

# Coordination of inbound and outbound transportation schedules with the production schedule



Utku Koç<sup>a</sup>, Ayşegül Toptal<sup>b,\*</sup>, Ihsan Sabuncuoğlu<sup>c</sup>

<sup>a</sup>Industrial Engineering Department, MEF University, İstanbul 34396, Turkey

<sup>b</sup>Industrial Engineering Department, Bilkent University, Ankara 06800, Turkey

<sup>c</sup>Rector Abdullah Gul University, Kayseri 38039, Turkey

## ARTICLE INFO

### Article history:

Received 24 March 2015

Received in revised form 8 November 2016

Accepted 18 November 2016

Available online 21 November 2016

### Keywords:

Supply chain scheduling

Transportation scheduling

Coordinated schedules

Beam search

## ABSTRACT

This paper studies the coordination of production and shipment schedules for a single stage in the supply chain. The production scheduling problem at the facility is modeled as belonging to a single process. Jobs that are located at a distant origin are carried to this facility making use of a finite number of capacitated vehicles. These vehicles, which are initially stationed close to the origin, are also used for the return of the jobs upon completion of their processing. In the paper, a model is developed to find the schedules of the facility and the vehicles jointly, allowing for effective utilization of the vehicles both in the inbound and the outbound. The objective of the proposed model is to minimize the sum of transportation costs and inventory holding costs. Issues related to transportation such as travel times, vehicle capacities, and waiting limits are explicitly accounted for. Inventories of the unprocessed and processed jobs at the facility are penalized.

The paper contributes to the literature on supply chain scheduling under transportation considerations by modeling a practically motivated problem, proving that it is strongly NP-Hard, and developing an analytical and a numerical investigation for its solution. In particular, properties of the solution space are explored, lower bounds are developed on the optimal costs of the general and the special cases, and a computationally-efficient heuristic is proposed for solving large-size instances. The qualities of the heuristic and the lower bounds are demonstrated over an extensive numerical analysis.

© 2016 Elsevier Ltd. All rights reserved.

## 1. Introduction and related literature

Supply, production and delivery are among the key functions for manufacturing companies. Although these functions are managed independently in many traditional systems, recent studies in supply chain management show that there is significant opportunity for savings if the related decisions are coordinated (Dawande, Geismar, Hall, & Sriskandarajah, 2006; Hall & Potts, 2003; Thomas & Griffin, 1996). Coordination of decisions among the various stages and functions of the supply chain is an issue that prevails at different phases of planning. Examples include coordination of decisions in the following areas: innovation, pricing at the strategic level; inventory control, lot sizing at the tactical level; scheduling at the operational level. Our focus in this study is on coordination of scheduling decisions involving production as well as inbound and outbound transportation.

We consider a setting consisting of two close warehouses—one for unprocessed jobs and the other for processed jobs, and a production facility far away from the warehouses. Shipment schedules of incoming materials and outbound delivery schedules in any system are linked to the production schedule through the inventories of unprocessed and processed jobs, respectively. In the specific setting of interest, the inventory holding costs for both types of jobs at the production facility, transportation costs and times between the facility and the warehouses are significant. Therefore, planning for effective interaction of the schedules for the production facility and the vehicles, serves as an important tool for lowering total inventory holding and transportation costs.

Our study is motivated by the practice of a worldwide home appliance manufacturer in Turkey, which imports a significant amount of its raw materials and exports a major portion of its end products. The company uses maritime transportation for import and export. The manufacturing facility is located inland whereas the two warehouses—one for holding the imported raw materials and one for holding the end products to be exported, are located at the harbor. Transportation of materials between

\* Corresponding author.

E-mail address: [toptal@bilkent.edu.tr](mailto:toptal@bilkent.edu.tr) (A. Toptal).

the manufacturing facility and the harbor is done via containers. Traditionally, the company arranges for transportation after the production schedule is made. This hierarchical decision making results in many containers being used only one way and traveling empty the other way. The company thinks that transportation costs can be reduced significantly if the inbound and outbound shipment schedules are coordinated so that the containers are utilized both ways. This kind of planning offers an opportunity but at the same time it turns out to be a challenge, because there is a limit on the time that a vehicle can be held at the facility, and the cost of holding the materials at the facility is high.

Motivated by the above practice, we study the problem of jointly finding the production schedule of the facility and the schedules of a finite number of capacitated vehicles subject to a waiting limit constraint at the facility. This waiting limit may originate in practice due to the length of the vehicle hire period. The objective is to minimize the total inventory holding and transportation costs for a certain number of unprocessed jobs to travel from an origin to a distant facility, get processed and return back to the origin. All vehicles are assumed to be identical, but jobs are allowed to occupy different amounts of space in the vehicles.

It is important to note that, the problem solved in this paper relates to a simplified version of the problem faced by the appliance manufacturer in question. As opposed to the real practice where multiple components form a final product, our model assumes that one inbound job is converted into one finished job after processing. This aspect of the proposed model makes it more applicable in a setting where jobs travel to and from a subcontractor for some of their operations to be performed. We refer to the subcontracting operations in the appliance manufacturer mentioned above and in the textile industry in U.S., as examples. The manufacturer in question outsources a portion of injection molding processes from subcontractors. Plastic fibers are sent to a subcontractor and the molded parts are then shipped back to the factory. As another example, some US textile manufacturers cut fabrics in the country and ship the cut fabrics to a low wage country, such as Mexico, for assembly. The assembled products are then returned back to US for finishing. Sen (2008) reports that, there are international agreements between US and Mexico on reducing the duty for outsourcing activities in the textile industry. In a setting where subcontracting is in place, our model may be of use if the objective is to minimize the sum of transportation costs and the inventory holding costs at the subcontractor.

Supply chain scheduling with transportation considerations has received significant attention over the past decade (e.g., Chang & Lee, 2004; Chen & Vairaktarakis, 2005; Li & Ou, 2005; Tang & Gong, 2008, 2009; Tang, Gong, Liu, & Li, 2014; Wang & Cheng, 2009). A common property of the studies in this area is that they model the factory as a single machine or parallel machines, and consider the scheduling of a group of jobs taking into account transportation times, capacities and/or costs in the inbound and/or the outbound. For the purpose of this paper, we classify the literature in terms of part of the supply chain where the transportation issues are modeled (i.e., inbound and/or outbound of the factory), and the objective function considered. As reviewed by the latest survey by Chen (2010), most papers focus on the delivery side (e.g., Agnetis, Aloulou, & Fu, 2014; Chang & Lee, 2004; Chen & Pundoor, 2006; Chen & Vairaktarakis, 2005; Gao, Qi, & Lei, 2015; Koc, Toptal, & Sabuncuoglu, 2013; Li, Vairaktarakis, & Lee, 2005; Toptal, Koc, & Sabuncuoglu, 2014; Wang & Cheng, 2006; Wang & Lee, 2005; Zhong, Dósa, & Tan, 2007). A few take into account inbound transportation (e.g., Tang & Gong, 2009; Tang et al., 2014), or both the inbound and the outbound transportation (e.g., Li & Ou, 2005; Tang & Gong, 2008; Wang & Cheng, 2009). Another feature that differentiates these studies from one another, is the objective function they consider. Many of the papers

reviewed, optimize a scheduling related objective such as functions of makespan, completion time of jobs, or total tardiness (e.g., Chang & Lee, 2004; Gao et al., 2015; Li & Ou, 2005; Li et al., 2005; Tang & Gong, 2009; Tang et al., 2014; Wang & Cheng, 2006, 2009; Zhong et al., 2007) whereas others take account of a combined measure of transportation costs and scheduling objectives (e.g., Chen & Pundoor, 2006; Chen & Vairaktarakis, 2005; Wang & Lee, 2005).

An important feature of our study is that we model transportation issues both in the inbound and the outbound of the production facility. To our best knowledge, Li and Ou (2005), Tang and Gong (2008), and Wang and Cheng (2009) are the only few papers with this consideration. Tang and Gong (2008) study a coordinated scheduling problem that involves a single batching machine with the objective of minimizing the sum of makespan and an increasing function of total number of batches. Li and Ou (2005) and Wang and Cheng (2009) consider minimization of makespan as an objective whereas our study aims to minimize total inventory holding and transportation costs. Moreover, our study differs from Li and Ou (2005), Tang and Gong (2008) and Wang and Cheng (2009) in the characteristics of the settings, concerning the number of vehicles used and the locations they operate in-between. Both Tang and Gong (2008) and Wang and Cheng (2009) assume that there are two vehicles—one for carrying items in the inbound from the warehouse to the factory, and one for carrying items in the outbound from the factory to a single customer location. It is important to emphasize that in both of these settings, different vehicles are utilized for the inbound and the outbound transportation, whereas, we consider a more restrictive case in which same vehicles handle the transportation in both directions. In this regard, Li and Ou (2005) is the only paper that exhibits some similarities to ours. They model the availability of one vehicle traveling between a factory and a warehouse where both the unprocessed and processed jobs are held.

Our paper builds on the idea of planning the schedules of a limited number of vehicles between two locations so that as many vehicles as possible are utilized both ways. Decreasing the number of trips made by the vehicles not only helps the total costs to be reduced, but it also brings down transport-related air pollution and the need for energy consumption. Although our primary focus is to solve the underlying production and transportation scheduling problems jointly in the specific setting of interest, we would like to note that some recent papers also study the different solution approaches resulting from sequentially solving the subproblems in varying settings (e.g., Agnetis et al., 2014; Toptal et al., 2014). In the paper, we show that the problem under consideration is NP-Hard in the strong sense. Therefore, analyzing this problem is both practically important and theoretically challenging.

In the next section, we begin with a detailed description of the problem and present a mixed integer linear programming formulation. In Section 3, we establish the computational complexity of the problem and present lower bounds on the optimal value of the objective function. We also present some properties of a class of solutions for the general case and a special case of the problem. This is followed by a description of the proposed heuristic in Section 4. In Section 5, we report the results of a computational study. Finally, in Section 6 we conclude the paper.

## 2. Problem definition and formulation

The system under consideration consists of two warehouses and a production facility. The warehouses, the first for unprocessed jobs and the second for end products, are close to each other. Therefore, they can be considered as in the same location, that is the origin. The production facility is far away from the warehouses.

Unprocessed jobs are transferred from the first warehouse to the production facility and end products are transported from the facility to the second warehouse with  $m$  identical vehicles. Job  $i$ , in its unprocessed state, occupies a space of  $s_i^1$  units in a vehicle. Similarly, it occupies a space of  $s_i^2$  units in a vehicle after it gets processed. The capacity of a vehicle is  $K$  units. Waiting time of a vehicle at the production facility is limited to  $l$  time units. A *tour* is referred to as the run made by a vehicle which starts and ends at the first warehouse, and visits the production facility and the second warehouse in that order. All vehicles are initially located in close proximity to the first warehouse. Total duration of a tour, excluding the waiting time, loading and unloading times, is called *tour time* and denoted by  $\tau$ . The production facility is modeled as a single machine. An unprocessed job  $i$  requires  $p_i$  time units of processing at the facility. Loading and unloading times are negligible.

A transportation cost  $c$  is incurred whenever a vehicle makes a tour, regardless of the number of jobs carried. Job  $i$  incurs an inventory holding cost of  $h_i^1$  per unit time it waits at the facility before its processing starts. Similarly, the inventory holding cost of job  $i$  per unit time it stays at the facility after it gets processed is  $h_i^2$ . No inventory holding cost is incurred for the jobs while they are being transported on the vehicles. The objective is to minimize the sum of inventory holding costs at the facility, and inbound and outbound transportation costs. A feasible solution to this problem should include the schedules of the vehicles and the production facility, and an assignment of the jobs to the vehicles for both inbound and outbound transportation. Before presenting the model, we introduce below additional notation for decision variables.

$\mathcal{N}$ : Set of jobs

$\sigma_i$ : Starting time of the processing of job  $i$ .  $\forall i \in \mathcal{N}$ .

$\alpha_i$ : Arrival time of job  $i$  to the facility.  $\forall i \in \mathcal{N}$ .

