl} = 0$.

 g'_{sm} : If supplier *s* uses the supplier's vehicle to deliver materials to manufacturer *m*, $g'_{sm} = 1$; otherwise, $g'_{sm} = 0$.

 g''_{mkp} : If vehicle *k* of manufacturer *m* is used to deliver products to warehouse *p*, $g''_{mkp} = 1$; otherwise, $g''_{mkp} = 0$.

2.2.1. Objective Function

According to the above description of VRPCSC, the whole logistics cost mainly consists of four parts. The first part is the purchase cost of materials (denoted as Z_{PUR}). The second part is the delivery cost of materials from suppliers to manufacturers (denoted as Z_{STM}). The third part is the processing cost for the manufacturers to process materials into products (denoted as Z_{PRO}). The fourth part is the delivery cost for the manufacturers to deliver the products to the warehouses (denoted as Z_{MTS}). Then, the calculation methods for each part and the optimization objective of this paper are described in detail.

The purchase cost of materials can be calculated by the weights and prices of the materials, and the calculation method is shown in Formula (1):

$$Z_{\text{PUR}} = \sum_{s \in SS} \sum_{m \in SM} \sum_{k \in SK} \sum_{p \in SP} \sum_{l \in SL} g_{smkpl} o'_{pl} c_{sl} \tag{1}$$

On the other hand, since there is a one round-trip path between a supplier and a manufacturer, then Z_{STM} can be calculated as follows:

$$Z_{\text{STM}} = 2 \times \sum_{s \in SS} \sum_{m \in SM} g'_{sm} d_{sm} b_s$$
⁽²⁾

It can be seen from Formula (2), that the unit delivery cost of the suppliers' vehicles and the delivery distances are related to Z_{STM} . Further, several varieties of materials will be processed into products by manufacturers. The processing cost is related to the materials' weights and processing prices, which is shown in Formula (3).

$$Z_{\text{PRO}} = \sum_{m \in SM} \sum_{k \in SK} \sum_{p \in SP} g''_{mkp} o_p c'_m \tag{3}$$

At last, the products can be delivered by the manufacturers' vehicles. Since multiple warehouses' products can be delivered by a vehicle of a manufacturer at the same time, then Z_{MTS} can be calculated by Formula (4).

$$Z_{\text{MTS}} = \sum_{m \in SM} \sum_{k \in SK} \sum_{p \in SP} (y_{mkp} + y''_{mkp}) d'_{mp} b'_{mk} + \sum_{m \in SM} \sum_{k \in SK} \sum_{p \in SP} \sum_{p' \in SP} y'_{mkpp'} d'_{pp'} b'_{mk}$$
(4)

From Formula (1) to Formula (4), it can be obtained that the objective function is:

$$Z_{\text{Total}} = \min(Z_{\text{PUR}} + Z_{\text{STM}} + Z_{\text{PRO}} + Z_{\text{MTS}})$$
(5)

The goal of this paper is to minimize the total cost of these four parts.

2.2.2. Constraints

After ensuring the objective of the VRPCSC, some constraints need to be confirmed in this model to make the VRPCSC fully satisfied with the problem description in Section 2.1. First, the relationship between products and materials satisfies linear constraints:

$$o_p = \frac{o'_{p1}}{\sigma_1} = \frac{o'_{p2}}{\sigma_2} = \dots = \frac{o'_{pL}}{\sigma_L}$$
(6)

where $\sigma_1, \sigma_2, \dots, \sigma_L$ are all known real number constants. According to the product demands of the warehouses, the demands for purchased materials can be calculated by Formula (6). However, in the process of purchasing materials, each material provided

by the supplier cannot exceed its maximum supply quantity. The constraint is shown in Formula (7):

$$\sum_{a \in SM} \sum_{p \in SP} g_{smkpl} o'_{pl} \le e_{sl}, \forall s \in SS, \forall k \in SK, \forall l \in SL$$
(7)

Further, in other to ensure that the supplier delivers the materials to the corresponding manufacturer at most by one vehicle, suppliers' vehicles should meet the following constraints:

n

$$g'_{sm} = \min\{1, \sum_{k \in SK} \sum_{p \in SP} \sum_{l \in SL} g_{smkpl}\}, \forall s \in SS, \forall m \in SM$$
(8)

When finished, the products will be delivered by the manufacturers' vehicles. In the same way, the weight of the products delivered by the manufacturers' vehicles cannot exceed their respective maximum loading capacities. The constraint is shown in Formula (9):

$$\sum_{p \in SP} g''_{mkp} o_p \le u_{mk}, \forall m \in SM, \forall l \in SL$$
(9)

In the process of delivering products to the warehouses, the manufacturers' vehicles should also meet the following constraints:

$$\sum_{m \in SM} \sum_{k \in SK} g''_{mkp} = 1, \forall p \in SP$$
(10)

$$\sum_{p \in SP} y_{mkp} \le 1, \forall m \in SM, \forall k \in SK$$
(11)

$$\sum_{p \in SP} \sum_{p' \in SP} y'_{mkpp'} \le \delta \sum_{p \in SP} y_{mkp'} \,\forall m \in SM, \forall k \in SK, \exists \delta \in R^+$$
(12)

$$\sum_{p \in SPT} \sum_{p' \in SPT} mkpp' \le SPT - \gamma(SPT), \forall SPT \in SP, SPT \neq \emptyset, \forall m \in SM, \forall k \in SK$$
(13)

where Formula (10) ensures that only one manufacturer delivers products to each warehouse; Formula (11) ensures that each manufacturer's vehicle can deliver products only once; Formula (12) ensures that the manufacturer's vehicle must start from the manufacturer's place to deliver products to the warehouse; Formula (13) ensures the continuous delivery process of one manufacturer's vehicle and $\gamma(SPT)$ denotes the minimum number of manufacturer's vehicles needed to serve all the warehouse in set *SPT*.

In addition, since the materials are related to the products, there are some constraints between the transportation of materials and corresponding products:

$$\sum_{s \in SS} \sum_{m \in SM} \sum_{k \in SK} \sum_{p \in SP} g_{smkpl} = 1, \forall l \in SL$$
(14)

$$\sum_{s \in SS} \sum_{l \in SL} g_{smkpl} = L \times g_{mkp}^{\prime\prime}, \forall m \in SM, \forall k \in SK, \forall p \in SP$$
(15)

where Formula (14) ensures that each material corresponding to the products needed by each warehouse is provided by only one supplier, processed by one manufacturer and delivered by one manufacturer's vehicle; Formula (15) indicates the relationship between material delivery and product delivery.

Above all, the VRPCSC model can be established from Formula (1) to Formula (15). Then the algorithm of this model will be solved in detail.

3. The Proposed HCDBAVNS Algorithm

BA is an intelligent evolutionary algorithm proposed by Yang X-S [23] that was inspired by the micro-bat echolocation principle. The BA simulates the process in nature where the bat searches for prey by echolocation, adjusts the wavelength of their emitted pulses and pulse emission rate depending on the proximity of the prey, and finally captures the prey, which is called a swarm optimization algorithm [24]. The BA mainly consists of four parts, i.e., bat position, bat velocity, bat frequency, and their corresponding update operations. Based on the characteristics of VRPCSC and our previous studies [20,21], this paper has proposed a HCDBAVNS algorithm to solve the model. The optimization framework of HCDBAVNS is shown in Figure 2.



