

Article

Integrated Scheduling of Picking and Distribution of Fresh Agricultural Products for Community Supported Agriculture Mode

Xujin Pu ^{1,*}, Yuchen Xu ¹ and Yaping Fu ²¹ Business School, Jiangnan University, Wuxi 214122, China² Business School, Qingdao University, Qingdao 266071, China

* Correspondence: puyiwei@ustc.edu

Abstract: Community Supported Agriculture (CSA), which offers two outstanding advantages, high-quality food and localized production, has come to the fore. In CSA, the output of picking scheduling is the input of delivery scheduling. Hence, only by scheduling the picking stage and distribution stage in a coordinated way can we achieve fresh agricultural products at minimum cost. However, due to asymmetric information in the picking and distribution stage, the integrated scheduling of picking and distribution may lead to an asymmetric optimization problem, which is suitable for solving with an iterative algorithm. Based on this, this work studies an integrated scheduling problem of the picking and distribution of fresh agricultural products with the consideration of minimizing picking and distribution costs as well as maximizing the freshness of orders. First, a nonlinear mixed-integer programming model for the problem under consideration is constructed. Second, a multi-objective multi-population genetic algorithm with local search (MOPGA-LS) is designed. Finally, the algorithm is compared with three multi-objective optimization algorithms in the literature: the non-dominated sorted genetic algorithm-II (NSGA-II), the multi-objective evolutionary algorithm based on decomposition (MOEA/D), and the multi-objective evolutionary algorithm based on decomposition that is combined with the bee algorithm (MOEA/D-BA). The comparison results show the excellent performance of the designed algorithm. Thus, the reported model and algorithm can assist managers and engineers in making well-informed decisions in managing the farm operation.

Keywords: integrated scheduling of picking and distribution; community supported agriculture; genetic algorithm; local search



Citation: Pu, X.; Xu, Y.; Fu, Y. Integrated Scheduling of Picking and Distribution of Fresh Agricultural Products for Community Supported Agriculture Mode. *Symmetry* **2022**, *14*, 2530. <https://doi.org/10.3390/sym14122530>

Academic Editor: José Carlos R. Alcantud

Received: 1 November 2022

Accepted: 25 November 2022

Published: 30 November 2022

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



Copyright: © 2022 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/>).

1. Introduction

Community Supported Agriculture (CSA) has boomed owing to the continuous improvement of living standards and the frequent occurrence of food safety crises. CSA represents a form of direct food selling where local farms provide agricultural products directly to community residents [1]. In the context of CSA, a farm is responsible for all relevant agricultural product supply chain activities, including planting, picking, and distribution. CSA can greatly shorten the time interval between picking and distribution to improve the freshness of fresh agricultural products, satisfying residents' demand for agricultural products [2]. Besides, disruptions of the agricultural supply chain are becoming more and more frequent due to COVID-19 and other emergencies. Growing attention has been devoted to localized production, which is one of the characteristics of CSA [3]. Hence, CSA is developing rapidly all over the world [4].

Picking and distribution are two highly related activities in CSA [5,6]. The former focuses on arranging for workers to pick agricultural products in the field, while the latter aims to deliver picked products to customers without unnecessary delay. Nowadays, a CSA farm mainly arranges picking scheduling based on experience. The distribution is characterized by scattered customer points and strict requirements for timeliness, which

leads to an inefficient use of resources and high distribution costs. To date, most studies have focused on the optimization of picking and distribution separately [7], which can achieve local optimization but fails in achieving global optimization. As we all know, fresh agricultural products are highly sensitive to time, and their freshness will decay exponentially over time after picking [8]. Generally, it is not suitable to store them in the inventory for a long time, which creates a great deal of challenges to effectively managing the fresh agricultural supply chain. The picked products must be delivered to customers without unnecessary delay to ensure freshness. Therefore, joint scheduling should be considered to achieve a seamless link between picking and distribution and achieve an overall optimization.

Thus, this work studies the integrated scheduling problem of the picking and distribution of fresh agricultural products with the consideration of minimizing picking and distribution costs as well as maximizing the freshness of the orders. At the picking stage, all products required by customers are picked according to varieties. The customer order may aggregate a variety of agricultural products planted in different blocks of the CSA farm due to different growth conditions. Therefore, in order to improve efficiency, the CSA farm will collect the order information and pick the products according to their varieties. In addition, different fresh agricultural products have different perishability. This work takes into account the differences in perishability that have an impact on joint scheduling. At the distribution stage, the departure of each vehicle from the farm is equal to the picking completion time of the last product in the vehicle's cargo. To sum up, there are two main contributions.

- (1) An integrated scheduling problem of the picking and distribution of fresh agricultural products with the consideration of minimizing picking and distribution costs as well as maximizing the freshness of the order is explored in this work. At the picking stage, a variety of agricultural products need to be assigned among multiple picking groups with different picking abilities. Then the orders are delivered to customers without unnecessary delay. A nonlinear mixed-integer programming model is constructed to formulate the problem.
- (2) A multi-objective multi-population genetic algorithm with local search (MOPGA-LS) is designed. The designed algorithm is compared with three multi-objective optimization algorithms in the literature: the non-dominated sorted genetic algorithm-II (NSGA-II) [9], the multi-objective evolutionary algorithm based on decomposition (MOEA/D) [10], and the multi-objective evolutionary algorithm based on decomposition that is combined with the bee algorithm (MOEA/D-BA) [11]. The comparison results demonstrate that the designed algorithm is a superior optimizer for tackling the integrated picking and distribution problem for agricultural products.

The rest of the paper is structured as follows: Section 2 reviews the relevant literature. In Section 3, the investigated problem is abstracted into a mathematical model. A multi-objective multi-population genetic algorithm is proposed in Section 4. The experimental results and analysis are presented in Section 5. In the end, the conclusion and future research direction are summarized.

2. Literature Review

There are two closely linked activities in the agricultural product supply chain, namely picking and distribution [12]. Only through joint scheduling can we achieve overall optimization. Next, this paper reviews the relevant literature from four perspectives: picking, distribution, integrated scheduling, and evolutionary algorithms for solving the integrated scheduling problem.

Literature about agricultural product picking is dedicated to harvesting decisions. Ferrer et al. [13] investigated the optimization of grape harvesting scheduling with the consideration of minimizing the operational cost as well as minimizing the quality loss. Bohle et al. [14] expanded the research of Ferrer by taking account into the uncertainties. González-Araya et al. [15] studied the tactical harvest planning of apple orchards, involving

the decision about the quantity of resource input and product to be harvested. The objective was to minimize harvesting costs. In their literature review on the supply chain for agricultural products related to crops, Kusumastuti et al. [16] concentrated on the integration of scheduling for harvesting and processing as well as the challenges of the related inventory control. Gómez-Lagos [17] proposed a MILP model to help large export companies manage harvest decisions for multiple orchards. This work focuses on the operational decisions in the picking stage, including which product should be picked by which group and the sequence of products in each group.

Fresh agricultural products are perishable, and their value is closely related to their freshness, which requires maximizing product freshness while meeting cost constraints. Thus, we take the features of agricultural products into account in the Vehicle Routing Problem [18]. Osvald et al. [19] studied the distribution of fresh vegetables with time windows and time-dependent travel times to reduce decay and distribution costs. Chen et al. [20] considered production scheduling and vehicle routing with regard to the time windows of perishable food products. The objective was to maximize the profits of the supplier. Song et al. [21] investigated the distribution of multi-commodity perishable food products while taking consumer satisfaction into account. It is worth mentioning that the vehicles are heterogeneous, involving refrigerated vehicles and general types of vehicles. Wang et al. [22] investigated the fresh agricultural product distribution route with the consideration of minimizing the time window penalty cost as well as maximizing customer satisfaction. In brief, the balance between cost and freshness is rarely considered in the existing literature.

Recent research has considered the integrated production and distribution scheduling problem (IPDSP) in the perishable products industry. The research of Amorim et al. [23] showed that the integrated approach can effectively improve the freshness of perishable products. Geismar et al. [24] studied an IPDSP of a product with short shelf life, where the product was processed at the factory with a single machine and the finished product was delivered to customers before expiration by a truck. The objective was to minimize the overall time. Devapriya et al. [25] expanded the research of Geismar, considering more realistic features, such as the decision of the fleet size and a planning horizon constraint. The scheduling objective was to minimize the cost. Belo-Filho et al. [26] studied an IPDSP of perishable products where the customer order with several products could be processed by multiple parallel capacitated production lines. The objective was to minimize the total operating cost through the joint decision. An adaptive large neighborhood search framework was proposed to deal with the problem. Kergosien et al. [27] investigated the coordinated scheduling of a chemotherapy product in order to decrease the tardiness of delivery. Philippe et al. [28] studied an IPDSP of a single perishable product and proposed a greedy random adaptive search procedure with an evolutionary local search to solve it. It was designed to minimize the overall time of production and distribution systems. Neves-Moreira [29] addressed an IPDSP in the meat supply chain, which comprised a single processing center with several production lines and a fleet of vehicles. The scheduling objective was to minimize global costs. By summarizing the above literature, we can find that research on the integrated production and distribution scheduling in the perishable products manufacturing industry has primarily concentrated on the perishability of products, while failing to investigate the fact that different kinds of agricultural products have different decay rates. Therefore, different from the integrated production and distribution scheduling in the manufacturing industry, where products are processed according to customer orders, it is significant to give priority to picking products according to the varieties when a variety of agricultural products with different decay rates are involved.

Evolutionary algorithms based on population are regarded as an effective approach to solving the integrated scheduling problem. Gharaei et al. [11] presented a novel multi-objective algorithm based on decomposition which had been combined with the bee algorithm (MOEA/D-BA) to solve the integrated scheduling problem in a multi-factory supply chain. Marandi et al. [30] solved the integrated scheduling problem with four different

types of population-based metaheuristics solution approaches, i.e., the Genetic Algorithm, PSO, Improved PSO, and Imperialist Competitive Algorithm. Mohammadi et al. [31] developed a hybrid particle swarm optimization (HPSO) algorithm to solve the integrated scheduling problem for medium and large sizes. In sum, evolutionary algorithms based on population can effectively solve the integrated scheduling problem in a reasonable time. However, one of the limitations of evolutionary algorithms is that they perform poorly in local search. Thus, it is appropriate to combine the evolutionary algorithm with local search. Qin et al. [32] developed an improved Particle Swarm Optimization (PSO) algorithm combined with a local optimal solution jumping mechanism to achieve infrared image enhancement. In order to perform the parameter estimation of photovoltaic systems, Tefek M. F. [33] proposed an ABC-Local Search (ABC-Ls) method by combining the standard artificial bee colony algorithm with a new local search. Ul Hassan N. [34] proposed an improved opposition-based Particle Swarm Optimization algorithm to balance global and local search, so as to achieve global optimization. Saad et al. [35] proposed an evolutionary multi-objective artificial bee colony (EMOABC) algorithm incorporating the characteristics of the simulated evolution (SE) algorithm for improved local search. Liu et al. [36] proposed a modified cuckoo search algorithm with variational parameters and logistic map (VLCS) to avoid falling into a local optimum and improve the rate of convergence. Fu et al. [37] proposed a new multi-objective brain storm optimization algorithm incorporating a stochastic simulation approach to solve a stochastic multi-objective distributed permutation flow shop scheduling problem. Fu et al. [38] proposed a multi-objective multiverse optimization algorithm with stochastic simulation to solve a disassembly sequence planning problem with a stochastic bi-objective.

