Hindawi
Scientific Programming
Volume 2021, Article ID 8863391, 17 pages
https://doi.org/10.1155/2021/8863391

Research Article

A Heuristic Solution Approach to Order Batching and Sequencing for Manual Picking and Packing Lines considering Fatiguing Effect

Xiaochun Feng[1] **and Xiangpei Hu**[2]

[1]*College of Economics and Management, Northwest A&F University, Yangling, Shaanxi 712100, China*
[2]*Institute of System Engineering, Dalian University of Technology, Dalian, Liaoning 116023, China*

Correspondence should be addressed to Xiaochun Feng; fxc11011@126.com

Received 15 September 2020; Revised 9 December 2020; Accepted 29 March 2021; Published 14 April 2021

Academic Editor: Shah Nazir

In this research, we study an extended version of the joint order batching and scheduling optimization for manual vegetable order picking and packing lines with consideration of workers' fatiguing effect. This problem is faced by many B2C fresh produce grocers in China on a daily basis which could severely decrease overall workflow efficiency in distribution center and customer satisfaction. In this order batching and sequencing problem, the setup time for processing each batch is volume-dependent and similarity dependent, as less ergonomic motion is needed in replenishing and picking similar orders. In addition, each worker's fatiguing effect, usually caused by late shift and repetitive operation, which affects order processing times, is assumed to follow a general form of logistic growth with respect to the start time of order processing. We develop a heuristic approach to solve the resultant NP-hard problem for minimization of the total completion time. For order batching, a revised similarity index takes into account not only the number of common items in any two orders but also the proportion of these items based on the vegetable order feature. To sequence batches, the genetic algorithm is adapted and improved with proposed several efficient initialization and precedence rules. Within each batch, a revised nondecreasing item quantity algorithm is used. The performance of the proposed heuristic solution approach is evaluated using numerical instances generated from practical warehouse operations of our partnering B2C grocer. The efficiency of the proposed heuristic approach is demonstrated.

1. Introduction

In this paper, we consider the optimization of order batching and sequencing on manual vegetable order picking and packing lines as a practical parallel machine serial-batch scheduling problem.

This problem is posed by our partner, an online B2C fresh produce grocery supplier in Beijing, China. In vegetable B2C direct-sales mode, vegetables are harvested from the production bases according to the orders online. Then, after the cleaning and precooling, SKU packing, order picking, and packing, the fresh products are able to be delivered to Intelligent Distribution Dispensers located in different communities on the same or early next day. To solve the issues of perishability and time sensitivity for fresh products, the company adopts JIT operation mode in their distribution center, aiming at zero inventory and processing customized orders. The order picking and packing for fresh products is operated on parallel manually operating lines rather than order picking in aisle system for other commercial products (see Section 2). In the distribution center, order picking and packing is the least efficient and highest cost operation in order processing and usually has to be scheduled to night shift for early delivery, which will lead to a large deviation from expectation if not taking pickers' fatiguing effect into account.

This study is further motivated by the common need to achieve excellent scheduling operations for manual vegetable order picking and packing lines. Previously, our industry partner's attempts in developing fast and effective

decision approaches for customer order scheduling had limited success. Less satisfactory performance is still observed, which is of a similar situation to many other vegetable order picking and packing lines in China. A common reason for such deficiency is that mathematical models adopted by these tools suffer from somewhat excessive simplification, in addition, order batching and scheduling is not optimized based on unique characters for fresh product. Many relevant aspects of manual picking lines have not been incorporated, including similarity dependency in batch setup time brought by the underlying convenience in replenishing and picking and work fatigue resulting from late shift work and repetitive operation. These practical aspects may affect the actual production plan significantly. In this paper, we accommodate them into an order batching and sequencing optimization model with the expectation to enhance order process efficiency for fresh products and better serve customers. In such systems, each order is considered as the smallest unit and most operations are labor-intensive.

Order batching and sequencing problems in warehouses have been studied for decades [1–5]. As mentioned, the order picking and packing system studied here is different from traditional order picking and scheduling, which is mostly based on picker's routing in aisle system, the picker needs to walk and search among the racks, and the objective is to minimize the total walking distance [6, 7], etc. However, human factors like learning and fatiguing effect are generally overlooked. In recent years, some researchers have proposed to consider such ergonomic effects [8–12]. Therefore, this study focuses on the special order picking system of fresh products with potential human fatiguing effects, which could be considered a practical exploration in this new area. Since orders will be batched and operated on the parallel picking and packing lines (treated as machines) without picker routing, the problem studied here could be categorized as order batching and scheduling on parallel machines, and relevant literature is reviewed next.

Order batching is a method of grouping a set of orders into a number of subsets, which can be retrieved or processed together in one operation. The related methods of order batching have also been studied for decades and classical solution approaches for order batching problems can be distinguished into priority rule-based algorithms, seed algorithms, saving algorithms, and data mining approaches [13–16]. Related batching operation is known as batch scheduling problem, which needs to determine optimal grouping and scheduling decisions about jobs that are about to be processed on capacitated machines. Current research has two main assumptions about the jobs in the same batch. One is that all jobs in the same batch are completed together upon the completion of the last job of the batch and, within each batch, the jobs are processed sequentially (also known as s-batch) [17–19]. Another assumption is made that the jobs in the same batch are completed together with the same starting time and same completion time (also known as p-batch) [20–23]. Based on the involved machines, current research about batch scheduling is reviewed from two main categories: single machine batch scheduling [24–27] and batch scheduling on multiple machines. Batch scheduling involved with multiple machines includes two sides: parallel batch machines [28–32] and flow shop batching and scheduling problems [33–35].

For single machine batching scheduling problems, most of the studies above seek polynomial-time solution methods on special machine scheduling cases. Although they provide theoretical bases for practice, they are not necessarily suitable for delivering practical solutions to many industry clients, including our partner in China. With a certain degree of similarity, studies about multiple machine batching and scheduling are faced with different operating systems, different setups, processing times, different resource constraints, and different optimizing objectives. The processing time of jobs is assumed to be constant values in reviewed literature so far. However, this assumption is not appropriate for the modeling of many manual processes where very often a job, executed in the same or almost the same conditions, has a varied processing time. In manual vegetable order picking and packing line, setup time depends on the similarity between adjacent batches and item types that needs to be prepared, the processing time depends on the current worker's fatigue and needed item types in the order. A worker has to assemble a large number of similar products and the processing time to assemble one product depends on his knowledge, repetition times, and other aspects which will change along with time. To bridge these research gaps, joint optimization of order batching and sequencing in the manual vegetable order picking and packing system is studied here while taking fatiguing effect of workers into account.

The efficiency of a worker decreases after working for long hours, which can lead to the increased processing time for the same task. This phenomenon is often referred to as the fatiguing effect (also known as the aging effect or efficiency deterioration). In addition, different workers may have different working habits, and thus their working efficiencies may deteriorate following different fatigue curves. It could lead to considerable differences in optimal scheduling for labor-intense tasks, such as different task allocation decisions. The fatiguing effect in the context of scheduling has been recognized in the literature, for instances, on different machine types: single machine scheduling [36–38], parallel machine scheduling [39, 40], flow shop scheduling [41, 42]; different types of fatigue: position related fatigue [43], linear (or piecewise linear), and time related fatigue [44, 45]. In recent years, the study of batch scheduling considering fatiguing effect has appeared [25, 46–52].

As reviewed, most of the research studies machine scheduling (or job sequencing) with consideration of machine deterioration effects and have not been extended to labor-intense operation systems with human fatiguing effects. In addition, for mathematical convenience, the relationship of job processing time with respect to job start time or position was set to linear or low-degree polynomial in the most relevant literature. Our problem is order batching and sequencing in the context of practical manual order picking and packing operations considering pickers' fatigue. In this

paper, we instead use a generalized logistic function to characterize the work fatiguing effect, and subsequently, job processing time. It is originally developed for growth modeling for various natural biological processes like population growth, weight growth, stress accumulation, etc. [53]. This fatigue trend could be captured very well by a general logistic function that has been used to simulate the relationship between human's stress and work time in human reliability analysis [54, 55]. Different from unlimited descending machine deterioration, human's working efficiency has an upper-bound in the beginning and a lower-bound at the end and has a different fatiguing rate. Hence, a general logistic relationship is used here to map human's operating efficiency into a certain range.

In summary, as an extension to traditional order batching and scheduling problems, the studied problem is proposed based on real world application with some new characters. Pickers' fatigue needs to be included in the picking operation, and setup time needs to be set based on features of fresh product. This motivates our innovations as follows: (1) We establish the order batching and scheduling model for vegetable order picking and packing system. In the model, the setup time is associated with the similarity degree of batches and the item type number in the batch. The processing time of the order is associated with pickers' fatigue and order size. This model can reflect the characteristics of vegetable picking more concretely. (2) The fatiguing effect curve of pickers based on logical function is proposed, which is not considered in previous literature, and it could reflect changes of human fatigue more naturally. (3) We propose a heuristic solution approach aiming at solving the scheduling problem in picking lines more efficiently. In the order sequencing stage, a revised nondecreasing item quantity (NDIQ) algorithm is proposed to sequence the orders within a single batch after brief prove while different batches are sorted using an improved genetic algorithm (GA). To overcome drawbacks that standard encoding schemes may result in, two-echelon real-number encoding structure is adopted and two precedence rules are combined to initialize the chromosomes more efficiently. Our contributions lie in two aspects: firstly, this study is the extension of the new research trend of considering human factors in the order picking system; secondly, our model, which is special order batching and scheduling model with similarity-based setup and pickers' fatiguing effect, enriches the batching scheduling field. For real world implication, the proposed solution approach could be applied and extended for a labor-intense order processing system.

This research is organized as follows. Section 2 introduces and describes the studied problem in detail. Section 3 presents the heuristic solution approach. Section 4 shows the performance of the proposed solution methods. Finally, Section 5 draws conclusions and outlines future research directions.

2. Problem Description

The problem initially arose from our collaborator, a fresh produce supplier in Beijing operating in B2C direct-selling mode. The basic process flow (see Figure 1) is that customers place orders online first and then vegetables are harvested from a planning base based on the day. After cleaning and precooling in the distribution center, fresh products of various types are first prepared as SKUs and stored in a temporary storage area. Then the orders are picked and packed according to customer order and dispatched to meet the requirement of "same day arrival" or "next day arrival." Following the process flow, the layout of the distribution center is designed in Figure 2, where the order is processed in parallel lines (seen as machines).