Figure 2. Optimization Framework of HCDBAVNS.

It can be seen from Figure 2, that there is a series of bat operations in HCDBAVNS, such as the updated operations of bat position, velocity, frequency, variation, loudness, and pulse emission rate. Then, HCDBAVNS also includes DCIS and VNS. In addition, the local search strategies, that is, 0-1 search, 1-1 search, and 2-Opt search are the critical parts of VNS. In the following sections, each component of the HCDBAVNS will be described in detail.

3.1. Definition of Parameters and Operations

3.1.1. Bat Position, Velocity and Frequency

The position of the bat is a key bridge between the solution model and the solution space. VRPCSC is a three-echelon supply chain structure, which consists of two stages of logistics and transportation scheduling. Then, the position of bat *i* is defined as follows:

$$x_i = [x_i^1, x_i^2]$$
(16)

$$x_i^1 = [x_{i1}^1, x_{i2}^1, \cdots, x_{iW}^1]$$
(17)

$$x_i^2 = [x_{i1}^2, x_{i2}^2, \cdots, x_{iW'}^2]$$
(18)

where the dimensions of the first and second bat position are respectively W, W'. The detailed encoding and decoding methods for bat positions are described in Section 3.2.

On the other hand, BA is related to the bat velocity and frequency in the process of updating bat position. Similarly, this paper defines the velocity and frequency of bats as a two-layer structure to prepare for the following update operations of bats. Then, the velocity of bat *i* is:

$$v_i = [v_i^1, v_i^2]$$
(19)

$$v_i^1 = [v_{i1}^1, v_{i2}^1, \cdots, v_{iW}^1]$$
(20)

$$v_i^2 = [v_{i1}^2, v_{i2}^2, \cdots, v_{iW'}^2]$$
(21)

where, v_{ij}^1 ($j = 1, 2, \dots, W$) is an integer and $1 \le v_{ij}^1 \le W$; v_{ij}^2 ($j = 1, 2, \dots, W'$) is an integer and $1 \le v_{ij}^1 \le W'$. Further, the frequency of bat *i* is:

$$f_i = [f_i^1, f_i^2]$$
(22)

where, $f_i^1 \in [0, 1]$; $f_i^2 \in [0, 1]$.

3.1.2. Update Operations of Bat Position, Velocity and Frequency

The update operations of bat position, velocity and frequency mainly include four steps: Supposing that, the position of bat *i* is $x_i = [x_i^1, x_i^2]$, $v_i = [v_i^1, v_i^2]$, $f_i = [f_i^1, f_i^2]$, the optimal global position $x_* = [x_*^1, x_*^2]$, and the updated bat position, velocity, and frequency are x'_i, v''_i and f'_i , respectively.

(1) $x_i - x_* = v'_i = [v'_i, v'_i]$: For any $x^1_{ij} \in x^1_i, x^1_{*j} \in x^1_*, v'_{ij} \in v'_i$ and $j = 1, 2, \dots, W$, if $x^1_{ij} = x^1_{*j}$, then $v'_{ij} = 0$; if $x^1_{ij} \neq x^1_{*j}$, then $v'_{ij} = x^1_{*j}$. v'_i is obtained from x^2_i and x^2_* in the same way.

(2) $v'_i \times f_i = v''_i = [v''_i, v''_i]$: Let $f'_i = [f'_i, f'^2_i]$, f_r is randomly generated and $f_r \in [0, 1]$, frequency factor $\theta^1 > 1$ and $\theta^2 > 1$. For any $v'_{ij} \in v'^{1}_i, v''_{ij} \in v''_i$ and $j = 1, 2, \dots, W$, if $f_r < f^1_i$, then $v''_{ij} = 0$, $f'^{1}_i = f^1_i$; if $f_r \ge f^1_i$, then $v''_{ij} = v'^{1}_{ij}, f'^{1}_i = f^1_i + (f_r - f^1_i)/\theta^1$. v''^{2}_i and f^2_i are obtained from f_r and θ^2 in the same way.

(3) $v_i + v''_i = v'''_i = [v'''_i, v'''_i]$: For any $v_{ij}^1 \in v_i^1, v''_{ij}^{\prime\prime\prime 1} \in v''_i, v''_{ij}^{\prime\prime\prime 1} \in v''_i, v'''_{ij}^{\prime\prime\prime 1} \in v''_i$ and $j = 1, 2, \dots, W$, if rand() < 0.5, then $v''_{ij}^{\prime\prime\prime 1} = v_{ij}^1$; if $rand() \ge 0.5$, then $v''_{ij}^{\prime\prime\prime 1} = v''_{ij}^{\prime\prime}$. v'''_i is obtained from v_i^2 and v''_i^2 in the same way.

(4) $x_i + v_{i''}^{\prime\prime'} = x_i' = [x_i'^1, x_i'^2]$: Let $x_i'^1 = x_i^1$, when $v_{ij}^{\prime\prime\prime'} \neq 0$, the component of $x_i'^1$ at position x_{ij}^1 is exchanged with its component at position $v_{ij}^{\prime\prime\prime'}$ ($j = 1, 2, \dots, W$). $x_i'^2$ is obtained from x_i^2 and $x_i^{\prime\prime\prime'}$ in the same way. A instance is shown in Figure 3.



Figure 3. An instance for the update operations of bat position, velocity, and frequency.

In Figure 3, assuming that $x_i^1 = \{1, 2, 3, 4\}$ and $x_*^1 = \{1, 4, 3, 2\}$, then $x_i^1 - x_*^1 = v_i'^1 = \{0, 4, 3, 2\}$. Second, assuming that $f_i^1 = \{0.2, 0.3, 0.6, 0.2\}$ and $f_r = 0.4$, then $v_i'^1 \times f_i^1 = v_i''^1 = \{0, 4, 0, 2\}$. Third, assuming that $v_i^1 = \{1, 3, 3, 4\}$ and the random values is $\{0.6, 0.2, 0.6, 0.7\}$, then $v_i'^1 + v_i''^1 = v_i'''^1 = \{0, 3, 0, 2\}$. Based on the x_i^1 and $v_i'''^1$, $x_i + v_i'' = x_i' = \{1, 4, 2, 3\}$.

3.1.3. Bat Variation Operation

Supposing that, the position of bat *i* is $x_i = [x_i^1, x_i^2]$. Let *j* and *w* be randomly generated unequal integers $(1 \le j \le W, 1 \le w \le W)$, and exchange the *j*-th component and *w*-th component of x_i^1 . x_i^2 is obtained in the same way.