We can conclude from the materials discussed in this section that previous studies have paid little attention to the integrated picking and distribution scheduling of fresh agricultural products, especially when considering a variety of fresh agricultural products with different decay rates. However, overall optimization can only be achieved through integrated scheduling. Meanwhile, owing to the intricacy of the problem under consideration, an efficient tool, namely metaheuristic methods, has been widely employed to cope with them. Thus, this work studies an integrated scheduling problem of the picking and distribution of fresh agricultural products with the consideration of minimizing the picking and distribution costs as well as maximizing the freshness of the orders. Then, a multi-objective multi-population genetic algorithm with local search (MOPGA-LS) is designed to solve this problem.

3. Problem Statement

Two pivotal stages are contained in the problem under consideration: the picking stage and the distribution stage. There are multiple picking groups with different picking speeds in the picking stage. A variety of agricultural products must be assigned among groups, and each product may only be assigned to one group. All products assigned to a group are picked by the group following the determined sequence. A group can pick only one product at a time, and interruption is not permitted.

The distribution stage can be abstracted into a vehicle routing problem centered on delivering the orders to the customers, where the vehicles have a defined maximum capacity. The CSA farm should determine how many vehicles to use, the vehicle assignments, and the sequence of customers for each trip. Each customer can only be visited once. Notice that the customer order may aggregate a variety of agricultural products that must be picked separately. Once picked, a product will be loaded into the corresponding vehicle according to the distribution scheduling. Then the departure time of each vehicle from the farm is equal to the picking completion time of the last product in the vehicle's cargo.

The freshness of agricultural products decays exponentially with time after picking, and different agricultural products decay at different speeds [39,40]. Based on the existing research on the freshness function, this work constructs a freshness function: $\theta(t) = C - e^{\beta t}$, $0 < \beta$, $0 \leq t \leq \ln C / \beta$, where β represents the decay rate of the agricultural

product, t represents the time between picking and distribution, and C is a constant. In the considered problem, a variety of agricultural products are divided into two types according to their decay rate. The agricultural products with the decay rate β_1 are more sensitive to time, and the others with the decay rate β_2 are less sensitive to time. An integrated picking and distribution schedule in which a farm serves multiple customers is depicted in Figure 1.

Table 1 provides the notations and definitions and the mathematical model is established as follows:

$$\min f_1 = \sum_{i \in J \setminus \{0\}} \sum_{j \in J} \sum_{l \in M} p_{jl} \cdot G_j \cdot t_{jl} \cdot X_{ijl} + \sum_{m \in N} \sum_{n \in N} \sum_{k \in K} \lambda \cdot Z_{mnk} \cdot d_{mn} + F \cdot \sum_{k \in K} \sum_{n \in D} Z_{0nk} \quad (1)$$

$$\max f_2 = \sum_{m \in D} \min \left\{ \min_{j \in J_{1m}} \left\{ C - e^{\beta_1(a_{mk} - c_j)} \right\}, \min_{j \in J_{2m}} \left\{ C - e^{\beta_2(a_{mk} - c_j)} \right\} \right\} \quad (2)$$

s.t

$$\sum_{j \in J \setminus \{0\}} X_{0jl} \leq 1, \forall l \in M. \quad (3)$$

$$\sum_{j \in J} \sum_{l \in M} X_{ijl} = 1, \forall i \in J \setminus \{0\}. \quad (4)$$

$$\sum_{i \in J} X_{ijl} = \sum_{i \in J} X_{jil}, \forall j \in J \setminus \{0\}, \forall l \in M. \quad (5)$$

$$r_i - r_j + \omega \cdot X_{ijl} \leq \omega - 1, \forall i, j \in J \setminus \{0\}, \forall l \in M. \quad (6)$$

$$c_j \geq c_i + t_{jl} \cdot G_j - B \cdot (1 - X_{ijl}), \forall i \in J, \forall j \in J \setminus \{0\}, i \neq j, \forall l \in M. \quad (7)$$

$$a_{0k} \geq c_j - B \cdot (1 - W_{mk}) - B \cdot (1 - q_{mj}), \forall j \in J \setminus \{0\}, \forall m \in D, \forall k \in K. \quad (8)$$

$$\sum_{k \in K} W_{mk} = 1, \forall m \in D. \quad (9)$$

$$W_{nk} = \sum_{m \in N} Z_{mnk}, \forall n \in D, \forall k \in K. \quad (10)$$

$$\sum_{m \in D} \sum_{j \in J} q_{mj} \cdot W_{mk} \leq Q, \forall k \in K. \quad (11)$$

$$\sum_{n \in D} Z_{0nk} = \sum_{m \in D} Z_{m0k} \leq 1, \forall k \in K. \quad (12)$$

$$\sum_{m \in N} Z_{mhk} - \sum_{n \in N} Z_{hmk} = 0, \forall h \in D, \forall k \in K. \quad (13)$$

$$a_{nk} \geq a_{mk} + \frac{d_{mn}}{v} + B \cdot (1 - Z_{mnk}), \forall m \in N, \forall n \in D, \forall k \in K. \quad (14)$$

$$u_m - u_n + \varphi \cdot Z_{mnk} \leq \varphi - 1, \forall m, n \in D, \forall k \in K. \quad (15)$$

$$X_{ijl}, W_{mk}, Z_{mnk} \in \{0, 1\}, \forall i, j \in J, \forall l \in M, i \neq j, \forall m, n \in N, m \neq n, \forall k \in K. \quad (16)$$

$$r_i, u_m, c_j, a_{mk} \geq 0, \forall i, j \in J, \forall m \in N, \forall k \in K. \quad (17)$$

where (1) is to minimize the total cost of picking and distribution. The picking cost mainly contains the labor cost of the picking process, and the distribution cost is made up of both fixed and variable costs. Constraint (2) is to maximize the freshness of all the orders. Constraints (3)–(6) guarantee the order relationship of two adjacent products and that each product should be picked by only one group. Constraint (7) guarantees that a group is allowed to pick only one product at a time and defines the completion time of the products at the picking stage. Constraint (8) specifies that the departure time of each vehicle from the farm is equal to the picking completion time of the last product in the

vehicle's cargo. Constraint (9) denotes that each customer should be assigned to only one vehicle. Constraint (10) restricts each customer in each tour to only one immediate previous customer. Constraint (11) guarantees that the capacity of each vehicle must be met. Constraint (12) specifies that each vehicle starts from the farm and finally returns to it. Constraint (13) indicates a flow balance. Constraint (14) defines the visiting time of customer n by vehicle k . Constraint (15) is employed to eliminate sub-tours in the distribution. Constraints (16) and (17) give the range of variables.

Table 1. Notations and symbols.

| Notations | Descriptions |
|-----------|---|
| | Indices |
| M | Set of groups, $M = \{1, 2, \dots, \gamma\}$, where γ is the number of groups. The group is indexed by the symbol $l \in M$. |
| J | Set of agricultural product varieties, $J = \{0, 1, 2, \dots, \omega\}$, where ω is the total number of agricultural product varieties and 0 is a dummy product. The products are indexed by symbols $i, j \in J$. |
| J_{1m} | Set of agricultural product varieties with decay rate β_1 required by customer m . |
| J_{2m} | Set of agricultural product varieties with decay rate β_2 required by customer m . |
| D | Set of customers, $D = \{1, 2, \dots, \varphi\}$, where φ is the total number of customers. The customers are indexed by symbols $m, n \in D$. |
| N | Set of the farm and customers, $N = \{0\} \cup D$, where 0 denotes the farm. |
| K | Set of vehicles, $K = \{1, 2, \dots, \delta\}$, where δ is the number of vehicles, which are indexed by symbols $k \in K$. |
| | Parameters |
| q_{mj} | Demand for product j of customer m . |
| G_j | Total demand for product j , $G_j = \sum_{m \in D} q_{mj}, \forall j \in J$. |
| U_{mj} | Customer-product matrix, which takes value 1 if customer m requires product j , and 0 otherwise. |
| t_{jl} | Picking time of product j with group l . |
| p_{jl} | Unit picking cost of product j with group l . |
| d_{mn} | Distance between customers m and n . |
| λ | Variable cost of the vehicle per unit of time. |
| v | Speed of a vehicle. |
| F | Fixed cost of a vehicle. |
| Q | Capacity of a vehicle. |
| B | An infinite constant. |
| | Decision variables |
| X_{ijl} | A binary variable that takes 1 if product j is picked immediately after product i by group l , and 0 otherwise. |
| W_{mk} | A binary variable that takes 1 if customer m is delivered by vehicle k , and 0 otherwise. |
| Z_{mnk} | A binary variable that takes 1 if customer n is delivered after customer m by vehicle k , and 0 otherwise. |
| r_i | Auxiliary variable that is employed to eliminate sub-tour in picking. |
| u_m | Auxiliary variable that is employed to eliminate sub-tour in distribution. |
| c_j | Picking completion time of product j . |
| a_{mk} | Visiting time of customer m by vehicle k . |

The problem under consideration contains two pivotal types of problem: unrelated parallel machine scheduling and vehicle routing problems. Both of them have proven to be NP-hard, and the considered problem is more complicated because it simultaneously optimizes two conflicting criteria. Consequently, it is also NP-hard, and we propose the MOPGA-LS to deal with it.

