Journal of Intelligent Manufacturing
https://doi.org/10.1007/s10845-019-01524-6

Production scheduling problem in a factory of automobile component primer painting

Ivan Kristianto Singgih[1] · Onyu Yu[1] · Byung-In Kim[1] · Jeongin Koo[2] · Seungdoe Lee[3]

Received: 20 August 2019 / Accepted: 10 December 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract
In a factory of automobile component primer painting, various automobile parts are attached to overhead hangers in a conveyor line and undergo a series of coating processes. Thereafter, the components are wrapped at a packaging station. The packaging process should be fully balanced by an appropriate sequence of components to prevent the bottleneck effect because each component requires different packaging times and materials. An overhead hanger has a capacity limit and can hold varying numbers of components depending on the component type. Capacity loss can occur if the hanger capacity is not fully utilized. To increase hanger utilization, companies sometimes mix two or more component types on the same hangers, and these hangers are called mixed hangers. However, mixed hangers generally cause heavy workload because different items require additional setup times during hanging and packing processes. Hence, having many mixed hangers is not recommended. A good production schedule requires a small number of mixed hangers and maximizes hanger utilization and packaging workload balance. We show that the scheduling problem is NP-hard and develop a mathematical programming model and efficient solution approaches for the problem. When applying the methods to solve real problems, we also use an initial solution-generating method that minimizes the mixing cost, set a rule for hanging the items on hangers considering eligibility constraint, and decrease the size of tabu list in proportion to the remaining computational time for assuring intensification in the final iterations of the search. Experimental results demonstrate the effectiveness of the proposed approaches.

Keywords Flow shop scheduling · Automobile component primer painting · Mixed-integer programming · Heuristic algorithms

Introduction

In recent years, an increasing number of studies have discussed specific scheduling cases (Fuchigami and Rangel 2018). Each case study has its unique characteristics that must be considered for developing appropriate scheduling methods (Baykasoğlu and Ozsoydan 2018; Dallasega et al. 2019; Mohammadi et al. 2020). In addition to the specific requirements of each system, many studies have focused on addressing sustainability (related to economic, environmental, and social aspects) and stochastic trends (Yin et al. 2015; Henao et al. 2019; Johansen 2019).

We study a scheduling problem in a factory of automobile component primer painting, in which thousands of automobile parts undergo anticorrosion electrodeposition coating. The company uses a continuous hanger line. Component items are hung on hangers and undergo the painting (coating) steps. After coating, the component items are wrapped at a packaging station.

All component items undergo the same painting processes in a pre-determined order, and the conveyor line moves at a constant speed. Thus, the line productivity depends on the hanger occupancy rate. An overhead hanger has a capacity limit and can hold different numbers of items depending on the item type. Low hanger occupancy rate and capacity loss will arise if the hanger capacity is not fully utilized. To increase the hanger occupancy rate or utilization, the company under study sometimes mixes two or more types

✉ Byung-In Kim
bkim@postech.ac.kr

[1] Department of Industrial and Management Engineering, Pohang University of Science and Technology, Pohang, South Korea

[2] Research Institute of Sustainable Manufacturing System, Korea Institute of Industrial Technology, Cheonan, South Korea

[3] Hankuk Metal Co., LTD, Gyeongju, South Korea

Published online: 02 January 2020

of item on the same hangers, and these are called mixed hangers. Placing mixed items on hangers increases the hanging workload for the coating process and additional setup time for the packaging process. If mixing is necessary, then items with similar characteristics must be hung on the same hangers to minimize the workload and packaging setup time. The conflict between maximizing the hanger occupancy rate and minimizing the number of mixed hangers and the dissimilarity of mixed items on the same hangers should be considered, and the component items of an order must be hung on consecutive hangers, such that the items of an order are processed and packaged together.

The workload in the packaging process must also be fully balanced during working hours because the company employs different levels of packaging workforce depending on the time of day. Different items require different workload levels. At the packaging step, each item is manually packed in accordance with its specifications. Packing specifications can be categorized into three groups depending on the workload. The packaging workload for several export items is considerably higher than that for others. Given the continuous movement of the hanger line, cumulative packaging workloads within a certain period or a certain number of consecutive hangers should be fully balanced to avoid heavy fatigue on workers at the packaging step.

A mixed-integer programming (MIP) model for the scheduling problem is developed. However, we cannot use the model for real-size problems because it requires extensive computation time due to its NP-hardness. Therefore, we develop a 2-Opt improvement algorithm and a tabu search metaheuristic algorithm. Computational results show that the proposed algorithms can effectively solve the problem.

The remainder of this paper is organized as follows. "Literature review" section reviews the related literature. "Problem description" section describes the problem in detail. "Mathematical formulation" section presents our mathematical model and its NP-hardness. "Solution approaches and computational results" section provides the solution approaches and their computational results. "Conclusions" section elaborates the conclusions.

Literature review

The problem studied in this work can be considered a permutation batch flow shop problem. Ruiz and Maroto (2005) explained that in a permutation flow shop problem, jobs are scheduled to be processed in a set of machines in the same order, and the jobs are processed in the same sequence in all machines. In our work, items hung on a hanger are considered a batch. All batches are processed in the same sequence in all machines because the hangers move at a constant speed through the coating line. Reviews of the flow shop problem were presented in Ruiz and Maroto (2005), Hejazi and Saghafian (2005), and Behnamian and Fatemi Ghomi (2016).

The batch scheduling problem has been studied extensively. A classification of batch scheduling based on batch size, batch processing time, and other factors was provided by Mendez et al. (2006). Potts and Kovalyov (2000) and Sun et al. (2011) reviewed this subject. The batch scheduling problem in a flow shop environment with two stages was discussed by Tang and Liu (2009a, b), Behnamian et al. (2012), Wang et al. (2012), and Liu et al. (2018). Batching problems in systems with more than two stages were also studied by Salmasi et al. (2011), Damodaran et al. (2013), Li et al. (2015), and Matin et al. (2017). Studies on flow shop batch scheduling with more than one item type, which is similar to the problem discussed in the present work, are also available (Sawik 2002; Kim et al. 2009; Masmoudi et al. 2016). Although these studies investigated the flow shop batch scheduling problem with multiple item types, waiting time was allowed between machines. By contrast, in our problem, no waiting time exists between machines, which is an important characteristic of a well-known, no-wait flow shop scheduling problem or flow shop scheduling problem with no in-process waiting (Selen and Hott 1986).

Product processing in chemical baths, in which each product must be immersed in baths in the same order, is similar to the problem in the present study. The problem is classified as a no-wait flow shop scheduling with batching considerations (Oulamara et al. 2005). Lin and Cheng (2001) studied a problem in steel and plastic production, in which jobs were processed as batches sequentially in two machines. Oulamara (2007) proposed an algorithm to solve a no-wait flow shop batch scheduling problem for minimizing makespan. Zhou et al. (2016) considered a problem of minimizing makespan with parallel and serial batch processing machines, non-identical job sizes, and unequal ready times. Stefansdottir et al. (2017) addressed a no-wait flow shop batch scheduling problem with consideration of setups and cleaning processes. The studied problems above only considered two stages of machines, whereas our problem includes more than two stages of machines.