3.1.4. Update Operations of the Bat Loudness and Pulse Emission Rate

Supposing $R_i^0 \in [0, 1]$ and $A_i^t \in [0, 1]$ are the initial pulse emission rate and loudness of bat *i* at iteration *t*. Then, at iteration t + 1, its loudness $A_i^{t+1} = \alpha A_i^t$, and pulse emission rate $R_i^{t+1} = R_i^0 \times [1 - \exp(-\gamma t)]$, where the loudness factor $0 < \alpha < 1$ and pulse emission rate factor $\gamma > 0$.

3.2. Encoding and Decoding Schemes

The encoding and decoding schemes are important steps to solve the optimization problem, which affects the quality of the final solution. Finally, they are described in detail.

3.2.1. Encoding Scheme

(1) Encoding of materials needed for products: The products are made from a variety of materials, but the materials of the products needed by the same warehouse can be provided by different suppliers. Therefore, a unique mark must be given to the material corresponding to the products needed by each warehouse. Since the number of warehouses is *P* and the number of material types needed for products *L*, the material *l* corresponding to the products needed by warehouse *p* is coded as $p \times l$. The material codes corresponding to all products will form a sequence $[1, 2, \dots, P \times L]$;

(2) Encoding of the first-part bat position: Let the dimension of x_i^1 be $W = P \times L + S - 1$, which is a permutation of the sequence of integers $[1, 2, \dots, W]$;

(3) Encoding of the second-part bat position: Let the dimension of x_i^2 be W' = P + M

 $\sum_{m=1}^{M} n_m - 1$, which is a permutation of the sequence of integers $[1, 2, \dots, W']$.

3.2.2. Decoding Scheme

The first-part bat position x_i^1 is used to determine the supplier of a certain material of the products needed by the warehouse.

(1) $x_{ij}^1 = 1, 2, \dots, P \times L$ ($1 \le j \le W$) represents the serial number of a certain material for the warehouse, so as to ensure that a certain material of the products needed by each warehouse is provided by only one supplier;

(2) $x_{ij}^1 > P$ ($1 \le j \le W$) splits x_i^1 into *S* segments. From left to right, the segment *s* represents the set of materials provided by supplier *s* corresponding to the products needed by the warehouse, and $s = 1, 2, \dots, S$.

The second-part bat position x_i^2 is used to determine the manufacturer, making the products needed by the warehouse and the manufacturer's vehicle delivering the products.

(1) $x_{ij}^2 \le P$ ($1 \le j \le W'$) represents the serial number of the warehouse, so as to ensure that the products needed by each warehouse are made by only one manufacturer and delivered by the manufacturer's vehicle;

(2) $x_{ij}^2 > P$ (1 $\leq j \leq W'$) splits x_i^2 into $\sum_{m=1}^M n_m$ segments. From left to right, the

segment $\begin{bmatrix} m-1 \\ \sum \\ i=1 \end{bmatrix}^m n_i$ represents the vehicle belonging to the manufacturer *m*, and *m* = 1, 2, ..., *M*. The *n*_m segments belonging to the manufacturer *m* represent the delivery path of vehicle *k* of the manufacturer *m* from left to right, and *k* = 1, 2, ..., *n*_m.

Since x_i^1 determines the supplier of a certain material of the products needed by the warehouse, and x_i^2 determines the manufacturer delivering the products needed by the warehouse, it is easy to determine the manufacturer processing a certain material provided by each supplier.

First, the materials corresponding to the products needed by each warehouse are encoded, as shown in Figure 4. After confirming the serial number of each material corresponding to the products needed by the warehouse, the bat position can be encoded, as shown in Figure 5.

In the first-part bat position, position components 12 and 11 act as supplier delimiters, dividing the first-part position into three segments, i.e., supplier 1 is responsible for providing materials 1, 3, 5, 10; supplier 2 is responsible for providing materials 2, 4, 6, 7; and supplier 3 is responsible for providing materials 8, 9.



Figure 4. An instance for encoding the materials of products.



Figure 5. An instance for encoding the bat position.

In the second-part position, position components 6, 7, and 8 act as the manufacturer's vehicle delimiters, dividing the second-part bat position into four segments, i.e., vehicle 1 of manufacturer 1 is responsible for delivering the products of warehouses 1 and 2; vehicle 2 of manufacturer 1 is an empty vehicle (i.e., this vehicle is not used); vehicle 1 of manufacturer 2 is responsible for delivering the products of warehouses 3 and 4; and vehicle 2 of manufacturer 2 is responsible for delivering the products of warehouses 5.

After determining the first-part and second-part bat positions, supplier 1 should deliver materials 1 and 3 to manufacturer 1, and materials 5 and 10 to manufacturer 2; supplier 2 should deliver materials 2 and 4 to manufacturer 1, and materials 6 and 7 to manufacturer 2; supplier 3 should deliver materials 8 and 9 to manufacturer 2. The delivery path after decoding is shown in Figure 6.



Figure 6. Instance for decoding the bat position.

3.3. Discrete Chaos Initialization Strategy

The chaos optimization algorithm is an optimization algorithm characterized by randomness, ergodicity, and regularity. Scholars often adopt it to generate a set of initialized random sequences with ergodicity and pseudo-randomness, so as to strengthen the prophase convergence rate and overall convergence effect of intelligent algorithms [25]. According to our previous works [26], DCIS is used to generate the pseudo-random sequence in this paper, which is based on Logistic mapping (shown in Formula (23)).

$$z_{i+1} = \mu z_i (1 - z_i) \tag{23}$$

where, $0 \le z_i \le 1, z_i \ne 0.25, 0.5, 0.75, i = 0, 1, 2, \cdots, \mu = 4$.

Further, we adopt the all-permutation structure theory [27] to establish the corresponding relation between the serial number and the path. Finally, the initial population of bats can be preferentially generated by Formula (23) using the characteristics of chaos.

3.4. Fitness Function

The delivery path among suppliers, manufacturers, and warehouses can be obtained through encoding and decoding the bat position. However, it cannot be proved that the derived delivery path is a feasible solution, because the path may violate the constraint conditions of Formulas (13) and (14). Therefore, this paper deals with the constraints of Formulas (13) and (14) using a punishment mechanism to punish the infeasible solution. Further, it is found that there is no quantitative relationship among Formulas (5), (13), and (14). If the sum of the three formulas is directly used for fitness comparison, it is not conducive to controlling the optimization direction of the algorithm. Second, it is found that the optimal path must first satisfy Formulas (13) and (14) and then converge to the optimal direction of Formula (5). Therefore, on the basis of the punishment mechanism, a vector mechanism has been added to the fitness function in this paper. After using the punishment and vector mechanisms, the fitness function of bat *i* can be derived from Formula (5):

$$G_i = \{G_{i1}, G_{i2}, G_{i3}\}$$
(24)

$$G_{i1} = \sum_{s \in SS} \sum_{l \in SL} \max\left\{0, \sum_{m \in SM} \sum_{k \in SK} \sum_{p \in SP} g_{smkpl} o'_{pl} - e_{sl}\right\}$$
(25)

$$G_{i2} = \sum_{m \in SM} \sum_{k \in SK} \max\left\{0, \sum_{p \in SP} g''_{mkp} o_p - u_{mk}\right\}$$
(26)

$$G_{i3} = \sum_{s \in SS} \sum_{m \in SM} \sum_{k \in SK} \sum_{p \in SP} \sum_{l \in SL} \sum_{g_{smkpl}o'} g_{pl}c_{sl} + 2 \times \sum_{s \in SS} \sum_{m \in SM} g'_{sm}d_{sm}b_s + \sum_{m \in SM} \sum_{k \in SK} \sum_{p \in SP} g''_{mkp}o_pc'_m + \sum_{m \in SM} \sum_{k \in SK} \sum_{p \in SP} (y_{mkp} + y''_{mkp})d'_{mp}b'_{mk} + \sum_{m \in SM} \sum_{k \in SK} \sum_{p \in SP} \sum_{p' \in SP} y'_{mkpp'}d'_{pp'}b'_{mk}$$

$$(27)$$

where G_{i1} and G_{i2} are the punishments for violating the constraint of Formulas (13) and (14) respectively, and G_{i3} is the original target.

