JOURNAL OF INDUSTRIAL AND
MANAGEMENT OPTIMIZATION
Volume **11**, Number **2**, April **2015**

doi:10.3934/jimo.2015.11.399

pp. **399–419**

COORDINATION OF PRODUCTION AND TRANSPORTATION IN SUPPLY CHAIN SCHEDULING

Jun Pei*[,1,2], Panos M. Pardalos[2,3], Xinbao Liu[1,4]
Wenjuan Fan[1,5] and Shanlin Yang[1,4]

[1]School of Management, Hefei University of Technology
Hefei 230009, China

[2]Center for Applied Optimization, Department of Industrial and Systems Engineering
University of Florida, Gainesville, FL 32611, USA

[3]Laboratory of Algorithms and Technologies for Networks Analysis, National Research
University Higher School of Economics, Niznhy Novgorod 603093, Russia

[4]Key Laboratory of Process Optimization
and Intelligent Decision-making of Ministry of Education
Hefei 230009, China

[5]Department of Computer Science, North Carolina State University
Raleigh 27695, USA

Ling Wang

Shanghai Key Laboratory of Power Station Automation Technology
School of Mechatronics and Automation
Shanghai University, Shanghai 200444, China

(Communicated by Eugene Levner)

Abstract. This paper investigates a three-stage supply chain scheduling problem in the application area of aluminium production. Particularly, the first and the third stages involve two factories, i.e., the extrusion factory of the supplier and the aging factory of the manufacturer, where serial batching machine and parallel batching machine respectively process jobs in different ways. In the second stage, a single vehicle transports jobs between the two factories. In our research, both setup time and capacity constraints are explicitly considered. For the problem of minimizing the makespan, we formalize it as a mixed integer programming model and prove it to be strongly NP-hard. Considering the computational complexity, we develop two heuristic algorithms applied in two different cases of this problem. Accordingly, two lower bounds are derived, based on which the worst case performance is analyzed. Finally, different scales of random instances are generated to test the performance of the proposed algorithms. The computational results show the effectiveness of the proposed algorithms, especially for large-scale instances.

2010 *Mathematics Subject Classification.* Primary: 90B35, 90C27; Secondary: 90C59.
Key words and phrases. Supply chain scheduling, batching, transportation, heuristic algorithm.
This work is supported by the National Natural Science Foundation of China (Nos. 71231004, 71171071, 71131002). Prof. Panos M. Pardalos is partially supported by LATNA laboratory, NRU HSE, RF government grant, ag. 11.G34.31.0057.
*Corresponding author. Email: peijun@ufl.edu.

1. **Introduction.** In recent years, effective supply chain management has become of greater and greater importance with the integration of global cooperation. A supply chain includes all stages that have added value to a product, and all interactions among suppliers, manufacturers, distributors, and customers [1]. Most of the literature about supply chain focuses on stochastic models to analyze inventory control issues at a strategic level, and there is a small amount of related research combining scheduling problems with supply chains until Hall and Potts's paper [2]. In their work, Hall and Potts considered a variety of scheduling, batching, and delivery problems that arose in an arborescent supply chain, where a supplier made deliveries to several manufacturers and a manufacturer also made deliveries to customers. They also derived efficient dynamic programming algorithms for a variety of problems, and identified incentives and mechanisms for cooperation. Afterwards, the problem of supply chain scheduling has attracted more attention of researchers.

In this paper, we study a three-stage supply scheduling problem, which arises from the real aluminium production supply chain. Based on the orders of a kind of special industrial aluminum profiles, the raw materials of aluminium ingots need to be processed through extrusion and aging by the factories located in this supply chain. We conclude the whole process as three stages, 1) in the upstream of the supply chain, the supplier (extrusion factory) produces the jobs of the orders from the manufacturer, 2) in the midstream, the vehicle carries the jobs from the supplier to the manufacturer after their processing is completed in the supplier, and 3) in the downstream, the manufacturer (aging factory) produces the jobs transported from the supplier. Based on shared economic interests, the supply chain participants try to cooperate more efficiently to increase the overall productivity by making better cooperative production decisions together. This is the motivation to conduct the supply chain scheduling problem on the integration of production and transportation considered in this paper.

In the first study of supply chain scheduling by Hall and Potts [2], they described the concept of supply chain scheduling as follows, "*Whereas much of the supply chain management literature focuses on inventory control or lot-sizing issues, this paper considers a number of issues that are important to scheduling in supply chains. Central to this literature is the idea of coordination between different parts of a supply chain. Where decision makers at different stages of a supply chain make decisions that are poorly coordinated, substantial inefficiencies can result. This paper considers the coordination of scheduling, batching, and delivery decisions, both at a single stage and between different stages of a supply chain, to eliminate those inefficiencies. The objective is to minimize the overall scheduling and delivery cost. This is achieved by forming batches of orders, each of which is delivered from a supplier to a manufacturer or from a manufacturer to a customer, in a single shipment.*" Since this problem is frequently encountered in many real-life supply chains, it has attracted the attention of many researchers over the last ten years. Basically, there are mainly three types of algorithms for solving supply chain scheduling problems: exact algorithms, heuristic algorithms, and metaheuristic approaches. Here we conduct a brief review of them, respectively.

Some exact algorithms, which are mainly dynamic programming and branch and bound algorithms, were used to solve supply chain scheduling problems. How optimization techniques were applied to scheduling problems, in particular dynamic programming and branch and bound, was described in [3]. A number of remarks with regard to areas of applications were also presented. Gordon and Strusevich [4] studied the supply chain scheduling problem derived between the manufacturer and customer. In their work, the job processing time depended on its position in the processing sequence, and the objective was to minimize the sum of the cost of changing the due dates and the total cost of discarded jobs. They developed dynamic programming algorithms with CON (constant flow allowance) and SLK (slack time) due date assignment methods. Grunder [5] addressed a production and transportation problem in a single-stage supply chain. The goal was to minimize the sum of production, transportation, and holding costs. They proposed the algorithm to solve the problem based on the efficient dynamic programming scheme. Cheng and Wang [6] considered the machine scheduling problems where the jobs of different classes needed to be processed and then delivered to customers. When the first job was processed on a machine or the following job belonged to another class, a setup time was required. Their goal was to minimize the weighted sum of the last arrive time of jobs and transportation cost. They proposed a dynamic programming algorithm to solve it optimally. Yeung et al. [7] studied a two-echelon supply chain scheduling problem, where the upstream suppliers first processed materials and delivered the semi-finished jobs to the manufacturer due to its time window and then the manufacturer processed these semi-finished jobs and delivered the finished jobs to the downstream retailers within its own production and delivery time windows. Some dominance properties were derived, and based on these properties fast pseudo-polynomial dynamic programming algorithms were developed to solve the problem optimally. Chretienne et al. [8] considered the integrated batch sizing and scheduling problem in a supply chain. Each customer order could be only handled by a single machine at a time. Their goal was to minimize the sum cost of the tardiness and the setup. They proposed dynamic programming algorithms based on some structural properties for the problem.

In addition to the mentioned dynamic programming algorithms, there were some studies which developed branch and bound algorithms for the proposed problem. One of them could refer to Mazdeh et al. [9] which studied the scheduling problem where the jobs were processed on a single machine and then transported to customers in batches for further processing. Their goal was to minimize the sum cost of flow time and delivery. They used a branch and bound algorithm to solve it and proved the proposed algorithm to be far more efficient than the only existing algorithm for this problem. Bard and Nananukul [10] investigated the scheduling problem including a single production facility, a set of customers, and a fleet of vehicles. Their goal was to minimize the total cost of production, inventory, and delivery. An approach combining heuristic algorithms and the branch-and-price algorithm was proposed, where a novel column generation heuristic algorithm and a rounding heuristic algorithm were developed to improve the algorithm. Rasti-Barzoki et.al [11] studied the integrated production and delivery scheduling problem in a supply chain, where a manufacturer first processed the orders from a customer on one or two machines and then the finished jobs were delivered to the customer. Their objective was to minimize the sum cost of the total weighted number of tardy jobs and the delivery costs. They designed a new branch and bound algorithm for solving both the single machine scheduling problem and the two-machine flow-shop problem. Rasti-Barzoki and Hejazi [12] proposed a supply chain scheduling problem with the due date assignment and the capacity-constrained deliveries for multiple customers, and the objective was to minimize the weighted number of tardy jobs in the single machine environment. The problem was formulated as an integer programming model, and a branch and bound algorithm was presented for solving it. Chang et al. [13] considered a supply chain scheduling problem integrating production and distribution, where the production and distribution stages are modelled as the parallel machines scheduling problem and the capacitated vehicle routing problem, respectively. They proposed column generation techniques in conjunction with a branch and bound approach to solve this problem. Despite the fact that these exact algorithms were proposed to solve various small-scale supply chain scheduling problems, they cannot obtain solutions for large-scale scheduling problems effectively.