Due to perishability and time sensitivity, grouped order picking is adopted in the processing where orders are grouped based on their similarities, and when the grouped orders arrive, the picker will pick different types and amount of SKUs for each order and pack it. This operating mode has integrated the advantages of low error rate from picking by order and efficiency from the grouped operation. Hence, before order picking and packing, similar orders are batched and scheduled in sequence for following assembly operations in picking lines. Thus, our problem can be summarized as a parallel machine batch scheduling problem considering workers' fatiguing effect, shown in Figure 3.

In this system, deterministic operation and optimization are considered. Although the order processing time is uncertain, it is estimated based on order size and working efficiency. Order size could be measured by the contained number of SKUs. Working efficiency could be defined as unit picking time in lean management which is the time spent on picking one SKU type. The worker has an initial unit picking time and this time will increase nonlinearly along with time, known as a fatiguing effect. Different from the fatigue of machines, workers' fatiguing effect is often caused by stress and strain with time, and the working efficiency will not continue degrading unlimitedly but with a bound. On the contrary, unit picking time will increase until a bound. The rate of the growth increases from an initial value reaches a maximum at a point of inflexion and then decreases towards zero at an upper asymptote. Furthermore, different workers may have different working habits, and thus their working efficiencies may deteriorate following different fatigue curves. It could lead to considerable differences in optimal scheduling for labor-intense tasks, such as different task allocation decisions. This fatigue trend could be captured very well by a general logistic function, which has been used to simulate the relationship between human's stress and work time in ergonomic studies. Hence, unit picking time of worker o could be described as $f_o(t) = A_o + (K_o - A_o)/(1 + e^{-B_o(t-4)})$, A_o and K_o are initial and final asymptote of unit picking time, B_o is deterioration rate. 4 is the time point of stabilization which means that the workers have maximum deterioration rate after about 4 hours (second could also be used) from starting time point (usually the working period is 8 hours). Different values could be used for time points of stabilization based on practice. Therefore, the unit picking time follows this fatigue curve to increase over time nonlinearly with an asymptotical stable value.

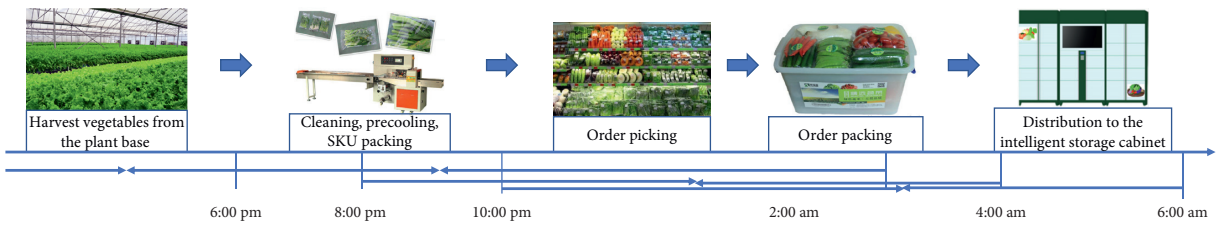

Figure 1: The process flow of the vegetable B2C direct-selling mode.

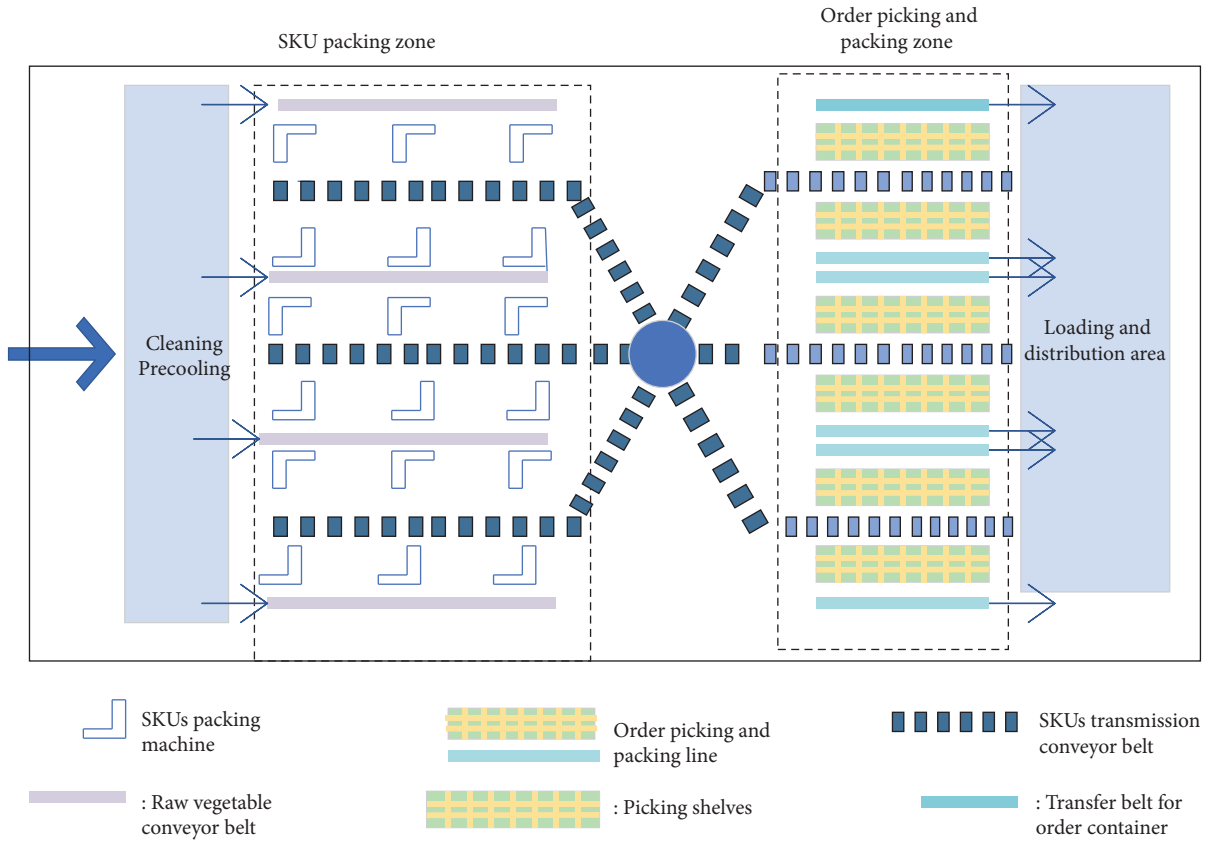

Figure 2: Layout of the vegetable distribution center.

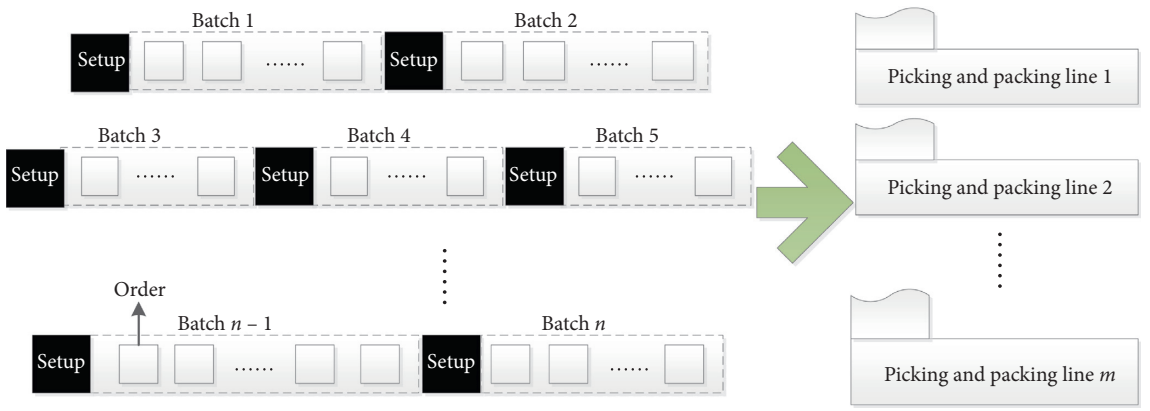

Figure 3: Order batching and sequencing process for parallel packaging lines.

Before processing one batch, a certain setup time is needed to preparing all the necessary items, for example, the packing materials, SKUs, etc. The more similar the orders in the batch are, the more consistent that is with requisite materials and SKUs, and less setup time will be needed. Contained item types determine the complexity of the batch, and the more complex the batch is, the longer the setup time will be. We assume that the setup time for batch j here follows $SP_b = V_b \cdot e^{-s_b}$. V_b is the number of item types in batch b. s_b is the overall similarity degree of batch b. Orders within each batch are processed sequentially, and our objective is to minimize total completion time.

Summarizing the above description, we report this scheduling problem below:

$$\begin{aligned} & m|s - \text{batch}, \\ & SP_b = V_b \cdot e^{-s_b}, \\ & p_{io} = Q_i \cdot f_o(t)|\sum_i C_i. \end{aligned} \tag{1}$$

For serial-batch scheduling problems on m picking lines, the setup time SP_b of batch b is volume and similarity dependent, the processing time p_{io} on order i of operator o equals item types quantity multiply current unit picking time and the objective is to minimize total completion time $\sum_i C_i$. Orders will be batched and then scheduled sequentially.

Note that the studied problem can be reduced to the order batching problem, which has proved to be NP-hard [56]. Thus, we resort to the heuristic solution.

3. The Heuristics Solution Approach

Since grouped order picking and scheduling is an NP-hard problem and plentiful orders will flow into the decision period, it is urgent to develop a fast solution approach that could be applied on order operation for fresh product suppliers; lots of studies have been conducted on heuristic algorithms for order scheduling [57–64]. The focus of this section is to develop an efficient heuristic solution approach to solve the joint order batching and sequencing problem formulated in the previous section. Because of the solving complexity, seeking a global optimum solution by solving this joint problem simultaneously is not realistic. Moreover, the decisions of such planning issues are often made sequentially at an operative level in practice. Naturally, the original problem is divided into two subproblems: order batching and batch sequencing. In the first phase, the orders are batched based on their similarities on common items in the proposed batching algorithm. Next, in the second phase, a revised nonincreasing item quantity algorithm is proposed after being proven to sequence orders within every single batch and then different batches are sorted using a hybrid genetic algorithm based on precedence rules derived from practical production experience. Order batching and batch scheduling are closely related and interact even though they are decomposed into two subproblems out of efficiency consideration. When certain orders are batched together, the similarity between two batches is certain and will determine their interval setup time. In reverse, the result of batch scheduling is feedback to order batching, based on which, orders need to be rebatched and resulted batches will be reevaluated. The flowchart of the overall solution procedure is sketched for proposed two-phase heuristic approach, Figure 4.