As can be seen from Formula (24), the fitness function G_{i1} is a three-dimensional value, and the commonly-used comparison method of one-dimensional numerical values is not suitable for fitness comparison in this paper. Hence, this paper has proposed a comparison method based on vector mechanism.

Supposing that the fitness of delivery path 1 is $G_1 = \{G_{11}, G_{12}, G_{13}\}$, the fitness of delivery path 2 is $G_2 = \{G_{21}, G_{22}, G_{23}\}$, and the optimal fitness is G_{best} , the flow chart is shown in Figure 7.



Figure 7. Flowchart of the fitness comparison based on vectorial mechanism.

3.5. Variable Neighborhood Search Strategy

The local search is a supplement to intelligent algorithms, and it can enhance the search capability of the algorithm [28]. Therefore, this paper introduces 2-Opt search, 0-1 search and 1-1 search, and proposes a VNS strategy [29]. In order to avoid excessive disturbance to the search direction, it is stipulated in this strategy that when one part of the bat position changes in the process of local optimization, the other part of bat position remains unchanged. The specific steps of VNS are as follows:

Step 1: 2-Opt search. 2-Opt search is carried out on the delivery path of a manufacturer's vehicle, until the vehicle's delivery path cannot be further improved.

Step 2: 0-1 search. For the first-part bat position, a material point of one supplier is inserted into that of another supplier to form a new bat position. If the new bat position is better, then the new bat position is accepted, and 0-1 search for this part of bat position, a warehouse point of one manufacturer's vehicle is inserted into that of another manufacturer's vehicle, and 2-Opt search is performed on the two vehicles, respectively, to form a new bat position. If the new bat position is better, then the new bat position is performed on the two vehicles, respectively, to form a new bat position. If the new bat position is better, then the new bat position is accepted, and 0-1 search for this part of bat position is repeated until no longer updated after *L* times.

Step 3: 1-1 search. It is similar to the 0-1 search. The difference lies in that 0-1 search is the insertion of points, while 1-1 search is the exchange of two points.

The VNS in this paper just exchanges the order of the components of bat position according to certain rules. The obtained bat position still conforms to the encoding definition of the bat position in Section 3.1, and a complete VRPCSC path can be obtained by decoding. The pseudocode of HCDBAVNS is shown as Algorithm 1.

Algorithm 1 HCDBAVNS

Input:

Parameter value;

Output:

- The global best position x_* of bat and its fitness G_*
- 1: Chaos Initialization Strategy is used to generate the bat population;
- Calculate the fitness G_i of bat *i* by the fitness function, then the optimal global position 2: of bats x_* and G_* are initialized, $i = 1, 2, \cdots, Q$.
- 3: T = 0;
- 4: while $T < N_{\text{max}}$ (the maximal number of iterations) do
- for i = 1 to Q do 5:
- Update operations of bat position, velocity, and frequency are used to generate 6: x_i^{new} , v_i^{new} and f_i^{new} , respectively. Then, $v_i = v_i^{new}$, $f_i = f_i^{new}$.
- if $R_i < rand()$ then 7:
 - Bat variation operation is used to generate x_i^{new} .
- end if 9:

8:

15: 16:

- Variable neighborhood search strategy is used to find the best bat position x_i^{pbest} in 10: the neighborhood of x_i^{new} .
- if $G_i^{pbest} < G_i$ and $A_i > rand()$ then 11:
- (1) $x_i = x_i^{pbest}$, and updating A_i through updated operations of loudness and 12: pulse emission rate.
- end if 13:
- 14:

if $G_i^{pbest} < G_*$ then (2) $x_* = x_i^{pbest}$ and $G_* = G_i^{pbest}$. end if

17: end for

N = N + 1.18:

19: end while

4. Experiments

4.1. Experimental Instances

Since the VRPCSC model has no benchmark instances, this paper randomly generates 20 test instances to evaluate the algorithm according to the size of instances in Table 1 and the parameter values of instances in Table 2. These instances are divided into four groups, each group has five instances, and each instance is marked as P01, P02, ..., P20, respectively.

Table 1. The scale of Instances.

Group	S	M	n _m	Р	L	Relationship between Materials and Products
G1	3	2	3	10	2	$o_p = \frac{o'_{p1}}{1} = \frac{o'_{p2}}{1}$ [t]
G2	4	3	3	20	2	$o_p = \frac{o'_{p1}}{1} = \frac{o'_{p2}}{1}$
G3	3	2	3	10	3	$o_p = \frac{o'_{p1}}{2} = \frac{o'_{p2}}{2} = \frac{o'_{p3}}{1}$
G4	4	3	3	20	3	$o_p = \frac{o'_{p1}}{2} = \frac{o'_{p2}}{2} = \frac{o'_{p3}}{1}$ [b]

Parameter	Range
X and Y coordinates	Rand (0, 500) [t]
e_s	Rand (20, 70)
C_S	Rand (1000, 1500)
b_s	Rand (10, 15)
u _{mk}	Rand (0, 3)*10
c'_m	Rand (1500, 1800)
b'_{mk}	Rand (3, 5)
o _p	Rand (1, 10) [b]

Table 2. The ranges of some parameters of instances.

4.2. Experimental Environment

The simulation environment of the experiment uses a Windows 7 operating system, i7 processor, and 4G memory. The programming language is C++. All the algorithms are implemented and executed in this experimental environment. Each instance is independently solved by the corresponding algorithm 10 times to obtain experimental results.

4.3. Key Experimental Parameters

The values of key experimental parameters have a great influence on the solving capability of intelligent algorithms. Thus, in order to enable the algorithm to solve the VRPCSC more effectively, four VRPCSC instances are solved using HCDBAVNS with different values of θ^1 , θ^2 , γ and α (other parameters: Q = 30, L = 100, $MIT \leq 3000$). The results are shown in Tables 3–5, where the unit of time is second.

ent θ^1 and θ^2 .
ent θ^1 and θ^2 .