Heuristic algorithms were widely used in supply chain scheduling problems. Chang and Lee [1] considered the machine scheduling and finishing product delivery, where jobs were processed on a single machine and delivered by a single vehicle to one custom area. They provided a proof of NP-hardness and a heuristic algorithm with worst-case analysis. The worst-case performance ratio for their heuristic algorithm was proven to be 5/3, and the bound was tight. Jetlund and Karimi [14] discussed the maximum-profit scheduling of the logistic chemicals manufacturing and delivery. They proposed a mixed-integer linear programming formulation and a heuristic decomposition algorithm. Their approach was illustrated on a real industrial case study and shown an increase of 37.2% in profit compared to the original plan. Selvarajah and Steiner [15] studied the batch scheduling problem in a two-level supply chain from the view of the supplier. In this problem, the supplier produced multiple products and delivered them in batches, and the objective was to minimize the sum of the total inventory holding cost and batch delivery cost of the supplier. They proposed an algorithm which exhibited polynomial complexity time to solve it. Agnetis et al. [16] considered the scheduling problem of finding an optimal supplier's schedule, an optimal manufacturer's schedule, and optimal schedules for both in a two-stage supply chain. They developed a polynomial algorithm to minimize total interchange cost and buffer storage cost. Averbakh and Xue [17] examined the on-line supply chain scheduling problem with preemption, where a manufacturer processed jobs and delivered them to the customers. The jobs were processed in batches and delivered to the customers as single shipments. Their objective was to minimize the sum cost of the total flow time and total delivery. They presented an on-line two-competitive algorithm for the single customer and considered an extension of the algorithm for the case of multiple customers. Lin et al. [18] studied a two-stage supply chain scheduling problem, including two machines in the first stage and one machine in the second stage. Before each batch was formed on the second-stage machine, a constant setup time was required. They proposed several heuristic algorithms to minimize the makespan. Lee et al. [19] addressed a scheduling problem with multi-attribute setup times, the objective of which was to minimize the total setup time on a single machine. They developed a constructive heuristic algorithm based on several theorems to solve the problem. Although many heuristic algorithms were applied in the various supply chain scheduling problems, there are no particular heuristic algorithms for our studied problem in this paper.

In recent years, the technological advancements in computer science have encouraged metaheuristic approaches such as genetic algorithm (GA) to be applied in supply chain scheduling problems. Chan et al. [20] studied distributed scheduling problems in a multi-factory and multi-product environment. They proposed an adaptive genetic algorithm, using a new crossover mechanism named as dominated gene crossover. They also carried out a number of experiments, in which significant improvement was obtained by the proposed algorithm. Torabi et al. [21] considered a supply chain scheduling problem where a single supplier produced components and delivered them to an assembly facility. They developed a mixed integer nonlinear program and proposed a hybrid genetic algorithm to solve it. Naso et al. [22] focused on the supply chain scheduling problem about the distribution of ready-mixed concrete industry in a real-world case. After developing a detailed model for the problem, they proposed a hybrid genetic algorithm combined with constructive heuristic algorithms to solve it. Zegordi et al. [23] investigated a two-stage supply chain scheduling problem. In the first stage, there were m suppliers producing jobs at different speeds. In the second stage, each of l vehicles had a different speed and different transport capacity. They developed a mixed integer programming model and proposed a gendered genetic algorithm to solve it. Yimer and Demirli [24] studied a build-to-order (BTO) supply chain scheduling problem for material procurement, components fabrication, product assembly, and distribution. They decomposed the problem into two subsystems and evaluated them sequentially. They also proposed a genetic algorithm based solution procedure to solve it. Cakici et al. [25] focused on a multi-objective supply chain scheduling problem which integrated production and distribution, and the objective was to minimize the total weighted tardiness and total distribution costs. They developed different heuristic algorithms based on a genetic algorithm for a Pareto-optimal set of solutions.

Some other metaheuristic approaches were also applied in supply chain scheduling problems. Moon et al. [26] considered the integration of planning and scheduling in a supply chain, which took into account sequence and precedence constraints. They developed a new evolutionary search approach based on a topological sort to minimize the makespan within a reasonable computing time. Pardalos et al. [27] proposed a new metaheuristic for the job shop scheduling problem. They used the backbone and "big valley" properties of the job shop scheduling problem and obtained new upper bounds for many problems. Zhang et al. [28] discussed the procedures of order planning in details and constructed a nonlinear integer programming model for the order planning problem. They designed a hybrid Particle Swarm Optimization (PSO) and Tabu Search (TS) algorithm, in which new heuristic rules to repair infeasible solutions were proposed. Mehravaran and Logendran [29] focused on a non-permutation flowshop scheduling problem with sequence-dependent setup times in a supply chain. Their goal was to minimize the inventory for the producer and maximize the service level for the customers. They developed a metasearch heuristic algorithm employing a new concept known as TS to solve it. Liu and Chen [30] considered the inventory, routing, and scheduling problem in a supply chain. After developing an integrated model, they proposed a metaheuristic of variable neighborhood search algorithm for the problem.

Another important research perspective, flowshop scheduling problems have been well studied during the last 60 years. Intuitively, the models of flowshop scheduling and the supply chain scheduling problem considered in this paper have some similarities. However, there are still some significant differences between them. In traditional flowshop scheduling problems, it is a common assumption that there is no transportation time between machines. In other words, the jobs completed on the previous machine can be processed on the new machine immediately. However, it might be unrealistic in the supply chains of the real-world industries. In recent years, there were several papers concerning flowshop scheduling problems considering the transportation. Hurink and Knust [31] addressed a flowshop scheduling problem, where if a job was transferred from one machine to another, then the job was delivered by a single robot and the transportation time was job-dependent or job-independent. Their main contribution is that they derived new complexity results for special cases with constant processing or transportation times. Allaoui and Artiba [32] investigated a hybrid flowshop scheduling problem subject to the machine maintenance, where setup, cleaning, and transportation times were considered simultaneously. They developed the approach by using three dispatching rules, the simulated annealing heuristic, and a flexible simulation model. Naderi et al. [33] pointed out that in the literature there was almost no work considering the features of sequence-dependent setup and transportation times simultaneously in hybrid flowshop scheduling problems, and the related studies with the minimization objectives based on due date were far less. Then, they investigated the hybrid scheduling problem with the above two features, and their objective was to minimize both total completion time and total tardiness. A simulated annealing was developed to solve this problem, where the optimum parameters with the experiments' least possible number were selected by using Taguchi method. Naderi et al. [34] investigated the permutation flowshop problem, where the transportation time of jobs from one machine to the next machine was considered. Both multi-transporter and single-transporter systems were considered, and six different mixed integer linear programs were formulated for them. Some well-known heuristics were provided, and the metaheuristics were developed based on artificial immune systems, combining an effective local search heuristic and simulated annealing. Khalili and Tavakoli-Moghadam [35] studied a bi-objective flowshop scheduling problem which was to minimize the makespan and total weighted tardiness, where jobs' transportation times between machines were considered. Based on the attraction-repulsion mechanism of electromagnetic theories, a new multi-objective electromagnetism algorithm was developed to solve the problem. Although there was some research work studying on flowshop scheduling problems with transportation times, specific job processing patterns were seldom considered in previous research, and most of them considered the situation that a single machine processes one job at one time. In our considered supply chain scheduling problem, two specific job processing patterns are considered in different factories which are located in the same supply chain.