3.1. Order Batching Algorithm upon Similarity (OBAS). To capture similarity more precisely, we consider not only the category number for common items but also the percentage of common items within each order. The similarity of two orders is calculated by the following formula to include the percentage of common items in this research:

$$S_{ij} = \frac{1}{2} \cdot \left(\frac{G}{H} \cdot \frac{\sum_g^G Q_{i,g}}{Q_i} + \frac{G}{V} \cdot \frac{\sum_g^G Q_{j,g}}{Q_j} \right), \quad \forall i, j = 1, \ldots, N. \tag{2}$$

Here, in equation (2), it is supposed that there are total H and V item categories in order i and j, respectively, with G as the category number for common items. $Q_{i,g}$, $Q_{j,g}$ denote quantity number of items in common category g for order i and j.

In current literature, the similarity for two orders is usually defined by included category number of common items [65], which is defined by equation (3), U denotes total item numbers in the union of order i and j.

$$S_{ij} = \frac{G}{U}, \quad \forall i, j = 1, \ldots, N. \tag{3}$$

As shown in equation (3), in current heuristic algorithms that rely on similarity index, shared item type is commonly focused and the impact of the percentage of common items within each order on their similarity has been neglected. Even though two orders have the same item types, a big quantity difference or, more precisely, a big percentage difference may cause lots of unnecessary picking motions when preparing items that are not effective. In addition, the study object in our problem is "order," not "job," so that the setup time between batches is similarity value dependent.

There is an instance to explain the difference between these two formulas. In Table 1, 10 items and 3 orders are considered. The value of equation (3) for these 3 orders is the same 2/3, whereas orders 1 and 3 are more similar intuitively because common items are the main components of both orders.

Based on the above discussion, order batching process in the first phase is presented as follows, Algorithm 1:

The main idea of this batching process is to batch orders with the highest similarity while satisfying the batch capacity constraint.

3.2. Sequencing between Batches: GA-Based Algorithm. It is proved that the parallel machine sequencing problem in the second phase here is NP-hard in the strong sense generally. For the past decades, genetic algorithm has received considerable attention for solving such difficult combinatorial

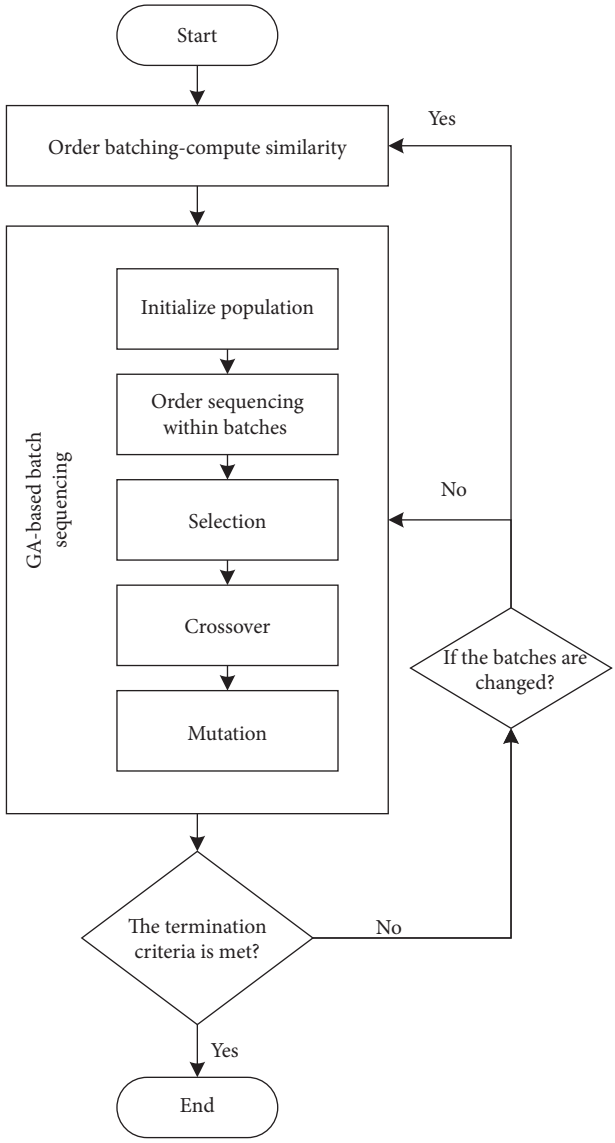

Figure 4: Overall solution procedure for proposed two-phase heuristics.

Table 1: Similarity comparison based on equations (2) and (3).

	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10
O1	8	1	0	0	9	10	0	0	1	1
O2	1	2	0	0	2	1	10	8	0	0
O3	10	2	1	1	11	8	0	0	0	0

optimization problems and demonstrated good performance to get satisfying solutions [66–68]. To obtain better sequencing solutions between batches, the conventional genetic algorithm is enhanced by including several techniques to improve solving efficiency and increase the exploitation ability. The main functional modules of the proposed enhanced genetic algorithm are interpreted as follows.

3.2.1. Encoding and Initialization. To overcome drawbacks (redundancy problem, insensitivity to crossover operators, etc.) that standard (item-oriented) encoding schemes may result in [69], a two-echelon real-number encoding structure is adopted to represent each chromosome or solution in consideration of problem characteristic. Suppose there are n batches and m picking lines, the upper echelon is encoded by a string of $n + m$ distinct genes, composed of n batch genes and m separate machine genes, meanwhile for each batch gene, its contained orders are expressed at the lower echelon. They evaluate the performance of each chromosome. Fitness function is defined as monotonic decreasing function about objective function. The negative exponential function is used here and the larger the fitness means better. The roulette wheel method is adopted for later selection operations.

For most searching algorithms, especially stochastic search techniques, the starting point could have non-negligible influence on the overall efficiency reflected by the subsequent iterative path and total computation time. To maintain the diversity of population and start from a relatively reasonable solution, two precedence rules, extracted from practical experience, are combined together with random initialization in chromosomes initialization module: (1) picking line with earliest completion time has the priority to be scheduled; (2) the batch with a larger amount of items is more likely to be assigned to high-efficiency picking lines. The details are provided as below, Algorithm 2.

Please note that rule 1 (line 6-line 8) and rule 2 (line 9-line 11) in Algorithm 2 are exclusive and will be applied to each chromosome in the population to choose the current pairwise batch and picking line in initial stage.

3.2.2. Crossover and Mutation. Corresponding to the two-echelon encoding scheme, classical crossover and mutation operators need to be modified here to reflect genetic evolution at different levels. The crossover and mutation procedures are depicted exemplarily for 3 picking lines and 7 batches in Figures 5 and 6.

3.2.3. Crossover Operation.

Step 1: Select two parental chromosomes P_1, P_2 from the population randomly and select a crossing section (shaded area) for each of them. Picking line genes (with a star) and batch genes may be included at the same time (see Figure 5(a));

Step 2: Interchange selected crossing sections between P_1 and P_2. The order expression at the lower echelon will also move with batch genes. Then check the redundancy of each offspring chromosome O_1, O_2, repeated genes (batch gene 4 and picking line gene $3*$) are deleted (see Figure 5(b));

Step 3: After deletion, the completeness of each chromosome should also be checked. At this same location where deletion happens, the missing packing

Step 1: Choose the seed order i^* randomly;
Step 2: Check batch capacity constraint, go to Step 3 if not exceeding capacity, go to Step 6 otherwise;
Step 3: Calculate similarity S_{i^*j} for orders j and sort orders by S_{i^*j} in descending way;
Step 4: Select order j with highest S_{i^*j}, if multiple orders have same similarity, randomly pick one;
Step 5: Append order j to the seed batch i^* and check batch capacity constraint, go to Step 6 if not exceeding capacity, go to Step 7 otherwise;
Step 6: Combine order j and i^* as a new order i, repeat Step 1 $\sim$ Step 6 until all orders are batched;
Step 7: Output the order batch.

Algorithm 1: Batching algorithm upon similarity.

Initialization: $b = 1$; unassigned batches $H = \varnothing$ with index u;
completion time $I_k = 0, \forall k \in$ total picking lines K;
vector $T = I_k | k \in K$, objective value $obj = 0$
While ($b \leq$ total batch number N) **do**
  if ($T = \varnothing$) **then** $T = \{I_k | k \in K\}$
  # follow rule 1 to choose picking line k^* and batch u^*
  get u^* randomly in unassigned batches R_u
  $k^* = *\text{argmin}\{I_k, k \in K\}$
  # follow rule 2 to choose picking line k^* and batch u^*
  choose subset SB_u from R_u randomly
  $u^* = *\arg\max$ {item quantity $Q_u, u \in SB_u$
  $k^* = *$ argminsetup time $\{E_k = f(I_K + \text{setup time}\, ST_{u^*}, I_k \in T)\}$
  $B_u := B_u / u^*; H := H \cup u^*$
  processing time $P_{u^*,k^*} = Q_{u^*} \cdot f(I_k + \text{setup time} ST_{u^*}$
  finishing time $FT_{u^*,k^*} = I_{k^*} + P_{u^*,k^*}$
  $T := T / \{I_{k^*}\}; I_{k^*} = FT_{u^*,k^*}$; obj = obj + FT_{u^*,k^*}
  $b := b + 1$
End

Algorithm 2: Initialization module in improved genetic algorithms.

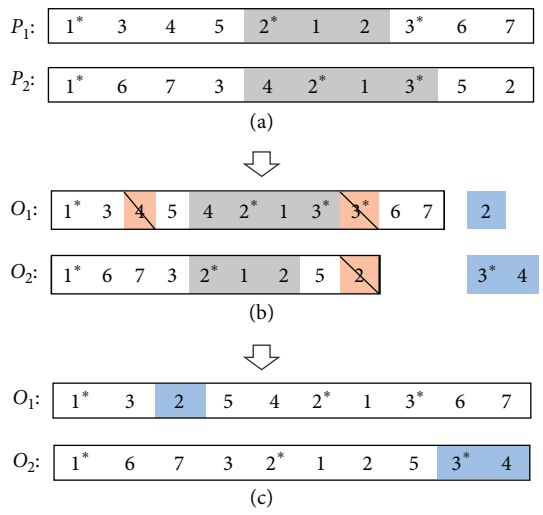

Figure 5: Crossover operation in the two-echelon encoding scheme.