$\delta_i$ : Departure time of job  $i$  from the facility.  $\forall i \in \mathcal{N}$ .

$s_{ij} = \begin{cases} 1, & \text{if job } i \text{ is to be processed before job } j \\ 0, & \text{otherwise} \end{cases} \quad \forall i, j \in \mathcal{N}$

$a_t$ : Arrival time of the vehicle in tour  $t$  to the facility.  
 $t = 1, 2, \dots, 2|\mathcal{N}|$

$d_t$ : Departure time of the vehicle in tour  $t$  from the facility.  
 $t = 1, 2, \dots, 2|\mathcal{N}|$

$\psi_t = \begin{cases} 1, & \text{if } t^{\text{th}} \text{ tour is utilized} \\ 0, & \text{otherwise} \end{cases} \quad t = 1, 2, \dots, 2|\mathcal{N}|$

$x_{it} = \begin{cases} 1, & \text{if job } i \text{ arrives at the facility with tour } t \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in \mathcal{N}, t = 1, 2, \dots, 2|\mathcal{N}|$

$y_{it} = \begin{cases} 1, & \text{if job } i \text{ departs from the facility with tour } t \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in \mathcal{N}, t = 1, 2, \dots, 2|\mathcal{N}|$

$M$ : A very big number

The problem is first modeled as a nonlinear integer program. Then, an effective way for its linearization is proposed.

$$\min \sum_{i \in \mathcal{N}} h_i^1 (\sigma_i - \alpha_i) + \sum_{i \in \mathcal{N}} h_i^2 (\delta_i - (\sigma_i + p_i)) + c \sum_{t=1}^{2|\mathcal{N}|} \psi_t$$

subject to

$$\sigma_j \geq \sigma_i + p_i - M(1 - s_{ij}) \quad \forall i, j \in \mathcal{N} \quad (1)$$

$$\sigma_i \geq \alpha_i \quad \forall i \in \mathcal{N} \quad (2)$$

$$\sigma_i + p_i \leq \delta_i \quad \forall i \in \mathcal{N} \quad (3)$$

$$s_{ij} + s_{ji} = 1 \quad \forall i, j \in \mathcal{N} \quad (4)$$

$$\sum_{t=1}^{2|\mathcal{N}|} x_{it} = 1 \quad \forall i \in \mathcal{N} \quad (5)$$

$$\sum_{t=1}^{2|\mathcal{N}|} y_{it} = 1 \quad \forall i \in \mathcal{N} \quad (6)$$

$$\sum_{i \in \mathcal{N}} x_{it} s_i^1 \leq K \psi_t \quad t = 1, 2, \dots, 2|\mathcal{N}| \quad (7)$$

$$\sum_{i \in \mathcal{N}} y_{it} s_i^2 \leq K \psi_t \quad t = 1, 2, \dots, 2|\mathcal{N}| \quad (8)$$

$$\psi_t \geq \psi_{t+1} \quad t = 1, 2, \dots, 2|\mathcal{N}| - 1 \quad (9)$$

$$a_{t+m} \geq d_t + \tau \quad t = 1, 2, \dots, 2|\mathcal{N}| - m \quad (10)$$

$$d_t \geq a_t \quad t = 1, 2, \dots, 2|\mathcal{N}| \quad (11)$$

$$d_t \leq a_t + l \quad t = 1, 2, \dots, 2|\mathcal{N}| \quad (12)$$

$$\alpha_i = \sum_{t=1}^{2|\mathcal{N}|} a_t x_{it} \quad \forall i \in \mathcal{N} \quad (13)$$

$$\delta_i = \sum_{t=1}^{2|\mathcal{N}|} d_t y_{it} \quad \forall i \in \mathcal{N} \quad (14)$$

$$\sigma_i, \alpha_i, \delta_i, a_t, d_t \geq 0 \quad \forall i \in \mathcal{N}, t = 1, 2, \dots, 2|\mathcal{N}| \quad (15)$$

$$s_{ij}, \psi_t, x_{it}, y_{it} \in \{0, 1\} \quad \forall i, j \in \mathcal{N}, t = 1, 2, \dots, 2|\mathcal{N}| \quad (16)$$

The first and the second terms of the objective function are inventory holding costs for unprocessed and processed jobs, respectively. The third term corresponds to the transportation costs. Constraint set (1) assures that there is no overlap of the processing of different jobs. The set of constraints in (2) and (3) restrict the processing of a job to be between its arrival and departure times. The sequence of jobs is maintained by Expression (4). Expressions (5) and (6) ensure that each job is assigned to a tour for its arrival to and departure from the production facility. Vehicle capacity constraints are modeled by (7) and (8). Expression (9) is used for an orderly indexing of the tours. Expressions (10)–(12) establish the link between arrival and departure times of the tours. Finally, (13) and (14) make sure that arrival and departure times of the jobs are consistent with the arrival and departure times of the tours they are assigned to. Even though the constraint sets as defined by Expressions (13) and (14) are nonlinear, they can easily be linearized as follows:

$$\alpha_i \geq a_t - (1 - x_{it})M \quad \forall i \in \mathcal{N}, t = 1, 2, \dots, 2|\mathcal{N}|$$

$$\alpha_i \leq a_t + (1 - x_{it})M \quad \forall i \in \mathcal{N}, t = 1, 2, \dots, 2|\mathcal{N}|$$

$$\delta_i \geq d_t - (1 - y_{it})M \quad \forall i \in \mathcal{N}, t = 1, 2, \dots, 2|\mathcal{N}|$$

$$\delta_i \leq d_t + (1 - y_{it})M \quad \forall i \in \mathcal{N}, t = 1, 2, \dots, 2|\mathcal{N}|$$

Since the vehicles are identical, there is no need to provide a different schedule for each vehicle. Instead, we index the tours and decide on the arrival and departure times of each tour. The maxi-

imum number of tours is  $2|\mathcal{N}|$ , in which case each job arrives and departs with a different tour. The indexed tours are assigned to vehicles in a uniform manner. If there are  $m$  vehicles, the first vehicle makes the 1st,  $(m + 1)$ st,  $(2m + 1)$ st, ... tours, the second vehicle makes the 2nd,  $(m + 2)$ nd,  $(2m + 2)$ nd, ... tour, etc. Without loss of generality, we assume that vehicle  $k$  makes the tours  $k + mj$  where  $j \in \mathbb{Z}^+ \cup \{0\}$ . An optimal solution of the above integer program is post-processed and translated to an optimal solution of the original problem. The post-processing is briefly assigning arrival and departure times of the tours to the vehicles. If tour  $k$  is utilized (i.e.,  $\psi_k = 1$ ), its arrival and departure times, to and from the production facility, are taken as those of vehicle  $k$  at the first time it is used. Similarly, if tour  $k + mj$  is utilized, then vehicle  $k$  is used at least  $j + 1$  times, and the  $(j + 1)$ st arrival and departure times of this vehicle can be inferred from those of tour  $k + mj$ .

We would like to remark that our formulation considers a case where vehicles are stationed at the origin. Expressions (10)–(12) in our model should be replaced with the following in case vehicles are stationed at the manufacturer. In addition,  $a_1$  should be set to  $\tau$ .

$$d_{t+m} \geq a_t \quad t = 1, 2, \dots, 2|\mathcal{N}| - m \quad (17)$$

$$a_t \geq d_t + \tau \quad t = 1, 2, \dots, 2|\mathcal{N}| \quad (18)$$

$$a_t \leq d_t + \tau + l \quad t = 1, 2, \dots, 2|\mathcal{N}| \quad (19)$$

### 3. Analysis of the problem

The problem in its most general form is clearly NP-Hard in the strong sense as it incorporates a bin-packing problem. Because, different sized items have to be placed in capacitated vehicles, and the total costs increase with number of vehicles utilized. In the proof of Theorem 1, we show that even in the special case in which all items occupy the same space in the vehicles, the problem is strongly NP-Hard. Therefore, the rest of our analysis aims at identifying some properties of optimal solutions, with the objective of reducing the set of feasible solutions. We also propose some lower bounds on the optimal objective function value. The proofs of all the theorems and propositions of the paper are presented in Appendix A.

**Theorem 1.** *The decision version of the problem (referred to as problem  $\mathcal{P}$ ) is NP – Complete in the strong sense even in the special case of one vehicle and identical inventory holding cost rates among jobs.*

The mathematical program in Section 2 formulates the problem of interest in its most general form. This leads to many alternative solutions. However, some of these solutions can be further eliminated by the following observation: Vehicles are allowed to wait  $l$  time units at the production facility. This may lead to alternative solutions in which some vehicles arrive early at the production facility or depart late without affecting the rest of the schedule and without exceeding the waiting time limit. In the rest of the paper, we do not consider such alternative solutions that involve unnecessary waiting of the vehicles at the production facility. More specifically, we look into only the feasible solutions with the following characteristics:

- Every tour  $t$  departs from the production facility at  $d_t = \max(a_t, \delta_{(t)})$ . Here,  $\delta_{(t)}$  is the latest completion time of processing among those of all the jobs that depart from the production facility with tour  $t$  (if no such job exists,  $\delta_{(t)}$  is taken as 0).
- Every tour  $t$  arrives at the production facility at  $a_t = \min(d_t, \sigma_{(t)})$  where  $\sigma_{(t)}$  is the earliest start time of processing among those of all the jobs that arrive to the production facility with tour  $t$  (if no such job exists,  $\sigma_{(t)}$  is taken as  $\infty$ ).

We note that a solution may be optimal even though  $d_t > \max(a_t, \delta_{(t)})$  for some tour  $t$  as long as  $d_t \leq a_t + l$ . Similarly, a solution may be optimal even though  $a_t < \min(d_t, \sigma_{(t)})$  for some tour  $t$  as long as  $a_t \geq d_t - l$ . However, we eliminate these solutions for practical purposes. Furthermore, due to the identicalness of the vehicles, indexing the tours with  $\psi_t = 1$  such that  $a_1 \leq a_2 \leq \dots$ , an assignment of vehicles to the tours can be made for any solution to also have  $d_1 \leq d_2 \leq \dots$ . We now continue our analysis with some lower bounds on the optimal objective function value.

#### 3.1. Lower bound scheme

In this section, we propose two lower bounds on the optimal value of the objective function. The first lower bound, which is presented in Corollary 1, concerns the general case where there may be more than one vehicle. The second lower bound, which is presented in Corollary 2, applies to the case of one vehicle. Recall that, the objective function is composed of inventory holding and transportation costs. Given the number of tours, which will be denoted by  $\omega$ , transportation cost is fixed and is equal to  $c \times \omega$ . Note that  $\omega$  may range from  $\max\left\{\left\lceil \frac{\sum_{i \in \mathcal{N}} s_i^1}{K} \right\rceil, \left\lceil \frac{\sum_{i \in \mathcal{N}} s_i^2}{K} \right\rceil\right\}$  to  $2|\mathcal{N}|$ . For a specified value of  $\omega$ , Theorems 2 and 3 introduce lower bounds on inventory holding costs considering the general case and the one-vehicle case, respectively. A lower bound on the objective function value of an optimal solution in each case is then given by the minimum, over all possible  $\omega$  values, of the sum of lower bound on inventory holding costs and the value  $c \times \omega$ . The lower bounds in Corollaries 1 and 2 rely on this fact.

We start with presenting a lower bound on inventory holding costs for the general case.

**Theorem 2.** *Given the number of tours, i.e.  $\omega$ , the following is a lower bound on the total inventory holding costs:*

$$LB'_T(\omega) = \sum_{i=1}^{|\mathcal{N}|} p_{(i)} \sum_{k=1}^{\lfloor \frac{i-1}{\omega} \rfloor} \left( h_{(|\mathcal{N}|+\omega k+1-i)}^1 + h_{(|\mathcal{N}|+\omega k+1-i)}^2 \right)$$

Here,  $\lfloor x \rfloor$  refers to the largest integer that is smaller than or equal to  $x$ ,  $p_{(i)}$  is the  $i^{\text{th}}$  longest processing time,  $h_{(i)}^1$  is the  $i^{\text{th}}$  largest inventory holding cost rate of any unprocessed job, and,  $h_{(i)}^2$  is the  $i^{\text{th}}$  largest inventory holding cost rate of any processed job.