Some studies addressed the no-wait flow shop scheduling problem with several machines, but considered job scheduling instead of the batches (Tasgetiren et al. 2011; Sapkal and Laha 2013; Allahverdi and Aydilek 2013, 2014; Samarghandi and Behroozi 2017; Koulamas and Panwalkar 2018). A survey of research on no-wait flow shop batching scheduling problems was conducted by Oulamara (2012) and Allahverdi (2016). These studies examined various no-wait flow shop batch scheduling problems; however, the characteristics of scheduling problems for product processing in chemical baths, which are important in the present study, were not considered.

We review the literature related to this study. The closest references are Lin and Cheng (2001), Oulamara (2007), Zhou et al. (2016), and Stefansdottir et al. (2017). These studies only consider two machines, whereas ours consider more than two machines. In addition, various characteristics make the present problem unique; these characteristics include item-hanger eligibility, different item sizes and packaging workloads, effect of a generated schedule on the packaging stage, continuous hanging requirement of orders, allowance for mixed items, and multiple conflicting objectives. Such characteristics will be explained in detail in the next section. To our knowledge, no study has been conducted on the flow shop batch scheduling problem considered in this work.

Problem description

Figure 1 shows an aerial view of the factory of automobile component painting. The numbers in the figure indicate the coating steps that the items move through after they are attached to hangers. Among the steps, five steps (labeled 4, 8, 10, 12, and 14) use coating pools in which the items are sunk. The lengths of the five liquid coating pools (labeled 4, 8, 10, 12, and 14 in Fig. 1) are accurately designed to obtain the desired thickness of the electrodeposition layer of items. The items are coated appropriately based on the conveyor's constant speed and pool lengths. A fixed number of hangers are hung to the conveyor rail line and pass through the painting areas. Figure 2 shows items on hangers attached to the conveyor line. After the coating process, the workers at the packaging station carry the items and pack them in accordance with their specifications.

The daily working period of the company is from 9 a.m. to 6 p.m., and the scheduling horizon is the same. At the end of work in a day, the conveyor rail is stopped, and the items already hung on the hanger are processed on the following day. Therefore, the hangers located in the five pools should be empty (that is, no items are hung on the hangers in the pools) or only contain eligible item types when the line stops to guarantee painting quality. Otherwise, items that stay too long in the pools cannot meet the quality requirements. In other words, restrictions should be observed for items on these hangers at the end of a working day. In addition, the number of packaging workers decreases during lunch and break times. Hence, delivering hard packing items to the packing area during these times should be avoided. With these constraints considered, the eligibility of hangers for items is determined. Given that the hangers circulate more than three times in a day, each physical hanger has multiple indices in our model. A total of 181 physical hangers are in the factory; however, the total number of logical hangers processed per day is set to 600.

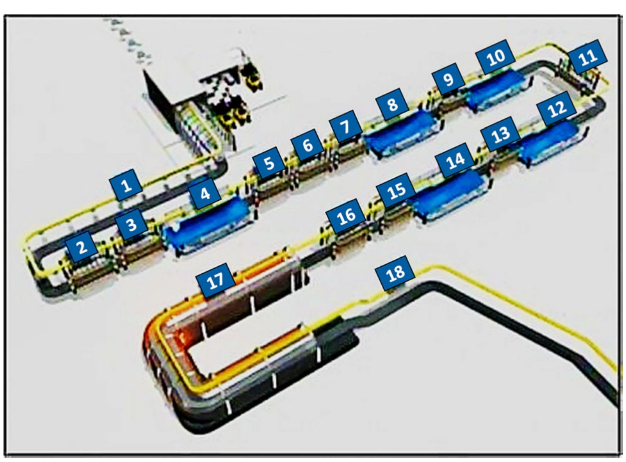

Fig. 1 Aerial view of the painting factory

(a) Large components attached to hangers

(b) Small components prepared to be hung

Fig. 2 Component items and hangers

The input data, objectives, and output form of the scheduling problem will be explained in the following subsections.

Input data

Items and hangers are given as input data. Several item types are considered, and each item has its own characteristics and unique item code, as shown in Table 1. The input amount is the number of items to be scheduled. Occupied hanger capacity pertains to the hanger area required by an item; if the occupied capacity is 1/5, then a hanger can contain up to 5 units of this item. Each item should be hung on one or two hangers. Long items, such as I1, I4, and I5, require two hangers, whereas a short item can be hung on a single hanger. The column "Required number of hangers per item" in Table 1 shows the information. If the number of hangers required to hang an item is two, then two consecutive hangers should be assigned to hang the item. If the required number is two, then the occupied hanger capacity also corresponds to two hangers. We use I1 in Table 1 as an example. Given that the occupied hanger capacity is 1/10 and the required number of hangers per item is 2, 2 hangers together can contain 10 items. The column "Packing type" in Table 1 is used to identify packing specifications.

The eligibility of logical hangers can be obtained by considering the physical hanger locations on the conveyor rail, the rail speed, the hanger stopping positions at the end of the day, and the workforce schedule. Morning and afternoon break times are from 10:00 a.m. to 10:10 a.m. and from 3:00 p.m. to 3:10 p.m., respectively, and lunch break is from 12:00 p.m. to 1:00 p.m. Given that 600 hangers are used in a day, an example of their eligibility data is shown in Table 2. Eligibility codes 1, 2, and 3 indicate that the hanger can be attached with all items, items with a packing workload equal to one, and no items at all, respectively. A point of consideration used in determining hanger eligibility is that certain items are not allowed to be immersed in the pools for more than the prescribed time for electrodeposition overnight.

Table 2 Hanger eligibility

Hanger	Eligibility code
1	3
2	3
3	3
…	…
250	2
251	2
…	…
515	1
516	1
…	…
599	2
600	2

Different items may be hung on the same hanger. However, mixed items on hangers affect the workload in the packaging area. Items that have the same packing specifications and belong to the same sub-assembly can be packed without an extra setup; otherwise, an additional setup is required. Moreover, mixed items on hangers increase the hanging workload for the coating process. Mixing costs (penalty) between pairs of items are used to identify the degree of differences between items. Table 3 presents these data. A high mixing penalty is imposed when different items are hung on the same hanger and cause difficulties in the packaging stage; this penalty contributes to the total mixing

Table 1 Input data of item order

Item code	Input amount	Occupied hanger capacity	Required number of hangers per item	Item name	Item type code	Sub-assembly name	Car model	Packing type	Packing load
I1	60	1/10	2	Large Etc	DH	Member Assy-Rear Floor Siderh	3A0	NC	1
I2	111	1/12	1	Fender	EH	Panel-Fend-erlh	2V0	BW	3
I3	34	1/14	1	Back Pnl	DH	Panel Assy-Back	3S0	FL	2
I4	109	1/6	2	Qtr	DH	Panel Assy-Quarter Outerrh	2V0	AF	2
I5	121	1/5	2	Qtr	DH	Panel Assy-Quarter Outerrh	2V0	AF	2