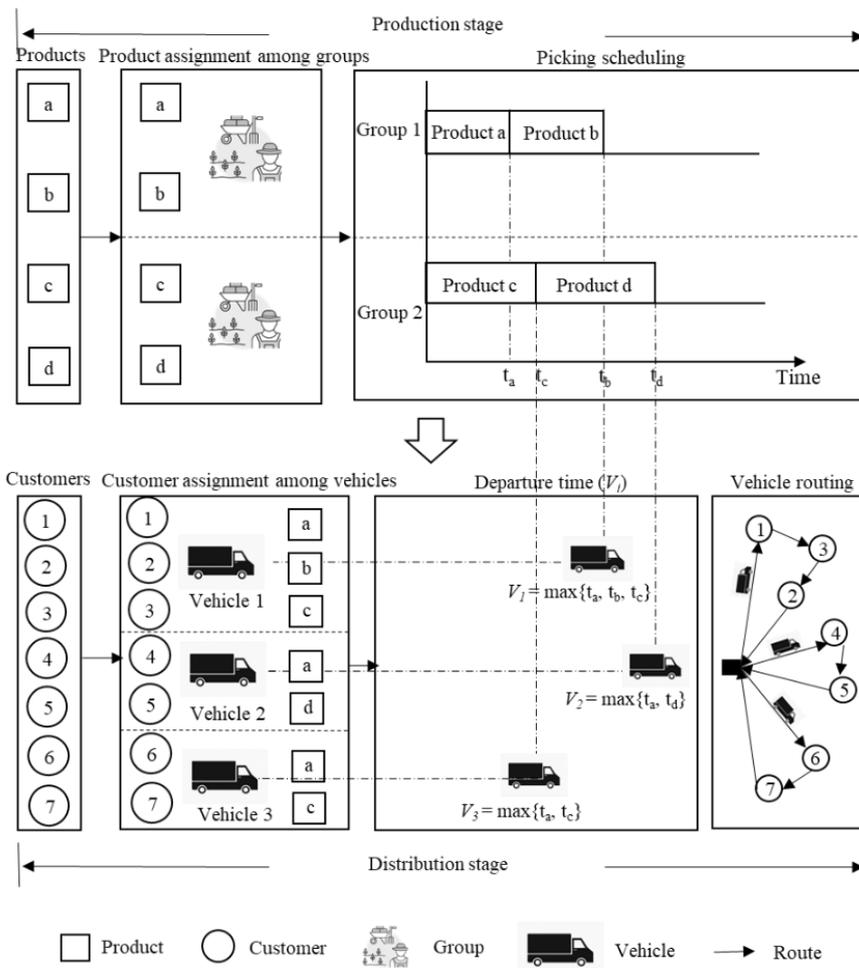


Figure 1. An example of integrated picking and distribution scheduling.

4. Proposed Algorithm

Multi-objective optimization problems have multiple Pareto optimal solutions, and the goal of solving them is to find as many Pareto optimal solutions as possible. Evolutionary algorithms can find a set of solutions with a single iteration, and thus they have been popularly regarded as an effective approach to solving multi-objective problems [41]. Extensive experimental comparisons verify their excellent performance in addressing complicated multi-objective optimization problems [42]. Motivated by their successful applications, this work proposes the MOPGA-LS to settle the problem under study.

4.1. Solution Representation

To solve the integrated picking and distribution problem, we need to make four decisions, i.e., the product assignment among groups, the picking sequence of the products at the picking stage, the customers served by each of the vehicles, and their delivery routes at the distribution stage. Thus, this work uses an integer string to represent a solution, i.e., a chromosome which consists of three parts. Part *a* represents the picking sequence, where each integer indicates a product index. Part *b* denotes the product assignment among groups, where each integer indicates the number of varieties picked by each group. Through parts *a* and *b*, we can clearly know which group picks which agricultural product and the picking sequence within each team. As is shown in Figure 2, group 1 picks two products in the order of product 3 and 6, group 2 picks three products in the order of product 5, 2, and 7, and group 3 picks two products in the order of product 1 and 4. Part *c* indicates the distribution sequence, where each integer indicates a customer. Notice that Part *c* just gives the delivery sequence of the customers and fails to provide the customer

assignment among vehicles. This work employs a decoding rule for Part *c* to minimize the number of vehicles [43]. In this method, the customers are sequentially assigned to a vehicle as their delivery sequence. If the load of a vehicle exceeds the maximum capacity, a new vehicle comes into use. Figure 2 illustrates the decoding rule. The capacity of the vehicle is set to 6. Thus, vehicle 1 serves customer 4, vehicle 2 serves three customers in the order of customer 3, 2, and 6, and vehicle 3 serves two customers in the order of customer 5 and 1. It is worth mentioning that the above encoding method avoids the generation of infeasible solutions.

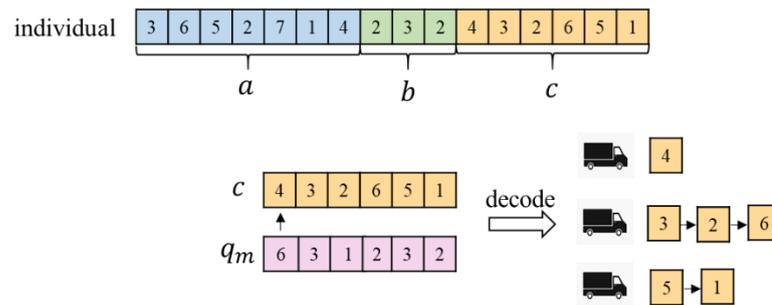


Figure 2. An example of coding.

4.2. Population Initialization

Two heuristic rules combine with the random generation mode to construct an initial population. This work employs two heuristic rules to generate an individual, which are, respectively, named the Sensitivity Priority Rule and the Mileage Saving Method [44]. The details are as follows.

The Sensitivity Priority Rule: In a group, all products are divided into two types J_1 and J_2 according to their decay rate. The products in J_1 are more sensitive to time and are sorted in non-decreasing order according to picking weights. The products in J_2 are less sensitive to time. Then the group first picks the products in J_2 as their original picking sequence, and finally picks the products in J_1 as their sorted sequence. The details are presented in Figure 3.

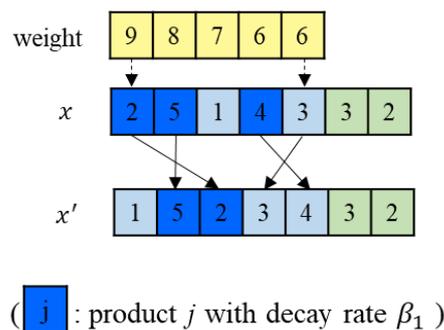


Figure 3. The sensitivity priority rule.

The Mileage Saving Method: Its core idea is to generate the initial solution with the shortest total distribution distance under the constraint that the vehicle has the maximum load limit. Its main approach is to make the two loops existing in transport problems $(0, m, 0)$ and $(0, n, 0)$ into one loop $(0, m, n, 0)$, reducing the total transport distance.

4.3. Multi-Population Construction

The solutions obtained in this work are called non-dominated solutions, i.e., Pareto solutions, because they simultaneously optimize two conflicting criteria. Thus, Fast Non-dominated Sorting (FNS), a crucial component of NSGA-II, is employed to sort a population into different non-domination levels [9], as is shown in Figure 4. The considered problem

is designed to minimize the picking and distribution cost and maximize the freshness of the orders. In the process of calculation, we take the reciprocal of the value of the second objective function. Then a population P in MOPGA-LS can be sorted into different non-domination levels (P_1, P_2, \dots, P_h). Based on this, we can perform a global search and local search, respectively, in different non-domination levels, i.e., perform a global search in all non-domination levels and perform a local search in non-domination level 1. Furthermore, the MOPGA-LS utilizes multiple non-domination levels to improve genetic operation. In short, it performs crossover operations on more promising solutions to enhance the algorithm's efficiency. The details are as follows.

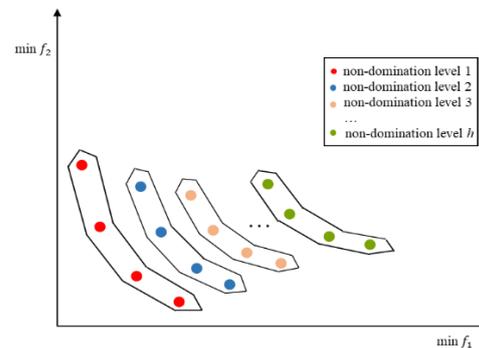


Figure 4. Ranking method.

4.4. Selection, Improved Crossover, and Mutation

The selection mechanism is described as follows. Firstly, through the FNS method, the Pareto-based rank and crowding distance are assigned to each individual in the parent population. Then, two individuals are selected randomly from the population every time, and compare their rank values. The individual with a smaller rank value is copied to an offspring. If the rank values of two individuals are the same, the individual with the bigger crowding degree value is copied to the next generation. Finally, repeat the above operations until the offspring population size reaches the original population size.

The individuals in the offspring population can be divided into the multi-population according to the Pareto-based rank. Based on this, MOPGA-LS improves the crossover operation. The following are the specific steps.

Step 1: Determine the non-domination level where the individuals for crossover are located. Randomly generate two integers in $[1, h]$, where h represents the number of non-domination levels. Then, the smaller integer value represents the non-domination level.

Step 2: Randomly select an individual as one of the parent individuals for crossover in the selected non-domination level.

Step 3: Repeat Steps 2 to 3 to generate another parent individual for crossover.

Step 4: Two parent individuals perform a partially mapped crossover.

The mutation takes place immediately after the improved crossover. A random number in $[0, 1]$ is generated. Perform the mutation operation if the generated number is equal to or smaller than r_m , where r_m represents the mutation probability. The three mutation operations are as follows.

(1) Randomly select a product in Part a , delete it from the current location, and then insert it into a location randomly.

(2) Randomly select two points (denoted as l_1, l_2) in Part b . If $l_1 < l_2$ and $l_1 \neq l_2$, then $l_1 = l_1 + 1$, and $l_2 = l_2 - 1$; otherwise, $l_1 = l_1 - 1$, and $l_2 = l_2 + 1$.

(3) Randomly select a customer in Part c , delete it from the current location, and then insert it to a location randomly.

Finally, the MOPGA-LS forms the next-generation population according to the elite-preservation strategy [9].

4.5. Local Search

After performing genetic operations, this work refines the non-dominated individuals by using a local search method (LSM) with probability p_l , $p_l = \tau_c / \tau$, where τ_c is the current number of function evaluations, and τ is the maximum number of function evaluations. If a random number r is equal to or smaller than p_l , the LSM is allowed to be performed. It uses a colling process like the simulated annealing method to avoid falling into a local optimum. To efficiently use the computation resource, the LSM is allowed to refine S individuals at most. Different from other similar works, this work combines a local search with a simulated annealing algorithm to determine the number of neighborhood solutions and maintain the diversity of the neighborhood solutions, so as to avoid falling into local optima and improve the performance of the proposed algorithm.

