

Article

Mixed Model Assembly Line Scheduling Approach to Order Picking Problem in Online Supermarkets

Minfang Huang ^{1,2,*}, Qiong Guo ¹, Jing Liu ¹ and Xiaoxu Huang ¹

¹ School of Economics and Management, North China Electric Power University, Beijing 102206, China; Guoqiong@ncepu.edu.cn (Q.G.); liujing@ncepu.edu.cn (J.L.); xxhuangncepu@126.com (X.H.)

² Beijing Key Laboratory of New Energy and Low-Carbon Development, North China Electric Power University, Beijing 102206, China

* Correspondence: huangmf@ncepu.edu.cn; Tel.: +86-10-6177-3126

Received: 20 September 2018; Accepted: 26 October 2018; Published: 29 October 2018



Abstract: Online retail orders, especially online supermarket orders, have been highlighted to have several distinguished features from traditional online retailers. These include a huge amount of daily orders and orders containing multiple items. Tens of thousands of Stock Keeping Units (SKUs) sold by online retailers have to be stored at multiple storage zones due to the limit capacity of one area, and ordered items should to be picked with a parallel picking strategy. What is the most efficient and accurate method of picking, sorting and packaging the ordered items from SKUs for online orders? This paper focuses on scheduling the three processes of order picking problems in a warehouse for an online supermarket. Referring to the principle of the mixed-model assembly line, it presents a new optimization method of group order picking. With an objective of minimizing the picking and packaging time, this paper studies order batching and order sequencing. In order batching, considering the workload balance, it builds a mathematical optimization model and applies a bi-objective genetic algorithm to solve it. Then an order batching sequencing model is built, and a solving algorithm based on Pseudo-Boolean Optimization is developed. Case study and sensitivity analyses are conducted to verify the effectiveness of the method.

Keywords: online retailing; warehouse management; order picking; genetic algorithm

1. Introduction

The rapid growth of online retailing has intensified the complexity of the order fulfilment business. Currently, thousands of orders are picked and packaged in warehouses by human pickers each day. Online supermarket shopping has gradually become one of the most popular online retailing models. Because of its advantages on operating costs (3% to 5% less than traditional supermarkets [1]), large online supermarkets appear constantly. Several big online retail giants in China (e.g., jingdong.com, suning.com, chaoshi.tmall.com), have been setting up business branches of online supermarkets. For large online supermarkets, the efficiency and expense of order picking is a key part of the survival and development, in addition to the competition with general retailers (e.g., Walmart) and other online retailers in commodity categories, quality and price. It was surveyed that the cost of order picking accounts for more than 55% of the operational costs of the distribution center, and the picking time accounts for 30% to 40% of the total operation time consumed in the distribution center [2]. Therefore, the optimization of the order picking process plays an important role in optimizing the whole order fulfilment process. Online supermarkets mainly sell the daily fast consuming goods with the highest frequency. Online supermarket orders have several distinguished features that differ from general online retailers, e.g., a huge amount of daily orders, orders containing multiple items (8–10 times the average of general online retailing orders [3]), and up to an average of 7–8 Stock Keeping Units (SKUs)

and 16.7 items per order [4]. Customer receiving time windows are also strict and varied (e.g., must be delivered within 10 h, JIT delivery, delivering three times in one day). Online supermarkets sell a large number of goods, and the millions of SKUs have to be stored at multiple storage zones. Therefore, ordered items need to be picked under a parallel picking strategy in order to combine the items from different storage zones into one order parcel. Under a parallel picking strategy, the picking staff can be more familiar with the goods and paths in the zone he/she is responsible for and thus shorten the picking distance, reduce the congestion and improve the efficiency. According to the survey, a large online supermarket in China sells up to one million kinds of SKUs, and each warehouse deposits tens of thousands of SKUs. Its daily orders have reached up to 60,000 in several super-large cities. In Beijing, this online supermarket has about 20 distribution centers and more than 200 delivery stations. Even if the orders are delivered four times a day, the size of the order picking problem has been decreased for one decision. However in theory, there will still be hundreds of millions of possible sorting and distribution schemes. Therefore, how to efficiently and accurately pick the ordered items from tens of thousands of products, then package and deliver them is a focal and difficult point for many online retailers and academic researchers.

Almost each order in an online supermarket needs to flow along the processes of grouping, segmenting, picking, sorting (collecting & checking), and packaging. Thus, the organizing and scheduling of the whole line of order picking will be complicated due to tremendously customized orders. Currently, idle or busy times exist in different processes. In the real operation, sometimes the staff will slow down to wait for the next batch of orders, resulting in the low efficiency of the whole order picking and packing process. We noticed that there are similarities between the process of order picking problems and mixed-model assembly lines. In other word, a parcel of an order can be regarded as a product, and the order of operations (order picking, sorting, and packaging) correspond to different processes of the product assembly line. Based on the above idea, this paper establishes a group picking and packaging method for large scale orders under a parallel picking strategy with a multi-warehouse storage network. After completing order batching, coordination among the three processes (order picking, sorting, and packaging) will be reinforced to eliminate the wasted time caused by the imbalance of the three processes, or the delay caused by one process as much as possible. Furthermore, the bottleneck can be identified, and finally the order assembly line balancing can be reached.

The mixed-model assembly line refers to the production of different types of products on the same production line for a period of time. It has been widely used in assembly production enterprises. This mode of production line can produce both standardized products and small batches of customized products. The latter can meet various customer needs, as well as some other requirements from the market. It is a flexible and diversified production and assembly method. Since the working time of each product on each machine of the assembly line might be different, this leads to frequent workload imbalances between machines. According to U.S. statistics, even in industrialized countries (e.g., the United States), an average of 5% to 10% of assembly time in industrial assembly production is wasted due to the delay caused by the imbalance. Huang [2] put forward that eliminating unnecessary movement could reduce stalling time by about 50%, and that improved operations could increase the average utilization of assembly lines from 73% to 90%.