Table 3 Item mixing costs

Mixing penalty	Same characteristics[a]				
	Item type code	Packing type	Item name	Sub-assembly name	Car model
0	1	1	1	1	1
0	1	1	1	1	0
2	1	1	0	0	1
2	1	1	0	0	0
3	1	0	1	0	1
3	1	0	1	0	0
4	1	0	0	0	1
4	1	0	0	0	0
5	0	1	1	1	1
5	0	1	1	1	0
6	0	1	0	0	1
6	0	1	0	0	0
7	0	0	1	0	1
7	0	0	1	0	0
1000	0	0	0	0	1
1000	0	0	0	0	0
10,000	1	1	1	0	1
10,000	1	1	1	0	0
10,000	1	0	1	1	1
10,000	1	0	1	1	0
10,000	0	1	1	0	1
10,000	0	1	1	0	0
100,000	Other cases				

[a] 1 = same; 0 = different

cost, as presented in "Objectives" section. The mixing penalty is 0 when all item characteristics are equal and is 10,000 when two items have the same characteristics, except for the sub-assembly names. An example of the calculation for the mixing cost is as follows. Supposing that four hangers exist, items A, B, and C are hung on hangers 1 and 2, items B and C are hung on hanger 3, and item B is hung on hanger 4, and the mixing penalty for items A and B, A and C, and B and C are 1000, 7, and 4, respectively. Then, the total mixing cost for hangers 1–4 is (1000 + 7 + 4), (1000 + 7 + 4), 4, and 0, respectively.

Objectives

The following objectives are considered in the scheduling problem: minimization of the total capacity loss of the hangers, total penalty for partially hung order items, total mixing cost, and maximum workload of packing workers on consecutive numbers of hangers. Total capacity loss is the remaining capacity on the hangers after assigning the items.

Order items may be hung partially. In other words, not all of the required numbers of an item order are hung on the hangers. Several items are left and handled in the following days. The remaining ones delay the packaging process because the order items should be packaged together and shipped to customers. Items are not considered to be partially hung when no or all items are hung on the hangers. For example, suppose that an item order consists of 100 units. When the number of scheduled units is between 1 and 99, the item order is considered partially hung. Imposing a penalty encourages generating schedules in which same-order items are processed on the same day to avoid delay in the packing process.

Given that the input number of an item is not an exact multiple of the occupied hanger capacity of the item, a few of these items should be mixed with other item types to minimize capacity loss. However, we should consider the packing workload caused by the mixed items. To ease the hanging and packaging processes, the mixing cost must be minimized by prioritizing hanging items with similar characteristics on the same hangers.

The packing workload must also be balanced throughout the work time by minimizing the total packing workload for each number of consecutive hangers. Each item is categorized into three packing workload levels (heavy, medium, and light), as shown in Table 1, in accordance with its packing specification. If items with heavy packing workload are assigned continuously, then the packing workload will greatly increase and cause fatigue to packing operators. The total packing workloads are measured for each g consecutive hangers. The solution quality is evaluated by obtaining the largest among all g consecutive total packing workloads (defined as $MaxP$). For example, we suppose that the total packing workloads on hangers 1–6 are 10, 30, 30, 10, 50, and 60, respectively, and g equals 3. The total packing workload of the first three hangers (hangers 1–3) equals 70. Similarly, the total packing workloads for hangers 2–4, 3–5, and 4–6 are 70, 90, and 120, respectively. Thus, $MaxP$ is 120. In this study, the value of g is set to 10.

Output

Table 4 presents the desired scheduling output format. The scheduling result contains information related to the number of items hung and their initial and final hanger numbers. We calculate the time for hanging each type of item using the hanger numbers.

Mathematical formulation

An MIP formulation is developed for the problem. We let i and j be the item indices (1, …, I) and h be the hanger index (1, …, H). H is 600 because a hanger on the conveyor rail passes the interval length between two consecutive hangers in 54 s, and the total work time equals 32,400 s (9 h). The parameters are as follows:

- a_i input number of items i
- b_{ij} penalty value given when items i and j are hung on the same hanger
- c_i occupied hanger capacity used by item i (0–1, real number)
- d_i required number of hangers per item i
- e_{ih} 1, if item i is allowed to be hung on hanger h while considering the hanger stopping position at the end of the day; 0, otherwise
- g number of consecutive hangers from which the workload is measured and the $MaxP$ value is calculated
- k_1 weight coefficient of the total capacity loss
- k_2 weight coefficient of the partially scheduled items
- k_3 weight coefficient of the total mixing value
- k_4 weight coefficient of the maximum workload of consecutive g hangers
- l_i packing workload level of item i

The decision variables are as follows:

- $MaxP$ maximum packing workload among g consecutive hangers
- r_i final hanger number that has item i
- s_i initial hanger number that has item i
- t_i 1, if the number of items i hung on the hangers is more than 0 and less than the required amount; 0, otherwise
- u_{ih} 1, if item i (with $d_i = 2$) is hung on h and h is the initial hanger for x_{ih} items; 0, otherwise
- v_{ijh} 1, if items i and j are hung together on hanger h; 0, otherwise
- w_i 1, if no item i is hung on any hangers; 0, otherwise
- x_{ih} number of items i on hanger h
- y_{ih} 1, if item i is hung on hanger h; 0, otherwise
- z_i 1, if item i is mixed with other items; 0, otherwise

The MIP model is formulated as follows.

$$\min k_1 \sum_h \left(1 - \sum_i c_i x_{ih}\right) + k_2 \sum_i t_i + k_3 \sum_h \sum_i \sum_{j>i} b_{ij} v_{ijh} + k_4 MaxP \tag{1}$$

$$\text{s.t } y_{ih} + y_{jh} - 1 \le v_{ijh} \quad \forall j > i, h \tag{2}$$

$$y_{ih} \le e_{ih} \quad \forall i, h \tag{3}$$

$$\sum_i c_i x_{ih} \le 1 \quad \forall h \tag{4}$$

$$y_{ih} \le x_{ih} \quad \forall i, h \tag{5}$$

$$c_i x_{ih} \le y_{ih} \quad \forall i, h \tag{6}$$

$$\sum_h x_{ih} \le a_i d_i \quad \forall i \tag{7}$$

$$\sum_h y_{ih} \le \left(\lceil c_i a_i \rceil + z_i\right) d_i \quad \forall i \tag{8}$$

$$M z_i \ge \sum_h \sum_j v_{ijh} \quad \forall i \tag{9}$$

$$\sum_h \sum_j v_{ijh} \ge z_i \quad \forall i \tag{10}$$

Table 4 Scheduled output data form

Item code	Amount	Start hanger	Finish hanger
I1	60	1	12
I10	53	13	23
I3	34	23	26

$$x_{ih} \geq x_{i,h+1} - M\left(1 - u_{i,h}\right) \quad \forall i \in \left\{i|d_i = 2\right\}, h = 1, \dots, H - 1 \tag{11}$$