Next, based on the above theorem, we present a lower bound on the objective function value of an optimal solution.

**Corollary 1.** *A lower bound on the total cost of an optimal solution is given by*

$$LB^1 = \max_{\left\{ \left\lceil \frac{\sum_{i \in \mathcal{N}} s_i^1}{K} \right\rceil, \left\lceil \frac{\sum_{i \in \mathcal{N}} s_i^2}{K} \right\rceil \right\} \leq \omega \leq 2|\mathcal{N}|} \{ LB'_T(\omega) + c\omega \}.$$

The following theorem provides a lower bound on inventory holding costs for the one-vehicle case.

**Theorem 3.** *Given the number of tours, i.e.  $\omega$ , the following is a lower bound on the total inventory holding costs when there is a single vehicle:*

$$LB''_T(\omega) = \sum_{i=1}^{|\mathcal{N}|} \left\{ I_i(\tau - p_i) \min(h_i^1, h_i^2) + \sum_{k=1}^{\lfloor \frac{i-1}{\omega} \rfloor} p_{(i)} h'_{(|\mathcal{N}|+\omega k+1-i)} \right\}.$$

Here,  $h'_i = |h_i^2 - h_i^1|$ ,  $h'_{(i)}$  is the  $i$ th largest value of  $h'_i$ ,  $p_{(i)}$  is the  $i$ th longest processing time and  $I_i$  is an indicator variable with the following value:

$$I_i = \begin{cases} 1, & \text{if } \tau > p_i > l \\ 0, & \text{otherwise.} \end{cases}$$

Based on **Theorem 3**, the following corollary provides a lower bound on the objective function value of an optimal solution when  $m = 1$ .

**Corollary 2.** *In case of a single vehicle, a lower bound on the total cost of an optimal solution is given by*

$$LB^2 = \max \left\{ \left[ \frac{\sum_{i \in \mathcal{N}} s_i^1}{K} \right], \left[ \frac{\sum_{i \in \mathcal{N}} s_i^2}{K} \right] \right\}_{\leq \omega \leq 2|\mathcal{N}|} \left\{ \max(LB'_x(\omega), LB''_x(\omega)) + c\omega \right\}. \tag{20}$$

**3.2. A special case: jobs occupying identical size on vehicles and having identical inventory holding cost rates**

A special case of the problem is when all jobs occupy the same amount of space in the vehicles (i.e.,  $s_i^1 = s^1$  and  $s_i^2 = s^2$  for all  $i \in \mathcal{N}$ ) and have the same inventory holding cost rates (i.e.,  $h_i^1 = h^1$ ,  $h_i^2 = h^2$  for all  $i \in \mathcal{N}$ ). Recall that, **Theorem 1** and its proof imply that even in this special case, the problem is NP-Hard in the strong sense. The next two corollaries, which will be presented without proof, follow from **Theorems 2 and 3**. In these corollaries, we provide lower bounds on total inventory holding costs given the number of tours for the cases of multiple vehicles and one vehicle. Lower bounds on total costs can in turn be found as in **Corollaries 1 and 2**. Following these corollaries, we continue our analysis for this special case by presenting some properties of optimal solutions.

**Corollary 3.** *Given the number of tours, i.e.  $\omega$ , the following is a lower bound on the total inventory holding cost when  $h_i^1 = h^1$ ,  $h_i^2 = h^2$  for all  $i \in \mathcal{N}$ :*

$$LB'_x(\omega) = \left\{ \sum_{i=1}^{[\omega]} \left[ \frac{i-1}{\omega} \right] p_{(i)} \right\} (h^1 + h^2).$$

Here,  $[\omega]$  refers to the largest integer that is smaller than or equal to  $\omega$ , and,  $p_{(i)}$  refers to the  $i$ th longest processing time.

**Corollary 4.** *Given the number of tours, i.e.  $\omega$ , the following is a lower bound on the total inventory holding costs when there is a single vehicle and  $h_i^1 = h^1$ ,  $h_i^2 = h^2$  for all  $i \in \mathcal{N}$ :*

$$LB''_x(\omega) = \sum_{i=1}^{[\omega]} \left\{ I_i(\tau - p_i) \min(h^1, h^2) + \left[ \frac{i-1}{\omega} \right] p_{(i)} |h^1 - h^2| \right\}.$$

Here,  $p_{(i)}$  refers to the  $i$ th longest processing time and  $I_i$  is an indicator variable with the following value:

$$I_i = \begin{cases} 1, & \text{if } \tau > p_i > l \\ 0, & \text{otherwise.} \end{cases}$$

The sequence of jobs in their nondecreasing order of arrival times to the facility is referred to as the *inbound transportation sequence*. As several items may arrive to the facility in the same vehicle, an inbound transportation sequence related to a production sequence may not be unique. The sequence of jobs in their nondecreasing order of departure times from the facility is referred to as the *outbound transportation sequence*. Similarly, an outbound transportation sequence related to a production sequence may not be unique. The following two theorems jointly imply that there is an optimal solution in which inbound and outbound transportation sequences are in compliance with the production sequence.

**Proposition 1.** *Every feasible solution can be converted to an alternative one in which for all job pairs  $(i, j)$ , if job  $i$  precedes job  $j$  in the production sequence, job  $i$  arrives at the facility no later than job  $j$ .*

**Proposition 2.** *Every feasible solution can be converted to an alternative one in which for all job pairs  $(i, j)$ , if job  $i$  precedes job  $j$  in the production sequence, job  $i$  departs from the facility no later than job  $j$ .*

**Proof.** Similar to that of **Proposition 1**. □

**Propositions 1 and 2** and their proofs imply that there exists an optimal solution in which if job  $i$  precedes job  $j$  in the production sequence, then job  $i$  arrives at the facility and departs from the facility no later than job  $j$  does. This can be accomplished by a pairwise interchange of job assignments to the vehicles for their inbound and outbound transportation. The following two propositions present additional properties involving the jobs that arrive at and depart from the production facility together.

**Proposition 3.** *If  $h^1 < h^2$ , there exists an optimal solution in which jobs that arrive at and depart from the production facility together, are processed in LPT (Longest Processing Time first) order.*

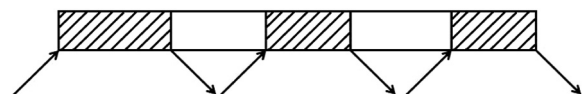
**Proposition 4.** *If  $h^1 > h^2$ , there exists an optimal solution in which jobs that arrive at and depart from the production facility together are processed in SPT (Smallest Processing Time first) order.*

**Proof.** Similar to that of **Proposition 3**. □

**4. Heuristic procedure**

A mathematical model has been presented in Section 2 for the problem of interest. Even in small-sized instances, this model has very long solution times (e.g., in the order of a week for 10 jobs). The proposed lower bounds and the characteristics of the optimal solutions decrease the computational times significantly. However, they are still too long to be considered as practical. In this section, we present a beam-search based heuristic to obtain solutions for large-size problems in shorter amount of time.

In the development of the heuristic approach, we restrict our analysis to the set of solutions which exhibit a certain property. Upon the analysis of the optimal solutions for some small sized instances (i.e., up to 10 jobs), we have detected that the following property reveals itself commonly: A job arriving with the  $t$ th tour either departs with the same tour (i.e., tour  $t$ ) or the next tour (i.e., tour  $t + 1$ ). Moreover, the sequence of jobs in the production schedule can be grouped into blocks such that the first block consists of the jobs that both arrive and depart with the first tour, the second block consists of the jobs that arrive with the first tour and depart with the second tour, and so on. We refer to this characteristic of a sequence as a *block structure*. In **Fig. 1**, an illustration of a sequence displaying this structure is presented. The arrows pointing inwards the figure coincide with inbound transportation times and the arrows pointing outwards coincide with the outbound transportation times.



**Fig. 1.** Block structure of a solution.

It is important to note that the set of solutions restricted to the above property does not always include an optimal one. However, numerical evidence shows that the cost of an optimal solution under this policy is close to that of a global optimum in many cases. In our computational tests, we observed that in more than 97% of the 960 5-job instances, there is an optimal solution satisfying the block structure. In the remaining instances, in which there is no optimal solution satisfying the block structure, the average and the maximum gap between the optimal objective function value and the cost of the best solution satisfying the block structure are 2.46% and 6.50%, respectively.

In the next three propositions, we present some characteristics of an optimal solution exhibiting the block structure.

**Proposition 5.** *In a lowest-cost solution among those satisfying the block structure, the jobs within the same block are processed in WSPT (Weighted Shortest Processing Time first) order where the weight of a job is calculated as  $w_i = h_i^2 - h_i^1$ , i.e., jobs are processed in nondecreasing order of  $\frac{h_i^2 - h_i^1}{p_i}$ . Furthermore, if there is some idle time in a block, the jobs with  $h_i^2 < h_i^1$  are processed consecutively before idle time, and the jobs with  $h_i^2 > h_i^1$  are processed consecutively after idle time.*

If all the jobs have the same inventory holding cost rate (i.e.,  $h_i^1 = h^1$  and  $h_i^2 = h^2$  for all  $i \in \mathcal{N}$ ), Proposition 5 implies that the jobs in the same block are processed in LPT (Longest Processing Time first) order if  $h^1 < h^2$ , and in SPT (Shortest Processing Time first) order if  $h^1 > h^2$ . Before we present further details about the heuristic, we provide two more properties that hold in the special case introduced in SubSection 3.2.

**Proposition 6.** *For a setting where  $h^1 \geq h^2$ , consider an optimal solution which exhibits the block structure. In this solution, if two jobs arrive at the facility together but depart from the facility with different tours, then the processing time of the job which departs later must be greater than that of the other.*

**Proposition 7.** *For a setting where  $h^2 \geq h^1$ , consider an optimal solution which exhibits the block structure. In this solution, if two jobs depart from the facility together but arrive at the facility with different tours, then the processing time of the job which arrives earlier must be greater than that of the other.*

**Proof.** Similar to that of Proposition 6. □

The main idea behind the proposed heuristic is to find a lowest-cost solution among those that satisfy the block structure, which is not necessarily optimal for the original problem. Furthermore, the procedure for finding an optimal solution that exhibits the block structure is based on beam search. Therefore, the output of the proposed procedure constitutes a heuristic solution for this problem as well.

The heuristic evolves over a search tree with the following characteristics: At level 0 of the search tree, there is a single node with no information, that is the root node. We first branch on the number of tours  $\omega$ . Note that  $\omega$  may range from  $\max \left\{ \left\lceil \frac{\sum_{i \in \mathcal{N}} s_i^1}{K} \right\rceil, \left\lceil \frac{\sum_{i \in \mathcal{N}} s_i^2}{K} \right\rceil \right\}$  to  $2|\mathcal{N}|$ . Fig. 2 illustrates part of the search tree for a sample problem with  $\max \left\{ \left\lceil \frac{\sum_{i \in \mathcal{N}} s_i^1}{K} \right\rceil, \left\lceil \frac{\sum_{i \in \mathcal{N}} s_i^2}{K} \right\rceil \right\} = 1$ . Conditioning on the value of  $w$ , Theorem 2 implies that  $LB'_T(\omega) + c\omega$  is a lower bound on total costs when  $m > 1$ . Similarly, Theorem 3 suggests that  $\max\{LB'_T(\omega), LB''_T(\omega)\} + c\omega$  is a lower bound on the total costs when  $m = 1$ . In subsequent parts of the search tree, we branch on different blocks for a given  $\omega$  value, and at each level, we consider the assignment of a job to one of the blocks. Fig. 2 shows how further branching is performed at the second level conditioning on  $\omega = 4$ . Note that, in this case, there are seven blocks, each block referring to a different pair of assignments of a job to a tour for its inbound and outbound transportation. For example, when a job is assigned to block 2–3, it is implied that the job arrives at the facility with the second tour and leaves the facility with the third tour. In general, if there are  $\omega$  tours, then there are  $2\omega - 1$  number of different blocks that a job can be assigned to.