One related literature is about the classical order picking problem (only including the process of item picking). The most commonly used optimization method is order batching methods. It's optimization strategies in the existing research are mainly focused on the time window of orders arriving [5], the proximity of storage zones or locations on shelves of the ordered commodities [6] and the minimization of the total completion time and order delay time [7,8]. The scholars proposed the exact solution methods [9] and heuristic algorithms to solve different order batching models, e.g., the classification method of a self-organizing neural network [10]. Among them, the heuristic algorithms proposed by Gong and De Koster [11] and Hong et al. [12] can solve the large-scale order batching problems, whose sizes have up to 1000 orders. However, most cases only contain one

type of commodity. The existing order batching strategies mainly contribute to minimizing item picking travel distance, seldom considering the workload balance among multiple storage zones under parallel picking strategy. Recently, several scholars have put forward several integrated optimizations for order picking, e.g., integration of order batching and sequencing [13,14], integration of order batching and picking routing [15], and integration of order batching, order batch assignment and picking routing [16]. Facing the problem of order splitting, Zhang et al. [17] present a powerful order fulfillment method—package consolidation. Increasing the number of transshipments, the model and algorithm are used to verify its superiority. Li et al. [18] are considering item correlation to classify items, that increases the parallel amount of goods in each picking zone and improves the picking efficiency. These related results only optimize the order picking operation in a single warehouse. More recently, several pieces of literature have been found for order batching problems using a parallel picking strategy [19]. From the perspective of the streamline process, the above results are only the optimization of the single picking activity, and the results cannot guarantee the efficiency of the subsequent order consolidation and packaging work. It still lacks the research results on the improvement of the whole process of order picking, sorting, and packaging.

Another related literature is the JIT production system. It is the focus of the research on the sequencing problems of assembly lines. It mainly optimizes two objectives: the workload balance of each machine in the assembly line and the main components' consumption rate. Recently, the assembly line sequencing problems are generally studied as multi-objective problems. Lin and Chu [20] constructed an integer programming model for a mixed-model assembly line sequencing problem in a door lock production company to minimize the production costs and used the Branch & Bound algorithm to solve it. Lu et al. [21] established a multi-objective model of mixed-model assembly line sequencing to minimize the total adjustment time, overload and free time, and used hybrid artificial bee colony algorithm to solve the model. There are several results which have studied joint optimization of the balancing and sequencing problems. Kucukkoc et al. [22], aimed at the two-sided assembly line problem of mixed-model assembly lines, put forward an optimal solution for the sequencing and balancing problem, and used ant colony-based optimization methods to minimize the length of the assembly line and the number of machines. Manavizadeh et al. [23] considered the task balance and model sequencing problem, put forward a new heuristic algorithm to minimize the cycle time and the waste of machines, and reduce the workload of machines. Swarm-based algorithms are popularly used algorithms for solving scheduling problems. Bottani et al. [24] developed an efficient heuristic based on firefly algorithm, defined two vectors for each firefly to represent a candidate solution and optimized by the objective function, the aim to formulate an objective function which takes into account both the throughput and the system unbalance, and the results show that the algorithm is feasible. Zhou et al. [25] presented a multi-population discrete firefly algorithm and combined with the k-opt algorithm to solve the traveling salesman problem. Wahab et al. [26] summarized the swarm-based algorithms including Genetic Algorithms, Ant Colony Optimization, Glowworm Swarm Optimization, etc. and they provided an in-depth survey of well-known optimization algorithms. Iwasaki et al. [27] initially presented the application of the idea of the assembly line to solve the order picking routing in the warehouse, in which the order groups to be picked were considered as jobs, storage positions (shelves) of goods as machines, goods picking orders as processing sequences, goods picking time as processing time. The above research results provide reference for this study to some extent. However, they rarely study multiple processes composed of the order picking and packaging from the perspective of JIT assembling.

In summary, the new picking problem (three processes included, item picking, sorting and packaging) studied in this paper is totally different from the classical one (only includes one process, item picking). Both the definition of parameters and the formulation of our problems are different. The whole process of order picking, sorting, and packaging could not be simplified to a mixed model assembly line. Even in a mixed model assembly line, it is still in a dilemma that multiple machines (corresponding to multiple picking zones in our problem) in one process operate in parallel, deliver

diversified parts (different commodities) to the next process at right time and assemble them into different products (different packages). Therefore, referring to the idea of mixed model assembly line scheduling approach, this paper aims to innovatively build a scientific solution procedure to organize and schedule the whole line of order picking in online supermarket. The following two key issues will be addressed: ensuring the balances of the workload of each picking zone and the total number of items in each batch while dealing with huge amount of daily orders within tight time windows; ensuring that all the picking zones only deliver the right goods to the packaging station at the right time.

In the next section we analyze the order picking process and present a description of the problem. Sections 2 and 3 present the order batching model considering the balance of the workload and order batch sequencing. In Section 4, numerical analyses will be conducted to demonstrate the performance of the proposed model and algorithm. The summary results and possible future research directions are presented in Section 5.

2. Order Picking Process Analysis and Problem Description

2.1. A General Order Picking Process of Online Supermarket

According to the investigation on the practical operations of order picking in the warehouse of an online supermarket, we summarize the current online supermarket order picking and logistics distribution process. The specific steps are as follows: (1) All orders placed online are put into an order pool and are classified by their required delivery time; (2) For every specified time period (usually 5–10 min), orders with similar attributes in the order pool are divided into several waves (each wave has about 12–15 orders) and picking lists corresponding to different storage zones are formed; (3) The picking staff will pick items according to the lists responsible for different storage zones from the whole warehouse, and then deliver the picked items to the packaging zone; (4) The checking staff are responsible for checking the packaged items; (5) The packing workers will choose the appropriate size of box and pack the reviewed items; (6) The packages will be placed at the loading locations corresponding to different distribution stations manually or by the automatic sorting machines.

Currently large online supermarkets apply wave picking in the whole warehouse to improve the order picking efficiency. However, with the increasing order scale, errors constantly occur in this multi-link process. This paper presents a group picking and packaging method for large scale orders under a parallel picking strategy with a multi-warehouse storage network. Considering the limited storage space in the packaging zone, it ensures the right items from picking zones would be delivered to the packaging zone at the right time. By applying the idea from JIT assembly, it attempts to totally utilize the capacity and ability of the packaging zone, increase the efficiency of the existing multi-link operational process, realize orderly production and improve the daily order handling ability.

2.2. Order Picking Process with a Parallel Picking Strategy and Problem Description

Due to the large number of commodities, online supermarkets have to store them in multiple categorized storage zones according to the correlations among them. The workers will be responsible for picking items in fixed zones separately. This strategy can reduce the picking area each worker is responsible for, and therefore reduce the congestion in the aisles and increase the worker's proficiency and efficiency. The items contained in a single order in an online supermarket are usually located in multiple storage zones. Therefore, it is necessary to pick and deliver all items involved in one order from different storage zones to the packaging facility. Then the package for one order could be packaged and finally delivered to the customer at one time. The flow chart of group order picking process in a warehouse of an online supermarket is shown in Figure 1.