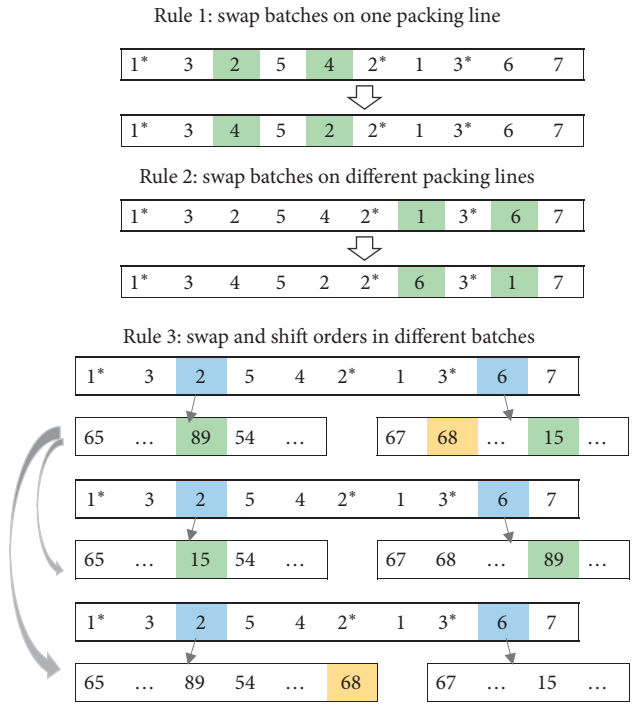

Figure 6: Mutation operation in the two-echelon encoding scheme.

line and batch genes will be inserted (picking line genes goes first) (see Figure 5(c)).

3.2.4. Mutation Operation.

(1) SWAP different batches on the same picking line (see Figure 6(a)).

(2) SWAP different batches on different picking lines (see Figure 6(b)).

(3) In addition, one or more orders may SWAP between batches or SHIFT from one batch to another batch (see Figure 6(c)).

The crossover and mutation happen with a predefined probability. Please note that these operations might lead to infeasible solutions due to violation of capacity constraints. Such situations are dealt with by applying Best-Fit-Rule [69], which moves the last customer order from an infeasible batch to another batch with sufficient capacity. The procedure terminates if no further improvement can be obtained in user-defined continuous iterations.

3.3. Sequencing within Batches: Revised Nondecreasing Item Quantity Algorithm. As mentioned, fatiguing effects of workers are considered and modeled in the picking process, which leads to a gradual change of order processing time. To sequence the orders within every single batch, a revised nondecreasing item quantity algorithm is proposed after being proved to be optimum scheduling. Some theorems are listed here to support it.

Lemma 1. *For the problem* $1|p_i = Q_i(a+bt)|\sum C$, *an optimal schedule can be obtained by arranging orders in order of nondecreasing orders of* Q_i.

Proof. This can be proved by interchanging adjacent jobs. Suppose under an optimal schedule π, there are two adjacent orders i and j, j followed by i, suppose item quantity in order i and j satisfy $Q_i > Q_j$. Let the starting time of order i is t, and the sum of completion time of order i and j is

$$C_i(\pi) + C_j(\pi) = t + Q_i(a + b\cdot t) + \left[t + \left(Q_i + Q_j + b\cdot Q_i\cdot Q_j\right)(a + b\cdot t)\right]. \tag{4}$$

If a nonoptimum schedule π' is obtained by performing a pairwise interchange on orders i and j, then similarly,

$$C_i(\pi') + C_j(\pi') = t + Q_i(a + b\cdot t) + \left[t + \left(Q_i + Q_j + b\cdot Q_i\cdot Q_j\right)(a + b\cdot t)\right]. \tag{5}$$

If $Q_i > Q_j$, it can be verified that

$$C_i(\pi') + C_j(\pi') < C_i^{\pi} + C_j^{(\pi)}. \tag{6}$$

This contradicts the optimality of π.

Remark 1. $f(t) = A + (K - A)/(C + Q\cdot e^{-B\cdot(t-M)})^{1/v}$ can be treated as the envelope of countless piecewise linear functions.

For the piecewise linear processing time problem, we use a heuristic algorithm presented by Sundararaghavan and Kunnathur [70] to solve it. The procedure is called Operation Exchange: let π be a schedule, define $E = i \in \pi$: $s_i \le D$

as a job set that includes all the jobs with its starting time less than D, and $L = \pi \backslash E$. Exchange a job $i \in E$ with a job $j \in L$ and reorder the jobs in $j \cup (E \backslash i)$ and reorder the jobs in $i \cup (L \backslash j)$.

To summarize, for one batch on the picking line, a revised nondecreasing item quantity (NDIQ) heuristic is proposed to schedule orders, Algorithm 3.

4. Numerical Experiments

In this section, sets of experiments are conducted to evaluate the performances of the proposed two-phase heuristic approach (HGA) in Section 3 for order batching and sequencing. First of all, the effectiveness and scalability of the approach are demonstrated by comparing it with other reported algorithms in relevant research. Then parameter settings for fatigue curves are carried out for the approach, corresponding results from the experiment have been shown. All experiments are implemented in [71] version on Windows 7 operating system with dual Intel Cores (CPU2.0 & 2.5Ghz) and 8 GB RAM.

4.1. Effectiveness of HGA. For effectiveness tests, experimental data are simulated based on transaction order data from an online retailer company in Beijing, China. Characteristics of the simulated dataset are summarized, including the total order number, total item types, maximum quantity for each kind of item in each order, maximum item types in each order, and batch capacity, from left to right in Table 2. Efficiency settings for picking lines are assumed in Table 3. The total number of workers is assumed to be 30. Their initial unit picking time, fatiguing rate and time point of stabilization are uniformly generated in according ranges.

The performance of the proposed HGA is illustrated on different possible order structures by comparing the other four representative algorithms in the area. To show the direct effectiveness of our proposed similarity formula in order batching phase (equation (2)) and two initialization rules in sequencing phase (Algorithm 2), two basic references are specially designed. GA-1 is HGA with the commonly adopted similarity (equation (3)) and GA-2 is HGA without two proposed rules in the initialization module. Besides, SA [72] and GSA [25] are chosen because of a very similar research problem. In the approach of SA and GSA, the orders are ranked and batched following their processing time and then scheduled by simulated annealing and gravitational search algorithms. GSA considers order batching and batch scheduling altogether, while SA batches the order first and then schedules the batches. Since the research problems in the citations are different, some minor revisions have been made when applying these two algorithms here. The MBF (Modified Best Fit) rule is changed to MITN (Minimum Item Type Number, first in our paper) rule to generate an initial solution for SA. For GSA, random encoding is used, and after orders are randomly assigned to processing lines, they are then batched based on order similarity. In each batch, orders are sequenced following the proposed revised NDIQ heuristic here. The parameter setting of HGA in the experiments is listed in Table 4.

Then, for a fixed total order number (800), five different order structures (A(60-12), B(80-16), C(100-20), D(150-30), E(200-40)) are tested here, for instance, A(60-12) means that in order structure A, total item types are assumed to be 60 and maximum item types in each order are 12. The other parameters are kept the same as Table 4 for all algorithms here. Mean values of objective value, setup time, and running time are recorded in Table 5 for each algorithm after ten runs. Imp% is the improvement percentage of HGA against other algorithms. "++" means the imp% less than −100%. Note that, in all following related results, objective value (total completion time) is at $1E+6$ scale, setup time is at $1E+2$ scale, running time is at $1E+2$ scale.

As shown in Table 5, along with increasing of total item types in different order structures, the mean value of three metrics raises up overall for all the algorithms. For the mean objective value under different order structures, generally speaking, HGA has the shortest total completion time followed by GA-1 (being improved at level 1.9% ~ 8.0%), SA (10.2% ~ 12.9%), and GSA (24.4% ~ 30.4%); GA-2 (42.5% ~ 49.9%) is the worst in most cases. Taking order structure C(100-20) as an example, HGA has an improvement of 6.7% and 18.2% on mean objective value and mean setup time but higher mean running time comparing with GA-1. Since GA-2 uses the same similarity formula with HGA, their setup time and obtained batch number are very close, see Figure 7. In the histogram of the mean setup time for each algorithm, the resulting batch number is also labeled. The needed setup time is related to the total item types within the next batch and their overall similarity. SA approach has the same batch number 57 always as the total order number is fixed here, and it groups orders based on their processing time; however, its mean setup time changes when the order structure changes.

In addition, to display the full range of variation (from min to max) for total completion times (natural log basis) in ten runs, a boxplot is applied in Figure 8. As shown, the performance of HGA is very stable. GSA has the maximum variability followed by GA-2, and sometimes the min value of GSA could match GA-1.

From this set of experiments on different order structures, it could be derived safely that the proposed two-phase heuristic method with according improvements on similarity formula and genetic algorithm is effective for order batching and sequencing problem.

4.2. Scalability Test. Besides the effectiveness at the current problem scale, the scalability of proposed two-phase heuristics is also explored by increasing total order numbers after fixing the order structure in Table 6.

Under this order structure, the total order number has been added up from 600 to 1500 to examine the performance of the algorithms. Same as last experiments, each algorithm has been run ten times for each order number, the mean value is recorded. Please note that the values are at the same scale in Table 5. In the results of the scalability test (Table 7), as it can be seen, along with increasing of order numbers, the mean value of total completion time, setup time, and

Step 1: Sort orders of one batch according to nondecreasing Q_i;
Step 2: Choose a middle point in the sorted sequence and divide the sequence into two parts E and D;
Step 3: Carry out an operation exchange with any order in E with any other order in the part D if it leads to a reduction in the objective function, then continue to carry out operation exchange until no such objective function reducing exchange exists;
Step 4: Keep the order sequence unchanged.

Algorithm 3: Revised nondecreasing item quantity algorithm (NDIQ).

Table 2: Order information simulated dataset.

Total order number	Item types	Max quantity/item/order	Max item types/order	Batch capacity
800	60	10	15	15

Table 3: Efficiency settings for picking lines.

Number of workers	Initial unit picking time	Fatiguing rate	Time point of stabilization	Final unit picking time
30	U (5,30)	U (0.0001,0.0005)	U (3000,10000)	U (100,200)

Table 4: Parameter setting for HGA.

Probability of mutation	Probability of crossover	Population size	Maximum iteration number	Allowed unimproved iteration number
0.15	0.6	60	500	30

Table 5: Results of all algorithms for different order structures.