We refer to the tree structure that emanates from a node at the first level a subtree. Notice that, there are at most  $2|\mathcal{N}|$  number of subtrees in a search tree. Our search for the best solution over the search tree gives full consideration to all the subtrees in the order of increasing  $\omega$ . However, only a certain number of nodes are kept for further consideration at each level of a subtree. Therefore, our search for the best solution conditioning on a value of  $\omega$ , unfolds in accordance with the beam search approach. The number of nodes that are explored further at each level of the subtree is a parameter of this approach, and is referred to as the *beam width*.

Since the subtrees corresponding to different values of  $\omega$  are explored sequentially, a feasible solution may be obtained from the search of each subtree. The total costs associated with such

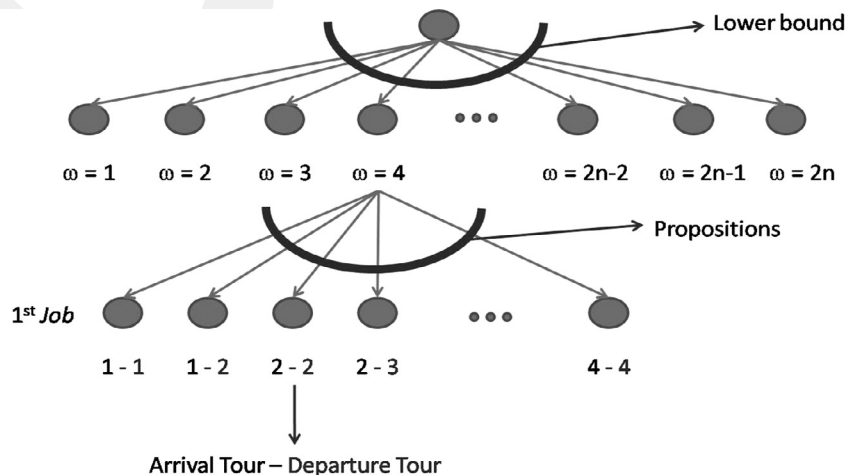


Fig. 2. An illustration of the search tree.

feasible solutions set upper bounds on the minimum cost. Therefore, if a lower bound at any node in upcoming steps of the search exceeds the smallest upper bound, then this node is pruned. The nodes at the first level of each subtree (i.e., the second level of the main search tree) store partial solutions incorporating the possible assignments of the job with the longest processing time to a block. In general, at level  $i$  ( $i = 1, \dots, |\mathcal{N}|$ ) of a subtree, an assignment of the  $i$ th longest job to a block is made. We would like to note that in assigning jobs to blocks, two issues are taken into account. First, the vehicle capacity constraints should not be exceeded. Secondly, the waiting time of a vehicle at the facility should be less than or equal to the limit  $l$ . When a new assignment is made to a block, the sequence of the jobs in that block are updated using Proposition 5, and if the new sequence improves the lower bound, it is revised based on the underlying approach of Theorems 2, 3 and their proofs.

The search for a solution conditioning on a  $\omega$  value, evolves using the following approach recursively at each level of the corresponding subtree: All the children nodes are created and their corresponding lower bounds are updated based on the partial solutions they carry. The children nodes with lower bounds greater than or equal to the objective value of the best known solution are eliminated. Remaining partial solutions in the promising nodes are then rapidly completed to a full solution. The completion algorithm is simply scheduling the next job to the position where the lower bound is minimum. The value of the global evaluation function for each child is the objective function value of the completed solution. The children nodes are then sorted according to the global evaluation function values. If a completed solution has a better objective value than the best known solution, the smallest upper bound is updated. When all the nodes at the current level are examined, the most promising beam-width number of them are chosen for further exploration. At this point, since more than one child node originating from the same parent node can be kept for further consideration, the proposed method constitutes a dependent beam search.

After all jobs are assigned to blocks, the assignments are converted to a schedule in terms of the arrival and departure times of vehicles, and start and end times of processing. As an example of such an assignment and how it is converted to a schedule, consider the illustrative representation in Fig. 3. There are 5 jobs with the following processing times:  $p_1 = 1$ ,  $p_2 = 2$ ,  $p_3 = 3$ ,  $p_4 = 4$  and  $p_5 = 5$ . Assume also that  $h_j^1 = h^1$  and  $h_j^2 = h^2$  for  $j = 1, \dots, 5$ . The jobs are assigned to 3 blocks, which implies that the number of tours is 2. Jobs 4 and 1 arrive at and depart from the facility with the same tour. Job 5 reaches to the facility with the same tour as of jobs 4 and 1, but it leaves the facility with the second tour. Jobs 3 and 2 arrive at and depart from the facility with the second tour. Fig. 3 also shows the sequence of processing among the jobs that are in the same block. That is, job 4 is processed before job 1, and job 3 is processed before job 2. Proposition 5 hints that in this example  $h^1 < h^2$ .

Let us first assume that there are 2 vehicles (i.e.,  $m = 2$ ), tour time is 5 units (i.e.,  $\tau = 5$ ), and waiting time limit is 5 (i.e.,  $l = 5$ ). First, the tours are assigned to vehicles. Vehicle 1 makes the odd numbered tours (1, 3, 5, ...) and vehicle 2 makes the even numbered tours (2, 4, 6, ...). Since there are only two tours, each vehicle makes a single tour. Jobs 4, 1 and 5 arrive at time 0 with vehicle 1, and the vehicle waits at the facility until the processing of job 1 finishes. At time 5, vehicle 1 departs from the facility with jobs

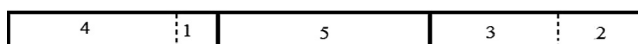


Fig. 3. An illustration of block assignments to jobs.

4 and 1. Job 5 is then processed until time 10. Vehicle 2 arrives at the facility with jobs 3 and 2 at time 10, and the processing of job 3 starts immediately. Job 2 follows job 3 starting at time 13 and jobs 5, 3 and 2 depart from the facility with vehicle 2 at time 15.

For the same assignment illustrated in Fig. 3, now assume that only the number of vehicles and the tour time attain different values, those are  $m = 1$  and  $\tau = 10$ . In this case, job 5 waits an extra 5 time units for the return of the vehicle and there is an inserted idleness in the production schedule in front of job 5. Since  $h^1 < h^2$ , idleness is inserted before job 5, otherwise the job has to wait for 5 time units after its processing is completed.

We close this section by noting that beam search is an approach that has been successfully used to solve various complex scheduling problems. We cite Erenay, Sabuncuoglu, Toptal, and Tiwari (2010) and Sabuncuoglu and Karabuk (1998) as examples of beam-search applications in the scheduling area.

## 5. Computational experiments

In this section, we discuss the design and the results of our numerical analysis. The objectives of this analysis are: (i) to test how the lower bounds affect the running time of the optimization model presented in Section 2, (ii) to assess the tightness of the lower bounds, and (iii) to evaluate the quality of the proposed heuristic.

With the above objectives in mind, we take the following parameters as factors of analysis: inventory holding costs of unprocessed jobs ( $h_i^1$ ), waiting-time limit of vehicles at the facility ( $l$ ), vehicle capacity ( $K$ ), number of vehicles ( $m$ ), tour time ( $\tau$ ), tour cost ( $c$ ), and number of jobs ( $|\mathcal{N}|$ ). In generating data, we follow the guide provided by Hall and Posner (2001). The different levels of the parameters used in the experimental analysis are summarized in Table 1. As a result of generating 10 instances for each combination of factor levels, 11,520 instances have been solved.

The number of jobs in an instance is taken from the set  $\{5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200\}$  and is referred to as the size of the problem. The processing times of the jobs are sampled from a discrete uniform distribution  $U[1, 100]$ . The space that a job occupies in a vehicle in the inbound and outbound, and the inventory holding cost that it incurs at the facility before and after processing are generated based on its processing time. More specifically, we set  $s_i^1 = p_i + U[0, 20]$  and  $s_i^2 = p_i + U[0, 20]$  where the second term in both expressions is a uniformly distributed random variable between 0 and 20. The inventory holding costs of all jobs in an instance before their processing start, take two levels; zero or positive. That is, either  $h_i^1 = 0$  for all jobs  $i$  in an instance, or  $h_i^1 = p_i + U[10, 50]$ . The value of an item is expected to increase as it moves down in the supply chain. This, in turn, leads to an increase in inventory holding costs. In our computational analysis,

Table 1  
Parameter settings.

Parameter	Levels
$p_i$	$U[1, 100]$
$s_i^1$	$p_i + U[0, 20]$
$s_i^2$	$p_i + U[0, 20]$
$h_i^1$	$0, p_i + U[10, 50]$
$h_i^2$	$p_i + U[10, 50] + U[10, 50]$
$l$	0, 101, 5000
$K$	182, 364
$m$	1, 3
$\tau$	51, 153
$c$	10,000, 40,000

$h_i^2$  values are generated to be greater than  $h_i^1$  by setting  $h_i^2 = p_i + U[10, 50] + U[10, 50]$ . Note that, this implies we have  $h_i^2 = h_i^1 + U[10, 50]$  when  $h_i^1 > 0$  for all jobs.

The maximum time that a vehicle can wait at the facility (i.e.,  $l$ ) assumes one of the following three values: 0, 101 and 5000. The first and the third values represent extreme cases in which vehicles are either not allowed to wait at all or the waiting-time limit is not constraining. In between these two extremes, the waiting-time limit is set to  $2E[p_i] = 101$ , meaning that a vehicle can wait at most two jobs on the average. In the experimental analysis,  $K$  takes two levels: low and high. In the low capacity case (i.e.,  $K = 182$ ), the size of a vehicle is enough to carry three jobs on the average ( $3E[s_i^1] = 3E[s_i^2] = 3E[p_i] + 30 \approx 182$ ). In the high capacity case ( $K = 364$ ), the size of a vehicle is large enough to carry six jobs on the average.

We consider two levels for the number of vehicles, those are  $m = 1$  and  $m = 3$ . The tour time of a vehicle also takes two levels: low ( $\tau = 51$ ) and high ( $\tau = 153$ ). The low level represents a situation where the average processing time of a job can barely be completed within the tour time. The high level represents a situation where the processing of three jobs, on the average, can be completed within the tour time. The last factor in our analysis, that is the tour cost, assumes two values; those are 10,000 and 40,000.

All the computational experiments have been carried out on a 2.6 GHz 8 x Intel Xeon E5430 Server running Debian Lenny (5.0.7) with 8 GBs of physical memory. GAMS version 22.6 has

been used to solve the mixed integer programming formulation of the problem.

5.1. The effects of the lower bounds on the computational time

The integer programming models provided in Section 2 can only be used to solve small size problems. This is due to the slow progress of the LP relaxations through the branch and bound tree. The progress through the branch and bound tree can be improved based on the lower bounds provided in Corollaries 1 and 2. Our objective in this section is to test the effects of the results provided in these corollaries on the computational time of the integer programming formulation, under different problem parameters.

In order to see the effects of the lower bounds, all instances are solved using the following two models:

Model I: Linearized version of the integer programming formulation presented in Section 2.

Model II: Linearized version of the integer programming formulation with the incorporation of the lower bounds.

In Model II, the following set of constraints are included in the formulation to employ the lower bounding scheme:

$$\sum_{i \in \mathcal{N}} h_i^1 (\sigma_i - \alpha_i) + \sum_{i \in \mathcal{N}} h_i^2 (\delta_i - (\sigma_i + p_i)) \geq (\psi_t - \psi_{t+1}) * LB(t)$$

$$t = 1, 2, \dots, 2|\mathcal{N}| - 1,$$

Table 2 Comparison of the computational times of the two models (CPU seconds).