For this large-scale and complex order process, we attempt to refer to the principle of the mixed-model assembly line. We first divide the orders arriving within a certain period of time into batches. Then we treat three processes of order picking, sorting, and packaging as a sequence of three independent steps to optimize the operation process, plan the operation time of the three steps, and finally greatly improve the utilization rate of personnel and facilities. Online supermarket order group picking problem with JIT mode studied in the paper contains two optimization problems, the order batch problem with a parallel picking strategy and order batch scheduling problem.

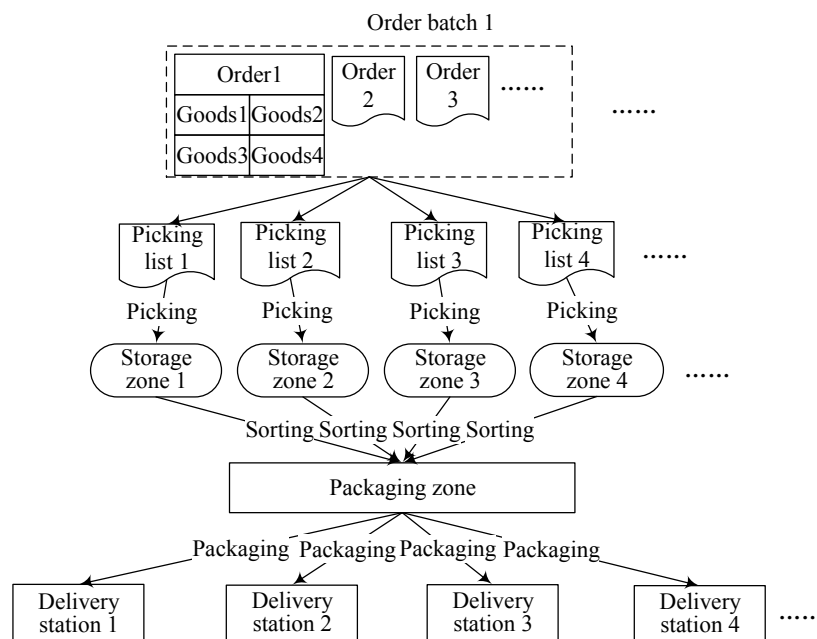


Figure 1. The flow chart of group order picking.

3. The Order Batching

3.1. Model Assumptions and Parameters

The assumptions of the model are as follows:

- (1) An order is not allowed to be split into different batches;
- (2) We assume one picker is only responsible for one picking zone. Considering the picking efficiency of the warehouse is proportional to the number of pickers, here we assume there is only one picker in one picking zone;
- (3) Order batches will be divided into multiple picking zones and they should be picked batch by batch, that is, one order batch can start to be picked only after the previous order batch has been completed picking in all storage zones;
- (4) Each kind of item can only be stored in one location on a shelf;
- (5) The orders are known in advance. After orders are batched, no new orders can be added.

The parameters and decision variables used in the model are presented in Table 1.

Table 1. Notations for parameters and decision variables.

Parameters for Picking Zones	Description
m	Storage zone $m, m \in M$
N_s	The total number of picking zones
T_m	Total working time of zone m
\bar{T}_m	Average working time in each picking zone
V	The capacity of the picking device
i	Item i
s_{mn}	The total number of SKUs in zone n of batch m
Parameters for Batching	Description
n	Batch $n, n \in N$
o	Order $o, o \in O$
I	The total number of orders to be picked
I_n	Total order quantity of batch n
I_{mn}	The number of items to be picked in the picking zone m of batch n
NP_n	The total number of picking zones for batch n
N_b	Batches of all orders
T_n	The total working time of batch n
C_n	The number of items in the batch n
N_c	Total number of items
\bar{N}_c	Average item number of each batch
S_o	Number of items in order o
Parameters for Time	Description
t_{pic}	The average picking time of each item
t_{prepic}	The average preparation time for a batch of orders in each zone
t_{mn}	The working time of the batch n in the picking zone m
t_{check}	Collecting and checking time of one item
α	The time factor of items collecting and checking, $\alpha \in (0, 1)$
t_{pack}	Fixed packaging time of one item
β	The time factor of packaging, $\beta \in (0, 1)$
t_n^{pick}	Picking time of batch n
t_n^{check}	Collecting and checking time of batch n
t_n^{pack}	Packaging time of batch n
Decision Variables	Description
x_{io}	0–1 variable, 1 if item i belongs to order o , 0 otherwise
y_{on}	0–1 variable, 1 if order o belongs to batch n , 0 otherwise

3.2. The Balance of the Workload

The idle waiting time on the order processing line is caused by the difference in working time of each batch of orders in the process and the difference of working time of a single batch order in different picking zones at the same process. For example, in the process of item picking, the items in one order batch assigned into different picking zones are different. Thus, for one batch of order, the workloads of item picking in all storage zones are unbalanced. On the other hand, imbalanced workload may also occurs due to the time difference spent on the whole process or on one step between two order batches. The possible solution is to reduce the idle waiting time by assimilating the scales of order batches and the workloads of a single batch order assigned in each picking zone.

As shown in Figure 2, there are N_b picking batches and N_s parallel picking zones. t_{mn} represents the picking time for completing picking the items of batch n assigned to the picking zone m . Since the items of each batch in different picking zones are picked simultaneously, picking time of n th batch t_n^{pick} can be defined as the longest picking time of the storage zone, that is, $t_n^{pick} = \max\{t_{1n}, t_{2n}, \dots, t_{mn}, t_{N_s n}\}$. While constructing an order batching optimization model under the parallel zone picking strategy, both the running efficiency of the whole assembly line and the balance of items among order batches and the workloads among picking zones should be considered.

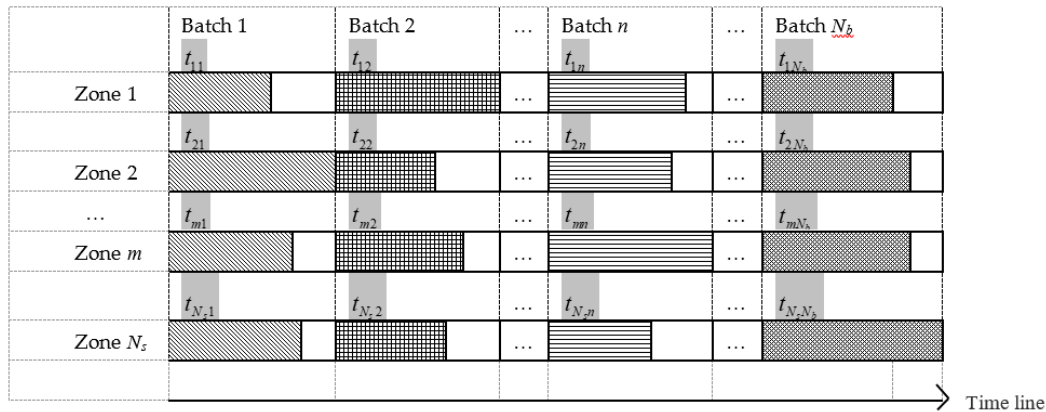


Figure 2. Picking time in different picking zones and for different order batches.