Instance	$ heta^1 = 0.5 imes W, heta^2 = 0.5 imes W', \ lpha = 0.999, \gamma = 0.001$			$\theta^1 = W, \theta^2 = 2 \times W', \ \alpha = 0.999, \gamma = 0.001$			$ heta^1=2 imes W, heta^2=W',\ lpha=0.999,\gamma=0.001$		
Instance	Best Solution	Average Solution	Time	Best Solution	Average Solution	Time	Best Solution	Average Solution	Time
P01	245,399	245,399.0	4.7	245,399	245,399.0	4.3	245,399	245,399.0	4.8
P06	454,497	454,891.2	43.5	450,664	453,527.2	45.9	452,069	453,797.8	44.6
P11	379,602	379,602.0	17.4	379,602	379,602.0	14.1	379,602	379,602.0	18.6
P16	939,685	941,535.4	56.7	938,095	940,699.0	59.2	938,336	941,260.2	59.6

1. In Table 3, α and γ have the same value. When $\theta^1 = W$ and $\theta^2 = W'$, the HCDBAVNS has the most balanced performance in optimization capability and time consumption.

Instance	$\theta^1 = W, \ \theta^2 = W',$ $\alpha = 0.9, \ \gamma = 0.001$		θ^1 $\alpha =$	$ heta^1 = W, heta^2 = W', \ lpha = 0.99, \gamma = 0.001$			$ heta^1 = W, heta^2 = W', \ lpha = 0.999, \gamma = 0.001$		
instance	Best Solution	Average Solution	Time	Best Solution	Average Solution	Time	Best Solution	Average Solution	Time
P01	245,399	245,399.0	5.1	245,399	245,399.0	4.7	245,399	245,399.0	4.3 [t]
P06	452,935	454,656.6	47.7	451,918	453,691.6	46.1	450,664	453,527.2	45.9
P11	379,602	379,602.0	18.3	379,602	379,602.0	16.3	379,602	379,602.0	14.1
P16	941,560	943,788.6	62.3	939,532	940,811.0	60.5	938,095	940,699.0	59.2 [b]

Table 4. The result of instances by HCDBAVNS with different α .

2. In Table 4, θ^1 , θ^2 and γ have the same value. As the value of α increases, HCDBAVNS has increasingly better optimization capability but slightly decreased time consumption.

Test	$ heta^1 = W, heta^2 = W', \ lpha = 0.999, \gamma = 0.001$		θ^1 $\alpha =$	$ heta^1 = W, heta^2 = W', \ lpha = 0.999, \gamma = 0.01$			$ heta^1 = W, heta^2 = W', \ lpha = 0.999, \gamma = 0.5$		
Instance	Best Solution	Average Solution	Time	Best Solution	Average Solution	Time	Best Solution	Average Solution	Time
P01	245,399	245,399.0	4.3	245,399	245,399.0	4.7	245,399	245,399.0	4.6 [t]
P06	450,664	453,527.2	45.9	452,229	453,827.0	43.7	453,026	454,184.4	41.6
P11	379,602	379,602.0	14.1	379,602	379,602.0	17.1	379,602	379,602.0	17.3
P16	938,095	940,699.0	59.2	939 <i>,</i> 359	941,121.6	53.2	942,008	942,730.2	55.7 [b]

Table 5. The result of instances by HCDBAVNS with different γ .

3. In Table 5, θ^1 , θ^2 and α have the same value. As the value of γ decreases, HCD-BAVNS has the generally the same time consumption but increasingly stronger optimization capability.

It can be seen that among the seven combinations of values that HCDBAVNS has the best performance when $\theta^1 = W$, $\theta^2 = W'$, $\alpha = 0.999$ and $\gamma = 0.001$. Therefore, Q = 30, L = 100, $MIT \leq 3000$, $\theta^1 = W$, $\theta^2 = W'$, $\alpha = 0.999$ and $\gamma = 0.001$ will be set in the following experiments.

4.4. Experimental Results and Analysis

In this part, 2 experiments are performed to prove the effectiveness and the optimization ability of the algorithm, respectively.

4.4.1. Experiment 1

In order to analyse the solving effect of HCDBAVNS in detail, this paper selects the experimental results of the instance P01 for specific analysis. The information of instance P01 is shown in Tables 6–8.

Table 6. Supplier information of P01.

Supplier	X Coordinate	Y Coordinate	b_s	c_{s1}	e_{s1}	c_{s2}	e_{s2}
SUP1	99	236	10	1027	38	1200	26 [t]
SUP2	342	29	14	1072	21	1079	30
SUP3	78	242	14	1067	24	1051	36 [b]

Table 7. Manufacturer information of P01.

Manufacturer Tab	V Coordinate	V Coordinate	b' ,	c'		Vehicle Information	
	X Coordinate	i Coordinate	v_{mk}	c_m	n_m	Vehicle Tab	u _{mk}
MAN1	304	266	3	1623	3	VEH1 VEH2 VEH3	20 [t] 30 30 [b]
MAN2	22	412	5	1512	3	VEH4 VEH5 VEH6	30 [t] 20 20 [b]

Warehouse	Y Caralianta	V Coordinata	0	Mate	erial 1	Material 2	
Tab	X Coordinate	i Coordinate	o _p	Tab	Demand	Tab	Demand
STO1	306	299	10	MAT1	10	MAT2	10 [t]
STO2	387	84	6	MAT3	6	MAT4	6
STO3	411	488	1	MAT5	1	MAT6	1
STO4	352	210	10	MAT7	10	MAT8	10
STO5	197	252	2	MAT9	2	MAT10	2
STO6	91	378	8	MAT11	8	MAT12	8
STO7	306	65	1	MAT13	1	MAT14	1
STO8	43	19	9	MAT15	9	MAT16	9
STO9	175	415	10	MAT17	10	MAT18	10
STO10	426	103	4	MAT19	4	MAT20	4 [b]

 Table 8. Warehouse information of P01.

Then, the best solution of P01 solved by the HCDBAVNS is shown in Tables 9–11 and Figure 8 which reflect the following characteristics.



Figure 8. The schematic diagram of optimal solutions of P01 by HCDBAVNS.

Item	Set Information	Supply Quantity of Material 1	Supply Quantity of Material 2	Material Cost
	SUP1: MAT3,			
	MAT11, MAT13,			
	MAT7, MAT 5,	20	25	69,026
	MAT16, MAT18,	50		
	MAT17, MAT4,			
The Material of Warehouses Served by Suppliers	MAT9			
The finite function of the choices better by suppliers	SUP3: MAT6,			
	MAT19, MAT2,		24	
	MAT1, MAT8,	22		(2.277
	MAT12, MAT15,	23	30	62,377
	MAT10, MAT20,			
	MAT14			
Total cost				131,403

Table 9. Warehouses served by suppliers in the optimal solution of P01 by HCDBAVNS.

Item	Route Information	Used Capacity	Processing Cost	Route Distance	Transportation Cost
The transportation from manufacturers to warehouses	VEH4: MAN2- > STO8- > STO7- > STO2- > STO10- > STO4- > MAN2	30	45,360	1301	6505
	VEH5: MAN2-> STO5-> STO1-> STO3->MAN2	13	19,656	967	4835
	VEH6: MAN2-> STO9-> STO6->MAN2	18	27,216	320	1600
Total cost			92,232		12,940

Table 10. The transportation from manufacturers to warehouses in the optimal solution of P01 by HCDBAVNS.