			1	2	3	4	5	6
			$h_i^1 = 0$ $l = 5000$	$h_i^1 = 0$ $l = 101$	$h_i^1 = 0$ $l = 0$	$h_i^1 > 0$ $l = 5000$	$h_i^1 > 0$ $l = 101$	$h_i^1 > 0$ $l = 0$
1	C = 10,000	$m = 1$	1.52	0.81	1.67	5.43	4.29	9.54
	K = 182	$\tau = 51$	0.83	0.74	0.89	5.39	2.68	4.43
2	C = 10,000	$m = 1$	2.59	1.13	3.30	14.15	5.43	14.92
	K = 364	$\tau = 51$	1.73	0.65	1.85	12.29	3.93	6.32
3	C = 10,000	$m = 3$	0.95	1.38	1.40	25.89	8.91	11.75
	K = 182	$\tau = 51$	0.46	1.00	1.16	19.76	7.50	7.73
4	C = 10,000	$m = 3$	2.62	1.16	3.93	82.72	9.20	21.12
	K = 364	$\tau = 51$	1.93	1.19	2.16	55.64	6.53	13.36
5	C = 40,000	$m = 1$	1.09	0.72	1.27	1.94	1.47	2.22
	K = 182	$\tau = 51$	0.72	0.64	0.58	1.90	1.00	2.14
6	C = 40,000	$m = 1$	0.35	0.35	0.31	1.68	0.95	2.27
	K = 364	$\tau = 51$	0.35	0.31	0.42	1.27	1.52	1.48
7	C = 40,000	$m = 3$	0.90	1.23	1.58	5.40	2.87	2.86
	K = 182	$\tau = 51$	0.54	0.85	1.23	3.18	2.09	3.49
8	C = 40,000	$m = 3$	0.28	0.34	0.37	10.96	1.26	2.91
	K = 364	$\tau = 51$	0.41	0.32	0.36	3.43	1.44	1.23
9	C = 10,000	$m = 1$	1.23	0.91	1.04	3.69	2.66	35.70
	K = 182	$\tau = 153$	0.67	0.80	0.63	4.16	2.85	0.79
10	C = 10,000	$m = 1$	2.45	1.20	3.29	10.45	5.17	59.11
	K = 364	$\tau = 153$	1.41	0.81	1.28	11.38	3.20	4.02
11	C = 10,000	$m = 3$	0.90	1.39	1.59	31.43	10.60	9.16
	K = 182	$\tau = 153$	0.59	1.09	1.07	25.11	6.46	8.45
12	C = 10,000	$m = 3$	3.01	1.86	3.65	112.29	9.10	21.43
	K = 364	$\tau = 153$	1.87	1.07	2.75	37.05	8.21	11.04
13	C = 40,000	$m = 1$	0.81	0.78	0.64	1.13	1.13	1.22
	K = 182	$\tau = 153$	0.58	0.67	0.68	1.15	1.15	0.58
14	C = 40,000	$m = 1$	0.31	0.39	0.31	0.98	2.01	0.74
	K = 364	$\tau = 153$	0.35	0.29	0.38	1.71	1.72	0.89
15	C = 40,000	$m = 3$	0.62	1.14	1.11	6.14	3.18	2.32
	K = 182	$\tau = 153$	0.59	0.99	1.13	6.33	3.00	2.87
16	C = 40,000	$m = 3$	0.35	0.38	0.38	12.41	1.08	2.71
	K = 364	$\tau = 153$	0.38	0.28	0.44	7.60	1.38	1.65

where  $LB(t) = \max(LB'(t), LB''(t))$ . The left hand side of the inequality is the total inventory holding cost. If the number of tours is  $t$ , then  $(\psi_t - \psi_{t+1}) = 1$  and the total inventory holding cost is bounded from below by  $LB(t)$ .

Table 2 presents the average solution times over 10 instances, each with 5 jobs, for different combinations of the remaining parameter values. The rows and the columns of the table correspond to different settings of  $K, \tau, m, C$  and  $h_i^1, l$  respectively. There are two values in each cell. The first value is the average time spent in CPU seconds to solve Model I and the second value is the average time to solve Model II. As can be seen in the table, in general (i.e., in more than 75% of the cells), the second value is smaller than the first one. This indicates that the lower bounds save from computational time. In some cases, the computational time that Model II requires is more than that of Model I. This may be due to the possibility that, in small problems, the additional computational burden of processing extra constraints for lower bounds does not justify their benefits. Another observation is that, if tour times and tour costs are high, the effects of the lower bounds diminish (see rows 13–16). Finally, our results imply that, with or without the lower bounds, solving the problem is easier in settings where  $h_i^1 = 0$  (see columns 1, 2 and 3).

Our findings in this section imply that, the lower bounds are quite effective in decreasing the running time of the model presented in Section 2.

5.2. Quality of the lower bound and the heuristic with respect to the optimal solution

In this section, we discuss our findings on the qualities of the proposed lower bound and the heuristic. For this purpose, we compare the lower bound and the cost of the heuristic solution to the optimal objective function value. Due to the computational difficulty in obtaining optimal solutions for larger size problems, we base our analysis in this section on 5-job instances.

Table 3 presents a summary of the results. The values of  $m, \tau, K, C$  are changed over the rows and the values of  $h_i^1, l$  are changed over the columns. In each cell, two statistics are reported based on 10 instances. The first statistic is the average gap between the optimal objective function value and the lower bound. This corresponds to the average percentage by which the lower bound is smaller than the optimal cost. The second value is the average gap between the heuristic solution and the optimal objective function value. Similarly, this corresponds to the average percentage by which the cost of the heuristic solution is higher than the optimal cost.

Considering all the 960 instances, those are the ones with 5 jobs, we find that the average deviation of the lower bound from the optimal solution amounts to 9.5% of the latter. The maximum gap over all the instances is 38.1%. Comparing columns 1–3 of Table 3 to columns 4–6, respectively, we observe that when raw

**Table 3**  
Average percentage gaps of the lower bound and the cost of heuristic solution.

			1	2	3	4	5	6
			$h_i^1 = 0$ $l = 5000$	$h_i^1 = 0$ $l = 101$	$h_i^1 = 0$ $l = 0$	$h_i^1 > 0$ $l = 5000$	$h_i^1 > 0$ $l = 101$	$h_i^1 > 0$ $l = 0$
1	$C = 10,000$ $K = 182$	$m = 1$ $\tau = 51$	5.76% 0.00%	8.32% 0.19%	4.45% 0.00%	12.36% 0.00%	12.92% 0.00%	9.93% 0.00%
2	$C = 10,000$ $K = 364$	$m = 1$ $\tau = 51$	10.34% 0.07%	10.80% 0.00%	7.98% 0.00%	12.76% 0.00%	13.32% 0.00%	10.24% 0.00%
3	$C = 10,000$ $K = 182$	$m = 3$ $\tau = 51$	5.76% 0.00%	8.32% 0.19%	4.45% 0.00%	9.66% 3.13%	11.62% 1.54%	9.86% 0.00%
4	$C = 10,000$ $K = 364$	$m = 3$ $\tau = 51$	10.34% 0.07%	10.80% 0.00%	7.98% 0.00%	10.08% 3.13%	12.03% 1.54%	10.18% 0.00%
5	$C = 40,000$ $K = 182$	$m = 1$ $\tau = 51$	5.36% 0.00%	7.81% 0.06%	4.20% 0.00%	7.77% 0.00%	9.99% 0.09%	5.99% 0.00%
6	$C = 40,000$ $K = 364$	$m = 1$ $\tau = 51$	7.09% 0.00%	17.68% 0.09%	4.80% 0.00%	15.81% 0.00%	20.57% 0.08%	11.62% 0.00%
7	$C = 40,000$ $K = 182$	$m = 3$ $\tau = 51$	5.36% 0.00%	7.81% 0.06%	4.20% 0.00%	6.64% 1.22%	9.44% 0.45%	5.99% 0.00%
8	$C = 40,000$ $K = 364$	$m = 3$ $\tau = 51$	7.09% 0.00%	17.68% 0.09%	4.80% 0.00%	14.41% 1.63%	20.42% 0.29%	11.62% 0.00%
9	$C = 10,000$ $K = 182$	$m = 1$ $\tau = 153$	5.76% 0.00%	8.32% 0.19%	4.45% 0.00%	12.36% 0.00%	12.92% 0.00%	2.25% 0.00%
10	$C = 10,000$ $K = 364$	$m = 1$ $\tau = 153$	10.34% 0.07%	10.80% 0.00%	7.98% 0.00%	12.76% 0.00%	13.32% 0.00%	5.45% 0.01%
11	$C = 10,000$ $K = 182$	$m = 3$ $\tau = 153$	5.76% 0.00%	8.32% 0.19%	4.45% 0.00%	9.66% 3.13%	11.62% 1.54%	9.86% 0.00%
12	$C = 10,000$ $K = 364$	$m = 3$ $\tau = 153$	10.34% 0.07%	10.80% 0.00%	7.98% 0.00%	10.08% 3.13%	12.03% 1.54%	10.18% 0.00%
13	$C = 40,000$ $K = 182$	$m = 1$ $\tau = 153$	5.36% 0.00%	7.81% 0.06%	4.20% 0.00%	7.77% 0.00%	12.07% 0.28%	3.36% 0.00%
14	$C = 40,000$ $K = 364$	$m = 1$ $\tau = 153$	7.09% 0.00%	17.68% 0.09%	4.80% 0.00%	15.96% 0.00%	24.74% 0.16%	3.57% 0.00%
15	$C = 40,000$ $K = 182$	$m = 3$ $\tau = 153$	5.36% 0.00%	7.81% 0.06%	4.20% 0.00%	6.64% 1.22%	9.44% 0.45%	5.99% 0.00%
16	$C = 40,000$ $K = 364$	$m = 3$ $\tau = 153$	7.09% 0.00%	17.68% 0.09%	4.80% 0.00%	14.41% 1.63%	20.42% 0.29%	11.62% 0.00%

material inventory holding cost  $h_i^1$  is zero, the lower bound is tighter in general. In fact, over all the instances with  $h_i^1 = 0$ , the average gap is 7.88%, whereas, it is 11.12% over the remaining instances where  $h_i^1 > 0$ . This is due to the way that the lower bound is constructed. Recall from Section 3.1 that, the lower bounds are computed based on minimization of costs assuming momentarily that either  $h_i^1 = 0$  or  $h_i^2 = 0$  for all  $i$ . Therefore, the lower bounds are tighter in these cases.

It can also be observed from Table 3 that the average percentage differences between the lower bound and the optimal costs in column 2 are higher than those in columns 1 and 3. Similarly, the same measures in column 5 are higher than those in columns 4 and 6. This implies that performance of the lower bound is the worst when the limit on the waiting time of vehicles, i.e.  $l$ , takes its mid value. In extreme cases, those are when  $l$  is very large or very small, lower bounds are closer to optimal costs. On a comparison of row 1 with row 3, row 2 with row 4, and so forth, we also conclude that the performance of the lower bounds increases with increasing number of vehicles. Comparing row 1 with row 2, row 3 with row 4, and so forth, we observe that the lower bounds get closer to optimal costs with decreasing vehicle capacity. Finally, we would like to point out that when  $h_i^1 > 0$ ,  $m = 1$  and  $\tau = 153$  (see the intersection of column 6 with rows 9, 10, 13, 14), the lower bound is particularly close to the optimal solution. What differentiates these instances from others is that, in finding a lower bound, the part of Expression (20) that concerns inventory holding costs is given by  $LB'_i(\omega)$  for some value of  $w$  rather than  $LB_i(\omega)$ .

In this section, we also report our results in terms of the quality of the beam-search heuristic with respect to the optimal solution for small-size problems. In the next section, we present the results of a more extensive study incorporating larger size problems. In order to decide the beam width parameter, pilot runs were taken on sample instances of all sizes. The objective function values of the heuristic solutions and CPU times spent for several beam width values, were recorded. It was observed that the solution time increases almost linearly as the beam width increases. However, the objective function value does not change for beam width values greater than 8. Furthermore, the marginal contribution of increasing the beam width beyond a value of 5, does not justify

the increase in the computational time. Thus, we decided to fix the beam width at a value of 5 in the remaining part of the analysis.