$$x_{ih} \leq x_{i,h+1} + M\left(1 - u_{i,h}\right) \quad \forall i \in \left\{i|d_i = 2\right\}, h = 1, \dots, H - 1 \tag{12}$$

$$x_{i1} \leq Mu_{i1} \quad \forall i \in \left\{i|d_i = 2\right\} \tag{13}$$

$$u_{iH} = 0 \quad \forall i \in \left\{i|d_i = 2\right\} \tag{14}$$

$$u_{ih} + u_{i,h+1} \leq 1 \quad \forall i \in \left\{i|d_i = 2\right\}, h = 1, \dots, H - 1 \tag{15}$$

$$MaxP \geq \sum_{h}^{h+g-1} \sum_{i} \frac{l_i}{d_i} x_{ih} \quad \forall i, h \leq H - g + 1 \tag{16}$$

$$s_i \leq H\left(1 - y_{ih}\right) + hy_{ih} \quad \forall i, h \tag{17}$$

$$r_i \geq hy_{ih} \quad \forall i, h \tag{18}$$

$$r_i - s_i \leq \sum_{h} y_{ih} - 1 \quad \forall i \tag{19}$$

$$t_i \leq \sum_{h} x_{ih} \quad \forall i \tag{20}$$

$$t_i \leq a_i d_i - \sum_{h} x_{ih} \quad \forall i \tag{21}$$

$$M\left(1 - w_i\right) \geq \sum_{h} x_{ih} \quad \forall i \tag{22}$$

$$M\left(w_i + t_i\right) \geq a_i d_i - \sum_{h} x_{ih} \quad \forall i \tag{23}$$

$$t_i, u_{ih}, v_{ijh}, w_i, y_{ih}, z_i \in \{0, 1\} \quad \forall i, j, h \tag{24}$$

$$x_{ih} \in Z^+ \cup \{0\} \quad \forall i, h \tag{25}$$

The objective function (1) minimizes the weighted sum of total capacity loss, total penalty for items that are partially hung, total mixing cost, and maximum workload for g consecutive hangers. Constraints (2) ensure that v_{ijh} is 1 if items i and j are hung on the same hanger. Constraints (3) restrict items that can be hung on hangers based on the eligibility of hangers. Constraints (4) indicate that the number of items i, which is hung on a hanger, cannot exceed the hanger capacity. Constraints (5) and (6) set y_{ih} to be 0 or 1 depending on whether any item is hung on hanger h. Constraints (7) indicate that the total number of scheduled items cannot be more than the demand amount.

Constraints (8)–(10) limit the maximum number of required hangers for each item. Given that item i has $c_i = 1/5$, $a_i = 8$, and $d_i = 1$ if item i is not mixed with any other item ($z_i = 0$), then item i can only be hung on a maximum of 2 hangers (the summation of y_{ih} cannot be larger than 2). Otherwise, item i can be hung on a maximum of three hangers (the summation of y_{ih} cannot be larger than three). When item i fully utilizes the hangers (e.g., $a_i = 10$ and $z_i = 1$), the item can be hung on a maximum of 3 hangers. Constraints (8) consider the values of d_i.

Constraints (11)–(15) ensure that items with $d_i = 2$ are hung on consecutive hangers. Constraints (16) set $MaxP$ to be the maximum value of the total packing workloads of each g consecutive hangers. Constraints (17) and (18) obtain the initial and final hanger indices, respectively. Constraints (17)–(19) ensure that each item is processed continuously from the initial to the final hanger, and its schedule cannot be interrupted by any other item. Constraints (20)–(23) identify the items that are partially hung on the hangers. Constraints (24) and (25) are the binary and integer constraints, respectively.

The scheduling problem is NP-hard, and we can prove this by restriction. If a special case of the considered problem generated by restriction is the same with a known NP-hard problem, then the considered problem is also NP-hard because it contains the hard problem (Garey and Johnson 1979). We can restrict the scheduling problem into a one-dimensional bin packing problem (1DBPP), which is NP-hard (Coffman et al. 1997). If we restrict the scheduling problem as follows, then the resulting special case is the same as the 1DBPP.

$$\begin{aligned} a_i &= 1, \quad \forall i \\ d_i &= 1, \quad \forall i \\ e_{ih} &= 1, \quad \forall i, h \\ k_1 &= 1 \\ k_2 &= 0 \\ k_3 &= 0 \\ k_4 &= 0 \end{aligned}$$

In the 1DBPP, given a set of items (each item has its weight) and an unlimited number of bins, each item is assigned to one bin to minimize the number of bins while following the capacity of each bin (Fleszar and Charalambous 2011). In the restrictions above, setting $a_i = 1$, $d_i = 1$, and $e_{ih} = 1$ for each item i and hanger h are equivalent to considering one item for each type; one item is inserted only into one bin, and any item can be placed into any bin in 1DBPP, respectively. $k_1 = 1$ is set to minimize the total capacity loss of the hangers that is equivalent to minimizing the number of bins in 1DBPP. Thus, our scheduling problem is NP-hard.

Solution approaches and computational results

To address the NP-hardness of the considered problem, we develop 2-Opt and tabu search metaheuristic algorithms. 2-Opt is one of the simplest and most popular local search methods, and it is utilized to solve various scheduling problems (Potts and Strusevich 2009). Tabu search has also been successfully used in many scheduling studies, including those on no-wait scheduling problems (Schuster 2006; Liaw 2008; Bożejko and Makuchowski 2009; Wang et al. 2010; Arabameri and Salmasi 2013; Ahani and Asyabani 2014; Ding et al. 2015). Later studies also confirmed that tabu search performs well in other scheduling problems (e.g., parallel machine scheduling problem; Ahonen and de Alvarenga 2017; Bektur and Saraç 2019).

The 2-Opt algorithm is shown in <2-Opt algorithm>. We denote v as an item sequence, v' as an updated item sequence after swapping the positions of two items in the sequence, and $n(I)$ as the number of items in set I. item(i) is an item with position i in v. Given an item sequence in v, the items starting from the first one are assigned to the hangers serially starting from the initial hanger.

```
<2-Opt algorithm>
v = given list of items in I
while (true)
  check = false
  for i = 1 to n(I) − 1 in I
    for j = i + 1 to n(I) in I
      if pair (i, j) does not meet skipping criterion then
        v′ = swap(item(i), item(j)) in v
        if (f(v′) < f_best) then
          v = v′
          f_best = f(v′)
          check = true
        end if
      end if
    end for
  end for
  if (check == false or the computational time exceeds time_limit) then
    stop
end while
```

An initial item sequence v is generated in both algorithms as follows. An item is selected randomly as the first one for a sequence and the subsequent items with the least mixing cost with the previously hung item. If more than one candidate is available for the next item, then the one with the least total mixing cost between the candidate and all the unhung items is selected.