Table 11. The transportation from suppliers to manufacturers in the optimal solution of P01 by HCDBAVNS.

Item	Route Information	Route Distance	Transportation Cost
The transportation from	SUP1 - > MAN2	384	3840
suppliers to manufacturers	SUP3- > MAN2	356	4984
Total cost			8824

1. The materials (i.e., MAT1, MAT2, ..., MAT10) corresponding to the products needed by all warehouses are supplied by SUP1 and SUP3 without any omission. For the SUP1, the supply quantity of material 1 is 38, and the supply quantity of material 2 is 25, both of which have not exceeded their maximum supply quantities (38 and 26). Similarly, for the SUP3, the supply quantity of material 1 is 23, and the supply quantities (24 and 36) either. It is worth noting that SUP2 is not selected in this best solution because the materials of SUP1 and SUP3 are cheaper than SUP2 and the material supply quantities of SUP1 and SUP3 have met the demand of all warehouses. In addition, SUP2 is further away from the MAN2 and MAN1 than SUP1 and SUP3.

2. All the products are made by MAN2 and delivered by three vehicles of MAN2. The actual loads of VEH4 are 30, 13, and 18, respectively, all of which have not exceeded their maximum loading capacities (30, 20, and 20). MAN1 is not selected for processing and delivery. Because the unit processing cost of MAN2 is cheaper than that of MAN1, and the position of MAN2 is closer to the SUP1 and SUP3 than MAN2.

This best solution satisfies the material supply constraint and maximum loading capacity constraint of VRPCSC, that is, it satisfies Formulas (9) and (10), so it is a feasible solution. Thus, HCDBAVNS can effectively solve VRPCSC.

4.4.2. Experiment 2

In order to assess the effectiveness HCDBAVNS, this experiment makes a comparison among HCDBAVNS without DCIS (denoted as HDBAVNS), VNS, Particle Swarm Optimization [30] with VNS (denoted as PSOVNS) and HCDBAVNS. This experiment selects 20 instances which have no known optimal solutions. The paper makes some assumptions to get a lower bound for each instance. Assuming that the purchase cost of materials can be calculated by the lowest price of materials of all suppliers in the Formula (1), and the lowest purchase cost of materials can be calculated as follows:

$$Z'_{PUR} = \sum_{p \in SP} \sum_{l \in SL} o'_{pl} \min_{s} \{c_{sl}\}$$

$$(28)$$

Secondly, in Formula (2), assuming that maximum supply quantities of all suppliers are set to the maximal value of e_{sl} , and the unit delivery costs and distances are a set value between suppliers and manufacturers, then the lowest delivery cost of suppliers' vehicles are shown as follows:

$$Z'_{\text{STM}} = 2 \times \left(\left[\sum_{p \in SP} \sum_{l \in SL} o'_{pl} / \max(e_{sl}) \right] \right) \times \min_{s,m} \{d_{sm}\} \times \min_{s} \{b_{s}\}$$
(29)

Thirdly, assuming the processing cost of materials can be calculated by the lowest processing price of all manufacturers in the Formula (3), and the lowest processing cost can be calculated by Formula (30).

$$Z'_{PRO} = \sum_{p \in SP} o_p \times \min\{c'_m\}$$
(30)

Lastly, assuming that maximum loading capacities of all manufacturers' vehicles are set to the maximal value of u_{mk} , and the unit delivery costs and distances are set the lowest value between manufacturers and warehouses, then the lowest delivery cost of manufacturers' vehicles are shown as follows:

$$Z'_{\text{MTS}} = 2 \times \left(\left[\sum_{p \in SP} o_p / \max_{m,k}(u_{mk}) \right] \right) \times \min\{d'_{mp}\} \times \min\{b'_{mk}\} + \left(P - \left[\sum_{p \in SP} o_p / \max_{m,k}(u_{mk}) \right] \right) \times \min\{d'_{pp'}\} \times \min\{b'_{mk}\}$$
(31)

From Formula (1) to Formula (5), a lower bound S_L for each instance is obtained, denoded as Formula (32).

$$S_{L} = Z'_{PUR} + Z'_{STM} + Z'_{PRO} + Z'_{MTS}$$

$$(32)$$

Then, the optimization rate *OP* is calculated as Formula (33) which asses the quality of the solutions of the algorithm. The experiment results are shown in Tables 12 and 13 and Figure 9.

$$OP = \frac{\text{Best solution} - S_L}{S_L} \times 100\%$$
(33)

It can be seen from Tables 12 and 13 that: when time consumption of all the algorithms are close, all the best solutions, average solutions, and worst solutions of 20 instances solved by HCDBAVNS are not worse than those solved by VNS and PSOVNS. On the other hand, there are seven and nine instances of HCDBAVNS obviously better than HDBAVNS in both the best solutions and average solutions, accounting for 35% and 45% of the total instances, respectively. Then, although DCIS made the optimization capacity of HCDBAVNS not increase significantly (all the examples were less than 1%), it made HCDBAVNS more steady.

Instance		MIT	S_L	VNS				PSOVNS					
				Best Solution	Average Solution	Worst solution	<i>OP</i> (%)	Time	Best Solution	Average Solution	Worst Solution	<i>OP</i> (%)	Time
G1	P01	200	227,607	245,399	245,399.0	245,399	7.82	4.3	245,399	245,399.0	245,399	7.82	4.5
	P02	200	211,615	238,905	238,905.0	238,905	12.90	4.5	238,905	238,905.0	238,905	12.90	4.3
	P03	200	259,559	286,191	286,191.0	286,191	10.26	4.3	286,191	286,191.0	286,191	10.26	4.2
	P04	200	213,048	240,149	240,149.0	240,149	12.72	4.2	240,149	240,149.0	240,149	12.72	4.1
	P05	200	241,592	269,959	269,959.0	269,959	11.74	3.9	269,959	269,959.0	269,959	11.74	3.8
G2	P06	2000	401,972	450,664	453,527.2	455,186	12.11	45.9	452,185	454,078.6	455,367	12.49	46.5
	P07	2000	482,338	562,425	564,518.4	566,501	16.60	23.3	563,563	564,828.2	566,140	16.84	23.5
	P08	2000	353,804	411,177	411,947.6	412,274	16.22	47.9	411,177	411,963.0	412,859	16.22	47.6
	P09	2000	462,224	539,301	540,791.6	541,875	16.68	18.8	539,351	541,266.4	542,524	16.69	19.2
	P10	2000	357,230	401,208	403,241.8	404,232	12.31	41.8	403,287	403,459.4	403,829	12.89	41.3
	P11	1000	342,420	379,602	379,602.0	379,602	10.86	14.1	379,602	379,602.0	379,602	10.86	14.9
G3	P12	1000	423,011	468,178	469,600.8	470,907	10.68	15.3	468,906	470,519.6	470,971	10.85	15.6
	P13	1000	341,952	371,543	371,543.0	371,543	8.65	19.5	371,543	371,543.0	371,543	8.65	19.1
	P14	1000	370,158	406,929	406,929.0	406,929	9.93	21.5	406,929	406,929.0	406,929	9.93	21.2
	P15	1000	532,112	588,222	588,222.0	588,222	10.54	15.6	588,222	588,222.0	588,222	10.54	15.7
G4	P16	3000	807,715	938,095	940,699.0	942,193	16.14	59.2	940,460	941,739.4	942 <i>,</i> 517	16.43	59.5
	P17	3000	958,872	1,132,475	1,136,530.4	11,373,88	18.10	50.1	1,133,556	1,139,009.8	1,143,398	18.22	50.6
	P18	3000	826,146	893,681	894,655.0	896,728	8.17	61.2	893,700	895,485.4	898,046	8.18	61.5
	P19	3000	625,755	675,812	677,138.0	678,534	8.00	82.7	676,342	677,294.8	678,405	8.08	82.3
	P20	3000	642,355	696,039	696,477.6	696,969	8.36	85.9	696,039	696,684.2	697,473	8.36	86.3