To generate a neighbor individual, this work designs five neighborhood structures as follows, and Figure 5 illustrates them.

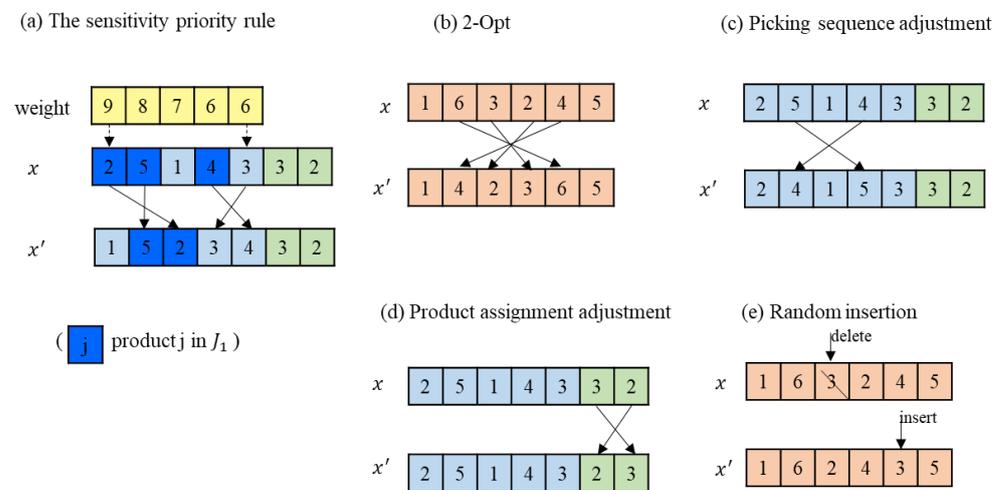


Figure 5. Neighborhood operations.

- (1) The Sensitivity Priority Rule: The details are described in Section 4.2.
- (2) 2-Opt: Select a segment in Part c of an individual and reverse the segment. A concrete example can be given in Figure 5, where the segment $\{6, 3, 2, 4\}$ in Part c is reversed to $\{4, 2, 3, 6\}$.
- (3) Product assignment adjustment: Randomly select two points in Part b to exchange.
- (4) Picking sequence adjustment: Randomly select two products in Part a to exchange.
- (5) Random insertion: Randomly select a customer in Part c , delete it from the current location, and then insert it to a location randomly.

The metropolis acceptance principle [45] is adopted to make the LSM jump out of the local optima. It is necessary to calculate the difference between the neighbor solution x' and the initial solution x of the first and second objective function values, i.e., $\Delta 1 = f_1(x') - f_1(x)$ and $\Delta 2 = f_2(x') - f_2(x)$. In addition, P_t , P_c and r are required, where $P_c = \exp(-\Delta 1 / KT)$, $P_t = \exp(-\Delta 2 / KT)$, r is a uniformly distributed random number, and the Boltzmann constant $K = 1$. Given that a bi-objective is being evaluated, there are five conditions for dealing with the new neighbor solution x' :

- (1) $\Delta 1 \leq 0$ and $\Delta 2 \leq 0$: x' is accepted;
- (2) $\Delta 1 > 0$ and $\Delta 2 \leq 0$: x' is accepted only if $r < P_t$;
- (3) $\Delta 1 \leq 0$ and $\Delta 2 > 0$: x' is accepted only if $r < P_c$;
- (4) $\Delta 1 > 0$, $\Delta 2 > 0$ and $\Delta 1 / f_1(x') < \Delta 2 / f_2(x')$: x' is accepted only if $r < P_t$;
- (5) $\Delta 1 > 0$, $\Delta 2 > 0$ and $\Delta 1 / f_1(x') \geq \Delta 2 / f_2(x')$: x' is accepted only if $r < P_c$.

The procedures of the LSM are as follows:

Step 1: Initialize the parameters, involving the maximum number of individuals selected for local search S_M , the initial temperature T_0 , the final temperature T_F , the coefficient that controls the cooling schedule α , and $T = T_0$.

- Step 2: Select an individual x randomly in R_1 .
- Step 3: Utilize five neighborhoods to create an individual x' based on x . If x' meets the Metropolis acceptance principle, then $\sigma = \sigma \cup \{x'\}$.
- Step 4: $T = T \cdot \alpha$.
- Step 5: Steps 3 to 4 must be repeated until T_F is reached.
- Step 5: If a specified termination criterion is fulfilled, output σ ; otherwise, go to Step 2.
- Step 6: Update the population by selecting Z individuals in σ randomly to return.

4.6. Procedure of MOPGA-LS

Having discussed the components of MOPGA-LS, we next summarize the concrete steps of MOPGA-LS, which are shown in Figure 6.

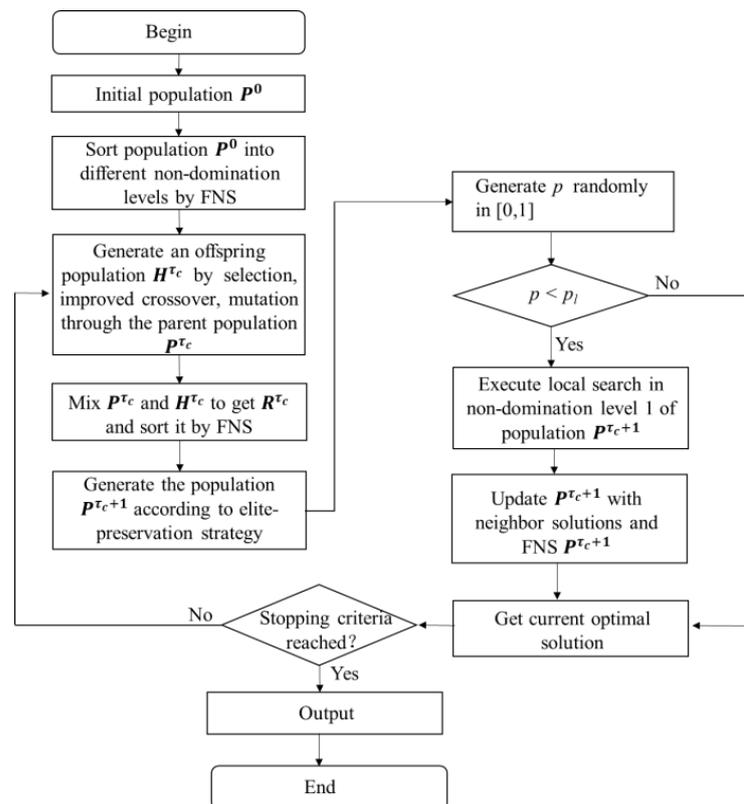


Figure 6. Flowchart of MOPGA-LS.

Step 1: Initialize the parameters of MOPGA-LS, such as the population size ρ , the mutation probability r_m , and the maximum number of function evaluations τ .

Step 2: Initialize the population, calculate the fitness and perform the Fast Non-dominated Sorting.

Step 3: Generate an offspring population by performing genetic operations.

Step 4: Generate a new generation population according to the elite-preservation strategy.

Step 5: If $p < P_l$, refine the non-dominated individuals by using a local search method, and perform the Fast Non-dominated Sorting.

Step 6: If a termination requirement is reached, that is the maximum number of function evaluations, so export non-dominated solutions; otherwise, go to Step 3.

5. Computational Experiments

Experiments are conducted on 32 different sets of benchmark instances. This work selects three multi-objective evolutionary algorithms: NSGA-II, MOEA/D, and MOEA/D-BA, as the peer approaches for comparisons. NSGA-II and MOEA/D are classic multi-objective

evolutionary algorithms that can handle routing planning and job shop scheduling [46,47]. MOEA/D-BA is also used to handle the integrated production and distribution scheduling problem that includes two pivotal problems: parallel machine scheduling and vehicle routing problems [11].

5.1. Test Instance Generation

There are no instances suitable for the problem under consideration, which contains a picking stage and a distribution stage, in the existing literature. Therefore, a set of instances in the picking stage are generated by using the approach described by Belo-Filho [27], and the distribution stage benchmark may be obtained at <http://www.coin-or.org/SYMPHONY/branchandcut/VRP/data/index.htm.old> (accessed on 1 August 2022). Table 2 outlines the information about the picking groups, mainly involving their unit picking speed and unit picking cost. The unit picking speed follows a uniform distribution and the unit picking cost is a constant. The fixed cost of utilizing each vehicle F is set to 150, the variable cost of the vehicle per unit of time λ is set to 1.5, and the vehicle speed is set to 30.

Table 2. Information on groups.

| Groups | Unit Picking Speed of Groups | Unit Picking Cost of Groups |
|--------|------------------------------|-----------------------------|
| 1 | $U[0.001, 0.005]$ | 100 |
| 2 | $U[0.006, 0.01]$ | 90 |
| 3 | $U[0.011, 0.015]$ | 80 |
| 4 | $U[0.016, 0.02]$ | 70 |
| 5 | $U[0.021, 0.025]$ | 60 |

The size of the test instances is determined by four factors, including the number of groups, the number of products with decay rate β_1 , the number of products with decay rate β_2 , and the number of customers. This work tests eight different sizes, and the details are described in Table 3. There are four instances for each size. Therefore, this work tests a total of 32 instances to evaluate the performance of the algorithm. In the following experiments, each algorithm runs 20 times for each instance, and the mean value is calculated to evaluate its performance. All the algorithms stop when $3 * |M| * |J| * |D|$ function evaluations are used up, where M represents the number of groups, J represents the number of products, and D represents the number of customers.

Table 3. Features of test instances.

| | Groups | Products with Decay Rate β_1 | Products with Decay Rate β_2 | Customers |
|-------------|--------|------------------------------------|------------------------------------|-----------|
| M2-J20-D20 | 2 | 10 | 10 | 20 |
| M2-J30-D40 | 2 | 15 | 15 | 40 |
| M3-J30-D40 | 3 | 15 | 15 | 40 |
| M3-J40-D60 | 3 | 20 | 20 | 60 |
| M4-J40-D60 | 4 | 20 | 20 | 60 |
| M4-J50-D80 | 4 | 25 | 25 | 80 |
| M5-J50-D80 | 5 | 25 | 25 | 80 |
| M5-J60-D100 | 5 | 30 | 30 | 100 |

5.2. Performance Metrics

This work selects two performance metrics, i.e., the hypervolume-metric and the IGD-metric, to compare the performance of different algorithms. HV and IGD are by far the most accepted performance metrics, as is proved by Riquelme N. (2015) [48] et al. In addition, this work utilizes two statistical methods, i.e., a t -test and a u -test, to analyze the statistical significance of the results obtained using the MOPGA-LS and the algorithms

in the comparison pool. The symbols +, −, and ~ denote statistical results with a significance level of 0.05. The results are given as +, −, or ~ if the MOPGA-LS is significantly better than, significantly worse than, or equivalent to each of the three multi-objective evolutionary algorithms.