We skip certain swapping pairs to reduce the searching space of the proposed algorithms. Pairs of items with the same l_i value, total required number of hangers, item and packing types, item and sub-assembly names, and car model are skipped. Exchanging such pairs does not improve the objective values in terms of the *MaxP* value, mixing cost, and capacity loss of hangers.

Items are hung on the hangers while considering their hanger capacity requirement, length, and eligibility to the hangers. If possible, then all amounts of an item order are hung, followed by the subsequent item order. However, if an item cannot be hung on a certain hanger due to the eligibility constraint, then the subsequent eligible hanger is searched and filled with the item. Given a previously hung item i with $d_i = 1$, the next item j with $d_j = 2$ in sequence v are hung on the subsequent empty new hanger to ensure an easy packaging operation. The total cost $f(v')$ is calculated using objective function (1). If the total cost of the swapped sequence is less than the incumbent solution, then the swapped sequence is stored as the current best solution (v).

The proposed tabu search algorithm is shown in <Tabu search algorithm>. When swapping is performed in the algorithm, the swapped item pair is stored in the tabu list. In the next iterations, a pair of items is not swapped if it is included in the tabu list. When the number of pairs in the tabu list (*size*(*tabu list*)) exceeds the size of the tabu list (*tabu_list_size_limit*), the earliest stored swapping pairs are removed. Given that $I =$ number of items and $combination = {}_IC_2$, the *tabu_list_size_limit* is calculated as $\alpha \cdot combination$. The value of α and the size of the tabu list decrease in proportion to the remaining computational time. As the computational time reaches the time limit, the value becomes 0. By decreasing the value, the final iterations of the search focus on intensification rather than exploration.

An acceptance rule in Ropke and Pisinger (2006), which is similar to the simulated annealing, is used to accept worse non-tabu solutions for escaping from local optima. Given a current sequence (solution) v and temperature T, a new worse sequence v' is accepted with the probability of $e^{-(f(v')-f(v))/T}$. The temperature T initially equals T_{start} and decreases by $T = T \cdot c_{rate}$ for each iteration, where c_{rate} is the cooling rate. The value of T_{start} is set to allow the acceptance of a ρ % worse solution to be 50%. If the best objective value is not updated for *max_same_sol_iterations* number of while loops or if the computation time reaches the given *time_limit*, then the algorithm stops.

```
<Tabu search algorithm>
v = given list of items in I, f = a very large number
while (true)
  new_best = 0
  v_nontabubest = ∅
  f_best = f_nontabubest = a very large number
  for i = 1 to n(I) − 1 in I
    for j = i + 1 to n(I) in I
      if pair (i, j) does not meet skipping criterion then
        update α and tabu_list_size_limit
        v′ = swap(i, j) in v
        if (f(v′) ≤ f_best) then
          add (item(i), item(j)) into tabu list
          if (size(tabu list) > tabu_list_size_limit) then
            delete the earliest pair from tabu list
          end if
          v_best = v′
          f_best = f(v′)
          new_best = 1
        end if
        if ((item(i), item(j)) is not in tabu list) and (f(v′) > f) then
          if the solution can be accepted then
            add (item(i), item(j)) into tabu list
            if (size(tabu list) > tabu_list_size_limit) then
              delete earliest pair from tabu list
            end if
            if (f(v′) ≤ f_nontabubest) then
              v_nontabubest = v′
              f_nontabubest = f(v′)
            end if
          end if
        end if
      end if
    end for
  end for
  if (new_best = = 1) then
    v = v_best
    f = f_best
  else if (v_nontabubest ≠ ∅) then
    v = v_nontabubest
    f = f_nontabubest
  end if
  if f does not change in max_same_sol_iterations then stop
  if the computational time exceeds time_limit then stop
end while
```

The MIP model is solved using CPLEX 12.9.0, and the 2-Opt heuristic and tabu search algorithms are implemented in C++ using Microsoft Visual Studio 2010. Experiments are conducted on a computer with an Intel® Core™ i5-6400 CPU at 2.70 GHz with 8 GB of RAM. In the experiments, various numbers of hangers, such as 30, 200, 400, and 600, and their corresponding working hours from 9:00 a.m. to 9:45 a.m., 12:00 p.m., 3:00 p.m., and 6:00 p.m., respectively, are tested. Break times are arbitrarily allocated to consider hanger eligibility in cases with 30 hangers. Table 5 provides an example of the data set. Codes, inputs, and solutions for all instances are publicly accessible from (http://logistics.postech.ac.kr/Painting_Line_Scheduling.html). The provided executable program can be used by researchers or practitioners to solve other instances after replacing the input data. Researchers can also improve the performance of the algorithms (e.g., through hybridization with other solution methods) by modifying the shared source code.

Table 5 Input data of small-size instances

Input order	Item code	Required number of hangers per item (d_i)	Amount (a_i)	Capacity ($1/c_i$)	Packing workload level (l_i)
1	I1	2	88	10	1
2	I2	1	122	12	2
3	I3	1	86	4	3
4	I4	1	47	6	2
5	I5	1	67	6	2

In the experiments, the values of k_1, k_2, k_3, and k_4 are 3000, 50, 1, and 1, respectively. The company sets these weights based on the importance of the objectives. Through preliminary experiments by using Instances 31–35 in Table 10, we set the parameters of tabu search α, *max_same_sol_iterations*, ρ, c_{rate}, and θ to 0.25, 20, 35, 0.99, and 3, respectively.

Tables 6 and 7 show the results of MIP and tabu search algorithm for the sample instance in Table 5. In the MIP result shown in Table 6, 36 of I4 items are hung on hangers 1–6, 70 of I1 items are hung on hangers 7–20, and 32 of I3 items are hung on hangers 23–30. The solution is represented with $v = \{I4, I1, I3, I2, I5\}$. The tabu search algorithm obtain the following solution for the sample instance, 47 of I4 items hung on hangers 1–8, 60 of I1 items on hangers 9–20, and 32 of I3 items on hangers 23–30 (Table 7). The total cost of MIP is smaller than those of the proposed algorithms. Table 8 shows the objective values obtained using all of the approaches. The total objective value (total value) consists of four parts, namely, total capacity loss (*cap*), total penalty for items that are partially hung (*part*), total mixing cost (*mix*), and maximum workload for g consecutive hangers (*MP*).

Instances with various numbers of items and hangers are generated and solved. Instances with 30 hangers and 5, 30, 60, and 86 items are experimented on, and the results are shown in Table 9. The example in Table 5 refers to Instance 4 in Table 9. The average gap between MIP and the 2-Opt and tabu search solutions are 3.22% and 1.40%, respectively.