Algorithms	Metric (sec)	A(60-12)		B(80-16)		C(100-20)		D(150-30)		E(200-40)	
		Value	Imp%	Value	Imp%	Value	Imp%	Value	Imp%	Value	Imp%
HGA	Obj	1.393	–	2.115	–	2.798	–	2.988	–	3.241	–
	Setup	7.490	–	10.134	–	12.758	–	19.665	–	23.722	–
	Run	12.968	–	14.524	–	15.862	–	16.936	–	18.091	–
GA-1	Obj	1.449	3.9	2.155	1.9	3.000	6.7	3.248	8.0	3.425	5.4
	Setup	8.488	11.8	12.049	15.9	15.593	18.2	23.775	17.3	28.404	16.5
	Run	4.343	++	5.564	++	5.862	++	6.682	++	7.697	++
GA-2	Obj	2.778	49.9	4.084	48.2	5.254	46.7	5.197	42.5	5.865	44.7
	Setup	7.491	–	10.133	–	12.758	–	19.665	–	23.722	–
	Run	12.205	−6.3	12.799	−13.5	13.475	−17.7	14.347	−18.1	15.139	−19.5
SA	Obj	1.599	12.9	2.522	16.1	3.373	17.0	3.731	19.9	3.61	10.2
	Setup	26.239	71.2	34.904	71.0	42.694	70.1	51.541	61.8	59.174	59.9
	Run	2.984	++	3.467	++	3.630	++	3.551	++	3.745	++
GSA	Obj	1.842	24.4	2.896	27.0	4.02	30.4	4.048	26.2	4.373	25.9
	Setup	27.646	72.9	36.349	72.1	44.881	71.6	52.895	62.8	59.930	60.4
	Run	19.592	33.9	21.636	32.9	24.724	35.8	37.337	54.6	41.727	56.6

running time has been multiplied several times, which is much faster than that of Table 5. However, the overall performance ranking of the algorithms still follows a similar pattern. From the other perspective, the distribution of completion time (natural log basis, see Figure 9) has exhibited the variance in multiple running at a different level of the total order number. Comparing with the grouped boxplot in Figure 8 for different order structures, all algorithms have shown less variability except GA-2 and GSA, but still, GA-2 has the worst performance on mean completion time. Sometimes, the best solution that GSA provides in these multiple runs could almost reach the best level of HGA (Figure 9). However, GSA is the most time consuming before the order number reaches 1200 (Table 7).

In Tables 5 and 7, the mean performance in ten runs of the five algorithms has been listed for different order structures and order numbers, and the variation range of completion time in the ten runs is captured in Figures 8 and 9. It is seen that the best scenario of the GSA algorithm in ten runs has the minimum completion time for order structure A(60-12) and A(600), then HGA starts to outperforms GSA when the order number increased. As seen from the box plot in Figures 8 and 9, HGA and GSA have the best case result than the other three algorithms. Hence, we chose the best

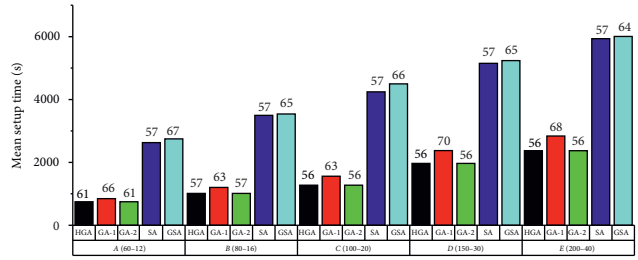

Figure 7: Mean setup time and batch number for each algorithm, different order structures.

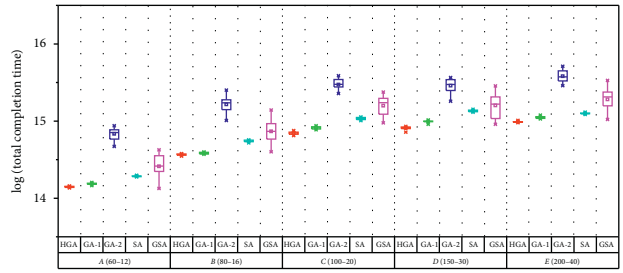

Figure 8: Range of variation for each algorithm in multiple times, different order structures.

Table 6: Fixed order structure for scalability test.

Item types	Max quantity/Item/Order	Max item types/Order	Batch capacity
60	10	15	15

Table 7: Results of all algorithms for different order numbers.

Algorithms	Metric (sec)	A(600)		B(800)		C(1000)		D(1200)		E(1500)	
		Value	Imp%	Value	Imp%	Value	Imp%	Value	Imp%	Value	Imp%
HGA	Obj	0.748	–	1.889	–	4.002	–	5.774	–	13.440	–
	Setup	5.879	–	8.158	–	9.68	–	10.981	–	14.068	–
	Run	5.731	–	12.939	–	25.032	–	50.881	–	92.383	–
GA-1	Obj	0.800	6.5	1.928	2.0	4.217	5.1	6.043	4.5	13.827	2.8
	Setup	6.915	15.0	8.898	8.3	10.831	10.6	12.716	13.6	15.638	10.0
	Run	2.907	−97.1	5.392	++	9.772	++	22.864	++	46.192	++
GA-2	Obj	1.294	42.2	4.052	53.4	8.075	50.4	11.921	51.6	24.129	44.3
	Setup	5.879	–	8.158	–	9.68	–	10.951	–	14.068	–
	Run	5.516	−3.9	12.529	−3.3	24.634	−1.6	50.338	−1.1[a]	91.452	−1.0
SA	Obj	0.905	17.4	2.140	11.7	4.522	11.5	6.317	8.6	14.588	7.9
	Setup	21.406	72.5	28.356	71.2	35.654	72.9	40.483	72.9	53.141	73.5
	Run	2.605	++	3.506	++	4.11	++	4.908	++	5.996	++
GSA	Obj	1.083	31.0	2.686	29.7	5.065	21.0	7.152	19.3	16.465	18.4
	Setup	23.934	75.4	30.27	73.0	37.181	74.0	41.101	73.3	52.863	73.4
	Run	22.558	74.6	33.36	61.2	40.686	38.5	50.581	−0.6	63.608	−45.2

scenario results in terms of total completion time for HGA and GSA under different order structures and numbers and summarize them for comparison in Table 8.

Besides performance test on different order structures and numbers, the dynamics of the proposed approach have also been evaluated with different settings for three parameters: batch capacity, crossover, and mutation operator in the GA algorithm. For this group of experiments, fixed simulation parameters are in Table 9. Here the solving process is repeated multiple times to obtain min–max and average value. Along with batch capacity increase from 5 to 30, the obtained total completion time and setup time are plotted in Figure 10. The total setup time is lower when the batch has a larger capacity in Figure 10(b), which leads to a smaller batch number. However, a larger batch needs a longer setup time than a smaller batch. Thus the completion

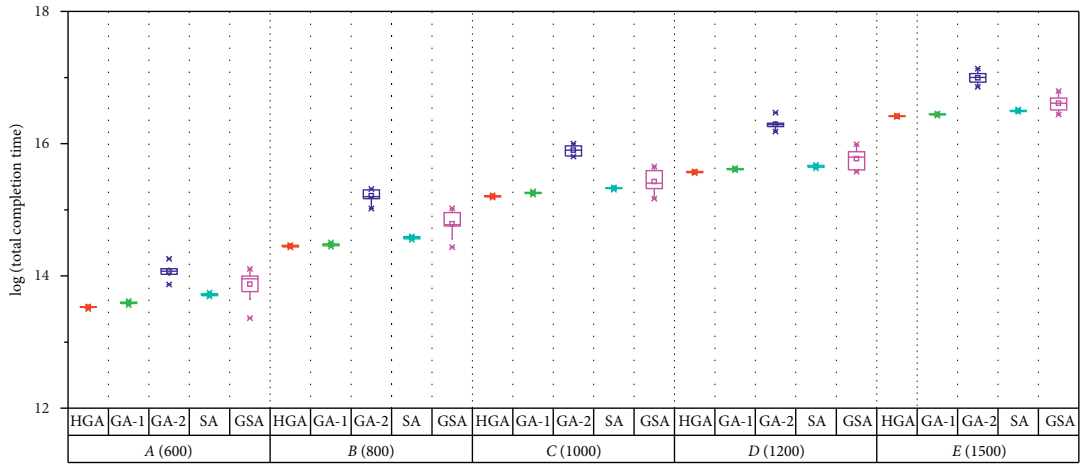

Figure 9: Range of variation for each algorithm in multiple times, different order numbers.

Table 8: Best scenario result of HGA & GSA in ten runs for different order numbers & structures.

Algorithms	Metric (sec)	A(60-12)	B(80-16)	C(100-20)	D(150-30)	E(200-40)	A(600)	B(800)	C(1000)	D(1200)	E(1500)
HGA	Obj	1.373	2.075	2.710	2.830	3.198	0.726	1.850	3.931	5.693	13.299
	Setup	7.490	10.133	12.758	19.665	23.722	5.879	8.158	9.680	10.951	14.068
	Run	12.960	13.820	14.823	15.117	18.285	5.724	12.924	24.995	50.832	92.351
	Batch	61	57	56	56	56	43	62	73	89	113
GSA	Obj	1.365	2.197	3.194	3.134	3.340	0.635	1.854	3.866	5.819	13.798
	Setup	27.106	35.139	43.844	52.147	59.230	23.261	29.020	36.728	40.633	52.298
	Run	19.794	20.134	24.281	36.563	40.289	25.415	30.438	39.788	55.859	68.674
	Batch	67	65	66	65	64	52	65	80	91	113

Table 9: Order information simulated dataset when batch capacity changes.

Total order number	Item types	Max quantity/item/order	Max item types/order	Number of workers
800	60	12	15	15

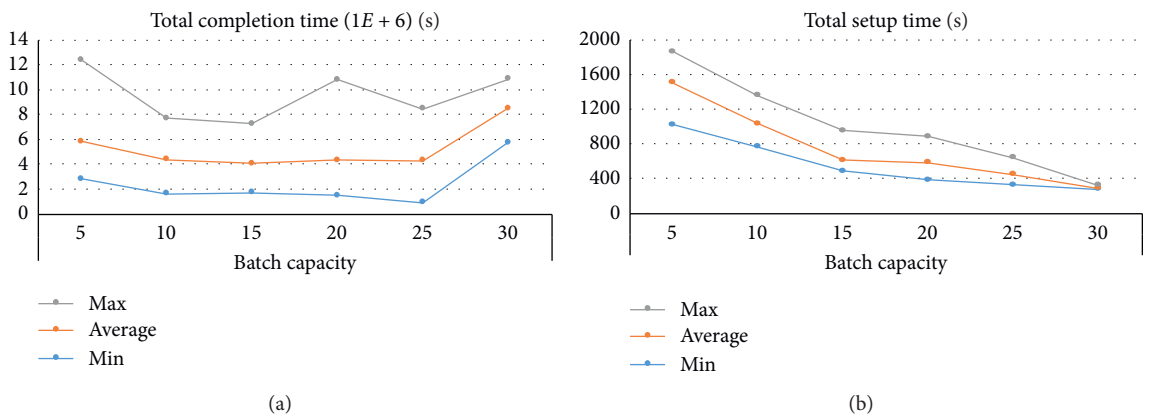

Figure 10: (a) Objective value changes when the batch capacity changes, (b) Total setup time changes when the batch capacity changes.

time for each sequenced order in the larger batch is increased accordingly, which leads to the rising of cumulative total completion time in Figure 10(a).