(1) The Hypervolume-metric: The HV measures the size of the objective space covered by an approximation set. A reference point must be used to calculate the mentioned covered space. In our experiments, (1, 1) is set to a reference point and all the objective vectors in approximation sets are normalized into [0, 1]. The bigger the HV, the better the obtained non-dominated solution set.

(2) Inverted Generational Distance: IGD calculates the minimum Euclidean distance between an approximation set and the Pareto optimal front. Because the Pareto optimal front of the investigated problem is unknown, the set of non-dominated solutions obtained by the four algorithms in all runs is regarded as the approximate Pareto optimal front, which refers to Saibal Majumder [49]. The specific steps are as follows.

Step 1: Mix all the solutions obtained by 20 independent executions of each of the algorithms.

Step 2: Perform the Fast Non-dominated Sorting of all the solutions to obtain the set of non-dominated solutions, which is regarded as the approximate Pareto optimal front.

5.3. Parameter Setting

To obtain the optimal parameter combination of the MOPGA-LS, an orthogonal experiment is carried out on the test instance with 3 groups, 15 products with decay rate β_1 , 15 products with decay rate β_2 , and 40 customers. The population size Q , the number of individuals selected for local search S , the simulated annealing rate α , and the number of neighbor solutions selected for return Z are the four most important parameters influencing algorithm performance. Each parameter has four levels, where $Q = \{50, 75, 100, 125\}$, $S = \{3, 6, 9, 12\}$, $\alpha = \{0.2, 0.4, 0.6, 0.8\}$, and $Z = \{5, 10, 15, 20\}$. Thus, an orthogonal array $L_{16}(4^4)$ is selected to carry out the orthogonal experiment. In addition, the mutation rate, the initial temperature T_0 , and the final temperature T_F in the MOPGA-LS are set to 0.25, 0.8, and 1500.

This experiment selects the hypervolume-metric as the performance metric [39]. The bigger the hypervolume, the better the obtained non-dominated solution set. The MOPGA-LS runs 20 times for each parameter combination independently, and the response value (RV) is calculated according to the average Hypervolume value over 20 independent runs. Additionally, the termination criterion, namely the maximum number of function evaluations, is set to $3 * |M| * |J| * |D|$, where M represents the number of groups, J represents the number of products, and D represents the number of customers. The results of the orthogonal experiment are displayed in Table 4, and the parameters' significance rankings are depicted in Table 5. Figure 7 illustrates the factor level trend of each parameter vividly. Based on these findings, we conclude that Q is the most important parameter and α has minimal impact on the algorithm's performance. Therefore, we conclude that, when $Q = 75$, $S = 3$, $\alpha = 0.8$, and $Z = 10$, the MOPGA-LS performs best.

The optimal parameter combination of NSGA-II, MOEA/D, and MOEA/D-BA is also determined by the orthogonal experiment. In NSGA-II, the population size, crossover probability, and mutation probability are the three key parameters. The factor level and the experimental result is shown in Figure 8. Therefore, we conclude that when the population size = 50, the crossover probability = 0.6, and the mutation probability = 0.8, NSGA-II performs best. In MOEA/D, the population size, neighborhood size, and crossover probability are the three key parameters. The factor level and the experimental result are shown in Figure 9. Therefore, we conclude that when the population size = 125, the neighborhood size = 15, and the crossover probability = 0.8, MOEA/D performs best. In MOEA/D-BA, the population size, neighborhood size, and a parameter related to the number of recruited bees are the three key parameters. The factor level and the experimental result is shown in Figure 10. Therefore, we conclude that when the population

size = 125, the neighborhood size = 10, and the parameter related to the number of recruited bees = 0.6, MOEA/D-BA performs best.

Table 4. Results from all parameter combinations of the proposed algorithm.

| ID | Q | S | α | Z | RV |
|----|-----|----|----------|----|--------|
| 1 | 50 | 3 | 0.2 | 5 | 0.7743 |
| 2 | 50 | 6 | 0.4 | 10 | 0.7668 |
| 3 | 50 | 9 | 0.6 | 15 | 0.7512 |
| 4 | 50 | 12 | 0.8 | 20 | 0.7531 |
| 5 | 75 | 3 | 0.4 | 15 | 0.7704 |
| 6 | 75 | 6 | 0.2 | 20 | 0.7598 |
| 7 | 75 | 9 | 0.8 | 5 | 0.7775 |
| 8 | 75 | 12 | 0.6 | 10 | 0.7710 |
| 9 | 100 | 3 | 0.6 | 20 | 0.7420 |
| 10 | 100 | 6 | 0.8 | 15 | 0.7391 |
| 11 | 100 | 9 | 0.2 | 10 | 0.7332 |
| 12 | 100 | 12 | 0.4 | 5 | 0.7117 |
| 13 | 125 | 3 | 0.8 | 10 | 0.7205 |
| 14 | 125 | 6 | 0.6 | 5 | 0.6420 |
| 15 | 125 | 9 | 0.4 | 20 | 0.7093 |
| 16 | 125 | 12 | 0.2 | 15 | 0.6814 |

Table 5. Average response value and significance ranking.

| Level | Q | S | α | Z |
|-------|--------|--------|----------|--------|
| 1 | 0.7613 | 0.7518 | 0.7372 | 0.7264 |
| 2 | 0.7697 | 0.7269 | 0.7396 | 0.7479 |
| 3 | 0.7315 | 0.7428 | 0.7265 | 0.7355 |
| 4 | 0.6883 | 0.7293 | 0.7475 | 0.7411 |
| Delta | 0.0814 | 0.0249 | 0.0210 | 0.0215 |
| rank | 1 | 2 | 4 | 3 |

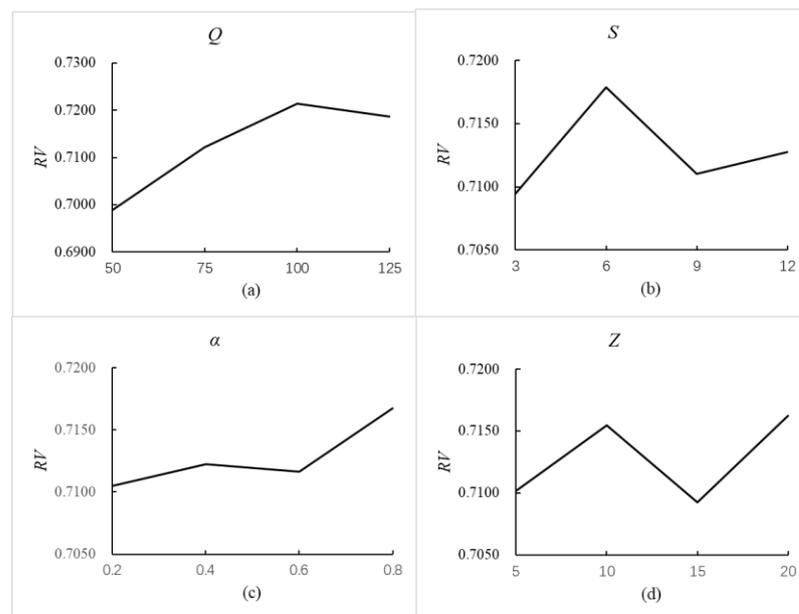


Figure 7. Trend of parameter level.

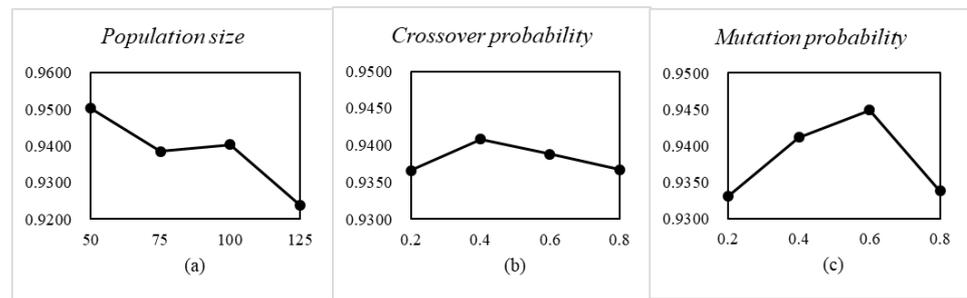


Figure 8. The orthogonal experimental result of NSGA-II.

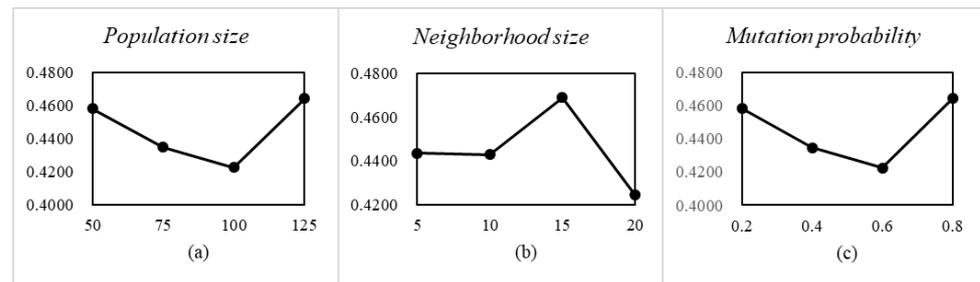


Figure 9. The orthogonal experimental result of MOEA/D.

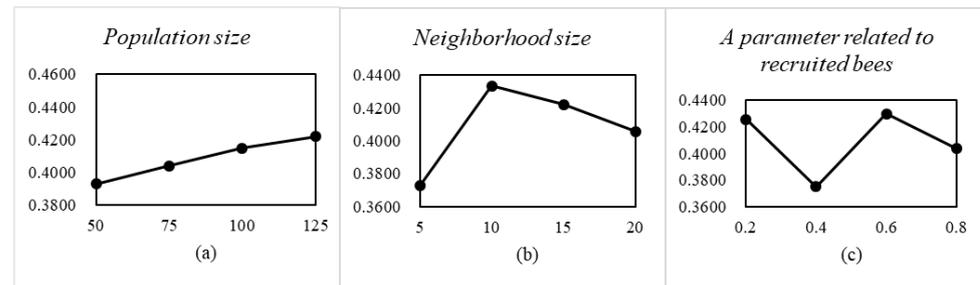


Figure 10. The orthogonal experimental result of MOEA/D-BA.