In the real situation, items with 86 types are hung onto 600 hangers. Additional instances with 200 and 400 hangers are also considered to assess the performance of the MIP model and the proposed heuristic algorithms. Table 10 provides the results. The MIP model cannot solve large-size instances. The MIP model can only hang items on some hangers in the instances while having a large amount of total capacity loss. It finds solutions in which all hangers are empty within 3600 s for Instances 28, 33, 34, and 35. The

Table 6 MIP results for small-size instance data

Item	Hanger																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	0	0	0	0	0	10	10	10	10	10	10	10	10	10	10	10	10	10	10	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	4	4	4	4	4	4
4	6	6	6	6	6	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7 Results of tabu search algorithm for small-size instance data

Item	Hanger																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	0	0	0	0	0	0	0	10	10	10	10	10	10	10	10	10	10	10	10	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	4	4	4	4	4	4
4	6	6	6	6	6	6	6	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8 Objective values of the example

Approach	Total value	*cap*	*part*	*mix*	*MP*
MIP	6246	2	3	0	96
2-Opt	6792	2.16	2	0	192
Tabu search	6704	2.16	2	0	104

average gap between MIP and each of 2-Opt algorithm and tabu search is −73.2% and −73.7%, respectively. The tabu search performs slightly better than 2-Opt. In conclusion, the proposed algorithms perform effectively when solving real-size instances.

Conclusions

We introduced a scheduling problem in a factory of automobile component primer painting, in which thousands of automobile parts undergo anticorrosion electrodeposition coating. The problem has various important and practical characteristics, such as item-hanger eligibility, different item sizes and packaging workloads, effect of a generated schedule on the packaging stage, continuous hanging requirement of orders, allowance for mixed items, and multiple conflicting objectives. These characteristics make the problem unique and have not been addressed previously. Our problem is classified as a no-wait flow shop batch scheduling problem. In our study, we use real data sets from a Korean company.

An MIP model was developed for the problem, and it shows that the problem is NP-hard. The MIP model cannot be used for real-size instances due to its complexity and long computational time. Therefore, we implemented customized versions of the well-known 2-Opt and tabu search algorithms. We also proposed a method to generate an initial solution for both methods, in which the first item was determined randomly. Then, the subsequent items with the least mixing cost with the previously hung item were selected. To decrease the computational time of the algorithms, we skip certain swapping pairs that have the same characteristics (e.g., packing workload, total required number of hangers, item and packing types, item and sub-assembly names, and car model) because exchanging such pairs does not improve the objective values in terms of maximum packing workload among consecutive hangers, mixing cost, and capacity loss of hangers. In the tabu search, we decreased the size of the tabu list in proportion to the remaining computational time to ensure that the final iterations of the search focus on intensification.

The proposed algorithms were confirmed to obtain good solutions when solving real-size instances. For small-size instances, the average gap between MIP and 2-Opt and the tabu search solutions was 3.22% and 1.40%, respectively. Meanwhile, the 2-Opt and tabu search algorithms performed well in solving real-size instances. The average gap between MIP and each of the 2-Opt algorithm and tabu search was −73.2% and −73.7%, respectively.

This study can be extended to various directions. First, the scheduling problem can be solved with direct consideration of the packaging process. In the present scheduling

Table 9 Comparison of MIP, 2-Opt, and tabu search algorithms on small-size instances (# of items = 5, 30, 60, 86; # of hangers = 30)

Instance no.	Items (hangers)	MIP model						2-Opt algorithm						Tabu Search						Gap (MIP and 2-Opt) (%)	Gap (MIP and tabu search) (%)
		Objective value					Comp time (s)	Objective value					Comp time (s)	Objective value					Comp time (s)		
		Total value	*cap*	*part*	*mix*	*MP*		Total value	*cap*	*part*	*mix*	*MP*		Total value	*cap*	*part*	*mix*	*MP*			
1	5 (30)	6140	2	2	0	40	1	6140	2	2	0	40	1	6140	2	2	0	40	1	0	0
2		6200	2	3	0	50	1	6200	2	3	0	50	1	6200	2	3	0	50	1	0	0
3		6616	2.12	3	4	105	1	6863	2.19	2	0	192	1	6774	2.19	2	0	103	1	3.74	2.39
4		6246	2	3	0	96	2	6792	2.17	2	0	192	1	6704	2.17	2	0	104	1	8.74	7.33
5		6150	2	2	0	50	2	6196	2	2	0	96	1	6150	2	2	0	50	1	0.75	0
6	30 (30)	6140	2	2	0	40	453	6995	2.17	2	21	344	1	6140	2	2	0	40	2	13.92	0
7		6104	2	1	0	54	41	7540	2.02	3	1004	306	1	6104	2	1	0	54	2	23.52	0
8		6185	2.1	1	12	103	125	6268	2	3	20	94	1	6185	2	1	12	103	2	1.33	0
9		6128	2	2	0	28	14	6128	2	2	0	28	1	6128	2	2	0	28	2	0	0
10		6576* (6236)	2	7	0	226	3600	7235	2.23	5	26	225	1	7650	2.40	3	14	261	2	10.02	16.33
11	60 (30)	6120* (6090)	2	2	0	20	3600	6116	2	2	0	16	2	6116	2	2	0	16	14	−0.07	−0.07
12		6537* (6157)	2	7	0	187	3600	6604	2.02	3	62	324	2	6589	2.02	3	56	324	20	1.02	0.80
13		6262* (6076)	2	4	0	62	3600	6240	2	3	0	90	1	6240	2	3	0	90	15	−0.35	−0.35
14		6151* (6122)	2	1	36	57	3600	6316	2	3	0	141	2	6316	2	3	0	141	10	2.68	2.68
15		6120	2	2	0	20	764	6120	2	2	0	20	2	6120	2	2	0	20	17	0	0
16	86 (30)	6120* (6119)	2	2	0	20	3600	6124	2	2	0	24	3	6124	2	2	0	24	52	0.07	0.07
17		6204* (6101)	2	3	0	54	3600	6148	2	2	0	48	3	6148	2	2	0	48	54	−0.90	−0.90
18		6116	2	2	0	16	389	6128	2	2	0	28	6	6116	2	2	0	16	81	0.20	0
19		6142* (6094)	2	2	0	42	3600	6130	2	2	0	30	5	6130	2	2	0	30	52	−0.20	−0.20
20		6112* (6094)	2	2	0	12	3600	6112	2	2	0	12	5	6112	2	2	0	12	56	0	0
Average																				3.22	1.40

*The best feasible solution is obtained within a limited time. (value) shows the best lower bound

Table 10 Comparison of MIP, 2-Opt, and tabu search algorithms on large-size instances (# of items = 86; # of hangers = 200, 400, and 600)