In summary, there are more and more algorithms proposed to solve supply chain scheduling problems. Most literature focused on the problem of coordination of production and transportation among the participants in supply chain. However, few authors considered the concrete production modes and constraints in the real factories, which is inadequate when confronted with practical situations. Also, most previous approaches cannot be directly applied in the large-scale scheduling problem, while this is particularly the reality in nowadays supply chain scheduling problems. Thus, this is our motivation to conduct such research. The main contributions of this paper can be summarized as follows:

(1) New research is pursued by considering the features of setup times and two typical processing ways of serial batching and parallel batching in a supply chain scheduling problem. To the best of our knowledge, this paper is the first time to consider the practical modes of production both in extrusion factory and aging factory and some particular production constraints simultaneously in a supply chain scheduling problem. The studied problem is also proved to be strongly NP-hard.

(2) The proposed problem is agilely analyzed in two different cases based on the relationship between the round-trip transportation time and the job processing time in the manufacturer. Two novel heuristic algorithms are developed for the problem, which can obtain high quality solutions for large-scale supply chain scheduling problems within a reasonable time.

(3) We derive the lower bounds of two different cases, based on which the worst case performance of two proposed heuristic algorithms is also analyzed.

The reminder of this paper is organized as follows: we start section 2 with the problem description. A mathematical model of the studied supply chain scheduling problem is proposed and the problem is also proved to be strongly NP-hard in section 3. In section 4 two heuristic algorithms are designed to solve the problem, and afterwards the performance of the heuristic algorithms is also analyzed. The computational experiments are conducted and discussed in section 5. We conclude the paper with a summary and give future research directions in section 6.

2. **Problem description.** The practical production system is complicated in the aluminium product supply chain, which is consisted of multiple participating enterprises. In this paper, we focus on the supply chain scheduling problem from the supplier to the manufacturer, i.e., extrusion factory and aging factory. Due to the large-scale feature of the problem, it is difficult for these enterprises to make an effective schedule for the whole supply chain. Until now, few enterprises can make the production plan considering upstream and downstream enterprises, and most enterprises even still rely on human experience to make manufacturing schedules. The poor performance on schedules reduces the competitiveness of the supply chain. Therefore, an effective solution for the scheduling problem is imperative.

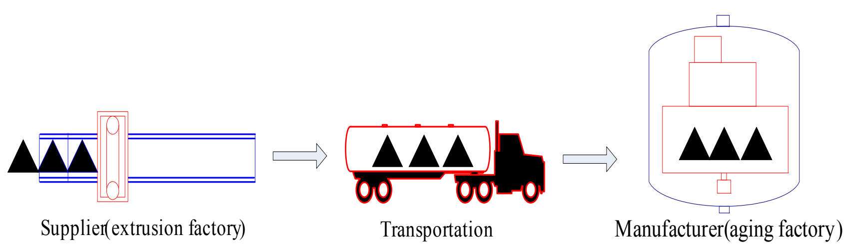

FIGURE 1. The structure of the supply chain scheduling between the supplier and the manufacturer

The structure of the studied scheduling problem is shown in Figure 1. A set $\{J_1, J_2, ..., J_n\}$ of n independent and non-preemptive jobs is available to be processed at time zero in the supplier. All jobs are processed in batches, and once a batch is initiated, no jobs in the batch can be released until the whole batch is completely processed. The machines and vehicle have the identical capacity of size, which is denoted as c, i.e., each machine and vehicle can handle no more than c jobs in a batch simultaneously. Our problem can be divided into 3 stages, i.e., the production stage on the supplier's machine, the transportation stage from the supplier to the manufacturer, and the production stage on the manufacturer's machine, which is described as follows:

(1) In the first stage, jobs are processed on the supplier's serial batching machine. The setup time is required before each batch is processed, which is denoted as s.

Let p_i and P_k denote the processing time of job J_i and batch b_k on the supplier's machine, respectively. In the serial batch production, jobs are processed one after another [36], so the processing time of batch b_k is $P_k = \sum_{J_i \in b_k} p_i (i = 1, 2, ..., n)$. There is no buffer for the processed batches, therefore the setup for a new batch cannot be started on the serial batching machine until the previous batch on the machine has been transported.

(2) In the second stage, all batches which have been processed are transported by only one vehicle. A round-trip transportation time of the vehicle is assumed to be a constant T, and one-way time is $T/2$. The vehicle can carry any one batch from the supplier in one-time shipment. We assume $T \geq p_i + s(i = 1, 2, ..., n)$ for emphasizing the importance of transportation.

(3) In the third stage, jobs are processed on the manufacturer's parallel batching machine, where all jobs in a batch are processed simultaneously [36]. The processing time of a batch on the parallel batching machine is determined by the batch processing time independent of jobs, which is supposed to be a constant P. It is also assumed that $P \geq p_i + s(i = 1, 2, ..., n)$ according to actual production condition.

Our objective is to minimize the makespan. Using the standard three-field notion $\alpha|\beta|\gamma$ introduced by Graham et al. [37], the problem can be denoted as $S \to M, 1|b = c, t_k = T|C_{max}$. In this notation, the symbols S and M stand for the supplier and the manufacturer respectively, and the symbol "1" represents that the number of machines in both the supplier and the manufacturer is one. The conditions $b = c$ and $t_k = T$ indicate that the capacity of the batching machines and vehicle is c, and the round-trip transportation time of each batch between supplier and manufacturer is equal to T, respectively. The symbol C_{max} denotes that the objective of the scheduling problem is to minimize the completion time of the last job.

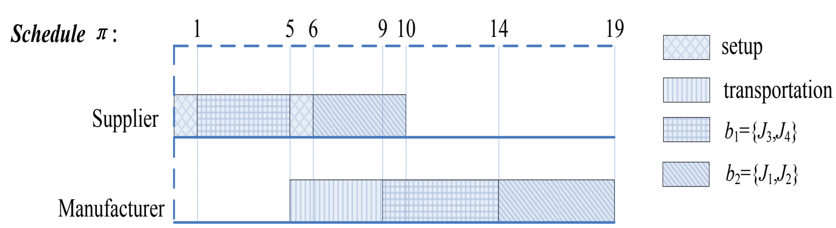

FIGURE 2. An example of the scheduling problem $S \to M, 1|b = c, t_k = T|C_{max}$

The following assumptions are considered for the problem formulation:

• All the facilities (i.e., machines and vehicle) are available at time zero in the usage time.

• The setup time on the supplier's machine is independent of the jobs sequence and batching.

• Pre-emption is prohibited, i.e., once a batch is initiated, no jobs can be released from the batch until the whole batch is completely processed.

To illustrate this problem, we give an example in Figure 2, where a set of four jobs with parameters $c = 2$, $s = 1$, $T = 8$, $P = 5$, $p_1 = 2$, $p_2 = 3$, and $p_4 = 1$ is considered. Figure 2 shows that the schedule π contains two batches, which are $b_1 = \{J_3, J_4\}$ and $b_2 = \{J_1, J_2\}$. The completion times for b_1 in the supplier and the manufacturer are 5 and 14, and for b_2 in the supplier and the manufacturer are 10 and 19, respectively. Then, the makespan is 19.

3. Model formulation and complexity analysis.

3.1. Model formalization.

In this section, a mixed integer programming model is formulated. The parameters and variables are described and the model is given below.

Parameters

n : total number of jobs;
i : job index, $i = 1, 2, ..., n$;
l : total number of batches, $\lceil \frac{n}{c} \rceil \leq l \leq n$;
k, f : batch index, $k, f = 1, 2, ..., l$;
c : the capacity of the batching machines and corresponding vehicle;
p_i : processing time of job J_i on the supplier's machine;
P_k : processing time of batch b_k on the supplier's machine;
P : processing time of each batch on the manufacturer's machine;
s : setup time on the supplier's machine;
T : round-trip transportation time of the vehicle between the supplier and the manufacturer;
M : a large enough positive constant.