The second value in each cell of Table 3 is the average gap between the heuristic and the optimal solution over 10 instances. Considering all the instances with 5 jobs, the maximum and average gaps were found as 6.50% and 0.29%, respectively. On a comparison of the first three columns of Table 3 with the last three columns, it can be observed that the heuristic performs better when  $h_i^1 = 0$ . In fact, the average gaps for cases with  $h_i^1 = 0$  and  $h_i^1 > 0$  are 0.034% and 0.55%, respectively. This may be due to the fact that the heuristic makes use of the lower bounds, and the lower bounds perform slightly better in these cases. Finally, when rows with  $m = 1$  are compared to rows with  $m = 3$ , it can be concluded that the performance of the heuristic decreases with the number of vehicles. The average gaps for single and multiple vehicle cases are 0.03% and 0.555%, respectively.

### 5.3. Quality of the heuristic for large size problems

In this section, the quality of the proposed heuristic is assessed in comparison to the lower bounds and over an extensive set of problems.

We first start with analyzing the effect of inventory holding costs on the performance of the heuristic. Fig. 4 shows the average percentage difference between the heuristic solution and the lower bound for varying levels of the job size and  $h_i^1$ . Recall from Section 5.2 that the average gap between the costs of the heuristic and optimal solution was found as 0.29% over 5-job problems. However, if the costs of the heuristic solutions are compared to the lower bounds over the same set of instances, it is found that the lower bound is less than the cost of the heuristic solution by an average of 9.36% and 14.11% of the latter for the cases of  $h_i^1 = 0$  and  $h_i^1 > 0$ , respectively. Thus, a significant portion of the gap when heuristic solution is tested against the lower bound, should be attributed to the difference between the lower bound and the optimal solution. Fig. 4 shows that the instances with  $h_i^1 > 0$  lead to a larger average deviation of the heuristic from the lower bound in comparison to the instances with  $h_i^1 = 0$ .

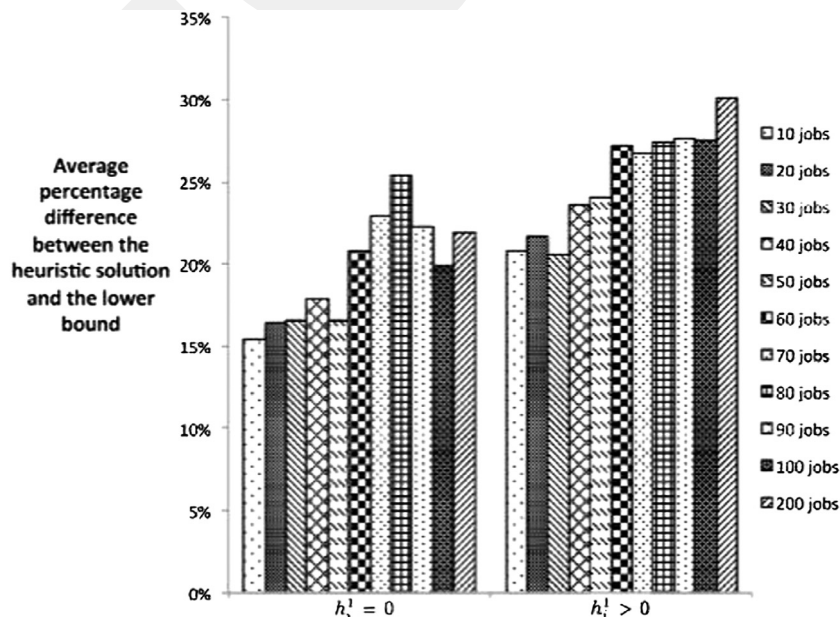


Fig. 4. Effect of inventory holding costs on the heuristic performance for different problem sizes.

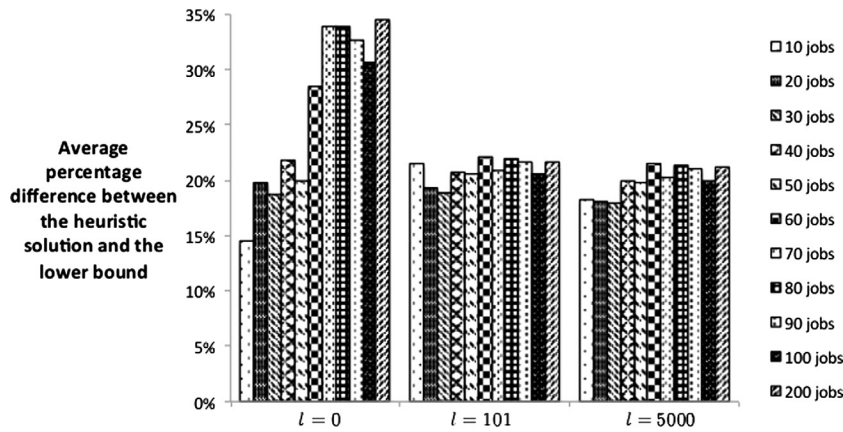


Fig. 5. Effect of waiting limit on the heuristic performance for different problem sizes.

Fig. 5 shows how the performance of the heuristic changes with respect to waiting time limit  $l$  at varying levels of job size. The average deviation of the cost of heuristic solution from the lower bound is the largest at mid levels of the waiting time limit for problems with up to 50 jobs. Recall from Section 5.2 that, this is also when the performance of the lower bound is the worst. For problems with 60 or more jobs, the average deviation is the largest at low level of the waiting time limit.

The effect of vehicle capacities on the performance of the heuristic is demonstrated in Fig. 6. When tour cost  $c$  is low, the capacities have little effect because the vehicles are not always fully utilized. When tour cost is high, vehicles are fully utilized to decrease the total number of tours. In this case, vehicle capacities become more constraining. The heuristic performs better at higher values of tour cost combined with small vehicle capacities.

Considering all 10,560 instances (all instances except for 5-job problems), we found that the cost of the heuristic solution deviates from the lower bound by 19% on the average for smaller problems (those with 10–50 jobs) and by 29% on the average for larger problems (those with 60–200 jobs). We would like to complete this section with some results on the running time of the heuristic. Table 4 reports the average running time of the heuristic over all the instances for each job size. A regression equation with the job size ( $n$ ) being the independent variable, implies that the running time can be explained by a third degree polynomial ( $0.0004n^3 - 0.0288n^2 + 0.55n$ ) with an  $R$ -squared value of 0.9977.

## 6. Conclusions and future research

This paper studies the joint problem of finding the production and vehicle schedules for inbound and outbound transportation of a single stage in the supply chain. In the specific setting of interest, a certain number of jobs are carried from an origin to a production facility at a distant location and returned back to the origin after their processing. There are multiple vehicles with limited capacities and they can be utilized for both inbound and outbound transportation. Inventory holding costs and transportation costs in this setting are high, therefore, coordination of the schedules for production and transportation is important.

Our paper falls into the area of supply chain scheduling with transportation considerations. While many of the studies in this area focus on just the delivery schedule and consider the joint scheduling problem for a scheduling related objective, our study models the shipment related constraints both in the inbound and the outbound, and aims to minimize the sum of inventory holding and transportation costs. In the paper, we first show that the problem under consideration is NP-Hard in the strong sense. We then prove some properties of the solution space and develop lower bounds on the optimal objective function value. Using these properties and lower bounds, we propose a heuristic based on beam-search approach. Over an extensive computational analysis, we demonstrate the qualities of the lower bounds and the heuristic. We would like to note that our model does not explicitly account

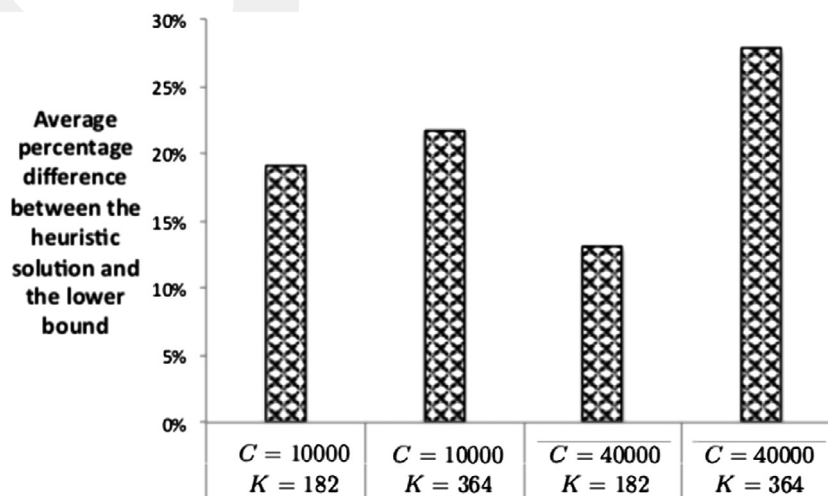


Fig. 6. Effect of tour cost and vehicle capacity on the heuristic performance.

**Table 4**  
Average running time of the heuristic in seconds.

Job size	Time spent
10	0.0121
20	0.4141
30	0.8727
40	2.9344
50	6.8372
60	16.7715
70	33.7393
80	60.6723
90	100.0391
100	166.5684
200	4012.2601

for deadlines. However, under a situation where jobs have common deadlines all our analytical results hold, and the proposed heuristic can easily be updated by assigning a job to a block only if it does not lead to deadline violation for any of the affected jobs in the current schedule.

The issue of coordinating the schedules for the production and a finite number of capacitated vehicles which can be utilized both in the inbound and the outbound, can be extended to other settings as well. Immediate extensions include modeling the production scheduling problem at a more detailed level and/or solving the problem for different objective functions. Last, but not least, conflict and cooperation issues can be investigated in this setting by modeling the existence of a decision maker, i.e., a trucking company, who owns the trucks and makes their scheduling decisions (see Dawande et al., 2006 as an example).

## Appendix A

### A.1. Proof of Theorem 1

In the proof we consider the special case in which there is a single vehicle, all jobs occupy the same amount of space within a vehicle (i.e.,  $s_i^1 = s^1$  and  $s_i^2 = s^2$  for all  $i \in \mathcal{N}$ ) and inventory holding costs of all jobs are equal (i.e.,  $h_i^1 = h^1$ ,  $h_i^2 = h^2$  for all  $i \in \mathcal{N}$ ). Clearly the generalization is also NP – Complete. Proof is done by a reduction from 3-Partition (3P) problem. 3P is defined as follows.

3P: Given a set  $\mathcal{A}$  of  $3t$  elements, a bound  $B \in \mathbb{Z}^+$ , and a size  $s(a) \in \mathbb{Z}^+$  for each  $a \in \mathcal{A}$  such that  $B/4 < s(a) < B/2$  and such that  $\sum_{a \in \mathcal{A}} s(a) = tB$ , can  $\mathcal{A}$  be partitioned into  $t$  disjoint sets  $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_t$  such that  $\sum_{a \in \mathcal{A}_i} s(a) = B$  for  $i = 1, 2, \dots, t$  (note that each  $\mathcal{A}_i$  must therefore contain exactly three elements from  $\mathcal{A}$ )?

REDUCTION: Given an instance of 3P, the instance of  $\mathcal{P}$  is constructed as follows: for each element  $a$  in set  $\mathcal{A}$ , a job  $a$  is defined in set  $\mathcal{N}$  with processing time equal to  $s(a)$ . Thus,  $\mathcal{N} = \mathcal{A}$ ,  $|\mathcal{N}| = 3t$ ,  $p_a = s(a)$ ,  $s^1 = s^2 = h^1 = h^2 = 1 \forall a \in \mathcal{A}$ ,  $\tau = B$ ,  $c = 4tB$ ,  $z^* = (t + 1)c + \frac{c}{2}$ ,  $K = 3$ ,  $l = 0$ . We prove that there is a solution to 3P if and only if there is a solution to  $\mathcal{P}$  with objective less than or equal to  $z^*$ .