3.3. The Order Batching Model Considering the Balance of the Workload

The mathematical model for the order batching problem considering the balances of the workload of each picking zone and the total number of items in each batch can be formulated as follows:

$$Min \sqrt{\sum_{m=1}^{N_b} (C_n - \bar{N}_c)^2 / N_c} \tag{1}$$

$$Min \sqrt{\sum_{m=1}^{N_s} (T_m - \bar{T}_m)^2 / N_s} \tag{2}$$

Subject to:

$$S_o = \sum_{i=1}^{N_c} x_{io}, o \in O \tag{3}$$

$$C_n = \sum_{o=1}^I S_o * y_{on} \tag{4}$$

$$\bar{N}_c = N_c / N_b \tag{5}$$

$$m = f(i), m = 1, \dots, N_s; i = 1, \dots, N_c \tag{6}$$

$$T = \sum_{n=1}^{N_b} T_n \tag{7}$$

$$\bar{T}_m = \sum_m^{N_s} T_m / N_s \tag{8}$$

$$T_n = \max\{t_{mn}\}, n = 1, 2, \dots, N_b; m = 1, 2, \dots, N_s \tag{9}$$

$$T_m = \sum_{n=1}^{N_b} t_{mn}, m = 1, 2, \dots, N_s \tag{10}$$

$$\sum_{n=1}^{N_b} y_{on} = 1 \tag{11}$$

$$\sum_{k=1}^I y_{ok} * I_{mn} \leq V \tag{12}$$

$$t_{mn} = t_{prepic} + t_{pic} * S_{mn} \tag{13}$$

$$t_n^{check} = t_{check} \cdot (1 + \alpha \cdot NP_n) \cdot s_{mn} \quad (14)$$

$$t_n^{pack} = t_{pack} \cdot (s_{mn} + \beta \cdot NP_n) \quad (15)$$

There are two objective functions. Function (1) represents the minimization of the standard deviation of the total number of items in each batch, namely, to balance the total number of items in each batch. Function (2) represents the minimization of the standard deviation of the workload of each picking zone, namely, to balance the picking time of each zone. Constraint (3) represents the total number of items in order o . Constraint (4) represents the number of items of batch n . Constraint (5) represents the average item number of each batch. Constraint (6) indicates that a commodity i belongs to the picking zone m . Constraint (7) represents the total completion time. Constraint (8) represents the average working time of each partition. Constraints (9) and (10) represent the completion time of each batch and the working time of each partition separately. Constraints (11) indicate that each order can only be assigned to one batch. Constraint (12) represents the order batch capacity constraints. They ensure the capacity of all items in one batch assigned into each picking zone shall not exceed the maximum capacity of the picking equipment. Constraint (13) represents the working time of the batch n in picking zone m . Constraint (14) represents the total items collecting and checking time of the batch n . Constraint (15) represents the packaging time of the batch n .

3.4. A Bi-Objective Genetic Algorithm for Order Batching Model

The genetic algorithm is appropriate to solve the order batching problem. It can obtain the optimal solution or the set of suboptimal solutions. Since the order batching model in the paper has two objectives, that is, balancing the total numbers of items and the workloads of picking zones, we adopt bi-objective genetic algorithm to solve it. In general, the multi-objective function can be transformed into a single objective function by assigning different weights to different objectives. According to the multi-objective solving principle of genetic algorithm, using the ranking-based representation matrix for finding different individual fitness, it attempts to determine the probability of its crossover mutation by comparing the solution of the individual fitness size and the overall performance of the population. Furthermore, it can control the population of individuals to parallelly approach to the front edge of the non-inferior solution set.

While using bi-objective genetic algorithm to solve the multi-objective decision-making model, it will randomly generate an initial population, which is a set of feasible solutions. After a series of genetic operations to optimize the initial population, the set of suboptimal solutions will be achieved in a relatively short period of time. The principle of this algorithm provides a favorable reference for decision-makers to determine production plans and product processing sequences. Due the order batching results are predictable and controllable, the solution obtained through genetic algorithm can not only optimizes the target function, but also satisfies the need of the sequential order. The specific process of bi-objective genetic algorithm is shown in Figure 3.

The important processes of bi-objective genetic algorithm in Figure 3 are explained as follows.

(1) Individual fitness

Let the value of objective Function (1) be $Obj1$, and the Function (2) $Obj2$. The values of $Obj1$ and $Obj2$ of chromosome i are $Obj1(i)$ and $Obj2(i)$, separately. Calculate $Obj1$ and $Obj2$ of all the chromosomes in the population and let the maximal $Obj1(i) + Obj2(i)$ be $MAXobj$. The fitness of chromosome i is denoted as $f(i)$, and $f(i) = k + MAXobj - (Obj1(i) + Obj2(i))$, $k \in (0, 1)$. The goal is to make the fitness not equal to 0.

(2) Crossover and variation

In this paper, two operations, single point crossover with a fixed probability and the variation with a fixed probability are applied. The single point crossover with a fixed probability is to select the individuals in the population with a fixed probability, and to replace or recombine several genes in the

chromosome of two parents, thus forming new individuals. The operation of the crossover is able to obtain a better new individual, which might be the optimal solution to the problem. The variation with a fixed probability is to select an individual in a population with a fixed probability and then change the value of several genes in the chromosome, or utilize some way of recombining the chromosome, for example, changing the sequence of genes. The specific algorithm for realizing the operations of crossover and variation is closely related to the coding scheme of chromosomes. For one same optimization problem, if the coding schemes of chromosomes are different, the specific algorithm of crossover and mutation can greatly be changed.

(3) Selecting the next generation of individuals

In this paper, we choose N individuals randomly selected by a roulette way to form the next generation. The greater fitness the individual has, the greater probability it can enter the next generation is with.