Decision variables

x_{ik} : 1, if job J_i is in batch b_k; 0, otherwise;
y_{kf} : 1, if batch b_k is processed before batch b_f on the supplier's machine; 0, otherwise;
S_{1k} : start time of batch b_k on the supplier's machine;
C_{1k} : completion time of batch b_k on the supplier's machine;
D_k : departure time of batch b_k from the supplier to the manufacturer;
S_{2k} : start time of batch b_k on the manufacturer's machine;
C_{2k} : completion time of batch b_k on the manufacturer's machine;
C_{max} : the makespan;

Mixed integer programming model

$$Minimize \quad C_{max} \tag{1}$$

Subject to

$$\sum_{k=1}^{l} x_{ik} = 1, i = 1, 2, ..., n \tag{2}$$

$$\sum_{i=1}^{n} x_{ik} \leq c, k = 1, 2, ..., l \tag{3}$$

$$\sum_{k=1}^{l}\sum_{i=1}^{n} x_{ik} = n \tag{4}$$

$$C_{1k} = S_{1k} + \sum_{J_i \in b_k} p_i, k = 1, 2, ..., l \tag{5}$$

$$D_k \geq C_{1k}, k = 1, 2, ..., l \tag{6}$$

$$S_{1(k+1)} = D_k + s, k = 1, 2, ..., l-1 \tag{7}$$

$$S_{2k} = D_k + \frac{T}{2}, k = 1, 2, ..., l \tag{8}$$

$$C_{2k} = S_{2k} + P, k = 1, 2, ..., l \tag{9}$$

$$C_{1k} - C_{1f} + s + P_f - (1 - y_{kf})M \leq 0, k, f = 1, 2, ..., l, k \neq f \tag{10}$$

$$C_{2k} - C_{2f} + P_f - (1 - y_{kf})M \leq 0, k, f = 1, 2, ..., l, k \neq f \tag{11}$$

$$C_{max} \geq C_{2k}, k = 1, 2, ..., l \tag{12}$$

$$x_{ik} \in 0, 1, \forall i, k \tag{13}$$

$$x_{kf} \in 0, 1, \forall k, f \tag{14}$$

The objective function (1) is to minimize the makespan. Constraint (2) ensures that any job should be contained in only one batch. Constraint (3) guarantees that the number of jobs in a batch cannot exceed the capacity of the batching machines and corresponding vehicle. Constraint (4) indicates that the total number of jobs in all batches should be equal to the total number of all jobs. By setting constraint (5), the completion time of each batch on the supplier's machine is defined. Constraint (6) specifies that any batch can be only transported from the supplier to the manufacturer after it is completely processed on the supplier's machine. Constraint (7) restricts that a constant setup time is required before each batch is processed on the supplier's machine, and the setup of a new batch cannot be started on the supplier's machine until the previously processed batch has been transported. Constraint (8) ensures that any batch cannot start to be processed on the manufacturer's machine until it reaches the manufacturer. Constraint (9) indicates the completion time of each job on the manufacturer's machine. Constraints (10) and (11) guarantee that there is no overlapping situation between any two batches on the supplier's and manufacturer's machines, respectively. Constraint (12) defines the property of the maximum completion time. Finally, the ranges of the decision variables are defined by constraints (13) and (14).

3.2. **Complexity analysis.** In the following content, we present the strongly NP-hard proof for the problem $S \to M, 1|b = c, t_k = T|C_{max}$.

Theorem 1. *The Problem $S \to M, 1|b = c, t_k = T|C_{max}$ is strongly NP-hard.*

Proof. We construct an instance to perform the reduction with respect to the following 3-PARTITION problem, which is known to be strongly NP-hard [37].

3-PARTITION: Given an integer M and a set A of $3h$ positive integers $\{x_1, x_2, ..., x_{3h}\}$, $\frac{M}{4} < x_i < \frac{M}{2}, 1 \leq i \leq 3h$, such as $\sum_{i=1}^{3h} x_i = hM$, does there exist a partition A_1,A_2,...,A_h of the set A such that $|A_l| = 3$ and $\sum_{x_i \in A_l} x_i = M, 1 \leq l \leq h$?

We construct the following instance of problem $S \to M, 1|b = c, t_k = T|C_{max}$.

Number of jobs: $n = 3h + 3$.

Capacity of the batching machines and vehicle: $c = 3$.

Setup time: $s = M$.

Processing time of the jobs on the extrusion factory's serial batching machine: $p_i = x_i$, $i = 1, 2, ..., 3h$. $p_i = 0$, $i = 3h + 1, 3h + 2, 3h + 3$.

Processing time of each job on the aging factory's parallel batching machine: $P = 2M$.

Round-trip transportation time: $T = 2M$.

Threshold value: $y = (2h + 4)M$.

We claim that there is a solution to 3-PARTITION problem if and only if there exists an optimal schedule for the instance of problem $S \to M, 1|b = c, t_k = T|C_{max}$ with makespan no greater than (2h+4)M.

($\Rightarrow$) We can construct $A_1, A_2, \ldots, A_h$ to be a partition for the set A in this 3-PARTITION problem, and the schedule is shown in Figure 3. In the constructed schedule, the first batch A_1 contains three jobs $\{J_{3h+1}, J_{3h+2}, J_{3h+3}\}$ and departs from the supplier at the time M. It is easy to derive that the makespan is $(2h + 4)M = y$.

$\Leftarrow$ Conversely, suppose that there exists an optimal schedule with makespan no greater than y for the problem $S \to M, 1|b = c, t_k = T|C_{max}$. The minimum number of possible batches is h due to the capacity of the batching machines and vehicle. Thus, the minimum sum of the setup time and processing time of all batches on the extrusion factory's serial batching machine is $(h + 1)M + hM$. The sum of the processing time of one batch on the aging factory's parallel batching machine and its one-way transportation time is $2M + \frac{T}{2}$. We have $2M + \frac{T}{2} = y - [(h+1)M + hm]$. Besides, the earliest possible departure time of the first batch from the supplier is M, the minimum sum of transportation time is $(2h+1)M$, and the processing time of one batch on the aging factory's parallel batching machine is $2M$. Then, we have $M + 2M = y - (2h + 1)M$. Therefore, for $y = (2h + 3)M$, (a) there is no idle time for the batches setup operation and processing on the extrusion factory's serial batching machine in the interval $[0, (2h + 1)M]$, and (b) there is no idle time for transportation in the interval $[M, (2h + 2)M]$.

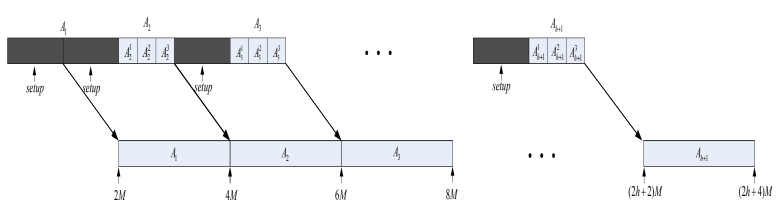

FIGURE 3. The optimal schedule in Theorem 1

We can prove this by contradiction. Suppose that there is a batch A_k, where $\sum_{J_i \in A_k} p_i \neq M$. There are two situations as follows:

If $\sum_{J_i \in A_k} p_i < M$, it is easy to infer from Figure 3 that there is idle time during the setup operation and processing on the extrusion factory's serial batching machine, and it contradicts (a).

If $\sum_{J_i \in A_k} p_i > M$, we can also see from Figure 3 that there is idle time during the transportation between the extrusion factory and aging factory, which contradicts (b).

Therefore, we can get a solution for 3-PARTITION problem.

Combining the "if" part and the "only if" part, we have proved the proposed theorem. □

4. **Heuristic algorithms and performance analysis.** In this section, we present two heuristic algorithms for two different cases. Also, two lower bounds for the studied problem and the worst case performance ratios of the proposed algorithms are established.

4.1. **Heuristic algorithms.** In this section, we will discuss two different cases.

4.11. **Case 1.** $T \geq P$

Heuristic Algorithm H_1

Step 1. Create $l = \lceil \frac{n}{c} \rceil$ batches based on the capacity c, and l should be the smallest number of possible batches. Index the jobs $\{J_1, J_2, \ldots, J_n\}$ in non-increasing order of the processing time p_i, i.e., $p_1 \geq p_2 \geq \ldots \geq p_n$;

Step 2. Set $i = 0$ and $P_f = 0 (f = 1, 2, \ldots, l)$, where P_f denotes the total processing time of jobs which have been assigned to batch b_f.