5.4. Experimental Results

This work makes a comparison among four algorithms in terms of their hypervolume-metric and IGD-metric. The experimental results are as follows. Table 6 shows the average of the HV-metric over 20 independent runs of the four algorithms. Though a *t*-test, we can conclude that the MOPGA-LS greatly outperforms NSGA-II in 28 of the 32 test instances, is equivalent to NSGA-II in the other four, and outperforms both MOEA/D and MOEA/D-BA. Though a *U*-test, we can conclude that the MOPGA-LS greatly outperforms the NSGA-II in 27 of the 32 test instances, is equivalent to the NSGA-II in the other five, and outperforms both MOEA/D and MOEA/D-BA. In addition, the boxplots of the hypervolume-metric, which vividly show the efficiency of the MOPGA-LS, are shown in Figure 11.

Table 7 shows the average of IGD-metric over 20 independent runs of the four algorithms. The results of the *t*-test and *U*-test show that the MOPGA-LS outperforms NSGA-II, MOEA/D and MOEA/D-BA. Similarly, the boxplots of the IGD-metric are shown in Figure 12.

Table 6. Experimental results of HV-metric.

| Set Name | Ins. | MOPGA-LS | NSGA-II | | MOEA/D | | | MOEA/D-BA | | | |
|-------------|------|----------|---------|--------|--------|---------|--------|-----------|---------|--------|--------|
| | | Average | Average | t-Test | U-Test | Average | t-Test | U-Test | Average | t-Test | U-Test |
| M2-J20-D20 | 1 | 0.9025 | 0.8671 | + | + | 0.7092 | + | + | 0.1965 | + | + |
| | 2 | 0.8963 | 0.8455 | + | + | 0.7260 | + | + | 0.2107 | + | + |
| | 3 | 0.8962 | 0.7470 | + | + | 0.6230 | + | + | 0.2050 | + | + |
| | 4 | 0.9289 | 0.7911 | + | + | 0.6358 | + | + | 0.1965 | + | + |
| M2-J30-D40 | 1 | 0.8981 | 0.7857 | + | + | 0.6746 | + | + | 0.2172 | + | + |
| | 2 | 0.8974 | 0.7503 | + | + | 0.6192 | + | + | 0.1759 | + | + |
| | 3 | 0.9550 | 0.8075 | + | + | 0.7168 | + | + | 0.1845 | + | + |
| | 4 | 0.8881 | 0.7648 | + | + | 0.6271 | + | + | 0.2136 | + | + |
| M3-J30-D40 | 1 | 0.9558 | 0.8978 | + | + | 0.8235 | + | + | 0.3552 | + | + |
| | 2 | 0.9423 | 0.9018 | + | + | 0.8410 | + | + | 0.3850 | + | + |
| | 3 | 0.9284 | 0.8929 | + | + | 0.8311 | + | + | 0.1566 | + | + |
| | 4 | 0.9258 | 0.8638 | + | + | 0.8026 | + | + | 0.1772 | + | + |
| M3-J40-D60 | 1 | 0.9451 | 0.9093 | + | + | 0.8689 | + | + | 0.1746 | + | + |
| | 2 | 0.9514 | 0.9287 | + | + | 0.8948 | + | + | 0.3135 | + | + |
| | 3 | 0.9653 | 0.9399 | + | + | 0.9072 | + | + | 0.1848 | + | + |
| | 4 | 0.9569 | 0.9336 | + | + | 0.8963 | + | + | 0.3169 | + | + |
| M4-J40-D60 | 1 | 0.9842 | 0.9590 | + | + | 0.9339 | + | + | 0.1181 | + | + |
| | 2 | 0.9808 | 0.9678 | + | + | 0.9463 | + | + | 0.2780 | + | + |
| | 3 | 0.9835 | 0.9679 | + | + | 0.9478 | + | + | 0.1179 | + | + |
| | 4 | 0.9816 | 0.9613 | + | + | 0.9358 | + | + | 0.2762 | + | + |
| M4-J50-D80 | 1 | 0.9904 | 0.9842 | + | + | 0.9725 | + | + | 0.1804 | + | + |
| | 2 | 0.9923 | 0.9884 | + | + | 0.9793 | + | + | 0.1857 | + | + |
| | 3 | 0.9917 | 0.9860 | + | + | 0.9762 | + | + | 0.1116 | + | + |
| | 4 | 0.9910 | 0.9851 | + | + | 0.9746 | + | + | 0.1121 | + | + |
| M5-J50-D80 | 1 | 0.9944 | 0.9883 | + | + | 0.9799 | + | + | 0.1994 | + | + |
| | 2 | 0.9938 | 0.9867 | + | + | 0.9801 | + | + | 0.1083 | + | + |
| | 3 | 0.9949 | 0.9936 | ~ | ~ | 0.9889 | + | + | 0.1119 | + | + |
| | 4 | 0.9944 | 0.9863 | + | + | 0.9844 | + | + | 0.1092 | + | + |
| M5-J60-D100 | 1 | 0.9920 | 0.9885 | + | ~ | 0.9845 | + | + | 0.1279 | + | + |
| | 2 | 0.9951 | 0.9944 | ~ | ~ | 0.9908 | + | + | 0.1268 | + | + |
| | 3 | 0.9956 | 0.9947 | ~ | ~ | 0.9915 | + | + | 0.1270 | + | + |
| | 4 | 0.9943 | 0.9936 | ~ | ~ | 0.9901 | + | + | 0.2052 | + | + |

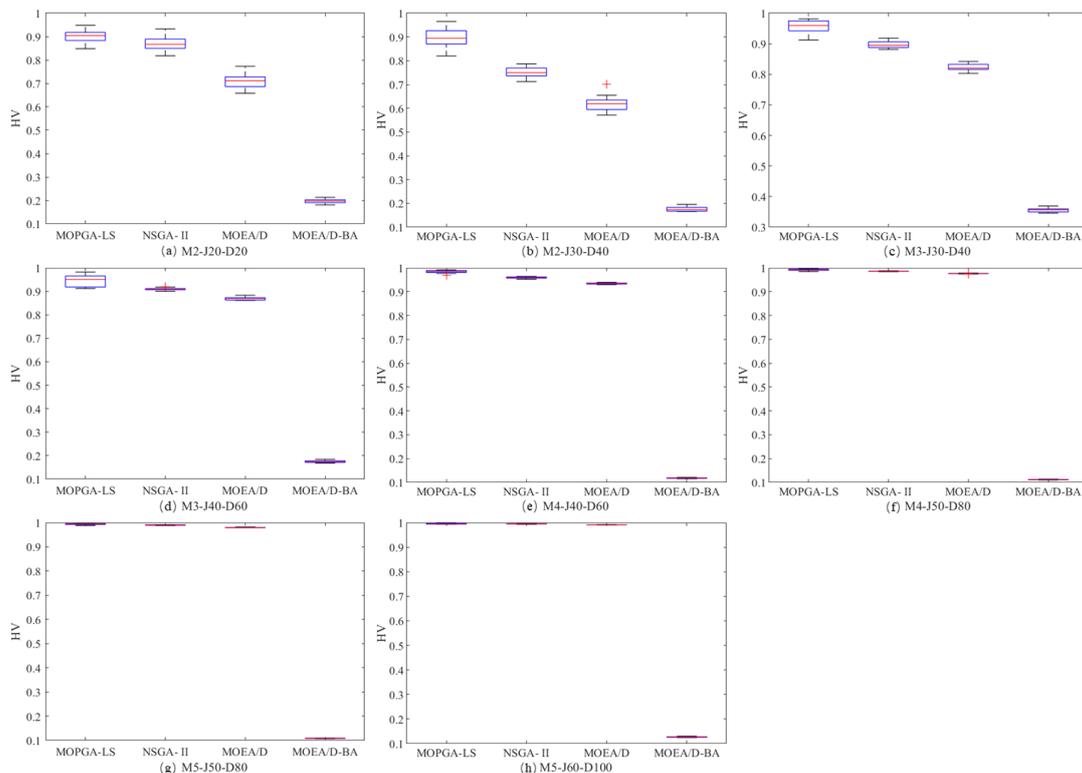


Figure 11. Boxplot of HV metric.

Table 7. Experimental results of IGD-metric.

| Set Name | Ins. | MOPGA-LS | | NSGA-II | | MOEA/D | | | MOEA/D-BA | | |
|-------------|------|----------|---------|---------|--------|---------|--------|--------|-----------|--------|--------|
| | | Average | Average | t-Test | U-Test | Average | t-Test | U-Test | Average | t-Test | U-Test |
| M2-J20-D20 | 1 | 0.1113 | 0.1931 | + | + | 0.7035 | + | + | 0.4493 | + | + |
| | 2 | 0.1231 | 0.3256 | + | + | 0.8105 | + | + | 0.6037 | + | + |
| | 3 | 0.1835 | 0.9014 | + | + | 1.6756 | + | + | 1.1924 | + | + |
| | 4 | 0.2830 | 0.3093 | + | + | 0.6614 | + | + | 0.5101 | + | + |
| M2-J30-D40 | 1 | 0.1541 | 0.4488 | + | + | 0.9139 | + | + | 0.7899 | + | + |
| | 2 | 0.1511 | 0.6358 | + | + | 1.2209 | + | + | 0.9799 | + | + |
| | 3 | 0.1354 | 0.4292 | + | + | 0.6814 | + | + | 0.7185 | + | + |
| | 4 | 0.3099 | 0.7114 | + | + | 1.0881 | + | + | 1.1647 | + | + |
| M3-J30-D40 | 1 | 0.0678 | 0.2347 | + | + | 0.5242 | + | + | 0.4527 | + | + |
| | 2 | 0.1557 | 0.3364 | + | + | 0.6253 | + | + | 0.6201 | + | + |
| | 3 | 0.0775 | 0.1632 | + | + | 0.3281 | + | + | 0.3093 | + | + |
| | 4 | 0.1423 | 0.4104 | + | + | 0.7467 | + | + | 0.7239 | + | + |
| M3-J40-D60 | 1 | 0.0604 | 0.1639 | + | + | 0.3129 | + | + | 0.3109 | + | + |
| | 2 | 0.1706 | 0.4405 | + | + | 0.8525 | + | + | 0.8265 | + | + |
| | 3 | 0.1034 | 0.3151 | + | + | 0.6426 | + | + | 0.6259 | + | + |
| | 4 | 0.0628 | 0.1331 | + | + | 0.2711 | + | + | 0.2797 | + | + |
| M4-J40-D60 | 1 | 0.0284 | 0.1204 | + | + | 0.2581 | + | + | 0.2600 | + | + |
| | 2 | 0.0534 | 0.1184 | + | + | 0.2507 | + | + | 0.3003 | + | + |
| | 3 | 0.0424 | 0.1405 | + | + | 0.2848 | + | + | 0.2868 | + | + |
| | 4 | 0.0402 | 0.1116 | + | + | 0.2227 | + | + | 0.2289 | + | + |
| M4-J50-D80 | 1 | 0.0304 | 0.0721 | + | + | 0.1745 | + | + | 0.1806 | + | + |
| | 2 | 0.0299 | 0.0711 | + | + | 0.1826 | + | + | 0.1791 | + | + |
| | 3 | 0.0246 | 0.0653 | + | + | 0.1563 | + | + | 0.1605 | + | + |
| | 4 | 0.0454 | 0.1140 | + | + | 0.2444 | + | + | 0.2924 | + | + |
| M5-J50-D80 | 1 | 0.0205 | 0.0594 | + | + | 0.1647 | + | + | 0.1594 | + | + |
| | 2 | 0.0278 | 0.0652 | + | + | 0.2020 | + | + | 0.2071 | + | + |
| | 3 | 0.0320 | 0.0649 | + | + | 0.1962 | + | + | 0.2043 | + | + |
| | 4 | 0.0370 | 0.0782 | + | + | 0.2372 | + | + | 0.2260 | + | + |
| M5-J60-D100 | 1 | 0.0179 | 0.0488 | + | + | 0.1304 | + | + | 0.1461 | + | + |
| | 2 | 0.0339 | 0.0614 | + | + | 0.2236 | + | + | 0.2091 | + | + |
| | 3 | 0.0190 | 0.0370 | + | + | 0.1379 | + | + | 0.1412 | + | + |
| | 4 | 0.0183 | 0.0355 | + | + | 0.1116 | + | + | 0.1191 | + | + |