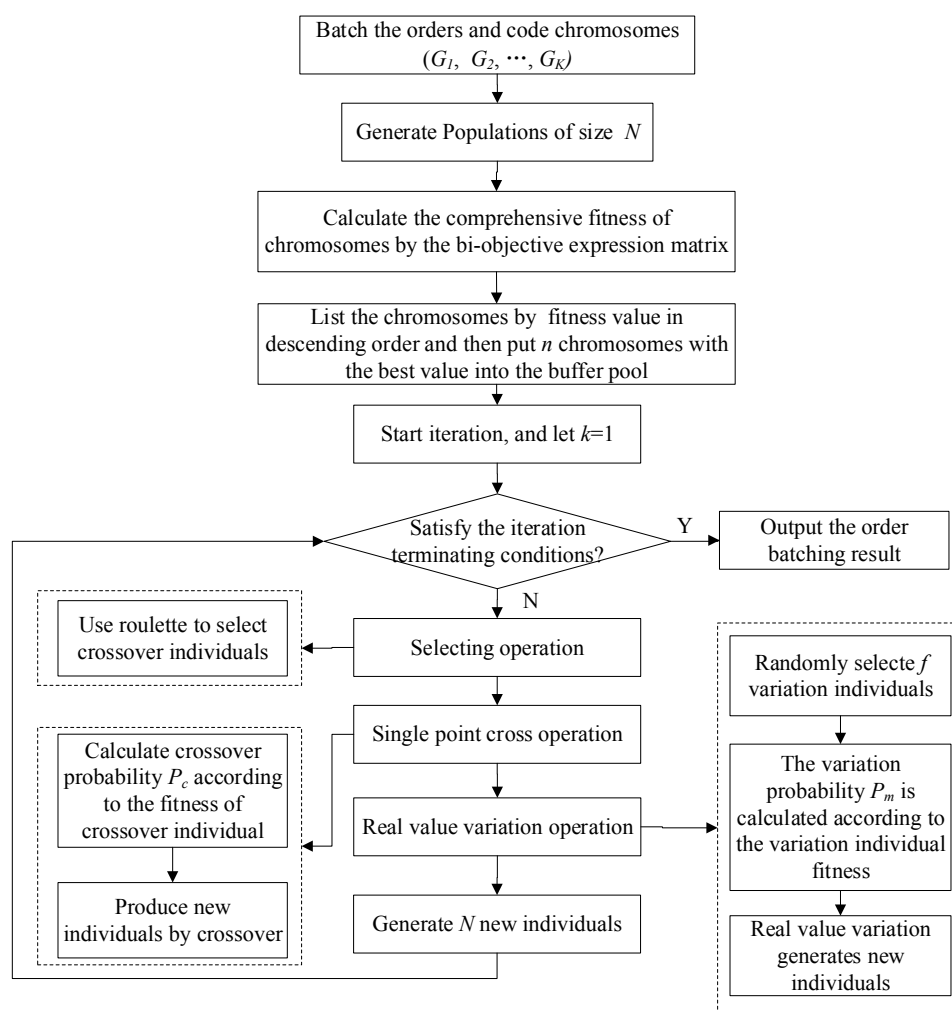


Figure 3. The bi-objective genetic algorithm for solving the order batching model.

If the descendant generation is superior to its parent generation, then it can replace the parent generation in the next generation. Otherwise, the descendant generation will be eliminated, and a new generation will be produced. If it cannot determine the parent generation or its descendant generation is better, then the diversity of the generation will be the second evaluation criterion.

In order to maintain the diversity of the solution set, a library for storing non-dominated solutions is also established in the algorithm, which is called the dominating library. If the descendant generation

is superior to any solution in the dominating library, the descendant generation will be the new parent generation and it will be removed from the library at the same time. If the descendant generation is not superior to the one in the dominating library, then the sparseness of the solutions both in the library and in the parent generation should be checked. If the descendant generation is in the least dense area of the library, the descendant will become the parent generation and is added to the library.

4. Order Batch Sequencing

4.1. Order Batch Sequencing Optimization Model

For the mixed-model assembly line, the production sequence of multiple products is an important issue to solve. An optimal sequence is beneficial to balance the workload of each process of the assembly line well and helps to minimize the waiting time needed for one order batch to enter each process. Thus, it guarantees no order batch will take a long time in one certain process which would lead to some idle time in any other processes.

In this paper, thousands of order parcels flowing along the picking process in the warehouse are regarded as different items produced in a mixed-model assembly line at a certain speed. After order batching, the entire assembly line has many batches in the same sequence through several processes, at each of which there is at least one worker. When the whole assembly line runs at a balanced state, the production efficiency will reach the best level. Therefore, besides the order batching, a better order batch sequence is another way to further improve the efficiency of the whole order picking process.

If there are N_b batches and N_p processes, then in a whole order process, there are $N_b + N_p - 1$ stages. As shown in Figure 4, there are 3 batches, A, B and C, and 3 processes, picking, sorting (items collecting and checking) and packaging. Assuming that the working time of n th order batch on k th processes is represented as following matrix S_{nk} :

$$s_{nk} = \begin{bmatrix} 4 & 3 & 2 \\ 3 & 2 & 1 \\ 4 & 3 & 2 \end{bmatrix} \tag{16}$$

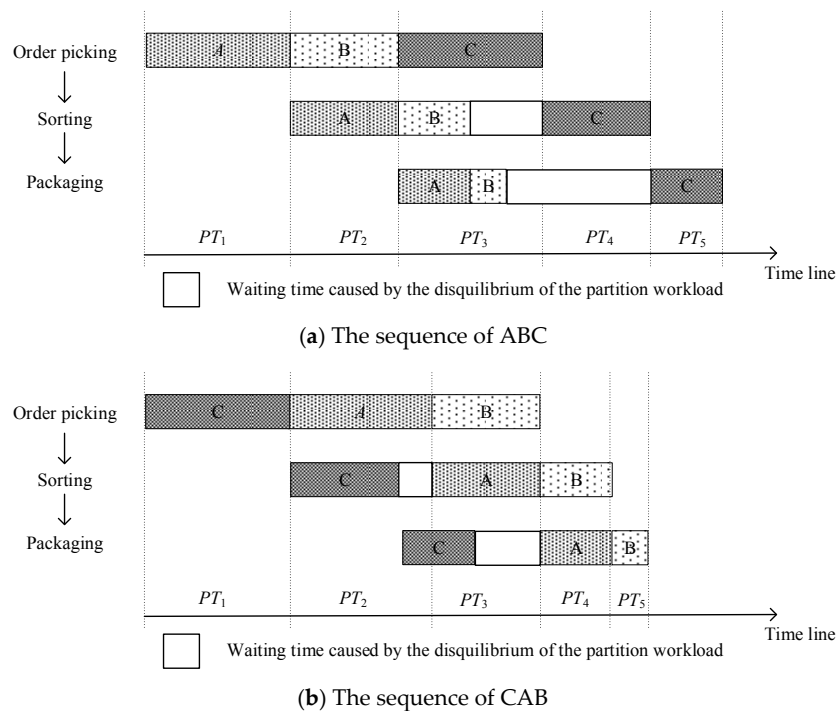


Figure 4. Unbalanced order batch results.