When the crossover operator is fixed to be 0.6, the trends of total completion time and setup time are shown in Figure 11 when the mutation operator has different values. Similarly, Figure 12 displays the corresponding trends for different values of the crossover operator when the mutation operator is set to 0.15. In Figure 11, it could be observed that when the mutation operator gets larger, the variation of total

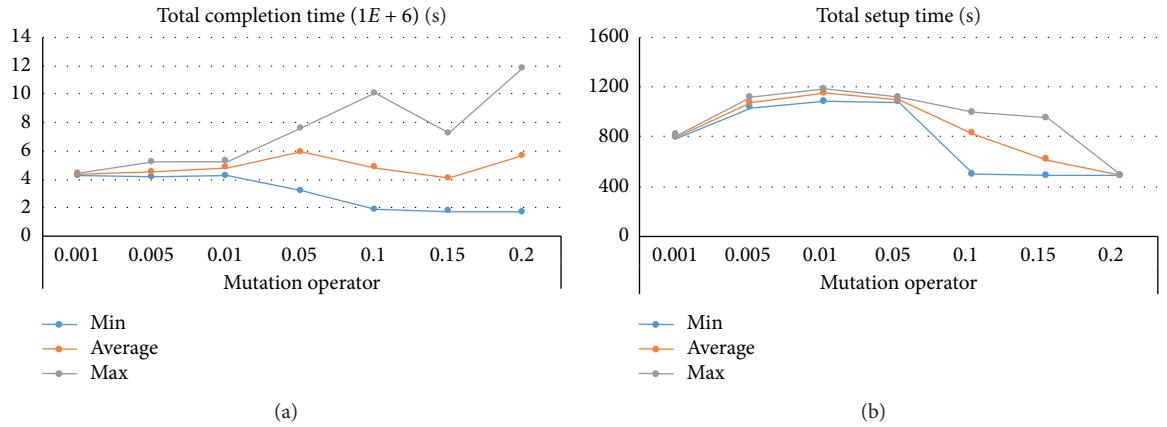

Figure 11: (a) Objective value changes when the mutation operator changes, (b) Total setup time changes when the mutation operator changes.

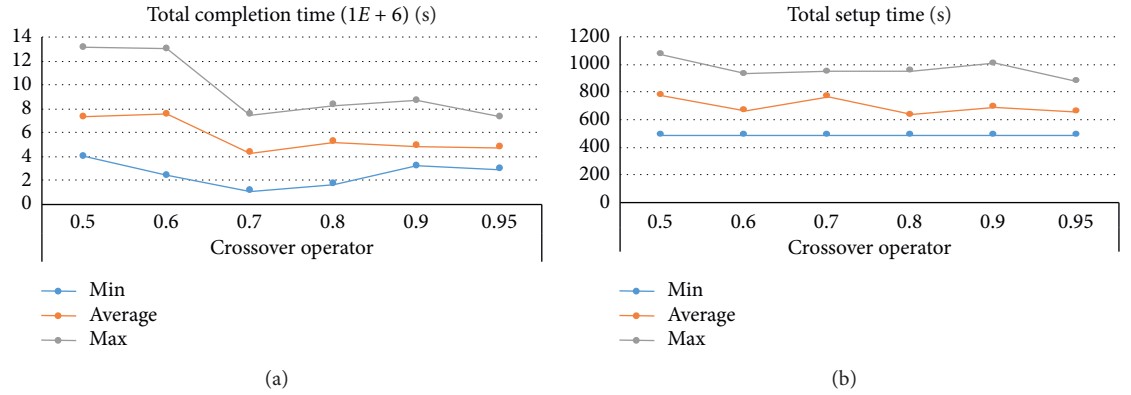

Figure 12: (a) Objective value changes when the crossover operator changes, (b) Total setup time changes when the crossover operator changes.

completion time increases while, on the other side, setup time reaches and stabilizes at best possible value it could find. The reason is that mutation operation swaps different batches (interbatches) and also shifts orders between batches (intrabatch), which could result in varied objective with a larger possibility. On the other hand, crossover operator interchanges selected crossing sections (interbatches), therefore the best possible objectives are lower when cross operator increases with fixed mutation operator in Figure 12, and the setup time, which is mostly caused by order numbers within batches (intrabatch), does not have many changes.

In summary, the proposed two-phase heuristic approach outperforms the other algorithms in terms of total completion time in conducted experiments up to now even though it may not be the fastest one in the chosen algorithms, which has proven its great effectiveness and scalability to batch orders firstly and then sequence the orders and batches in the second phase. Next, to understand how different fatigue habits of workers may affect the task allocation decisions, further analysis about the fatigue and other parameter settings are present in the following part.

4.3. Fatigue Analysis. As assumed and stated in Section 2, fatigue accumulation may follow the generalized logistic function. However, different workers will have different working habits. Some workers probably have a higher initial efficiency but higher deterioration rate, while others may be slower in the beginning and have smaller degradation. Thus, different fatigue situations are considered here and depicted in Figure 13.

In the experiment here, total workers are divided into two groups evenly with different fatigue curves. One group is assumed to have the reference curve (Figures 13(a) and 13(b) with 50s and 200s as initial and final unit picking time in about 8 hours, and the other group will have one curve among the four options. Order number and batch capacity are set to be 1500 and 15 in the proposed HGA algorithm. At the end of one day, the orders assigned for the two groups are compared in Table 10. As it can be seen, different groups are preferred when final efficiency or initial efficiency is fixed. For instance, the reference group is given fewer orders (723 out of 1500 orders) comparing with curve A1(70-180) even though it has higher picking efficiency at first. It is indicated that 40s difference of initial working efficiency may be

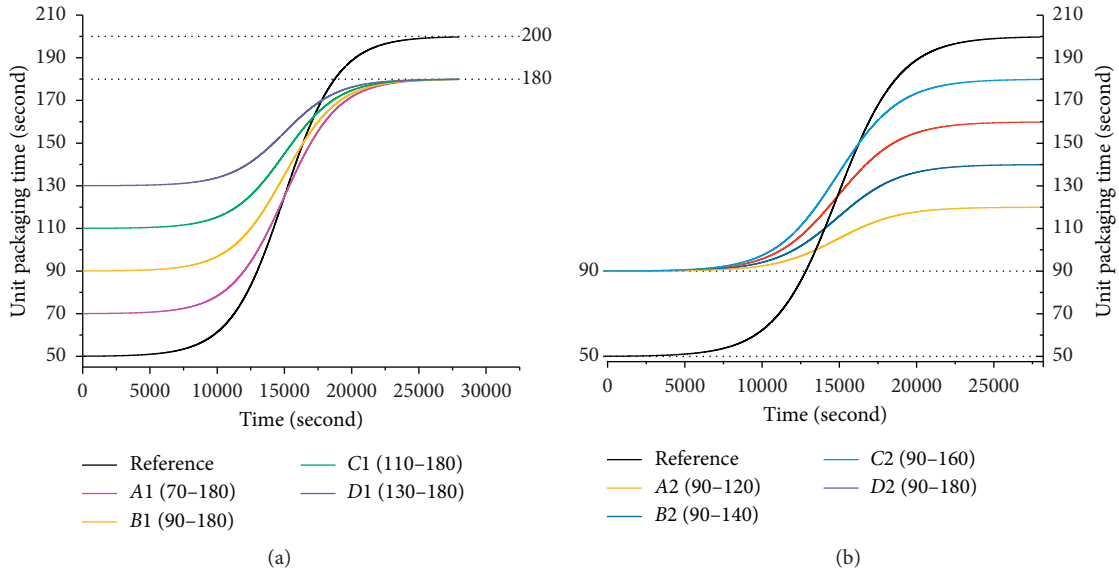

Figure 13: (a) Fatigue curves with different initial unit packaging times, (b) Fatigue curves with different final unit packaging times.

Table 10: Assigned orders for different groups with different fatigue curves.

Reference	*A*1(70-180)	*B*1(90-180)	*C*1(110-180)	*D*1(130-180)	*A*1(90-120)	*B*1(90-140)	*C*1(90-160)	*D*1(90-180)
(50-200)	723/777	737/763	754/746	763/737	696/804	714/785	722/778	735/765

Table 11: Fatigue parameter setting for picking lines.

Picking line ID	*1*	*2*	*3*	*4*	*5*
Initial unit picking time	5	6	7	5	10
Time point of stabilization	3227	3227	10000	10000	5300
Fatiguing rate	0.11	0.45	0.11	0.45	0.24
Final unit picking time	109	115	189	195	150
Picking line ID	*6*	*7*	*8*	*9*	*10*
Initial unit picking time	17	18	17	18	20
Time point of stabilization	3227	3227	10000	10000	8215
Fatiguing rate	0.00011	0.00045	0.00011	0.00045	0.00038
Final unit picking time	104	113	196	199	150
Picking line ID	*11*	*12*	*13*	*14*	*15*
Initial unit picking time	29	28	27	28	29
Time point of stabilization	3227	3227	10000	10000	10000
Fatiguing rate	0.00011	0.00045	0.00011	0.00045	0.0005
Final unit picking time	110	105	191	195	150

compensated with a slightly higher base level (20s gap). The comparison here could provide some managerial insights to decision makers in assigning orders to pickers according to current personnel fatigue effect and also provide some managerial suggestions for a distribution center in hiring and managing: evaluate the candidate's fatigue curve and make more incentives or interval reasonable rest policy before the time point reaches highest fatigue rate.

To compare the order assignment with and without considering the fatigue effect of workers, we split the 15 pickers into three groups in Table 11. Picking line ID 1-5 has a small initial unit picking time in range (1, 10). Picking line ID 6-10 has medium-scale initial unit picking time in range (11, 20). Picking line ID 11-15 has a large initial unit picking time in range (21, 30). Within each group, there are different scale values of other parameters. Order information is based on Table 9. The order assignment to each picking line is shown in Figures 14 and 15. It is seen that, without considering fatigue, a large amount of orders is shifted to picking line ID 1-5 because they have a relatively small initial unit picking time. In contrast, the assignment becomes more balanced if their fatiguing effect is considered because the worker's working efficiency may drop very quickly even though they had some advantage at the initial stage.