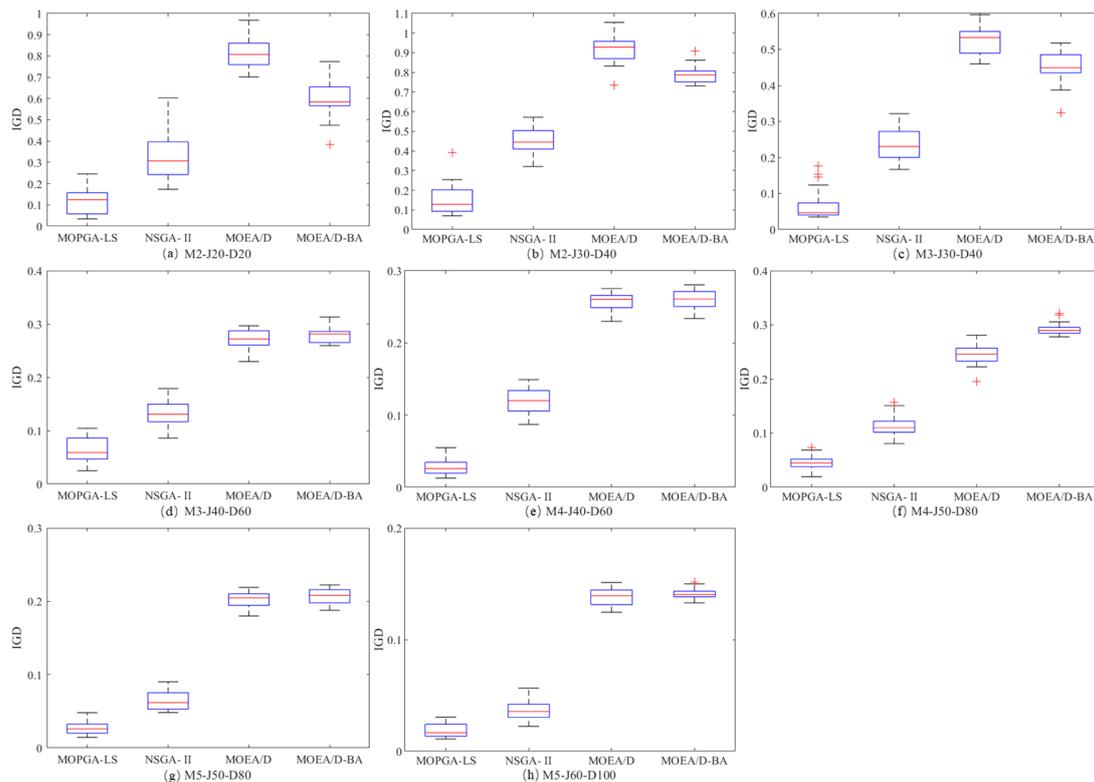


Figure 12. Boxplot of IGD metric.

Finally, the distribution of these non-dominated solution sets obtained by the four algorithms in the instance of 3 groups, 15 products with decay rate β_1 , 15 products with decay rate β_2 , and 40 customers is depicted in Figure 13. It is clear that the solution sets obtained by the MOPGA-LS have superior approximation and distribution to those obtained by the other algorithms. Therefore, we conclude that the proposed algorithm is a superior optimizer for addressing the integrated picking and distribution problem of agricultural products. The experimental results in terms of the hypervolume-metric and IGD-metric show that the MOPGA-LS has better approximation and distribution than other algorithms as peers. The advantages of the MOPGA-LS over NSGA-II, MOEA/D, and MOEA/D-BA illustrate that the multi-population effectively balances the global search and local search. In addition, the improved crossover operation and the combination of local search and the simulated annealing algorithm improve the efficiency of the MOPGA-LS.

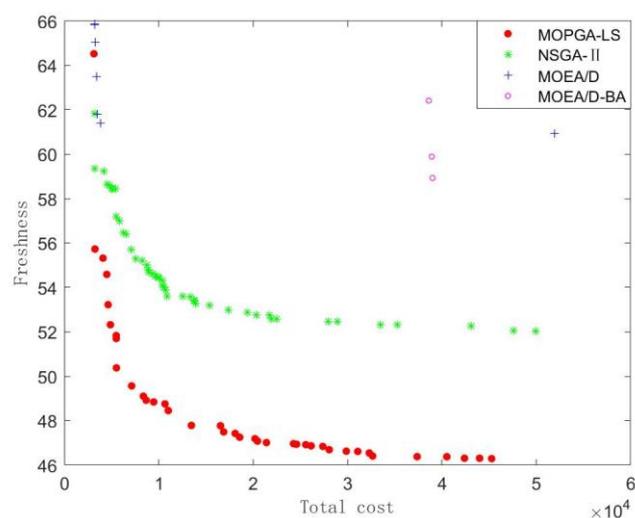


Figure 13. Non-dominated solution sets obtained by the four algorithms.

6. Conclusions

Referring to the research of Devapriya et al. [25], Belo-Filho et al. [26], Hou et al. [50], and so on, we can conclude that the integrated scheduling of the production and distribution is conducive to achieving global optimization. Meanwhile, evolutionary algorithms can effectively solve these NP-hard problems and provide multiple Pareto optimal solutions that provide diverse options for managers. However, there are few studies on the integrated scheduling of the picking and distribution of agricultural products, which is not conducive to the development of the agricultural product supply chain. Therefore, in view of the asymmetric information in the picking stage and the distribution stage, this work addresses an integrated scheduling problem of the picking and distribution of fresh agricultural products with the consideration of minimizing the picking and distribution costs as well as maximizing the freshness of orders. It considers in detail the heterogeneity of the picking groups, the product classification, and the exponential decay of the freshness of fresh agricultural products with time. A mathematical model is provided to formulate this problem, which is suitable for solving with the iterative algorithm. Then, a multi-objective multi-population genetic algorithm with local search is designed to handle it. The comparison experiment proves that the designed algorithm is an effective tool. The reported model and algorithm can provide a basis for making well-informed decisions for the managers and engineers managing farm operation.

In the future, we intend to design a scheduling method via assigning the products among picking groups dynamically. In addition, uncertainties in the integrated scheduling process of picking and distribution need to be considered.

Author Contributions: Conceptualization, X.P. and Y.X.; methodology, Y.F.; software, Y.X.; validation, X.P., Y.X. and Y.F.; formal analysis, X.P.; investigation, Y.X.; resources, X.P.; data curation, Y.X.; writing—original draft preparation, Y.X.; writing—review and editing, X.P.; visualization, X.P.; supervision, Y.F.; project administration, X.P.; funding acquisition, X.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China grant number 72271109.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: For helpful comments and discussions, we thank Shuling Xu.