Suppose that there is a feasible solution to  $\mathcal{P}$  such that the cost  $z$  is less than or equal to  $z^*$ . We show that there also exists a feasible solution to 3P. Since  $l = 0$ , the vehicle is not allowed to wait at the facility. Therefore, the first tour departs from the facility empty. As  $K = 3$ , the vehicle makes at least  $t + 1$  tours, with a transportation cost of  $c(t + 1)$ . Since  $z \leq z^* < c(t + 2)$ , the vehicle makes exactly  $t + 1$  tours. Therefore, tour  $i$  ( $i = 1, \dots, t$ ) carries exactly 3 jobs (whose total processing times is denoted by  $\tilde{p}_i$ ) to the facility, which should be processed by the time when the next arrival of the vehicle. At tour  $i$ , whatever the processing sequence is, the inventory holding cost incurred is at least  $2\tilde{p}_i$ . This is because, each job waits for the other two either after or before being processed and  $h^1 = h^2 = 1$ . The total inventory holding cost is at least

$2\sum_{i=1}^t \tilde{p}_i = 2\sum_{a \in \mathcal{A}} p_a = 2tB = c/2$ , which means  $z = z^*$ , which in turn implies the total inventory holding cost is exactly  $c/2$ . Note that  $\tilde{p}_i \leq \tau$ ,  $\forall i$ . Otherwise, there would be an extra inventory holding cost incurred by all three jobs waiting after or before being processed.  $\sum_{i=1}^t \tilde{p}_i = tB$ , thus, we should have  $\tilde{p}_i = \tau$ ,  $\forall i$ . Then, one can obtain a feasible solution to 3P by taking  $\mathcal{A}_i$  as the set which includes the processing times of the jobs arriving with tour  $i$ . Conversely, if there exists a feasible solution to 3P, a feasible solution to  $\mathcal{P}$  can be obtained by assigning the jobs whose processing times are the numbers in  $\mathcal{A}_i$  to arrive with tour  $i$  and depart with tour  $(i + 1)$ . Note that the parameter settings in the reduction are polynomial in the size of the problem. Consequently, decision version of  $\mathcal{P}$  is NP – Complete in the strong sense. □

### A.2. Proof of Theorem 2

Total inventory holding costs are composed of inventory holding costs for unprocessed jobs and processed jobs. For the proof of the theorem, we will first find lower bounds individually for each component, and later, we will sum them up. In reaching a lower bound for unprocessed jobs, we will ignore the effect of any scheduling decision on the inventory holding costs of the processed jobs. This is equivalent to momentarily assuming that  $h_i^2 = 0$  for all  $i \in \mathcal{N}$ . Likewise, in deriving a lower bound for processed jobs, we will assume that  $h_i^1 = 0$  for all  $i \in \mathcal{N}$ .

Let us start with the inventory holding costs of the unprocessed jobs. As the cost of unprocessed jobs is positive but  $h_i^2 = 0$  for all  $i \in \mathcal{N}$ , in an optimal solution, the production facility will never be idle as long as there is some job waiting to be processed. Therefore, the inventory holding costs of unprocessed jobs are given by  $\sum_{i \in \mathcal{N}} \sum_{j \in \mu_i^1} h_j^1 p_i$ , where  $\mu_i^1$  is the set of jobs that wait for job  $i$  as unprocessed. Since there are  $\omega$  tours, we have at most  $\omega$  jobs with  $\mu_i^1 = \emptyset$ , at most  $\omega$  jobs with  $|\mu_i^1| = 1$ , and so on. The expression  $\sum_{i \in \mathcal{N}} \sum_{j \in \mu_i^1} h_j^1 p_i$  is minimized when the jobs with longer processing times have smaller  $\sum_{j \in \mu_i^1} h_j^1$  values. That is, when the longest  $\omega$  jobs are chosen to have  $\mu_i^1 = \emptyset$ , the next longest  $\omega$  jobs are chosen to have  $|\mu_i^1| = 1$ , and so on. Moreover, we distribute  $h_j^1$ 's to the sets  $\mu_i^1$  so that the smallest  $h_j^1$  is assigned to the longest job among all with  $|\mu_i^1| = 1$ , the second smallest  $h_j^1$  is assigned to the second longest, and so on. That is, the  $\omega$  jobs with smallest  $h_j^1$  values are distributed to the jobs with  $|\mu_i^1| = 1$ . For the jobs with  $|\mu_i^1| > 1$ , we assign one job from the smallest  $\omega$   $h_j^1$ 's, one job from the second smallest  $h_j^1$ 's, and so on. This leads to

$$\sum_{i \in \mathcal{N}} \sum_{j \in \mu_i^1} h_j^1 p_i \geq \sum_{i=1}^{\lfloor \frac{|\mathcal{N}|}{\omega} \rfloor} p_{(i)} \sum_{k=1}^{\lfloor \frac{|\mathcal{N}|}{\omega} \rfloor} \left( h_{(|\mathcal{N}| + \omega k + 1 - i)}^1 \right) \quad (21)$$

where  $\lfloor \frac{|\mathcal{N}|}{\omega} \rfloor$  represents  $|\mu_{(i)}^1|$ ,  $p_{(i)}$  is the  $i$ th longest processing time and  $h_{(j)}^1$  is the  $j$ th largest value among all  $h_i^1$ 's. Hence, the right side of the above inequality is a lower bound on the inventory holding costs of unprocessed jobs.

A lower bound on the inventory holding costs of the processed jobs can be derived in a similar way. Let  $\mu_i^2$  be the set of jobs that wait for job  $i$  as processed. Then, the inventory holding costs of the processed jobs are given by  $\sum_{i \in \mathcal{N}} \sum_{j \in \mu_i^2} h_j^2 p_i$ . With a similar argument as in the case of unprocessed jobs, we have

$$\sum_{i \in \mathcal{N}} \sum_{j \in \mu_i^2} h_j^2 p_i \geq \sum_{i=1}^{\lfloor \frac{|\mathcal{N}|}{\omega} \rfloor} p_{(i)} \sum_{k=1}^{\lfloor \frac{|\mathcal{N}|}{\omega} \rfloor} \left( h_{(|\mathcal{N}| + \omega k + 1 - i)}^2 \right)$$

where  $\lfloor \frac{i-1}{\omega} \rfloor$  represents  $\lfloor \mu_{(i)}^2 \rfloor$ ,  $p_{(i)}$  is the length of the  $i$ th longest job and  $h_{(j)}^2$  is the  $j$ th largest value among all  $h_i^2$ 's. The right side of the above inequality is a lower bound on the inventory holding costs of the processed jobs. Therefore, its summation with the right side of Inequality (21) gives a lower bound on the total inventory holding costs for a given value of number of tours (i.e.,  $\omega$ ).  $\square$

### A.3. Proof of Theorem 3

The proof of Theorem 3 follows based on a similar idea which underlies the proof of Theorem 2. In general, a job may contribute to the total inventory holding costs in two ways; one is due to the waiting of the job for its delivery to the warehouse until the departure of next available vehicle (it may wait processed or unprocessed), and the other is the inventory holding cost of a job while it waits for the processing of the other jobs. Note that some of these waiting times may overlap. Theorem 2 and its proof build on a consideration of the second cause for waiting of any job. Herein, we will also take into account the waiting of jobs for their pickup until a vehicle becomes available. Notice that, this is easier to do in case of one vehicle, because in this case, we know that the time between the drop-off and pick-up of a job, if  $\tau > p_j > l$ , is at least  $\tau$ . The remaining part of the proof relies on this observation and accounts for the two reasons of waiting.

If  $\tau > p_j > l$  for some job  $j$ , the job has to wait for the return of the vehicle as long as at least  $\tau - p_j$  time units. Ignoring other jobs at the facility momentarily, if  $h_j^1 < h_j^2$  for some job  $j$ , the inventory holding cost due to the waiting of this job for the return of the vehicle can be minimized if the job is held unprocessed during its waiting time. That is, the job contributes to the total inventory holding costs in an amount of at least  $h_j^1(\tau - p_j)$ . If  $h_j^2 < h_j^1$ , the job's contribution to the total inventory holding costs is decreased if it is held processed. This, in turn, leads to an inventory holding cost of at least  $h_j^2(\tau - p_j)$ . Thus, the inventory holding cost incurred by this job due to the first reason is at least  $(\tau - p_j)\min(h_j^1, h_j^2)$ , and this is valid for all jobs for which  $\tau > p_j > l$ .

Note that summing up  $(\tau - p_j)\min(h_j^1, h_j^2)$  for all jobs, we already include the waiting time of a job either in its unprocessed or processed state. Recall that Theorem 2 proposes  $\sum_{i=1}^{\lfloor \frac{N}{\omega} \rfloor} p_{(i)} \sum_{k=1}^{\lfloor \frac{i-1}{\omega} \rfloor} (h_{(\lfloor N \rfloor + \omega k + 1 - i)}^1 + h_{(\lfloor N \rfloor + \omega k + 1 - i)}^2)$  as a lower bound on inventory holding costs due to the waiting of the jobs for one another. The cost of waiting due to the vehicle unavailability is incorporated in the above calculations by considering a job's state at which the inventory holding cost rate is minimum. Therefore, the waiting of jobs in their minimum cost state is already penalized. To that, we add the term  $\sum_{k=1}^{\lfloor \frac{i-1}{\omega} \rfloor} p_{(i)} h'_{(\lfloor N \rfloor + \omega k + 1 - i)}$  for each job to account for the incremental cost of waiting of jobs for one another, which has not been incorporated in the  $(\tau - p_j)\min(h_j^1, h_j^2)$  term.  $\square$

### A.4. Proof of Proposition 1

Let  $S$  be a feasible solution such that job  $i$  precedes job  $j$  in the production sequence but arrives at the facility later (i.e.,  $\sigma_i < \sigma_j$  and  $\alpha_j < \alpha_i$ ). We have  $\alpha_j < \alpha_i \leq \sigma_i < \sigma_j$ . Consider a new solution  $S'$  in which job  $i$  and job  $j$  are swapped for their assignment to vehicles in the inbound transportation. That is, we now have  $\alpha'_i = \alpha_j$  and  $\alpha'_j = \alpha_i$ , where  $\alpha'_i$  and  $\alpha'_j$  are the arrival times of jobs  $i$  and  $j$  in solution  $S'$ , respectively. Note that  $S$  and  $S'$  have the same outbound transportation and production schedules. Let  $TC(S)$  denote the cost of solution  $S$ .  $TC(S)$  and  $TC(S')$  differ only in terms of inventory

holding costs of jobs  $i$  and  $j$  while they are waiting as unprocessed at the production facility. It follows that  $TC(S) - TC(S') = (\sigma_i - \alpha_i + \sigma_j - \alpha_j)h^1 - (\sigma_i - \alpha'_i + \sigma_j - \alpha'_j)h^1 = 0$ . Thus,  $S'$  is equivalent to  $S$  in its objective function value. Continuing in this fashion and swapping the inbound vehicle assignments of all such  $(i, j)$  in  $S$ , results in another feasible solution with the same objective function value.  $\square$

### A.5. Proof of Proposition 3

We know from Propositions 1, 2 and their proofs that there exists an optimal solution in which if job  $i$  precedes job  $j$  in the production sequence, then job  $i$  arrives at the facility and departs from the facility no later than job  $j$ . The proof of the current theorem will follow by showing that, if  $h^1 < h^2$ , in such an optimal solution, jobs that arrive to and depart from the facility together are processed in LPT order. Hence, in case of  $h^1 < h^2$ , there exists an optimal solution with the property stated in the theorem.

Take an optimal solution  $S$  in which inbound, outbound and production sequences are in compliance. Note that, in this solution, jobs that arrive to and depart from the production facility together are processed consecutively. Assume, by contradiction, that  $S$  does not comply with the theorem. Therefore, there exists at least a pair of adjacent jobs  $i$  and  $j$  in the production schedule that arrive to and depart from the facility together ( $\alpha_i = \alpha_j$ ,  $\delta_i = \delta_j$ ), however, job  $i$  precedes job  $j$  in the production schedule ( $\sigma_i < \sigma_j = \sigma_i + p_i$ ) despite  $p_i < p_j$ .