Step 3. $i = i + 1$. Assign job J_i to batch b_k, where $k = argmin_{f=1,2,\ldots,l}\{P_f \left| |b_f| < c, s + \sum_{J_i \in b_f} p_i \leq T \right|\}$. If there are multiple batches with the same least processing time, then the smallest indexed batch is selected. If there exists no such batch, then we create a new batch and assign J_i to it. Set $l = l + 1$ and $P_l = p_i$;

Step 4. If $i < n$, then return to step 3. Otherwise, turn to step 5;

Step 5. Re-index all batches in SPT order (the smallest processing time first order), i.e., $P_1 \leq P_2 \leq \ldots \leq P_l$. Schedule the batches on the machines and the vehicle according to the sequence of the batches.

Theorem 2. *Given $T \geq P$, we have $C_{max}(H_1) = s + P_1 + (l - 1)T + \frac{T}{2} + P$.*

Proof. According to the rule of H_1, we have $s + P_1 \leq s + P_2 \leq \ldots \leq s + P_l$. There are three situations with respect to P, T, and $s + P_k, k = 1, 2, \ldots, l$.

(1) $P \leq s + P_1 \leq s + P_2 \leq \ldots \leq s + P_l \leq T$;

(2) $s + P_1 \leq \ldots \leq P \leq \ldots \leq s + P_l \leq T$.

(3) $s + P_1 \leq s + P_2 \leq \ldots \leq s + P_l \leq P \leq T$.

In the first situation, for the first batch b_1, $d_1 = s + P_1$. Then, $C_{21} = d_1 + \frac{T}{2} + P = s + P_1 + \frac{T}{2} + P$. For the second batch b_2, $d_2 = d_1 + T = s + P_1 + T$, and $C_{22} = d_2 + \frac{T}{2} + P = s + P_1 + T + \frac{T}{2} + P$. Intuitively, the completion time of the kth batch b_k is $C_k = s + P_1 + (k - 1)T + \frac{T}{2} + P (k = 1, 2, \ldots, l)$. Therefore, we can get the makespan of the schedule $C_{max}(H_1) = s + P_1 + (l - 1)T + \frac{T}{2} + P$.

The derivation of the second and third situations is similar to the first one, so we can get the makespan of the schedule $C_{max}(H_1) = s + P_1 + (l - 1)T + \frac{T}{2} + P$.

The proof is completed. □

4.12. **Case 2.** $T < P$

Heuristic Algorithm H_2:

Step 1. Create $l = \lceil \frac{n}{c} \rceil$ batches based on the capacity c, and l should be the smallest number of possible batches. Index the jobs $\{J_1, J_2, \ldots, J_n\}$ in non-increasing order of the processing time p_i, i.e.,$p_1 \geq p_2 \geq \ldots \geq p_n$;

Step 2. Set $i = 0$ and $P_f = 0 (f = 1, 2, \ldots, l)$, where P_f denotes the total processing time of jobs which have been assigned to batch b_f;

Step 3. $i = i + 1$. Assign job J_i to batch b_k, where $k = argmin_{f=1,2,\ldots,l}\{P_f \Big| |b_f| < c, s + \sum_{J_i \in b_f} p_i \leq P \Big|\}$. If there are multiple batches with the same least processing time, then the smallest indexed batch is selected. If there exists no such batch, then we create a new batch and assign J_i to it. Set $l = l + 1$ and $P_l = p_i$;

Step 4. If $i < n$, then return to step 3. Otherwise, turn to step 5;

Step 5. Re-index all batches in SPT order, i.e., $P_1 \leq P_2 \leq \ldots \leq P_l$. Schedule the batches on the machines and the vehicle according to the sequence of the batches.

Theorem 3. *Given $T < P$, we have $C_{mac}(H_2) = s + P_1 + \frac{T}{2} + lP$.*

Proof. According to the rule of H_2, we have $s + P_1 \leq s + P_2 \leq \ldots \leq s + P_l$. There are three situations with respect to P, T, and $s + P_k, k = 1, 2, \ldots, l$.

(1) $s + P_1 \leq \ldots \leq s + P_{\theta-1} \leq T \leq s + P_\theta \leq \ldots \leq s + P_l \leq P (2 \leq \theta \leq l)$;

(2) $T \leq s + P_1 \leq \ldots \leq P \leq \ldots \leq s + P_l \leq P$;

(3) $s + P_1 \leq s + P_2 \leq \ldots \leq s + P_l \leq T < P$.

In the first situation, for the first batch b_1, we have $d_1 = s + P_1$. Then, $C_{21} = d_1 + \frac{T}{2} + P = s + P_1 + \frac{T}{2} + P$. For the $(\theta - 1)$th batch $b_{\theta-1}$, we can deduce that $d_{\theta-1} = s + P_1 + (\theta - 2)T$, and $C_{2(\theta-1)} = s + P_1 + \frac{T}{2} + (\theta - 1)P$. The θth batch b_θ departs from the supplier at the time $d_\theta = s + P_1 + (\theta - 2)T + s + P_\theta$, and it reaches the manufacturer at the time $d_\theta + \frac{T}{2} = s + P_1 + (\theta - 2)T + s + P_\theta + \frac{T}{2}$. The earliest time when the manufacturer's machine can process the θth batch b_θ is $s + P_1 + \frac{T}{2} + (\theta - 1)P$, where $s + P_1 + \frac{T}{2} + (\theta - 1)P \geq s + P_1 + (\theta - 2)T + s + P_\theta + \frac{T}{2}$, so we have $C_{2\theta} = s + P_1 + \frac{T}{2} + (\theta - 1)P + P = s + P_1 + \frac{T}{2} + \theta P$. Intuitively, the completion time of the last batch b_l is $C_{2l} = s + P_1 + \frac{T}{2} + lP$. Therefore, the makespan of the schedule $C_{max}(H_2) = s + P_1 + \frac{T}{2} + lP$ when $T < P$.

The derivation of the second and third situations is similar to the first one, so we can get the makespan of the schedule $C_{max}(H_2) = s + P_1 + \frac{T}{2} + lP$.

The proof is completed. □

4.2. **Lower bounds.** In order to evaluate the performance of the proposed heuristic algorithms, we derive two lower bounds of the makespan under two situations. When $T \geq P$, we assume that there is no additional idle time on the vehicle. The lower bound in the first case can be derived as the sum of the completion time of the first batch on the supplier's serial batching machine, the round-trip transportation time of $(l - 1)$ batches, the one-way transportation time of the last batch, and the processing time of the last batch on the manufacturer's parallel batching machine, i.e. $s + P_1 + (l - 1)T + \frac{T}{2} + P$, where l denotes the total number of the batches. Since $l \geq \lceil \frac{n}{c} \rceil$ and $P_1 \geq min_{i=1,2,\ldots,n}\{p_i\}$, we have the lower bound for the first case as $LB_1 = min_{i=1,2,\ldots,n}\{p_i\} + s + (\lceil \frac{n}{c} \rceil - 1)T + \frac{T}{2} + P$ when $T \geq P$.

When $T < P$, we assume that there is no additional idle time on the manufacturer's parallel batching machine. The lower bound in the second case can be derived as the sum of the completion time of the first batch on the supplier's serial batching machine, the one-way transportation time of the first batch, and the total processing time of all batches on the manufacturer's parallel batching machine, i.e. $s + P_1 + \frac{T}{2} + lP$, where l denotes the total number of the batches. Since $l \geq \lceil \frac{n}{c} \rceil$ and $P_1 \geq min_{i=1,2,\ldots,n}\{p_i\}$, the lower bound for the second case is $LB_2 = min_{i=1,2,\ldots,n}\{p_i\} + s + \frac{T}{2} + \lceil \frac{n}{c} \rceil P$ when $T < P$.

4.3. **Worst case analysis.** Let the symbol C^* denote the optimal makespan, and we have the makespan $C_{max}(H_1)$ and $C_{max}(H_2)$ when $T \geq P$ and $T < P$ according to the Theorems 2 and 3, respectively. The following theorems are proposed to analyze heuristic algorithms H_1 and H_2.