Figure 9. *OP* of HCDBAVNS, HDBAVNS, VNS and APSOVNS.

Instance		MIT	S_L	VNS				PSOVNS					
				Best Solution	Average Solution	Worst solution	<i>OP</i> (%)	Time	Best Solution	Average Solution	Worst Solution	<i>OP</i> (%)	Time
G1	P01	200	227,607	245,399	247,606.8	250,222	7.82	4.9	245,399	245,850.2	246,527	7.82	4.5
	P02	200	211,615	239,446	241,699.0	243,061	13.15	4.1	238,905	239,255.2	239,807	12.90	4.4
	P03	200	259,559	286,882	287,921.6	289,545	10.53	4.6	286,291	286,291.0	286,291	10.30	4.3
	P04	200	213,048	245,069	248,848.2	250,150	15.03	3.9	240,149	242,305.0	245,069	12.72	4.3
	P05	200	241,592	269,959	270,079.4	270,329	11.74	4.1	269,959	269,959.0	269,959	11.74	3.8
G2	P06	2000	401,972	461,985	465,777.2	473,112	14.93	46.1	454,971	456,304.6	457,961	13.18	46.5
	P07	2000	482,338	573,135	577,617.6	581,841	18.82	23.6	569,162	569,866.8	570,764	18.00	24.1
	P08	2000	353,804	419,927	422,849.0	425,984	18.69	29.5	413,760	413,848.0	413,931	16.95	49.9
	P09	2000	462,224	562,461	563,011.0	564,653	21.69	20.5	546,047	550,210.4	552,288	18.13	20.3
	P10	2000	357,230	408,072	414,620.6	419,073	14.23	39.5	407,489	407,716.2	407,829	14.07	40.9
-	P11	1000	342,420	379,602	379,903.6	380,142	10.86	15.6	379,602	379,602.0	379,602	10.86	14.8
G3	P12	1000	423,011	471,805	474,410.4	475,715	11.53	15.8	469,544	464,248.7	471,044	11.00	15.4
	P13	1000	341,952	371,543	371,655.4	371,907	8.65	19.2	371,543	371,543.0	371,543	8.65	19.3
	P14	1000	370,158	406,929	406,936.2	406,947	9.93	20.6	406,929	406,929.0	406,929	9.93	21.2
	P15	1000	532,112	588,222	588,222.0	588,222	10.54	15.8	588,222	588,222.0	588,222	10.54	15.8
G4	P16	3000	807,715	948,560	954,319.4	957,402	17.44	54.7	948,136	948,838.6	949,841	17.38	56.3
	P17	3000	958,872	1,149,885	1,154,544.0	1,156,343	19.92	49.5	1,148,604	1,150,104.4	1,152,467	19.79	61.5
	P18	3000	826,146	902,756	904,639.6	907,400	9.27	63.2	899,041	899,651.4	900,260	8.82	62.2
	P19	3000	625,755	690,032	692,122.6	694,282	10.27	86.4	684,229	685,327.8	686,814	9.34	88.4
	P20	3000	642,355	706,239	707,694.0	709,278	9.95	90.4	702,318	704,458.8	705,547	9.33	92.4

Table 13. Result of VNS and APSOVNS.

In Figure 9, it can be sees that HCDBAVNS has a higher possibility of finding a better solution than VNS. Especially, the *OP* of HCDBAVNS, which is 4% less than that of VNS in P09. In addition, HDBAVNS is also better than PSOVNS. It proves that BA has better global optimization ability than PSO.

Further, the paired two-sample mean analysis method in the *t*-Test is used to test whether the three algorithms are significantly different. Supposing that the significance level is 0.05 and mean deviation is 0, The experimental results are shown in Table 14. In the *t*-Test, p > 0.05 indicates no significant difference; 0.01 indicates a significant difference; <math>p < 0.01 indicates an extreme significant difference.

Table 14. S	tudent's	t-test.
-------------	----------	---------

	t Test	<i>p</i> Value				
lab	t-lest –	Best Solution	Average Solution			
W1	HCDBAVNS vs. HDBAVNS	0.01149	0.01514 [t]			
W2	HCDBAVNS vs. VNS	0.00034	0.00005			
W3	HDBAVNS vs. VNS	0.0005	0.00005			
W4	HDBAVNS vs. PSOVNS	0.00148	0.00610 [b]			

In Table 14, it can be seen from W1 test that the p values of the best solution and average solution are 0.01149 and 0.01514 (both less than 0.05). In other words, both the best solution and average solution of HCDBAVNS show significant differences to HDBAVNS.

In addition, compared with VNS, HDBAVNS, and HCDBAVNS have shown extreme significant differences. Furthermore, HDBAVNS shows a significant difference to PSOVNS. This also proves that HCDBAVNS finally exhibits strong optimization capability and stability.

5. Conclusions

Based on the vehicle routing problem in the three-echelon supply chain, this paper proposes a VRPCSC model and constructs HCDBAVNS to solve it. Two sets of experiments were conducted, which showed the effectiveness of the proposed algorithm in solving the VRPCSC. In the future, more complex factors will be considered to establish a more comprehensive supply chain model, such as the manufacturer's maximum processing volume and the manufacturer's processing time. Accordingly, we can use Taguchi or RSM to tune the parameters and compare the influence of different methods of parameter values. In addition, we can apply the algorithm to solve actual examples and compare the actual scheme, so as to judge the quality of the solutions obtained by the algorithm.

Author Contributions: Writing—original draft preparation, Y.Q.; Writing—review and editing, Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by the Science and Technology Program of Tianhe District, Guangzhou under grant No. 2018CX005, the Key Project in Higher Education of Guangdong Province, China under grant No. 2020ZDZX3030 and the Science and Technology Foundation of Guangdong Province under grant No. 2021A0101180005.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors would like to thank Peng Hou for his contributions to the algorithm development and all authors' help of comments which makes the paper more readable.