If the sequence of order batches is arranged as ABC, shown in Figure 4a, the total completion time is 16. If the sequence of order batches is arranged as CAB, shown in Figure 4b, the total completion time is 14. It can be seen that an appropriate sequence could shorten the total processing time.

We also define the stage time as the longest completion time of all the parallel working processes. As shown in Figure 4a, $PT_1 = S_{11}$, $PT_2 = \text{Max}\{S_{21}, S_{12}\}$, $PT_3 = \text{Max}\{S_{31}, S_{22}, S_{13}\}$, $PT_4 = \text{Max}\{S_{32}, S_{23}\}$, $PT_5 = S_{33}$. So the total completion time is $\sum_{j=1}^{N_b+N_p-1} PT_j = PT_1 + PT_2 + PT_3 + PT_4 + PT_5$.

The parameters in the model in this chapter are as follows:

n : Order batch n

N_b : The total number of batches

k : Process k

N_p : The total number of processes

r : Sequence r

X_{nr} : The sequence of batch n is r , $r = 1, 2, \dots, N_b$; $k = 1, 2, \dots, N_b$

S_{nk} : The working time of the batch n in the process k , $n = 1, 2, \dots, N_b$; $k = 1, 2, \dots, N_p$

j : Work stage j

PT_j : Time spent on work stage j

The mathematical model of this problem can be formulated as:

$$f = \min \sum_{j=1}^{N_b+N_p-1} PT_j \quad (17)$$

$$\sum_{n=1}^{N_b} X_{nr} = 1, r = 1, 2, \dots, N_b \quad (18)$$

$$PT_j \geq \sum_{n=1}^{N_b} S_{nk} X_{n(j-k+1)} = 1, j = 1, 2, 3, 4; k = 1, 2, \dots, N_p \quad (19)$$

$$j, k \in JK \begin{cases} j = 1, \dots, N_p - 1, k = 1, \dots, j \\ j = N_p, \dots, N_b, k = 1, \dots, N_p \\ j = N_p + 1, \dots, N_p + N_b + 1, k = j - N_b + 1, \dots, N_p \end{cases} \quad (20)$$

Function (17) is the objective function with a minimization of the completion time of all batches. Formula (18) represents only one sequence r . Formula (19) represents the working time of the stage and should be the longest completion time of all the parallel working processes. Formula (20) represents the relationship between process j and process k . There are three batches and three processes, $N_b = 3$, $N_p = 3$.

4.2. Pseudo-Boolean Optimization (PBO) for Order Batch Sequencing

The order batch sequencing problem can be transformed into Pseudo-Boolean Optimization (PBO) problem [28]. PBO is a special case of integer programming problems. The problem consists of objective function, constrained variables and constraints, where the constrained variables are limited to Boolean variables (0 or 1). The solution of a PBO problem must satisfy all the constraints and minimize the objective function. Currently, PBO problem is widely used to solve the problems in the areas of engineering design and automation. We will apply one open source solution software, SCIP, to solve the instances.

The optimization problem that can be solved by traditional algorithms (e.g., genetic algorithm) are corresponding to the optimal solution of the Pseudo-Boolean function. Thus, the optimization problem can be solved by finding the optimal solution of the Pseudo-Boolean function. The solution strategy is first to build an objective function with Boolean variables and a set of constrained inequalities, and then to find the solution that satisfies the constrained inequalities and minimizes the value of the objective

function. By using polynomial Pseudo-Boolean function, the structural features of Pseudo-Boolean function can be described. PBO is a powerful problem solution tool which can represent complex constraints with simple expressions.

While solving small size problems, SCIP software can quickly achieve the accurate solutions for the problems. However, it usually consumes more time to solve the larger scale problems and can sometimes only achieve satisfied solutions. A new solution method is designed to improve the solution efficiency. That is, it attempts to compromise the computing time and the quality of the solution by setting a fixed maximum computing time. The optimal solution or a satisfied solution could be achieved within the maximum solution time.

5. Numerical Study

5.1. A Numerical Example

A numerical example is constructed to verify the model and solution algorithm proposed in this paper. As shown in Table 1, there are five picking zones to store different categories of SKUs in a warehouse, each of which has 2000 kinds of SKUs. There are 10 picking staff in each picking zone. The maximum capacity of a picking container is 10. It randomly generates 1000 orders, and the number of items in one order obeys the random distribution [1,15]. The average picking time for each item is 8 s, the average setup time for picking a batch of orders in each zone is 50 s, item sorting time t_{check} is 0.5, item sorting time factor α is 0.5, packaging time constant t_{pack} is 1.6, and packaging time factor β is 1. The parameters in bi-objective genetic algorithm are: the population size is 100, iterative times is 1000, crossover ratio is 0.6, and mutation ratio is 0.001.

In the batching, as the number of iterations of genetic algorithm increases, the value of the objective function of the optimal individual in each generation decreases gradually. The changing trend of the value of the objective function is shown in Figure 5.

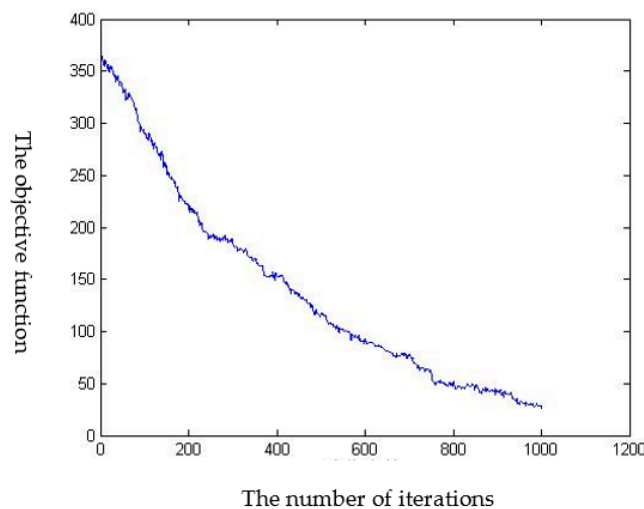


Figure 5. The objective function values changing with the number of iterations of GA.

The batching results of individuals with an optimal objective function are shown in Table 2. 1000 orders were divided into 16 batches. As the number of iterations increase to 1000, the value of the objective function decreases gradually. The standard deviation of the working time of each picking zone of the optimal solution was reduced to 8.76, and the standard deviation of the number of items in each batch was near 17.96, while the two standard deviations of the initial population were 58.98 and 305.76, respectively. It can be found that both standard deviations decrease significantly.