Theorem 4. *Given $T \geq P$, the worst case ratio of H_1 is no more than $3c$.*

Proof. Let the lower bound of the first case in the supply chain scheduling problem be C^{LB_1}. We obtain $C_{max}(H_1)$ and C^{LB_1} as follows:

$C_{max}(H_1) = s + P_1 + (l-1)T + \frac{T}{2} + P,$

$C^{LB_1} = min_{i=1,2,\ldots,n}\{p_i\} + s + (\lceil \frac{n}{c} \rceil - 1)T + \frac{T}{2} + P.$

Hence,

$$\frac{C_{max}(H_1)}{C^*} \leq \frac{C_{max}(H_1)}{C^{LB_1}} \leq \frac{s+P_1+(l-1)T+\frac{T}{2}+P}{min_{i=1,2,\ldots,n}\{p_i\}+s+(\lceil \frac{n}{c} \rceil-1)T+\frac{T}{2}+P}$$
$$\leq \frac{lT+\frac{1}{2}T}{(\lceil \frac{n}{c} \rceil-\frac{1}{2})T} \leq \frac{n+\frac{1}{2}}{\frac{n}{c}-\frac{1}{2}} \leq \frac{(1+\frac{1}{2n})c}{1-\frac{c}{2n}} \leq 2(1+\frac{1}{2n})c \leq 3c. \qquad \square$$

Theorem 5. *Given $T < P$, the worst case ratio of H_2 is no more than $2c$.*

Proof. Let the lower bound of the second case in the supply chain scheduling problem be C^{LB_2}. We obtain $C_{max}(H_2)$ and C^{LB_2} as follows:

$C_{max}(H_2) = s + P_1 + \frac{T}{2} + lP,$

$C^{LB_2} = min_{i=1,2,\ldots,n}\{p_i\} + s + \frac{T}{2} + \lceil \frac{n}{c} \rceil P.$

Hence,

$$\frac{C_{mac}(H_2)}{C^*} \leq \frac{C_{max}(H_2)}{C^{LB_2}} \leq \frac{s+P_1+\frac{T}{2}+lP}{min_{i=1,2,\ldots,n}\{p_i\}+s+\frac{T}{2}+\lceil \frac{n}{c} \rceil P}$$
$$\leq \frac{P+lP}{\lceil \frac{n}{c} \rceil P} \leq \frac{n+1}{\lceil \frac{n}{c} \rceil} \leq (1+\frac{1}{n})c \leq 2c. \qquad \square$$

5. **Computational experiments.** In this section, we conduct computational experiments to evaluate the performance of H_1 when $T \geq P$ and H_2 when $T < P$, respectively. We designed two types of computational tests for both situations with a number of random instances. One type of computational tests involved small-scale random instances, and the other one was performed with large-scale random instances. The parameters for the test problems are randomly generated based on the real aluminium production as follows:

(a) Number of jobs n:

For small-scale random instances, we take $n \in \{50, 60, 70, 80, 90, 100\}$;

For large-scale random instances, we take $n \in \{500, 600, 700, 800, 900, 1000\}$.

(b) Capacity of the batching machines and vehicle c, job processing time on the supplier's serial batching machine p_i, and setup time on the supplier's serial batching machine s are generated from the continuous uniform distributions $U[6, 8]$, $U[1, 15]$, and $U[2, 4]$, respectively.

(c) Transportation time between the supplier and manufacturer T is generated from the continuous uniform distributions $U[50, 60)$, and $U[60, 70]$:

For $T \geq P$, we take $T \in U[60, 70]$;

For $T < P$, we take $T \in U[50, 60)$.

(d) Job processing time on the manufacturer's parallel batching machine P is generated from the continuous uniform distributions $U[50, 60]$ and $U[60, 70]$:

For $T \geq P$, we take $P \in U[50, 60]$;
For $T < P$, we take $P \in U[60, 70]$.

To evaluate the performance of the proposed heuristic algorithms, the solutions reported by H_1 and H_2 were also compared with two other approaches applied in previous literature, namely algorithm LOE (Last-Only-Empty) and LPT (longest processing time), and algorithm FOE (First-Only-Empty) and SPT (shortest processing time) [38]. These different approaches are compared by measuring the relative gap between the makespan reported by each approach and the lower bounds derived in section 4.2. The relative gap percentage (Gap_H) between the approach H and the lower bound (LB) is calculated as in Eq. 15.

$$Gap_H = \frac{C_{max}^H - LB}{LB} \times 100\% \tag{15}$$

A factorial experiment was designed to determine the impact of the factors on the performance of the obtained solution. There are two factors in the factorial experiment, which characterize the number of jobs (i.e., n), and the capacity of machines and vehicles (i.e., c). Each treatment for the combination (n, c) was replicated fifty times, and both average gap and maximum gap were analyzed.

All the algorithms were coded in PowerBuilder 9.0 language and their code was run on a Pentium(R)-4, 300 MHz PC with 2GB of RAM.

Case 1. $T \geq P$

(1) Analysis of results for small-scale random instances

Figure 4 summarizes the computational results of the average relative gaps for small-scale random instances when $T \geq P$. As seen in Figure 4, it is clear that the average gap percentage of H_1 decreased with the number of jobs increasing except when $n = 90$, and it ranged from approximately 4.45% to 7.77%. H_1 outperformed the other two algorithms in this case, and the algorithm FOE and SPT performed better than the algorithm LOE and LPT. The algorithm LOE and LPT reported an average gap of 9.98% in the 100-job problem instance. However, H_1 reported an average gap of only 4.45%.

Figure 5 summarizes the computational results of the maximum relative gaps for small-scale random instances when $T \geq P$. From Figure 5, we observe that H_1 worked better than the other two algorithms, and the maximum gaps ranged from approximately 5.51% to 16.59%.

(2) Analysis of results for large-scale random instances

In Figure 6, we compare the impact of problem scale on the average relative gaps for large-scale random instances when $T \geq P$. From the results, we observe that the effectiveness of H_1 increased with the number of jobs increasing, and it ranged from approximately 0.48% to 1.09%. Compared to H_1, the other two algorithms performed poorly. The algorithm FOE and SPT also performed better than the algorithm LOE and LPT. The algorithm LOE and LPT reported an average gap of 10.29% in the 1000-job problem instance. However, H_1 reported an average gap of only 0.48%.

Figure 7 shows the maximum relative gaps for large-scale random instances when $T \geq P$. The maximum gaps appeared in a decreasing trend as the number of jobs increased except when $n = 900$, and it ranged from approximately 0.61% to 7.47%. The proposed heuristic algorithm also worked better than the other two algorithms.

Case 2. $T < P$

(1) Analysis of results for small-scale random instances

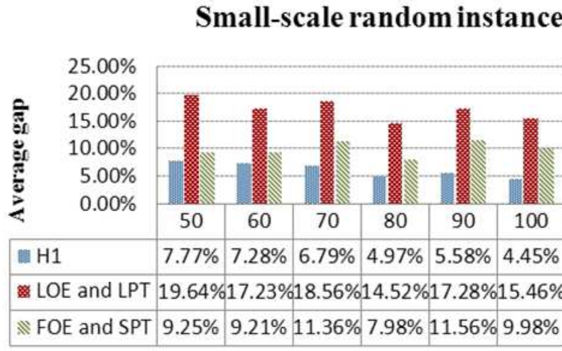

Figure 4. Computational results of the average gap percentage for $c \in U[6,8]$,$p_i \in U[1,15]$,$s \in U[2,4]$,$T \in U[60,70]$,$P \in U[50,60]$,$n \in \{50,60,70,80,90,100\}$

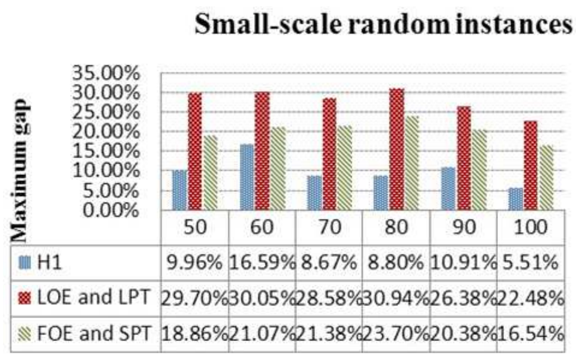

Figure 5. Computational results of the maximum gap percentage for $c \in U[6,8]$,$p_i \in U[1,15]$,$s \in U[2,4]$,$T \in U[60,70]$,$P \in U[50,60]$,$n \in \{50,60,70,80,90,100\}$

Figure 8 presents the comparison of the different approaches on the average gap percentage for small-scale random instances when $T < P$. It can be seen that the average gap percentage of H_2 decreased with the number of jobs increasing except when $n = 70$, and it ranged from approximately 4.67% to 9.61%. H_2 obtained better solutions than the other two algorithms, and the algorithm FOE and SPT worked better than the algorithm LOE and LPT. The algorithm FOE and SPT reported an average gap of 10.81% in the 100-job problem instance. However, H_2 reported an average gap of only 4.67%.