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

References

1. Savarese, M.; Chamberlain, K.; Graffigna, G. Co-creating value in sustainable and alternative food networks: The case of community supported agriculture in New Zealand. *Sustainability* **2020**, *12*, 1252. [[CrossRef](#)]
2. Yu, Z.; Rehman Khan, S.A. Evolutionary game analysis of green agricultural product supply chain financing system: COVID-19 pandemic. *Int. J. Logist. Res. Appl.* **2021**, *25*, 1115–1135. [[CrossRef](#)]
3. Pulighe, G.; Lupia, F. Food First: COVID-19 Outbreak and Cities Lockdown a Booster for a Wider Vision on Urban Agriculture. *Sustainability* **2020**, *12*, 5012. [[CrossRef](#)]
4. Medici, M.; Canavari, M.; Castellini, A. Exploring the economic, social, and environmental dimensions of community-supported agriculture in Italy. *J. Clean. Prod.* **2021**, *316*, 128233. [[CrossRef](#)]
5. Zhu, A.; Bian, B.; Jiang, Y.; Hu, J. Integrated Tomato Picking and Distribution Scheduling Based on Maturity. *Sustainability* **2020**, *12*, 7934. [[CrossRef](#)]
6. Wang, Z.; Yao, D.-Q.; Yue, X. E-business system investment for fresh agricultural food industry in China. *Ann. Oper. Res.* **2017**, *257*, 379–394. [[CrossRef](#)]
7. Yan, B.; Chen, X.; Cai, C.; Guan, S. Supply chain coordination of fresh agricultural products based on consumer behavior. *Comput. Oper. Res.* **2020**, *123*, 105038. [[CrossRef](#)]
8. Liang, Y.; Liu, F.; Lim, A.; Zhang, D. An integrated route, temperature and humidity planning problem for the distribution of perishable products. *Comput. Ind. Eng.* **2020**, *147*, 106623. [[CrossRef](#)]
9. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [[CrossRef](#)]
10. Qingfu, Z.; Hui, L. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731. [[CrossRef](#)]
11. Gharaei, A.; Jolai, F. A multi-agent approach to the integrated production scheduling and distribution problem in multi-factory supply chain. *Appl. Soft Comput.* **2018**, *65*, 577–589. [[CrossRef](#)]
12. Ahumada, O.; Villalobos, J.R. Application of planning models in the agri-food supply chain: A review. *Eur. J. Oper. Res.* **2009**, *196*, 1–20. [[CrossRef](#)]
13. Ferrer, J.-C.; Mac Cawley, A.; Maturana, S.; Toloza, S.; Vera, J. An optimization approach for scheduling wine grape harvest operations. *Int. J. Product. Econ.* **2008**, *112*, 985–999. [[CrossRef](#)]
14. Bohle, C.; Maturana, S.; Vera, J. A robust optimization approach to wine grape harvesting scheduling. *Eur. J. Oper. Res.* **2010**, *200*, 245–252. [[CrossRef](#)]
15. González-Araya, M.C.; Soto-Silva, W.E.; Espejo, L.G.A. Harvest Planning in Apple Orchards Using an Optimization Model. In *Handbook of Operations Research in Agriculture and the Agri-Food Industry*; Plà-Aragónés, L.M., Ed.; Springer: New York, NY, USA, 2015; pp. 79–105. [[CrossRef](#)]
16. Kusumastuti, R.D.; Donk, D.P.v.; Teunter, R. Crop-related harvesting and processing planning: A review. *Int. J. Product. Econ.* **2016**, *174*, 76–92. [[CrossRef](#)]
17. Gómez-Lagos, J.E.; González-Araya, M.C.; Soto-Silva, W.E.; Rivera-Moraga, M.M. Optimizing tactical harvest planning for multiple fruit orchards using a metaheuristic modeling approach. *Eur. J. Oper. Res.* **2021**, *290*, 297–312. [[CrossRef](#)]
18. Augerat, P.; Belenguer, J.M.; Benavent, E.; Corberán, A.; Naddef, D. Separating capacity constraints in the CVRP using tabu search. *Eur. J. Oper. Res.* **1998**, *106*, 546–557. [[CrossRef](#)]
19. Osvald, A.; Stirn, L.Z. A vehicle routing algorithm for the distribution of fresh vegetables and similar perishable food. *J. Food Eng.* **2008**, *85*, 285–295. [[CrossRef](#)]
20. Chen, H.-K.; Hsueh, C.-F.; Chang, M.-S. Production scheduling and vehicle routing with time windows for perishable food products. *Comput. Oper. Res.* **2009**, *36*, 2311–2319. [[CrossRef](#)]

21. International Journal of Lower Extremity Wounds Song, B.D.; Ko, Y.D. A vehicle routing problem of both refrigerated- and general-type vehicles for perishable food products delivery. *J. Food Eng.* **2016**, *169*, 61–71. [[CrossRef](#)]
22. Wang, H.; Xu, Y.; Wang, Z.; Zhou, T.; Tian, D. Distribution Routing Optimization of Fresh Agricultural Products Based on Road Conditions. *J. Syst. Simul.* **2019**, *31*, 126. [[CrossRef](#)]
23. Amorim, P.; Almada-Lobo, B. The impact of food perishability issues in the vehicle routing problem. *Comput. Ind. Eng.* **2014**, *67*, 223–233. [[CrossRef](#)]
24. Geismar, H.N.; Laporte, G.; Lei, L.; Sriskandarajah, C. The integrated production and transportation scheduling problem for a product with a short lifespan. *INFORMS J. Comput.* **2008**, *20*, 21–33. [[CrossRef](#)]
25. Devapriya, P.; Ferrell, W.; Geismar, N. Integrated production and distribution scheduling with a perishable product. *Eur. J. Oper. Res.* **2017**, *259*, 906–916. [[CrossRef](#)]
26. Belo-Filho, M.A.F.; Amorim, P.; Almada-Lobo, B. An adaptive large neighbourhood search for the operational integrated production and distribution problem of perishable products. *Int. J. Prod. Res.* **2015**, *53*, 6040–6058. [[CrossRef](#)]
27. Kergosien, Y.; Gendreau, M.; Billaut, J.C. A Benders decomposition-based heuristic for a production and outbound distribution scheduling problem with strict delivery constraints. *Eur. J. Oper. Res.* **2017**, *262*, 287–298. [[CrossRef](#)]
28. Lacomme, P.; Moukrim, A.; Quilliot, A.; Vinot, M. Supply chain optimisation with both production and transportation integration: Multiple vehicles for a single perishable product. *Int. J. Prod. Res.* **2018**, *56*, 4313–4336. [[CrossRef](#)]
29. Neves-Moreira, F.; Almada-Lobo, B.; Cordeau, J.-F.; Guimarães, L.; Jans, R. Solving a large multi-product production-routing problem with delivery time windows. *Omega* **2019**, *86*, 154–172. [[CrossRef](#)]
30. Marandi, F.; Fatemi Ghomi, S.M.T. Integrated multi-factory production and distribution scheduling applying vehicle routing approach. *Int. J. Prod. Res.* **2018**, *57*, 722–748. [[CrossRef](#)]
31. Mohammadi, S.; Al-e-Hashem, S.M.J.M.; Rekik, Y. An integrated production scheduling and delivery route planning with multi-purpose machines: A case study from a furniture manufacturing company. *Int. J. Product. Econ.* **2020**, *219*, 347–359. [[CrossRef](#)]
32. Qin, C.; Gu, X. Improved PSO Algorithm Based on Exponential Center Symmetric Inertia Weight Function and Its Application in Infrared Image Enhancement. *Symmetry* **2020**, *12*, 248. [[CrossRef](#)]
33. Tefek, M.F. Artificial bee colony algorithm based on a new local search approach for parameter estimation of photovoltaic systems. *J. Comput. Electron.* **2021**, *20*, 2530–2562. [[CrossRef](#)]
34. Ul Hassan, N.; Bangyal, W.H.; Ali Khan, M.S.; Nisar, K.; Ag Ibrahim, A.A.; Rawat, D.B. Improved Opposition-Based Particle Swarm Optimization Algorithm for Global Optimization. *Symmetry* **2021**, *13*, 2280. [[CrossRef](#)]
35. Saad, A.; Khan, S.A.; Mahmood, A. A multi-objective evolutionary artificial bee colony algorithm for optimizing network topology design. *Swarm Evol. Comput.* **2018**, *38*, 187–201. [[CrossRef](#)]
36. Liu, L.; Liu, X.; Wang, N.; Zou, P. Modified cuckoo search algorithm with variational parameters and logistic map. *Algorithms* **2018**, *11*, 30. [[CrossRef](#)]
37. Fu, Y.; Tian, G.; Fathollahi-Fard, A.M.; Ahmadi, A.; Zhang, C. Stochastic multi-objective modelling and optimization of an energy-conscious distributed permutation flow shop scheduling problem with the total tardiness constraint. *J. Clean. Prod.* **2019**, *226*, 515–525. [[CrossRef](#)]
38. Fu, Y.; Zhou, M.; Guo, X.; Qi, L.; Sedraoui, K. Multiverse optimization algorithm for stochastic biobjective disassembly sequence planning subject to operation failures. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *52*, 1041–1051. [[CrossRef](#)]
39. Chan, F.T.S.; Wang, Z.X.; Goswami, A.; Singhanian, A.; Tiwari, M.K. Multi-objective particle swarm optimisation based integrated production inventory routing planning for efficient perishable food logistics operations. *Int. J. Prod. Res.* **2020**, *58*, 5155–5174. [[CrossRef](#)]
40. Fan, T.; Xu, C.; Tao, F. Dynamic pricing and replenishment policy for fresh produce. *Comput. Ind. Eng.* **2020**, *139*, 106127. [[CrossRef](#)]
41. Yang, Y.; Liu, J.; Tan, S.; Wang, H. A multi-objective differential evolutionary algorithm for constrained multi-objective optimization problems with low feasible ratio. *Appl. Soft Comput.* **2019**, *80*, 42–56. [[CrossRef](#)]
42. Ojsteršek, R.; Brezocnik, M.; Buchmeister, B. Multi-objective optimization of production scheduling with evolutionary computation: A review. *Int. J. Ind. Eng. Comput.* **2020**, *11*, 359–376. [[CrossRef](#)]
43. Xuping, W.; Jun, Z.; Caiyu, Y. An Integrated Production and Delivery Scheduling Model and Algorithm for Online Meal Ordering. *J. Syst. Manag.* **2020**, *29*, 158. [[CrossRef](#)]
44. Guo, F.; Huang, Z.; Huang, W. Integrated sustainable planning of fast-pick area network and vehicle routing with simultaneous delivery and pick-up. *Syst. Eng. Theory Pract.* **2021**, *41*, 962–978. [[CrossRef](#)]
45. Lin, S.-W.; Ying, K.-C. Minimizing makespan and total flowtime in permutation flowshops by a bi-objective multi-start simulated-annealing algorithm. *Comput. Oper. Res.* **2013**, *40*, 1625–1647. [[CrossRef](#)]
46. Fatemi Ghomi, S.; Karimi, B.; Behnamian, J.; Firoozbakht, J. A multi-objective particle swarm optimization based on pareto archive for integrated production and distribution planning in A Green supply chain. *Appl. Artif. Intell.* **2021**, *35*, 133–153. [[CrossRef](#)]
47. Jiang, E.-d.; Wang, L. An improved multi-objective evolutionary algorithm based on decomposition for energy-efficient permutation flow shop scheduling problem with sequence-dependent setup time. *Int. J. Prod. Res.* **2019**, *57*, 1756–1771. [[CrossRef](#)]

48. Riquelme, N.; Von Lücken, C.; Baran, B. Performance metrics in multi-objective optimization. In Proceedings of the 2015 Latin American Computing Conference (CLEI), Arequipa, Peru, 19–23 October 2015; pp. 1–11. [[CrossRef](#)]
49. Majumder, S.; Barma, P.S.; Biswas, A.; Banerjee, P.; Mandal, B.K.; Kar, S.; Ziemba, P. On multi-objective minimum spanning tree problem under uncertain paradigm. *Symmetry* **2022**, *14*, 106. [[CrossRef](#)]
50. Fu, Y.; Hou, Y.; Chen, Z.; Pu, X.; Gao, K.; Sadollah, A. Modelling and scheduling integration of distributed production and distribution problems via black widow optimization. *Swarm Evol. Comput.* **2022**, *68*, 101015. [[CrossRef](#)]