Instance no.	Items (hangers)	MIP model						2-Opt algorithm						Tabu search						Gap (MIP and 2-Opt) (%)	Gap (MIP and TS) (%)
		Objective value					Comp time (s)	Objective value					Comp time (s)	Objective value					Comp time (s)		
		Total value	*cap*	*part*	*mix*	*MP*		Total value	*cap*	*part*	*mix*	*MP*		Total value	*cap*	*part*	*mix*	*MP*			
21	86 (200)	182,216* (84,063)	59	19	0	2218	3600	90,342	28	5	3064	1790	43	90,108	28	5	3064	1556	300	−50.4	−50.5
22		94,366* (84,063)	31	11	0	693	3600	86,665	28	7	1026	1098	47	85,638	28	5	42	1112	300	−8.1	−9.2
23		92,583* (84,095)	29	33	2008	1631	3600	87,199	28	7	1066	1087	138	85,500	28	6	11	970	300	−5.8	−7.6
24		384,832* (84,080)	128	1	0	240	3600	87,416	29	8	134	1348	95	85,628	28	9	46	936	300	−77.2	−77.7
25		171,133* (84,053)	56	10	0	525	3600	87,565	29	6	21	412	65	87,565	29	6	21	412	300	−48.8	−48.8
26	86 (400)	1176,062* (117,037)	392	1	0	12	3600	127,575	41	10	2175	2240	88	123,287	40	10	156	2240	300	−89.1	−89.5
27		1147,108* (117,067)	382	1	0	58	3600	122,411	40	8	2098	1098	91	120,574	40	9	158	1301	300	−89.3	−89.4
28		1200,000 (117,013)	400	0	0	0	3600	127,072	40	9	4082	1485	228	123,967	40	7	1082	1485	300	−89.4	−89.6
29		1192,824* (117,039)	397	0	0	24	3600	144,678	44	9	12,119	1433	300	123,831	40	10	3128	1164	300	−87.8	−89.6
30		1182,053* (117,028)	394	1	0	3	3600	126,752	41	12	2089	1809	94	119,524	39	9	75	758	300	−89.2	−89.8
31	86 (600)	1704,564* (150,018)	597	39	0	8	3600	127,091	40	7	5093	2134	234	126,194	40	7	4109	1932	300	−92.5	−92.6
32		1793,575* (0)	597	40	0	5	3600	125,008	40	9	2120	1284	138	122,886	40	7	112	1284	300	−93.0	−93.1
33		1800,000 (0)	600	0	0	0	3600	129,795	42	8	3060	1193	97	127,320	41	8	2074	1482	300	−92.7	−92.9
34		1800,000 (0)	600	0	0	0	3600	147,780	45	8	12,143	1433	300	124,112	40	8	1142	1433	300	−91.7	−93.1
35		1800,000 (150,007)	600	0	0	0	3600	128,700	42	10	2096	1602	122	127,514	41	8	1084	1602	300	−92.8	−92.9
Average																				−73.2	−73.7

*The best feasible solution is obtained within a limited time. (value) shows the best lower bound problem, the packaging process is indirectly considered by controlling the packaging workload balance. If the scheduling problems of the coating and packaging processes can be solved together, then the total productivity can be improved. Second, other metaheuristic approaches (e.g., adaptive large neighborhood search) and computational intelligence algorithms (e.g., deep neural networks and evolutionary computation) may be tested to improve the quality of obtained solutions. Third, uncertainty related to the quality of the processed item can be incorporated into the problem. After the possibility of defects is considered, a robust schedule can be generated to minimize the production loss caused by defective items.

Acknowledgments This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (No. 2016R1A2B4012177 and No. NRF-2019R1F1A1058165) and the Brain Korea 21 Postdoctoral Program.

Compliance with ethical standards

Conflict of interest The authors declare no conflict of interest.

References

Ahani, G., & Asyabani, M. (2014). A tabu search algorithm for no-wait job shop scheduling problem. *International Journal of Operational Research, 19*(2), 246–258.

Ahonen, H., & de Alvarenga, A. G. (2017). Scheduling flexible flow shop with recirculation and machine sequence-dependent processing times: Formulation and solution procedures. *The International Journal of Advanced Manufacturing Technology, 89,* 765–777.

Allahverdi, A. (2016). A survey of scheduling problems with no-wait in process. *European Journal of Operations Research, 255,* 665–686.

Allahverdi, A., & Aydilek, H. (2013). Algorithms for no-wait flowshops with total completion time subject to makespan. *The International Journal of Advanced Manufacturing Technology, 68,* 2237–2251.

Allahverdi, A., & Aydilek, H. (2014). Total completion time with makespan constraint in no-wait flowshops with setup times. *European Journal of Operations Research, 238,* 724–734.

Arabameri, S., & Salmasi, N. (2013). Minimization of weighted earliness and tardiness for no-wait sequence-dependent setup times flowshop scheduling problem. *Computers & Industrial Engineering, 64*(4), 902–916.

Baykasoğlu, A., & Ozsoydan, F. B. (2018). Dynamic scheduling of parallel heat treatment furnaces: a case study at a manufacturing system. *Journal of Manufacturing Systems, 46,* 152–162.

Behnamian, J., & Fatemi Ghomi, S. M. T. (2016). A survey of multi-factory scheduling. *Journal of Intelligent Manufacturing, 27*(1), 231–249.

Behnamian, J., Fatemi Ghomi, S. M. T., Jolai, F., & Amirtaheri, O. (2012). Realistic two-stage flowshop batch scheduling problems with transportation capacity and times. *Applied Mathematical Modelling, 36,* 723–735.

Bektur, G., & Saraç, T. (2019). A mathematical model and heuristic algorithms for an unrelated parallel machine scheduling problem with sequence-dependent setup times, machine eligibility restrictions and a common server. *Computers & Operations Research, 103,* 46–63.

Bożejko, W., & Makuchowski, M. (2009). A fast hybrid tabu search algorithm for the no-wait job shop problem. *Computers & Industrial Engineering, 56*(4), 1502–1509.

Coffman, E. G., Garey, M. R., & Johnson, D. S. (1997). Approximation algorithms for bin-packing: a survey. In D. S. Hochbaum (Ed.), *Approximation algorithms for NP-hard problems* (pp. 46–93). Boston: PWS Publishing Company.

Dallasega, P., Rojas, R. A., Bruno, G., & Rauch, E. (2019). An agile scheduling and control approach in ETO construction supply chains. *Computers in Industry, 112,* 103122.

Damodaran, P., Rao, A. G., & Mestry, S. (2013). Particle swarm optimization for scheduling batch processing machines in a permutation flowshop. *The International Journal of Advanced Manufacturing Technology, 64,* 989–1000.

Ding, J.-Y., Song, S., Gupta, J. N. D., Zhang, R., Chiong, R., & Wu, C. (2015). An improved iterated greedy algorithm with a tabu-based reconstruction strategy for the no-wait flowshop scheduling problem. *Applied Soft Computing, 30,* 604–613.

Fleszar, K., & Charalambous, C. (2011). Average-weight-controlled bin-oriented heuristics for the one-dimensional bin-packing problem. *European Journal of Operational Research, 210,* 176–184.

Fuchigami, H. Y., & Rangel, S. (2018). A survey of case studies in production scheduling: Analysis and perspectives. *Journal of Computational Science, 25,* 425–436.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: a guide to the theory of NP-completeness*. New York: WH Freeman.