Figure 9 summarizes the computational results of the maximum relative gaps for small-scale random instances when $T < P$. From Figure 9, we observed that H_2 worked better than the other two algorithms, and the maximum gap ranged from approximately 6.67% to 25.22%.

(2) Analysis of results for large-scale random instances

Figure 10 displays the performance comparison of these three approaches in terms of the average gap percentage for large-scale random instances when $T < P$. The results indicated that the average gap percentage of H_2 also decreased with the

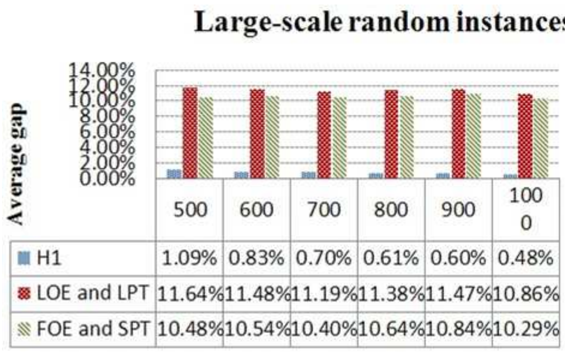

FIGURE 6. Computational results of the average gap percentage for $c \in U[6,8]$,$p_i \in U[1,15]$,$s \in U[2,4]$,$T \in U[60,70]$,$P \in U[50,60]$,$n \in \{500,600,700,800,900,1000\}$

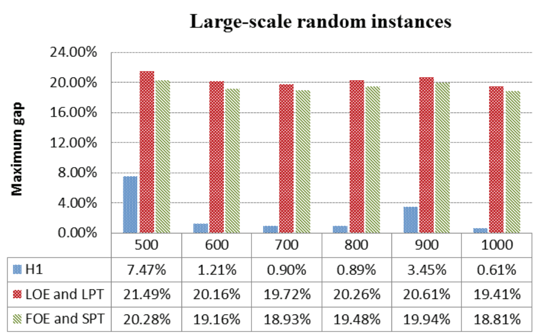

FIGURE 7. Computational results of the maximum gap percentage for $c \in U[6,8]$,$p_i \in U[1,15]$,$s \in U[2,4]$,$T \in U[60,70]$,$P \in U[50,60]$,$n \in \{500,600,700,800,900,1000\}$

number of jobs increasing except when $n = 900$, and it ranged from approximately 0.80% to 1.67%. It is clear that H_2 performed better than the other two algorithms, and the algorithm FOE and SPT also performed better than the algorithm LOE and LPT. The algorithm FOE and SPT reported an average gap of 12.36% in the 1000-job instance. However, H_2 reported an average gap of only 0.80%.

Figure 11 shows the maximum relative gaps in large-scale random instances when $T < P$. We observed that H_2 performed better than the other two algorithms, and its maximum gaps ranged from approximately 6.57% to 13.33%.

In summary, H_1 and H_2 outperformed the other two algorithms on either small-scale or large-scale problems. Even when $n \geq 500$, for both situations of $T \geq P$ and $T < P$, the average gaps between the proposed heuristic algorithms and the lower bounds were below 1.70%, indicating that H_1 and H_2 are quite effective for large-scale random instances. It is also worthwhile to note that the proposed heuristic algorithms can obtain solutions within 6 seconds for each problem instance, even for the instance up to 1000 jobs. Thus, the computational burden of the procedure is low enough to run the proposed heuristic algorithms for large-scale random instances.

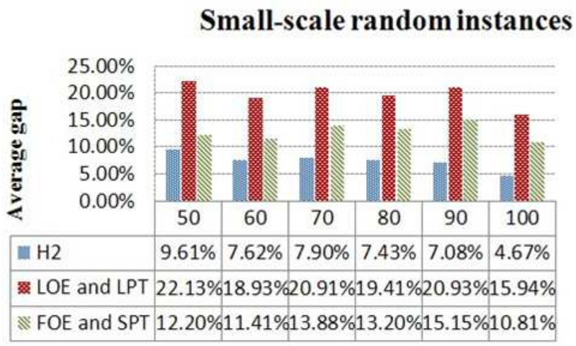

FIGURE 8. Computational results of the average gap percentage for $c \in U[6,8]$,$p_i \in U[1,15]$,$s \in U[2,4]$,$T \in U[50,60)$,$P \in U[60,70]$,$n \in \{50,60,70,80,90,100\}$

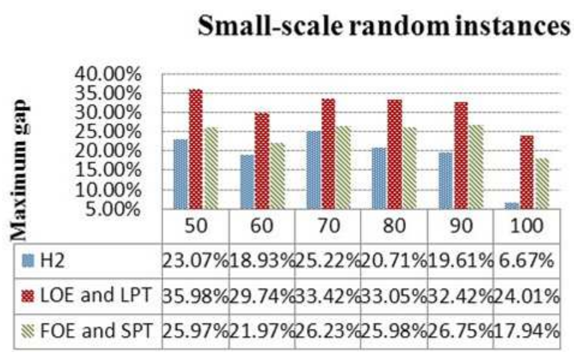

FIGURE 9. Computational results of the maximum gap percentage for $c \in U[6,8]$,$p_i \in U[1,15]$,$s \in U[2,4]$,$T \in U[50,60)$,$P \in U[60,70]$,$n \in \{50,60,70,80,90,100\}$

6. **Conclusions.** In this paper, we focus on a scheduling problem in an aluminum production supply chain. The strongly NP-hard problem can be characterized into the features such as different batching operations, setup time, capacity constraints, and numerical relationship between processing time and transportation time. In order to address the problem, two different heuristic algorithms were proposed respectively for the cases that $T \geq P$ and $T < P$. Meanwhile, we developed two lower bounds to analyze their worst case performance and verify the effectiveness of the proposed heuristic algorithms. The experimental results showed that the proposed heuristic algorithms outperformed the other two algorithms in the previous literature, and they can solve both small-scale and large-scale problems effectively and efficiently.

Despite this completed work, there are several interesting future directions based on this research. The first interesting one is to combine other objective functions,

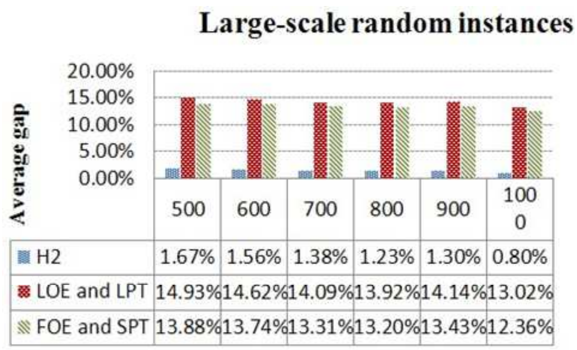

FIGURE 10. Computational results of the average gap percentage for $c \in U[6,8]$,$p_i \in U[1,15]$,$s \in U[2,4]$,$T \in U[50,60)$,$P \in U[60,70]$,$n \in \{500,600,700,800,900,1000\}$

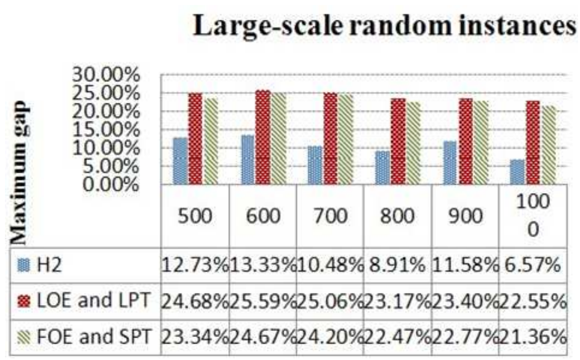

FIGURE 11. Computational results of the maximum gap percentage for $c \in U[6,8]$,$p_i \in U[1,15]$,$s \in U[2,4]$,$T \in U[50,60)$,$P \in U[60,70]$,$n \in \{500,600,700,800,900,1000\}$

such as minimizing the sum of completion time, minimizing maximum lateness, and minimizing the number of tardy jobs. The second future exploration may study the case with different job release times in the first stage. Last but not least, the other future research may consider multiple batching machines and vehicles in each stage to extend the practice of the model.