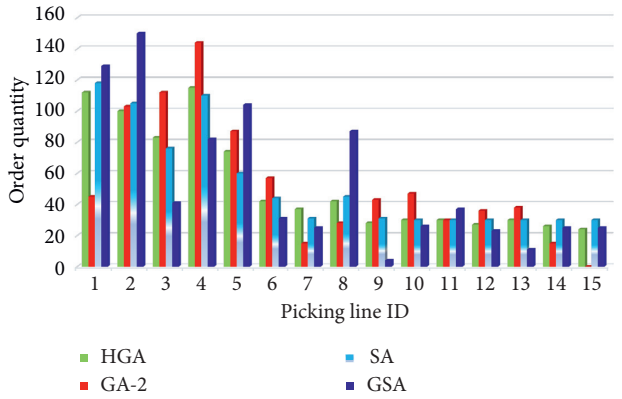

Figure 14: Orders assigned to picking lines without considering fatigue.

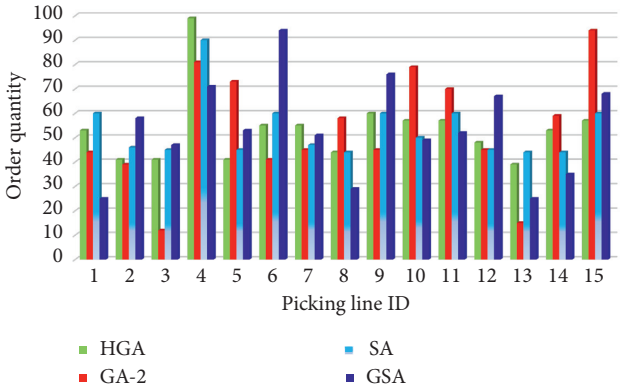

Figure 15: Orders assigned to picking lines considering fatigue.

5. Conclusions

In this research, order batching and scheduling problem has been modeled for picking lines in the context of e-commerce warehouses with the consideration of operators' nonlinear fatiguing effect. Due to the NP-hard solving complexity of this problem, an efficient two-phase heuristic method has been developed based on proposed new similarity-based batching rules and an improved genetic algorithm for order and batch scheduling after batching. Numerical instances have been generated from practical warehouse operations of an online vegetable retailer to testify and demonstrate the validity and scalability of the proposed solution approach. The experiments results have shown that the proposed algorithm outperforms the other comparable algorithms with less variance for different order structures and different levels of the total order number. In addition, possible preferences of task assignment have been analyzed for different fatigue curves, and it could shed some light on training and management (e.g., shift policy, incentive policy).

For future study, we will extend the decision model of batching and scheduling using the framework of stochastic programming or online optimization to include uncertainty of ordering in real time. On the other hand, to develop an efficient solution approach for an intelligent decision support system, data-driven order batching and scheduling algorithms based on statistical learning techniques are worth to be explored.

Data Availability

The data for different order structures and different order numbers are available in Github repository https://github.com/Xchunf/order-instances.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Annual Social Science Foundation Project of Shaanxi Province under Grant 2020R002, Natural Science Foundation Research Project of Shaanxi Province under Grant 2020JQ-281, Scientific Research Startup Foundation of Northwest A&F University under Grant 2452019167, and Basic Operating Expense under Humanities & Social Sciences Program of Northwest A&F University under Grant 2452020068.

References

[1] A. R. F. Pinto and M. S. Nagano, "An approach for the solution to order batching and sequencing in picking systems," *Production Engineering*, vol. 13, no. 3-4, pp. 325–341, 2019.

[2] X. Jiang, Y. Zhou, Y. Zhang, L. Sun, and X. Hu, "Order batching and sequencing problem under the pick-and-sort strategy in online supermarkets," *Procedia Computer Science*, vol. 126, pp. 1985–1993, 2018.

[3] B. Menéndez, M. Bustillo, E. G. Pardo, and A. Duarte, "General variable neighborhood search for the order batching and sequencing problem," *European Journal of Operational Research*, vol. 263, no. 1, pp. 82–93, 2017.

[4] X. Wang, J. Ruan, J. Ruan, and J. Ruan, "On-line order batching and sequencing problem with multiple pickers: a hybrid rule-based algorithm," *Applied Mathematical Modelling*, vol. 45, pp. 271–284, 2017.

[5] F. Chen, Y. Wei, and H. Wang, "A heuristic based batching and assigning method for online customer orders," *Flexible Services and Manufacturing Journal*, vol. 30, no. 4, pp. 640–685, 2018.

[6] M. Çelik and H. Süral, "Order picking in parallel-aisle warehouses with multiple blocks: complexity and a graph theory-based heuristic," *International Journal of Production Research*, vol. 57, no. 3, pp. 1–19, 2018.

[7] I. Žulj, C. H. Glock, E. H. Grosse, and M. Schneider, "Picker routing and storage-assignment strategies for precedence-constrained order picking," *Computers and Industrial Engineering*, vol. 123, pp. 338–347, 2018.

[8] D. Battini, C. H. Glock, E. H. Grosse, A. Persona, and F. Sgarbossa, "Human energy expenditure in order picking storage assignment: a bi-objective method," *Computers & Industrial Engineering*, vol. 94, pp. 147–157, 2016.

[9] E. H. Grosse, C. H. Glock, and W. P. Neumann, "Human factors in order picking: a content analysis of the literature," *International Journal of Production Research*, vol. 55, no. 5, pp. 1260–1276, 2017.

[10] C. H. Glock, E. H. Grosse, H. Abedinnia, and S. Emde, "An integrated model to improve ergonomic and economic performance in order picking by rotating pallets," *European Journal of Operational Research*, vol. 273, no. 2, pp. 516–534, 2019.

[11] R. M. Elbert, T. Franzke, C. H. Glock, and E. H. Grosse, "The effects of human behavior on the efficiency of routing policies in order picking: the case of route deviations," *Computers & Industrial Engineering*, vol. 111, pp. 537–551, 2017.

[12] E. H. Grosse, C. H. Glock, M. Y. Jaber, and W. P. Neumann, "Incorporating human factors in order picking planning models: framework and research opportunities," *International Journal of Production Research*, vol. 53, no. 3, pp. 695–717, 2015.

[13] Ç. Cergibozan and A. S. Tasan, "Order batching operations: an overview of classification, solution techniques, and future research," *Journal of Intelligent Manufacturing*, vol. 30, no. 1, pp. 335–349, 2019.

[14] F. Gzara, S. Elhedhli, U. Yildiz, and G. Baloch, "Data-driven modeling and optimization of the order consolidation problem in E-warehousing," *INFORMS Journal on Optimization*, vol. 2, no. 4, pp. 273–296, 2020.

[15] K. H. Leung, C. K. Lee, and K. L. Choy, "An integrated online pick-to-sort order batching approach for managing frequent arrivals of B2B e-commerce orders under both fixed and variable time-window batching," *Advanced Engineering Informatics*, vol. 45, pp. 1–16, 2020.

[16] P. Yang, Z. Zhao, and H. Guo, "Order batch picking optimization under different storage scenarios for e-commerce warehouses," *Transportation Research Part E: Logistics and Transportation Review*, vol. 136, pp. 101897–101918, 2020.

[17] X. Qi and J. Yuan, "A further study on two-agent scheduling on an unbounded serial-batch machine with batch delivery cost," *Computers & Industrial Engineering*, vol. 111, pp. 458–462, 2017.

[18] M. A. Aloulou, A. Bouzaiene, N. Dridi, and D. Vanderpooten, "A bicriteria two-machine flow-shop serial-batching scheduling problem with bounded batch size," *Journal of Scheduling*, vol. 17, no. 1, pp. 17–29, 2014.

[19] C. He, H. Lin, and Y. Lin, "Bounded serial-batching scheduling for minimizing maximum lateness and makespan," *Discrete Optimization*, vol. 16, pp. 70–75, 2015.

[20] B. Q. Fan, T. C. E. Cheng, S. S. Li, and Q. Feng, "Bounded parallel-batching scheduling with two competing agents," *Journal of Scheduling*, vol. 16, no. 3, pp. 261–271, 2013.

[21] A. Condotta, S. Knust, and N. V. Shakhlevich, "Parallel batch scheduling of equal-length jobs with release and due dates," *Journal of Scheduling*, vol. 13, no. 5, pp. 463–477, 2010.

[22] S. Li and J. Yuan, "Unbounded parallel-batching scheduling with two competitive agents," *Journal of Scheduling*, vol. 15, no. 5, pp. 629–640, 2012.

[23] S.-S. Li and R.-X. Chen, "Single-machine parallel-batching scheduling with family jobs to minimize weighted number of tardy jobs," *Computers & Industrial Engineering*, vol. 73, pp. 5–10, 2014.

[24] M. Jin, X. Liu, and W. Luo, "Single-machine parallel-batch scheduling with nonidentical job sizes and rejection," *Mathematics*, vol. 8, pp. 1–8, 2020.

[25] J. Pei, B. Cheng, X. Liu, P. M. Pardalos, and M. Kong, "Single-machine and parallel-machine serial-batching scheduling problems with position-based learning effect and linear setup time," *Annals of Operations Research*, vol. 272, no. 1-2, pp. 217–241, 2019.

[26] J. Cheng, F. Chu, M. Liu, P. Wu, and W. Xia, "Bi-criteria single-machine batch scheduling with machine on/off switching under time-of-use tariffs," *Computers & Industrial Engineering*, vol. 112, pp. 721–734, 2017.

[27] Z. Li, H. Chen, R. Xu, and X. Li, "Earliness-tardiness minimization on scheduling a batch processing machine with non-identical job sizes," *Computers & Industrial Engineering*, vol. 87, pp. 590–599, 2015.

[28] A. Bilyk, L. Mönch, and C. Almeder, "Scheduling jobs with ready times and precedence constraints on parallel batch machines using metaheuristics," *Computers & Industrial Engineering*, vol. 78, pp. 175–185, 2014.

[29] J.-Q. Li, M.-X. Song, L. Wang et al., "Hybrid artificial bee colony algorithm for a parallel batching distributed flow-shop problem with deteriorating jobs," *IEEE Transactions on Cybernetics*, vol. 50, no. 6, pp. 2425–2439, 2020.