Table 2. The results of order batching.

Batch/Data	Working Hours of Each Picking Zone(s)					Item Number	Number of Goods
	1	2	3	4	5		
1	794	834	778	722	770	456	473
2	850	914	674	754	818	470	489
3	722	810	882	706	906	472	489
4	1034	826	874	834	802	515	536
5	762	706	762	946	682	451	460
6	890	786	746	738	834	468	484
7	866	770	858	890	842	497	513
8	682	842	778	778	1010	480	500
9	794	826	842	818	818	481	497
10	778	850	866	786	722	469	482
11	770	826	874	738	802	470	488
12	818	890	922	882	794	507	530
13	810	738	754	858	914	478	498
14	746	794	786	842	722	455	475
15	810	834	850	866	770	485	505
16	866	754	746	834	770	465	481
Total	12,292	13,000	12,292	12,292	12,976		
Deviation	8.76	17.96	-	722	770	456	473

:- no optimal solution was found within 1 CPU hour.

The algorithm also outputs the picking time, sorting time and packaging time of each batch of orders corresponding to the optimal order batch solution, as shown in Table 3.

Table 3. The detailed results of order batching.

Batch	Picking Time	Sorting Time	Packaging Time
1	834	798	830
2	914	823	861
3	906	826	858
4	1034	901	926
5	946	789	811
6	890	819	854
7	890	870	901
8	1010	840	869
9	842	842	880
10	866	821	850
11	974	823	850
12	922	887	907
13	914	837	861
14	842	796	822
15	866	849	872
16	866	814	838

According to the results in Table 2, the Pseudo-Boolean Optimization problem is constructed and solved by SCIP software. The optimal solution is calculated by about 14 s. And the optimal batch sequence is 1, 9, 4, 2, 8, 7, 11, 3, 15, 12, 6, 13, 16, 10, 14, 5. The total time to complete the whole order picking process is 16,049 s.

5.2. Sensitivity Analysis

We further execute several sensitivity analyses to test the solution efficiency of the solution algorithm. We increase the number of orders from 500 to 2000 to study the impact of the problem size on computing time. The results are shown in Table 4. It can be seen that with the order size increases, the computing time increases rapidly. This is because the problem studied in this paper is NP-hard, and the expansion of solution space inevitably leads to the increase of computation time.

Table 4. Total working time and computation time varying with the problem size.

Total Orders	Total Working Time (s)	Computing Time (s)
500	9332	2.013
1000	16,049	154.831
2000	30,951	732.285

We set the number of items per order changing in 10, 15, and 30. Table 5 indicates that with the number of items per order increases, the total working time increases due to the increasing picking time. But it doesn't have as greater impact on two-time indicators as the problem size.

Table 5. Total working time and computation time varying with the problem size.

Number of Items Per Order	Total Working Time (s)	Computing Time (s)
10	12,545	4.121
15	16,049	14.592
30	38,186	90.628

We set the number of pickers changing in 5, 10, 15 to depict how the total working time and the computing time change. The results are shown in Table 6. It can be seen that the increase in the number of pickers can effectively reduce the time cost. This is because the increasing number of pickers will lead to decrease the average order picking time, thus reducing the total working time.

Table 6. Total working time and computation time varying with the number of pickers.

The Number of Pickers	Total Working Time (s)	Computing Time (s)
5	25,743	54.312
10	16,049	40.009
15	10,128	42.732

6. Conclusions

At present, large online supermarkets are faced with huge amounts of customized orders within tight time windows every day. The general order fulfilment measurements for online retailers, picking by order or picking by batch in the whole warehouse, have been inadequate for dealing with such situations. It will be effective to utilize the principle of assembly line to operate the whole process of order picking. In this paper, the JIT assembly theory is innovatively introduced into the area of order picking problems, which is helpful to maximize the processing abilities of item picking, sorting and packaging. And it is also beneficial for online retailers to make decisions on the design and the control of order picking systems at the tactical or operational level.

This paper aims to establish a group picking and packaging method under a parallel picking strategy with a multi-warehouse storage network. Utilizing the idea of mixed-model assembly line, with an objective of minimizing the total order processing time, this paper presents a two-stage sequential optimization method for online supermarket order picking problems, that is, batching first, sequencing second. It is beneficial to realize efficient and balanced operations of each process in the whole order picking process, which includes three steps, order picking, sorting and packaging. Considering workload balance, the proposed order batching model has two objective functions. One is to balance the picking time of each batch in each picking area, and the other is to balance the total number of items between order batches. The first objective can help to achieve similar picking times in different storage zones for one batch whose items are assigned into these zones, thus to ensure that all the picking zones only deliver the right goods to the packaging station at the right time. It improves the efficiency of the packaging station. The second objective is to balance the total number of items between order batches so that the times required for batches on picking, sorting, and packaging could

be similar. In determining the order batches production sequence, in which the orders of different batches that enter into the assembly line, the optimal combination of order batches is optimized to further minimize the total completion time of the whole process of order picking. Meanwhile, production bottleneck could be determined and finally be eliminated or alleviated.

The limitation of this study is related with the assumptions for ideal conditions under which the new order picking optimization method is carried out. The efficiency of a JIT production system depends on the automation level of the whole process and the punctuality of each process. In reality, due to the varieties of volumes and weights of commodities sold by online supermarkets, it is not appropriate to fully install automatic production lines in the warehouse. Order picking systems still involve human operators on a large scale. Sometimes manual operations fail to guarantee the regular timetable precision, which may result in disruptions of any process, and finally affect the seamless connection of the three processes.

There are still several further studies for this research: (1) If adding one more facility or working staff in the process of item sorting, or adding more working staff in the processes of packaging, order batching and sequencing problems are a bigger issue for assembly lines with multiple machines (2). While facing a super large size of orders, the problem-solving complexity will be greatly increased, and the efficiency of the solution method and the generality of the solution method will need to be improved. (3) A more applicable solution-based software should be designed for real-world operations, to realize scheme-generating quickly and support quick decision-making.

Author Contributions: M.H. and X.H. contributed the central idea and wrote the initial draft of the paper. M.H. also has taken charge of writing-review and editing and supervision. Q.G. and J.L. did the analyses and contributed to refining the ideas, carrying out additional analyses and finalizing this paper.

Funding: This research was supported by program of the National Natural Science Foundation of China (71571067) and the innovative research group program of the National Natural Science Foundation of China (71421001), and China Scholarship Council (201706735001).