Acknowledgments. We would like to thank the editors and referees for their valuable comments and suggestions.

REFERENCES

[1] Y. C. Chang and Y. C. Lee, Machine scheduling with job delivery coordination, *European Journal of Operation Research*, **158** (2004), 470–487.
[2] N. G. Hall and C. N. Potts, Supply chain scheduling: Batching and delivery, *Operations Research*, **51** (2003), 566–584.
[3] P. M. Pardalos and G. C. R. Mauricio (Eds), *Handbook of Applied Optimization*, Oxford University Press, 2002.

[4] V. S. Gordon and V. A. Strusevich, Single machine scheduling and due date assignment with positionally dependent processing times, *European Journal of Operation Research*, **198** (2009), 57–62.

[5] O. Grunder, Lot sizing, delivery and scheduling of identical jobs in a single-stage supply chain, *International Journal of Innovative Computing Information and Control*, **6** (2010), 3657–3668.

[6] T. C. E. Cheng and X. Wang, Machine scheduling with job class setup and delivery considerations, *Computers & Operations Research*, **37** (2010), 1123–1128.

[7] W. K. Yeung, T. M. Choi and T. C. E. Cheng, Supply chain scheduling and coordination with dual delivery modes and inventory storage cost, *International Journal of Production Economics*, **132** (2011), 223–229.

[8] P. Chretienne, O. Hazir and S. Kedad-Sidhoum, Integrated batch sizing and scheduling on a single machine, *Journal of Scheduling*, **14** (2011), 541–555.

[9] M. M. Mazdeh, M. Sarhadi and K. S. Hindi, A branch-and-bound algorithm for single-machine scheduling with batch delivery and job release times, *Computers & Operations Research*, **35** (2008), 1099–1111.

[10] J. F. Bard and N. Nananukul, A branch-and-price algorithm for an integrated production and inventory routing problem, *Computers & Operations Research*, **37** (2010), 2202–2217.

[11] M. Rasti-Barzoki, S. R. Hejazi and M. M. Mazdeh, A branch and bound algorithm to minimize the total weighed number of tardy jobs and delivery costs, *Applied Mathematical Modelling*, **37** (2013), 4924–4937.

[12] M. Rasti-Barzoki and S. R. Hejazi, Minimizing the weighted number of tardy jobs with due date assignment and capacity-constrained deliveries for multiple customers in supply chains, *European Journal of Operational Research*, **228** (2013), 345–357.

[13] Y. C. Chang, K. H. Chang and T. K. Chang, Applied column generation-based approach to solve supply chain scheduling problems, *International Journal of Production Research*, **51** (2013), 4070–4086.

[14] A. S. Jetlund and I. A. Karimi, Improving the logistics of multi-compartment chemical tankers, *Computers & Chemical Engineering*, **28** (2004), 1267–1283.

[15] E. Selvarajah and G. Steiner, Batch scheduling in a two-level supply chain - A focus on the supplier, *European Journal of Operational Research*, **173** (2006), 226–240.

[16] A. Agnetis, N. G. Hall and D. Pacciarellir, Supply chain scheduling: Sequence coordination, *Discrete Applied Mathematics*, **154** (2006), 2044–2063.

[17] I. Averbakh and Z. Xue, On-line supply chain scheduling problems with pre-emption, *European Journal of Operational Research*, **181** (2007), 500–504.

[18] B. M. T. Lin, T. C. E. Cheng and A. S. C. Chou, Scheduling in an assembly-type production chain with batch transfer, *Omega-International Journal of Management Science*, **35** (2007), 143–151.

[19] C. H. Lee, C. J. Liao and C. W. Chao, Scheduling with multi-attribute setup times, *Computers & Industrial Engineering*, **63** (2012), 494–502.

[20] F. T. S. Chan, S. H. Chung and P. L. Y. Chan, An adaptive genetic algorithm with dominated genes for distributed scheduling problems, *Expert Systems with Applications*, **29** (2005), 364–371.

[21] S. A. Torabi, S. M. T. Fatemi Ghomi and B. Karimi, A hybrid genetic algorithm for the finite horizon economic lot and delivery scheduling in supply chains, *European Journal of Operational Research*, **173** (2006), 173–189.

[22] D. Naso, M. Surico, B. Turchiano and U. Kaymak, Genetic algorithms for supply-chain scheduling: A case study in the distribution of ready-mixed concrete, *European Journal of Operational Research*, **177** (2007), 2069–2099.

[23] S. H. Zegordi, I. N. Kamal Abadi and M. A. Beheshti Nia, A novel genetic algorithm for solving production and transportation scheduling in a two-stage supply chain, *Computers & Industrial Engineering*, **58** (2010), 373–381.

[24] A. D. Yimer and K. Demirli, A genetic approach to two-phase optimization of dynamic supply chain scheduling, *Computers & Industrial Engineering*, **58** (2010), 411–422.

[25] E. Cakici, S. J. Mason and M. E. Kurz, Multi-objective analysis of an integrated supply chain scheduling problem, *International Journal of Production Research*, **50** (2012), 2624–2638.

[26] C. Moon, Y. H. Lee, C. S. Jeong and Y. Yun, Integrated process planning and scheduling in a supply chain, *Computers & Industrial Engineering*, **54** (2008), 1048–1061.

[27] P. M. Pardalos, O. V. Shylo and A. Vazacopoulos, Solving job shop scheduling problems utilizing the properties of backbone and "big valley", *Computational Optimization and Applications*, **47** (2010), 61–76.

[28] T. Zhang, Y. J. Zhang, Q. P. Zheng and P. M. Pardalos, A hybrid particle swarm optimization and tabu search algorithm for order planning problems of steel factories based on the make-to-stock and make-to-order management architecture, *Journal of Industrial and Management Optimization*, **7** (2011), 31–51.

[29] Y. Mehravaran and R. Logendran, Non-permutation flowshop scheduling in a supply chain with sequence-dependent setup times, *International Journal of Production Economics*, **135** (2012), 953–963.

[30] S. C. Liu and A. Z. Chen, Variable neighborhood search for the inventory routing and scheduling problem in a supply chain, *Expert Systems with Applications*, **39** (2012), 4149–4159.

[31] J. Hurink and S. Knust, Makespan minimization flow-shop problems with transportation times and a single robot, *Discrete Applied Mathematics*, **112** (2001), 199–216.

[32] H. Allaoui and A. Artiba, Integrating simulation and optimization to schedule a hybrid flow shop with maintenance constraints, *Computers & Industrial Engineering*, **47** (2004), 431–450.

[33] B. Naderi, M. Zandieh, A. Khaleghi Ghoshe Balagh and V. Roshanaei, An improved simulated annealing for hybrid flowshops with sequence-dependent setup and transportation times to minimize total completion time and total tardiness, *Expert Systems with Applications*, **36** (2009), 9625–9633.

[34] B. Naderi, A. Ahmadi Javid and F. Jolai, Permutation flowshops with transportation times: Mathematical models and solution methods, *International Journal of Advanced Manufacturing Technology*, **46** (2010), 631–647.

[35] M. Khalili and R. Tavakoli-Moghadam, A multi-objective electromagnetism algorithm for a bi-objective flowshop scheduling problem, *Journal of Manufacturing Systems*, **31** (2012), 232–239.

[36] H. Xuan and L. Tang, Scheduling a hybrid flowshop with batch production at the last stage, *Computers & Operations Research*, **34** (2007), 2718–2733.

[37] R. L. Graham, E. L. Lawler, J. K. Lenstra and A. H. G. Rinnooy Kan, Optimization and approximation in deterministic machine scheduling: A survey, *Annals of Discrete Mathematics*, **5** (1979), 287–326.

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

Received June 2013; 1st revision January 2014; 2nd revision March 2014.

E-mail address: peijun@ufl.edu
E-mail address: pardalos@ufl.edu
E-mail address: lxinbao@126.com
E-mail address: wfan3@ncsu.edu
E-mail address: hgdysl@gmail.com
E-mail address: wangling.sh@gmail.com