[30] M. Kong, X. Liu, J. Pei, Z. Zhou, and P. M. Pardalos, "Parallel-batching scheduling of deteriorating jobs with non-identical sizes and rejection on a single machine," *Optimization Letters*, vol. 14, no. 4, pp. 857–871, 2020.

[31] Z. Zhao, S. Liu, M. Zhou, D. You, and X. Guo, "Heuristic scheduling of batch production processes based on petri nets and iterated greedy algorithms," *IEEE Transactions on Automation Science and Engineering*, pp. 1–11, 2020.

[32] J. Pei, Q. Song, B. Liao, X. Liu, and P. M. Pardalos, "Parallel-machine serial-batching scheduling with release times under the effects of position-dependent learning and time-dependent deterioration," *Annals of Operations Research*, vol. 298, no. 1-2, pp. 407–444, 2020.

[33] L. M. Liao, "Tabu search heuristic for two-machine flowshop with batch processing machines," *Computers & Industrial Engineering*, vol. 60, no. 3, pp. 426–432, 2011.

[34] O. Shahvari and R. Logendran, "Hybrid flow shop batching and scheduling with a bi-criteria objective," *International Journal of Production Economics*, vol. 179, pp. 239–258, 2016.

[35] C. Hertrich, C. Weiß, H. Ackermann, S. Heydrich, and S. O. Krumke, "Scheduling a proportionate flow shop of batching machines," *Journal of Scheduling*, vol. 23, no. 5, pp. 575–593, 2020.

[36] W.-C. Lee, C.-C. Wu, and Y.-H. Chung, "Scheduling deteriorating jobs on a single machine with release times," *Computers & Industrial Engineering*, vol. 54, no. 3, pp. 441–452, 2008.

[37] W.-C. Lee, J.-B. Lin, and Y.-R. Shiau, "Deteriorating job scheduling to minimize the number of late jobs with setup times," *Computers & Industrial Engineering*, vol. 61, no. 3, pp. 782–787, 2011.

[38] G. Moslehi and A. Jafari, "Minimizing the number of tardy jobs under piecewise-linear deterioration," *Computers & Industrial Engineering*, vol. 59, no. 4, pp. 573–584, 2010.

[39] S. Li and J. Yuan, "Parallel-machine scheduling with deteriorating jobs and rejection," *Theoretical Computer Science*, vol. 411, no. 40-42, pp. 3642–3650, 2010.

[40] A. J. Ruiz-Torres, G. Paletta, and E. Pérez, "Parallel machine scheduling to minimize the makespan with sequence dependent deteriorating effects," *Computers & Operations Research*, vol. 40, no. 8, pp. 2051–2061, 2013.

[41] A.-A. Jafari, H. Khademi-Zare, M. M. Lotfi, and R. Tavakkoli-Moghaddam, "A note on "On three-machine flow shop scheduling with deteriorating jobs"" *International Journal of Production Economics*, vol. 191, pp. 250–252, 2017.

[42] M. Cheng, P. R. Tadikamalla, J. Shang, and B. Zhang, "Two-machine flow shop scheduling with deteriorating jobs: minimizing the weighted sum of makespan and total

completion time," *Journal of the Operational Research Society*, vol. 66, no. 5, pp. 709–719, 2015.

[43] S.-J. Yang and D.-L. Yang, "Minimizing the makespan on single-machine scheduling with aging effect and variable maintenance activities," *Omega*, vol. 38, no. 6, pp. 528–533, 2010.

[44] C.-C. Wu and W.-C. Lee, "Single-machine group-scheduling problems with deteriorating setup times and job-processing times," *International Journal of Production Economics*, vol. 115, no. 1, pp. 128–133, 2008.

[45] J.-B. Wang, X. Huang, Y.-B. Wu, and P. Ji, "Group scheduling with independent setup times, ready times, and deteriorating job processing times," *The International Journal of Advanced Manufacturing Technology*, vol. 60, no. 5-8, pp. 643–649, 2012.

[46] J. Pei, X. Liu, P. M. Pardalos, W. Fan, and S. Yang, "Single machine serial-batching scheduling with independent setup time and deteriorating job processing times," *Optimization Letters*, vol. 9, no. 1, pp. 91–104, 2014.

[47] J. Pei, P. M. Pardalos, X. Liu, W. Fan, and S. Yang, "Serial batching scheduling of deteriorating jobs in a two-stage supply chain to minimize the makespan," *European Journal of Operational Research*, vol. 244, no. 1, pp. 13–25, 2015.

[48] J. Pei, X. Liu, P. M. Pardalos, A. Migdalas, and S. Yang, "Serial-batching scheduling with time-dependent setup time and effects of deterioration and learning on a single-machine," *Journal of Global Optimization*, vol. 67, no. 1-2, pp. 251–262, 2017c.

[49] J. Pei, X. Liu, W. Fan, P. M. Pardalos, and S. Lu, "A hybrid BA-VNS algorithm for coordinated serial-batching scheduling with deteriorating jobs, financial budget, and resource constraint in multiple manufacturers," *Omega*, vol. 82, pp. 55–69, 2019b.

[50] Y. Gao, J. Yuan, C. T. Ng, and T. C. E. Cheng, "A further study on two-agent parallel-batch scheduling with release dates and deteriorating jobs to minimize the makespan," *European Journal of Operational Research*, vol. 273, no. 1, pp. 74–81, 2019.

[51] Y.-B. Woo and B. S. Kim, "Matheuristic approaches for parallel machine scheduling problem with time-dependent deterioration and multiple rate-modifying activities," *Computers & Operations Research*, vol. 95, pp. 97–112, 2018.

[52] J. Ding, L. Shen, Z. Lü, and B. Peng, "Parallel machine scheduling with completion-time-based criteria and sequence-dependent deterioration," *Computers & Operations Research*, vol. 103, pp. 35–45, 2019.

[53] A. S. Martinez, R. S. González, and C. A. S. Terçariol, "Continuous growth models in terms of generalized logarithm and exponential functions," *Physica A: Statistical Mechanics and Its Applications*, vol. 387, no. 23, pp. 5679–5687, 2008.

[54] Z. S. Givi, M. Y. Jaber, and W. P. Neumann, "Modelling worker reliability with learning and fatigue," *Applied Mathematical Modelling*, vol. 39, no. 17, pp. 5186–5199, 2015.

[55] S. Nagashima, Y. Suwazono, Y. Okubo et al., "Working hours and mental and physical fatigue in Japanese workers," *Occupational Medicine*, vol. 57, no. 6, pp. 449–452, 2007.

[56] O. Shahvari and R. Logendran, "An Enhanced tabu search algorithm to minimize a bi-criteria objective in batching and scheduling problems on unrelated-parallel machines with desired lower bounds on batch sizes," *Computers & Operations Research*, vol. 77, pp. 154–176, 2017.

[57] A. M. Fathollahi-Fard, M. Hajiaghaei-Keshteli, and S. Mirjalili, "A set of efficient heuristics for a home healthcare problem," *Neural Computing and Applications*, vol. 32, 2019.

[58] A. M. Fathollahi Fard and M. Hajiaghaei-Keshteli, "A bi-objective partial interdiction problem considering different defensive systems with capacity expansion of facilities under imminent attacks," *Applied Soft Computing*, vol. 68, pp. 343–359, 2018.

[59] A. M. Fathollahi-Fard, M. Hajiaghaei-Keshteli, and R. Tavakkoli-Moghaddam, "A bi-objective green home health care routing problem," *Journal of Cleaner Production*, vol. 200, pp. 423–443, 2018.

[60] A. M. Fathollahi-Fard, M. Hajiaghaei-Keshteli, and S. Mirjalili, "Hybrid optimizers to solve a tri-level programming model for a tire closed-loop supply chain network design problem," *Applied Soft Computing*, vol. 70, pp. 701–722, 2018.

[61] A. M. Fathollahi-Fard, M. Hajiaghaei-Keshteli, and S. Mirjalili, "Multi-objective stochastic closed-loop supply chain network design with social considerations," *Applied Soft Computing*, vol. 71, pp. 505–525, 2018.

[62] A. M. Fathollahi-Fard and M. Hajiaghaei-Keshteli, "A stochastic multi-objective model for a closed-loop supply chain with environmental considerations," *Applied Soft Computing*, vol. 69, pp. 232–249, 2018.

[63] M. Hajiaghaei-Keshteli and A. M. Fathollahi-Fard, "A set of efficient heuristics and metaheuristics to solve a two-stage stochastic bi-level decision-making model for the distribution network problem," *Computers & Industrial Engineering*, vol. 123, pp. 378–395, 2018.

[64] N. Sahebjamnia, A. M. Fathollahi-Fard, and M. Hajiaghaei-Keshteli, "Sustainable tire closed-loop supply chain network design: hybrid metaheuristic algorithms for large-scale networks," *Journal of Cleaner Production*, vol. 196, pp. 273–296, 2018.

[65] J. Li, R. Huang, and J. B. Dai, "Joint optimisation of order batching and picker routing in the online retailer's warehouse in China," *International Journal of Production Research*, vol. 55, no. 2, pp. 447–461, 2016.

[66] C.-Y. Tsai, J. J. H. Liou, and T.-M. Huang, "Using a multiple-GA method to solve the batch picking problem: considering travel distance and order due time," *International Journal of Production Research*, vol. 46, no. 22, pp. 6533–6555, 2008.

[67] T.-L. Chen, C.-Y. Cheng, Y.-Y. Chen, and L.-K. Chan, "An efficient hybrid algorithm for integrated order batching, sequencing and routing problem," *International Journal of Production Economics*, vol. 159, pp. 158–167, 2015.

[68] C. M. Joo and B. S. Kim, "Hybrid genetic algorithms with dispatching rules for unrelated parallel machine scheduling with setup time and production availability," *Computers & Industrial Engineering*, vol. 85, pp. 102–109, 2015.

[69] S. Koch and G. Wäscher, "A grouping genetic algorithm for the order batching problem in distribution warehouses," *Journal of Business Economics*, vol. 86, no. 1-2, pp. 131–153, 2016.

[70] P. S. Sundararaghavan and A. S. Kunnathur, "Single machine scheduling with start time dependent processing times: some solvable cases," *European Journal of Operational Research*, vol. 78, no. 3, pp. 394–403, 1994.

[71] The MathWorks, Inc., MATLAB and Statistics Toolbox, The MathWorks, Inc., Natick, MA, USA, 2016.

[72] P. Damodaran and M. C. Vélez-Gallego, "A simulated annealing algorithm to minimize makespan of parallel batch processing machines with unequal job ready times," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1451–1458, 2012.