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

References

1. Yousept, I.; Li, F. Online Supermarkets: Emerging strategies and business models in the UK. In Proceedings of the 17th Bled eCommerce Conference, Bled, Slovenia, 21–23 June 2004; pp. 1–22.
2. Huang, D. *Application of Lean Production in Escalator Assembly Line Project*; Shanghai Jiaotong University: Shanghai, China, 2010.
3. Chen, F.; Wei, Y.; Wang, H. A heuristic based batching and assigning method for online customer orders. *Flex. Serv. Manuf. J.* **2017**. [[CrossRef](#)]
4. Xu, P.J.; Allgor, R.; Graves, S.C. Benefits of reevaluating real-time order fulfillment decisions. *Manuf. Serv. Oper. Manag.* **2009**, *11*, 340–355. [[CrossRef](#)]
5. Van Nieuwenhuysse, I.; Koster, R.B.M. Evaluating order throughput time in 2-block warehouses with time window batching. *Int. J. Prod. Econ.* **2009**, *121*, 654–664. [[CrossRef](#)]
6. De Koster, R.; Le-Duc, T.; Roodbergen, K.J. Design and control of warehouse order picking: A literature review. *Eur. J. Oper. Res.* **2007**, *182*, 481–501. [[CrossRef](#)]
7. Wang, X.; Zhang, J.; Yi, C. Integrated order picking and delivery scheduling optimization in B2C e-commerce environment. *Chin. J. Manag. Sci.* **2016**, *24*, 101–109.
8. Bukchin, Y.; Khmelnitsky, E.; Yakuel, P. Optimizing a dynamic order-picking process. *Eur. J. Oper. Res.* **2012**, *219*, 335–346. [[CrossRef](#)]
9. Muter, I.; Öncan, T. An Exact Solution Approach for the Order Batching Problem. *IIE Trans.* **2015**, *47*, 728–738. [[CrossRef](#)]
10. Hsieh, L.F.; Huang, Y.C. New batch construction heuristics to optimise the performance of order picking systems. *Int. J. Prod. Econ.* **2011**, *131*, 618–630. [[CrossRef](#)]
11. Gong, Y.; de Koster, R. A polling-based dynamic order picking system for online retailers. *IIE Trans.* **2008**, *40*, 1070–1082. [[CrossRef](#)]

12. Hong, S.; Johnson, A.L.; Peters, B.A. Large-scale order batching in parallel-aisle picking systems. *IIE Trans.* **2012**, *44*, 88–106. [[CrossRef](#)]
13. Henn, S. Order Batching and Sequencing for the Minimization of the Total Tardiness in Picker-to-Part Warehouses. *Flex. Serv. Manuf.* **2015**, *27*, 86–114. [[CrossRef](#)]
14. Zhang, J.; Wang, X.; Chan, F.T.S. On-line order batching and sequencing problem with multiple pickers: A hybrid rule-based algorithm. *Appl. Math. Model.* **2017**, *45*, 271–284. [[CrossRef](#)]
15. Valle, C.A.; Beasley, J.E. Optimally solving the joint order batching and picker routing problem. *Eur. J. Oper. Res.* **2017**, *262*, 817–834. [[CrossRef](#)]
16. Chen, T.L.; Cheng, C.Y.; Chen, Y.Y. An Efficient Hybrid Algorithm for Integrated Order Batching, Sequencing and Routing Problem. *Int. J. Prod. Econ.* **2015**, *159*, 158–167. [[CrossRef](#)]
17. Zhang, Y.; Huang, M.; Hu, X.; Sun, L. Package consolidation approach to the split-order fulfillment problem of online supermarkets. *J. Oper. Res. Soc.* **2017**, *69*, 127–141. [[CrossRef](#)]
18. Li, M.; Wu, Y.; Chen, N. Items assignment optimization for array automated picking device based on serial order accumulation. *Comput. Integr. Manuf. Syst.* **2016**, *22*, 2127–2134.
19. Wang, X.; Zhang, J.; Yi, C. Optimizing Operatio efficiency of e-commerce manual synchronized zone order picking system. *J. Ind. Eng. Eng. Manag.* **2017**, *2*, 209–215.
20. Lin, D.Y.; Chu, Y.M. The mixed-product assembly line sequencing problem of a door-lock company in Taiwan. *Comput. Ind. Eng.* **2013**, *64*, 492–499. [[CrossRef](#)]
21. Lu, J.; Weng, Y.; Li, X.; Tang, H.; Dong, Q. Application of hybrid artificial bee colony algorithm in mixed assembly lines sequencing. *Comput. Integr. Manuf. Syst.* **2014**, *1*, 121–127.
22. Kucukkoc, I.; Zhang, D.Z. Mathematical model and agent based solution approach for the simultaneous balancing and sequencing of mixed-model parallel two-sided assembly lines. *Int. J. Prod. Econ.* **2014**, *158*, 314–333. [[CrossRef](#)]
23. Manavizadeh, N.; Rabbani, M.; Radmehr, F. A new multi-objective approach in order to balancing and sequencing U-shaped mixed model assembly line problem: A proposed heuristic algorithm. *Int. J. Adv. Manuf. Technol.* **2015**, *79*, 415–425. [[CrossRef](#)]
24. Bottani, E.; Centobelli, P.; Cerchione, R.; Gaudio, L.D.; Murino, T. Solving machine loading problem of flexible manufacturing systems using a modified discrete firefly algorithm. *Int. J. Ind. Eng. Comput.* **2017**, *8*, 363–372. [[CrossRef](#)]
25. Zhou, L.; Ding, L.; Qiang, X.; Luo, Y. An improved discrete firefly algorithm for the traveling salesman problem. *J. Comput. Theor. Nanosci.* **2015**, *12*, 1184–1189. [[CrossRef](#)]
26. Wahab, M.N.A.; Nefti-Meziani, S.; Atyabi, A.A. Comprehensive Review of Swarm Optimization Algorithms. *PLoS ONE* **2015**, *10*, e0122827. [[CrossRef](#)] [[PubMed](#)]
27. Iwasaki, Y.; Suzuki, I. Job-shop scheduling approach to order-picking problem. In Proceedings of the ASME/ISCIE 2012 International Symposium on Flexible Automation (ISFA2012), St. Louis, MI, USA, 18–20 June 2012.
28. Huang, X. An Mixed Assembly Line-Based Optimization Method of Order Picking Operations of Online Supermarkets. Master’s Thesis, North China Electric Power University, Beijing, China, March 2018.



© 2018 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 (<http://creativecommons.org/licenses/by/4.0/>).