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

References

- 1. Karimi, B.; Ghare Hassanlu, M.; Niknamfar, A.H. An integrated production-distribution planning with a routing problem and transportation cost discount in a supply chain. *Assem. Autom.* **2019**, *39*, 783–802. [CrossRef]
- Saif-Eddine, A.S.; El-Beheiry, M.M.; El-Kharbotly, A.K. An improved genetic algorithm for optimizing total supply chain cost in inventory location routing problem. *Ain Shams Eng. J.* 2019, 10, 63–76. [CrossRef]
- 3. Apte, U.M.; Viswanathan, S. Effective cross docking for improving distribution efficiencies. *Int. J. Logist. Res. Appl.* 2000, 3, 291–302. [CrossRef]
- 4. Setak, M.; Habibi, M.; Karimi, H.; Abedzadeh, M. A time-dependent vehicle routing problem in multigraph with FIFO property. J. Manuf. Syst. 2015, 35, 37–45. [CrossRef]
- 5. Govindan, K.; Jafarian, A.; Nourbakhsh, V. Designing a sustainable supply chain network integrated with vehicle routing: A comparison of hybrid swarm intelligence metaheuristics. *Comput. Oper. Res.* **2019**, *110*, 220–235. [CrossRef]
- 6. Qi, Y.; Cai, Y.; Cai, H.; Yang, L.; Yeboah, Y. Two-Level Bat Algorithm with Variable Neighborhood Search for Capacitated Vehicle Routing Problem in Supply Chain. *Acta Electron. Sin.* **2019**, *47*, 1434–1442.
- Wen, M.; Larsen, J.; Clausen, J.; Cordeau, J.F.; Laporte, G. Vehicle routing with cross-docking. J. Oper. Res. Soc. 2009, 60, 1708–1718. [CrossRef]
- Rohrer, M. Simulation and cross docking. In Proceedings of the Winter Simulation Conference Proceedings, Arlington, VA, USA, 3–6 December 1995; pp. 846–849.
- 9. Yu, W.; Egbelub, P.J. Scheduling of inbound and outbound trucks in cross docking systems with temporary storage. *Eur. J. Oper. Res.* **2008**, *184*, 377–396. [CrossRef]
- 10. Lim, A.; Ma, H.; Miao, Z. Truck dock assignment problem with operational time constraint within crossdocks. *Eur. J. Oper. Res.* **2009**, *192*, 105–115.
- 11. Boysen, N. Truck scheduling at zero-inventory cross docking terminals. Comput. Oper. Res. 2010, 37, 32–41. [CrossRef]
- 12. 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]
- 13. Zuluaga, J.P.S.; Thiell, M.; Perales, R.C. Reverse cross-docking. *Omega* 2017, 66, 48–57. [CrossRef]
- 14. Liu, S. On the integrated production, inventory, and distribution routing problem. *IIE Trans.* 2006, 38, 955–970.
- 15. Qiu, Y.; Qiao, J.; Pardalos, P.M. A branch-and-price algorithm for production routing problems with carbon cap-and-trade. *Omega* **2017**, *68*, 49–61. [CrossRef]
- 16. Kumar, R.S.; Kondapaneni, K.; Dixit, V.; Goswami, A.; Thakur, L.S.; Tiwari, M.K. Multi-objective modeling of production and pollution routing problem with time window. *Comput. Ind. Eng.* **2016**, *99*, 29–40. [CrossRef]

- 17. Darvish, M.; Archetti, C.; Coelho, L.C. Trade-offs between environmental and economic performance in production and inventory-routing problems. *Int. J. Prod. Econ.* **2019**, *217*, 269–280. [CrossRef]
- 18. Wang, Z.; Ren, X.; Ji, Z.; Huang, W.; Wu, T. A novel bio-heuristic computing algorithm to solve the capacitated vehicle routing problem based on Adleman Lipton model. *Biosystems* **2019**, *184*, 103997. [CrossRef]
- 19. Liu, J.; Wan, J.; Jia, D.; Zeng, B.; Li, D.; Hsu, C.H.; Chen, H. High-Efficiency Urban Traffic Management in Context-Aware Computing and 5G Communication. *IEEE Commun. Mag.* **2017**, *55*, 34–40.
- 20. Qi, Y.; Cai, Y.; Cai, H.; Huang, H. Discrete bat algorithm for vehicle routing problem with time window. *Acta Electron. Sin.* **2018**, 46, 672–679. [CrossRef]
- 21. Qi, Y.; Hou, P.; Yang, L.; Yang, G. Simultaneous optimisation of cable connection schemes and capacity for offshore wind farms via a modified bat algorithm. *Appl. Sci.* **2019**, *9*, 265. [CrossRef]
- 22. Qi, Y.; Cai, Y.; Cai, H.; Tang, Y.; Lv, W. Chaotic hybrid bat algorithm for travelling salesman problem. *Acta Electron. Sin.* **2016**, 44, 2543–2547. [CrossRef]
- Yang, X.S. A new metaheuristic bat-inspired algorithm. In Nicso 2010: Nature Inspired Cooperative Strategies for Optimization; Gonzalez, J.R., Pelta, D.A., Cruz, C., Terrazas, G., Krasnogor, N., Eds.; Springer: Berlin, Germany, 2010; Volume 284, pp. 65–74.
- 24. Osaba, E.; Yang, X.S.; Diaz, F.; Lopez-Garcia, P.; Carballedo, R. An improved discrete bat algorithm for symmetric and asymmetric Traveling Salesman Problems. *Eng. Appl. Artif. Intell.* **2016**, *48*, 59–71. [CrossRef]
- El-Semary, A.M.; Azim, M.M.A.; Diab, H. SPCBC: A Secure Parallel Cipher Block Chaining Mode of Operation based on logistic Chaotic Map. KSII Trans. Internet Inf. Syst. 2017, 11, 3608–3628. [CrossRef]
- Cai, Y.; Qi, Y.; Cai, H.; Huang, H.; Chen, H. Chaotic discrete bat algorithm for capacitated vehicle routing problem. *Int. J. Auton. Adapt. Commun. Syst.* 2019, 12, 91–108. [CrossRef]
- 27. Gao, S. Solving traveling salesman problem by chaos ant colony optimization algorithm. *Syst. Eng. Theory Pract.* **2005**, 25, 100–104. [CrossRef]
- 28. Alvarez, A.; Munari, P.; Morabito, R. Iterated local search and simulated annealing algorithms for the inventory routing problem. *Int. Trans. Oper. Res.* **2018**, 25, 1785–1809. [CrossRef]
- 29. Hiermann, G.; Puchinger, J.; Ropke, S.; Hartl, R.F. The electric fleet size and mix vehicle routing problem with time windows and recharging stations. *Eur. J. Oper. Res.* **2016**, 252, 995–1018. [CrossRef]
- Zhan, Z.H.; Zhang, J.; Li, Y.; Chung, H.S.H. Adaptive Particle Swarm Optimization. *IEEE Trans. Syst. Man Cybern. Part B-Cybern.* 2009, 39, 1362–1381. [CrossRef]