Hejazi, S. R., & Saghafian, S. (2005). Flowshop-scheduling problems with makespan criterion: a review. *International Journal of Operational Research, 43,* 2895–2929.

Henao, R., Sarache, W., & Gómez, I. (2019). Lean manufacturing and sustainable performance: trends and future challenges. *Journal of Cleaner Production, 208,* 99–116.

Johansen, S. G. (2019). Emergency orders in the periodic-review inventory system with fixed ordering costs and stochastic lead times for normal orders. *International Journal of Production Economics, 209,* 205–214.

Kim, Y.-D., Joo, B.-J., & Shin, J.-H. (2009). Heuristics for a two-stage hybrid flowshop scheduling problem with ready times and a product-mix ratio constraint. *Journal of Heuristics, 15,* 19–42.

Koulamas, C., & Panwalkar, S. S. (2018). New index priority rules for no-wait flow shops. *Computers & Industrial Engineering, 115,* 647–652.

Li, D., Meng, X., Liang, Q., & Zhao, J. (2015). A heuristic-search genetic algorithm for multi-stage hybrid flow shop scheduling with single processing machines and batch processing machines. *Journal of Intelligent Manufacturing, 26*(5), 873–890.

Liaw, C. F. (2008). An efficient simple metaheuristic for minimizing the makespan in two-machine no-wait job shops. *Computers & Operations Research, 35*(10), 3276–3283.

Lin, B. M. T., & Cheng, T. C. E. (2001). Batch scheduling in the no-wait two-machine flowshop to minimize the makespan. *Computers & Operations Research, 28,* 613–624.

Liu, C., Chen, H., Xu, R., & Wang, Y. (2018). Minimizing the resource consumption of heterogeneous batch-processing machines using a copula-based estimation of distribution algorithm. *Applied Soft Computing, 73,* 283–305.

Masmoudi, O., Yalaoui, A., Ouazene, Y., & Chehade, H. (2016). Multi-item capacitated lot-sizing problem in a flow-shop system with energy consideration. *IFAC-PapersOnLine, 49*(12), 301–306.

Matin, H. N. Z., Salmasi, N., & Shahvari, O. (2017). Makespan minimization in flowshop batch processing problem with different batch compositions on machines. *International Journal of Production Economics, 193,* 832–844.

Mendez, C. A., Cerda, J., Grossmann, I. E., Harjunkoski, I., & Fahl, M. (2006). State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Computers & Chemical Engineering, 30,* 913–946.

Mohammadi, S., Mirzapour Al-e-Hashem, S. M. J., & Rekik, Y. (2020). An integrated production scheduling and delivery route planning with multi-purpose machines: A case study from a furniture manufacturing company. *International Journal of Production Economics, 219,* 347–359.

Oulamara, A. (2007). Makespan minimization in a no-wait flow shop problem with two batching machines. *Computers & Operations Research, 34,* 1033–1050.

Oulamara, A. (2012). No-wait scheduling problems with batching machines. In R. Z. Ríos-Mercado & Y. A. Ríos-Solís (Eds.), *Just-in-time systems* (pp. 147–168). New York: Springer.

Oulamara, A., Kovalyov, M. Y., & Finke, G. (2005). Scheduling a no-wait flow shop containing unbounded batching machines. *IIE Transactions, 37,* 685–696.

Potts, C. N., & Kovalyov, M. Y. (2000). Scheduling with batching: a review. *European Journal of Operations Research, 120,* 228–249.

Potts, C. N., & Strusevich, V. A. (2009). Fifty years of scheduling: a survey of milestones. *Journal of the Operational Research Society, 60*(1), 41–68.

Ropke, S., & Pisinger, D. (2006). An adaptive large neighbourhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science, 40,* 455–472.

Ruiz, R., & Maroto, C. (2005). A comprehensive review and evaluation of permutation flowshop heuristics. *European Journal of Operations Research, 165,* 479–494.

Salmasi, N., Logendran, R., & Skandari, M. R. (2011). Makespan minimization of a flowshop sequence-dependent group scheduling problem. *The International Journal of Advanced Manufacturing Technology, 56,* 699–710.

Samarghandi, H., & Behroozi, M. (2017). On the exact solution of the no-wait flow shop problem with due date constraints. *Computers & Operations Research, 81,* 141–159.

Sapkal, S. U., & Laha, D. (2013). A heuristic for no-wait flow shop scheduling. *The International Journal of Advanced Manufacturing Technology, 68,* 1327–1338.

Sawik, T. (2002). An exact approach for batch scheduling in flexible flow lines with limited intermediate buffers. *Mathematical and Computer Modelling, 36,* 461–471.

Schuster, C. J. (2006). No-wait job scheduling: tabu search and complexity of subproblems. *Mathematical Methods of Operations Research, 63*(3), 473–491.

Selen, W. J., & Hott, D. D. (1986). A new formulation and solution of the flowshop scheduling problem with no in-process waiting. *Applied Mathematical Modelling, 10,* 246–248.

Stefansdottir, B., Grunow, M., & Akkerman, R. (2017). Classifying and modeling setups and cleanings in lot sizing and scheduling. *European Journal of Operations Research, 261,* 849–865.

Sun, Y., Zhang, C., Gao, L., & Wang, X. (2011). Multi-objective optimization algorithms for flow shop scheduling problem: A review and prospects. *The International Journal of Advanced Manufacturing Technology, 55,* 723–739.

Tang, L., & Liu, P. (2009a). Minimizing makespan in a two-machine flowshop scheduling with batching and release time. *Mathematical and Computer Modelling, 49,* 1071–1077.

Tang, L., & Liu, P. (2009b). Two-machine flowshop scheduling problems involving a batching machine with transportation or deterioration consideration. *Applied Mathematical Modelling, 33,* 1187–1199.

Tasgetiren, M. F., Pan, Q.-K., Suganthan, P. N., & Chua, T. J. (2011). A differential evolution algorithm for the no-idle flowshop scheduling problem with total tardiness criterion. *International Journal of Operational Research, 49,* 5033–5050.

Wang, C., Li, X., & Wang, Q. (2010). Accelerated tabu search for no-wait flowshop scheduling problem with maximum lateness criterion. *European Journal of Operations Research, 206*(1), 64–72.

Wang, I.-L., Yang, T., & Chang, Y.-B. (2012). Scheduling two-stage hybrid flow shops with parallel batch, release time, and machine eligibility constraints. *Journal of Intelligent Manufacturing, 23*(6), 2271–2280.

Yin, S., Nishi, T., & Zhang, G. (2015). A game theoretic model for coordination of single manufacturer and multiple suppliers with quality variations under uncertain demands. *International Journal of Systems Science: Operations & Logistics, 3*(2), 79–91.

Zhou, S., Li, X., Chen, H., & Guo, C. (2016). Minimizing makespan in a no-wait flowshop with two batch processing machines using estimation of distribution algorithm. *International Journal of Operational Research, 54*(16), 4919–4937.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.