Construct another feasible solution  $S'$  from  $S$  by interchanging jobs  $i$  and  $j$  in the production sequence. We now have  $\sigma'_j = \sigma_i$ ,  $\sigma'_i = \sigma_j + p_j$ , where  $\sigma'_i$  and  $\sigma'_j$  are the starting times of processing of jobs  $i$  and  $j$  in  $S'$ , respectively. Note that,  $S$  and  $S'$  are only different in their production schedules of these two jobs. Let  $TC(S)$  denote the total cost of solution  $S$ . We have

$$TC(S) - TC(S') = \left[ (\sigma_i - \alpha_i + \sigma_j - \alpha_j)h^1 + (\delta_i - (\sigma_i + p_i) + \delta_j - (\sigma_j + p_j))h^2 \right] - \left[ (\sigma'_i - \alpha_i + \sigma'_j - \alpha_j)h^1 + (\delta_i - (\sigma'_i + p_i) + \delta_j - (\sigma'_j + p_j))h^2 \right],$$

which leads to

$$TC(S) - TC(S') = (\sigma_i + \sigma_j - \sigma'_i - \sigma'_j)(h_1 - h_2) = (p_j - p_i)(h^2 - h^1).$$

Since  $p_j > p_i$  and  $h^2 > h^1$ , the above expression is greater than zero. This implies  $TC(S') < TC(S)$ , which contradicts with the optimality of  $S$ . Therefore, if  $h^1 < h^2$ , jobs that arrive to and depart from the production facility together in solution  $S$ , should be processed in LPT order in every optimal solution.  $\square$

### A.6. Proof of Proposition 5

We first show that if there is any idle time in a block, the last job processed before the idle time starts, say job  $i$ , must have  $h_i^2 \leq h_i^1$ . Similarly, the first job processed after the idle time ends must have  $h_i^2 \geq h_i^1$ . If there is some idle time in a block after (before) some job  $i$ , starting the processing of this job later (earlier) will decrease the total costs by  $h_i^2 - h_i^1 (h_i^1 - h_i^2)$  per time unit. Thus, for any job with  $h_i^2 > h_i^1$ , there will be no idle time after job  $i$ , and for any job with  $h_i^2 < h_i^1$ , there will be no idle time before job  $i$ .

Now, consider an optimal solution  $S$  with two consecutive jobs  $j$  and  $k$  such that  $j$  precedes  $k$  in the processing sequence and  $\frac{h_j^2 - h_j^1}{p_j} > \frac{h_k^2 - h_k^1}{p_k}$ . If there is idle time between these jobs, then  $h_j^2 \leq h_j^1$

and  $h_k^2 \geq h_k^1$  according to the above discussion. This implies  $\frac{h_j^2 - h_j^1}{p_j} \leq 0 \leq \frac{h_k^2 - h_k^1}{p_k}$ , which contradicts with  $\frac{h_j^2 - h_j^1}{p_j} > \frac{h_k^2 - h_k^1}{p_k}$ . If there is no idle time between these jobs in  $S$ , define another solution  $S'$  where only jobs  $j$  and  $k$  are swapped in the production sequence. The difference between the total costs of solutions  $S$  and  $S'$  is given by

$$TC(S) - TC(S') = (p_j h_k^1 + p_k h_j^2) - (p_k h_j^1 + p_j h_k^2) \\ = p_k (h_j^2 - h_j^1) - p_j (h_k^2 - h_k^1),$$

which is greater than zero as  $\frac{h_j^2 - h_j^1}{p_j} > \frac{h_k^2 - h_k^1}{p_k}$ . Thus,  $S$  is not an optimal solution. Note that together with the above discussion on idle time, the WSPT property implies that, if there is idle time in a block, the jobs with  $h_i^2 < h_i^1$  are processed consecutively before idle time and the jobs with  $h_i^2 > h_i^1$  are processed consecutively after idle time in the WSPT sequence. □

**A.7. Proof of Proposition 6**

Consider an optimal solution  $S$  which exhibits the block structure. Assume, in contradiction to the proposition, that there exist two jobs  $u$  and  $v$  that arrive at the facility together,  $u$  departs earlier than  $v$ , and  $p_u > p_v$ . Fig. 7 is an illustration of such a solution. A, B, C and D in the figure refer to sets of jobs with certain common characteristics. More specifically, A and B are groups of jobs that arrive at the facility with job  $u$  at time  $t_0$  and leave the facility with job  $u$  at time  $t_1$ . Jobs in A are processed before job  $u$  and jobs in B are processed after job  $u$ . Jobs in C and D also arrive at the facility with job  $u$ , however, they leave the facility with job  $v$  and at time  $t_2$ . In mathematical terms,

$$\alpha_i = t_0 \quad \forall i \in A \cup B \cup C \cup D \cup \{u, v\},$$

$$\delta_i = t_1 \quad \forall i \in A \cup B \cup \{u\},$$

$$\delta_i = t_2 \quad \forall i \in C \cup D \cup \{v\}.$$

Note that, any of the sets A, B, C and D may be empty.

Now, consider a new solution  $S'$  that is formed by interchanging the positions of jobs  $u$  and  $v$  in the production sequence and their assignments to vehicles in the outbound transportation. Fig. 8 is an illustration of such a solution.

Denoting  $\sigma_i$  as the starting time of processing of job  $i$  in solution  $S$ , in the new schedule  $S'$  we have

$$\alpha'_i = t_0 \quad \forall i \in A \cup B \cup C \cup D \cup \{u, v\},$$

$$\sigma'_i = \sigma_i \quad \forall i \in A \cup D,$$

$$\sigma'_i = \sigma_i - p_u + p_v \quad \forall i \in B \cup C,$$

$$\delta'_i = t_1 - p_u + p_v \quad \forall i \in A \cup B \cup \{v\},$$

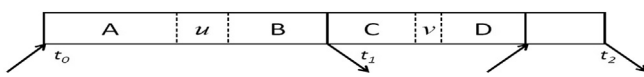


Fig. 7. An illustration of a solution in contradiction to Proposition 6.

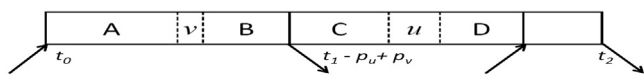


Fig. 8. An illustration of the updated solution  $S'$ .

$$\delta'_i = t_2 \quad \forall i \in C \cup D,$$

$$\delta'_u = t_2, \quad \sigma'_u = \sigma_v - p_u + p_v, \quad \sigma'_v = \sigma_u.$$

As the number of tours in  $S'$  remains the same as the one in  $S$ , the total costs of the two solutions differ only in their inventory holding cost component, and the difference is

$$TC(S) - TC(S') = \sum_{i \in A \cup B \cup C \cup D \cup \{u, v\}} (\sigma_i - t_0) h^1 + (\delta_i - \sigma_i - p_i) h^2 \\ - (\sigma'_i - t_0) h^1 - (\delta'_i - \sigma'_i - p_i) h^2,$$

which reduces to

$$TC(S) - TC(S') = \sum_{i \in A \cup B \cup C \cup D \cup \{u, v\}} (\sigma_i - \sigma'_i) h^1 + \{(\delta_i - \delta'_i) + (\sigma'_i - \sigma_i)\} h^2.$$

When the values of  $\delta_i, \sigma'_i, \delta'_i$  are plugged in the above expression for each group of jobs, it can be rewritten as

$$TC(S) - TC(S') = \sum_{i \in A} (\sigma_i - \sigma_i) h^1 + \{t_1 - (t_1 - p_u + p_v) \\ + (\sigma_i - \sigma_i)\} h^2 + \sum_{i \in B} (\sigma_i - (\sigma_i - p_u + p_v)) h^1 \\ + \{(t_1 - (t_1 - p_u + p_v)) + ((\sigma_i - p_u + p_v) - \sigma_i)\} h^2 \\ + \sum_{i \in C} (\sigma_i - (\sigma_i - p_u + p_v)) h^1 \\ + \{(t_2 - t_2) + ((\sigma_i - p_u + p_v) - \sigma_i)\} h^2 \\ + \sum_{i \in D} (\sigma_i - \sigma_i) h^1 + \{(t_2 - t_2) + (\sigma_i - \sigma_i)\} h^2 \\ + (\sigma_u - (\sigma_v - p_u + p_v)) h^1 \\ + \{(t_1 - t_2) + ((\sigma_v - p_u + p_v) - \sigma_u)\} h^2 \\ + (\sigma_v - \sigma_u) h^1 + \{(t_2 - (t_1 - p_u + p_v)) + (\sigma_u - \sigma_v)\} h^2.$$

After some cancelations and rearrangement of terms, the above expression reduces to

$$TC(S) - TC(S') = \sum_{i \in A} (p_u - p_v) h^2 + \sum_{i \in B} (p_u - p_v) h^1 \\ + \sum_{i \in C} (p_u - p_v) (h^1 - h^2) + (p_u - p_v) h^1,$$

which is equivalent to

$$TC(S) - TC(S') = (p_u - p_v) (|A| h^2 + |B| h^1 + |C| (h^1 - h^2) + h_1).$$

Note that under the  $h^1 \geq h^2$  condition of this proposition, we assume  $h^1 > 0$  because, otherwise we would have  $h^1 = h^2 = 0$ , which would be trivial. Combining with  $p_u > p_v$ , we conclude that  $TC(S) - TC(S') > 0$ . This contradicts with the optimality of  $S$ . □

**References**

Agnetis, A., Aloulou, M. A., & Fu, L.-L. (2014). Coordination of production and interstage batch delivery with outsourced distribution. *European Journal of Operational Research*, 238, 130–142.

Chang, Y.-C., & Lee, C.-Y. (2004). Machine scheduling with job delivery coordination. *European Journal of Operational Research*, 158, 470–487.

Chen, Z.-L. (2010). Integrated production and outbound distribution scheduling: Review and extensions. *Operations Research*, 58, 130–148.

Chen, Z.-L., & Pundoor, G. (2006). Order assignment and scheduling in a supply chain. *Operations Research*, 54, 555–572.

Chen, Z.-L., & Vairaktarakis, G. L. (2005). Integrated scheduling of production and distribution operations. *Management Science*, 51, 614–628.

Dawande, M., Geismar, H. N., Hall, N. G., & Sriskandarajah, C. (2006). Supply chain scheduling: Distribution systems. *Production and Operations Management*, 15, 243–261.

Enenay, F. S., Sabuncuoglu, I., Toptal, A., & Tiwari, M. K. (2010). New solution methods for single machine bicriteria scheduling problem: Minimization of average flowtime and number of tardy jobs. *European Journal of Operational Research*, 201, 89–98.

- Gao, S., Qi, L., & Lei, L. (2015). Integrated batch production and distribution scheduling with limited vehicle capacity. *International Journal of Production Economics*, 160, 13–25.
- Hall, N. G., & Posner, M. E. (2001). Generating experimental data for computational testing with machine scheduling applications. *Operations Research*, 49, 854–865.
- Hall, N. G., & Potts, C. N. (2003). Supply chain scheduling: Batching and delivery. *Operations Research*, 51, 566–584.
- Koc, U., Toptal, A., & Sabuncuoglu, I. (2013). A class of joint production and transportation planning problems under different delivery policies. *Operations Research Letters*, 41, 54–60.
- Li, C.-L., & Ou, J. (2005). Machine scheduling with pickup and delivery. *Naval Research Logistics*, 52, 616–630.
- Li, C.-L., Vairaktarakis, G. L., & Lee, C.-Y. (2005). Machine scheduling with deliveries to multiple customer locations. *European Journal of Operational Research*, 164, 39–51.
- Sabuncuoglu, I., & Karabuk, S. (1998). A beam search-based algorithm and evaluation of scheduling approaches for flexible manufacturing systems. *IIE Transactions*, 30, 179–191.
- Sen, A. (2008). The US fashion industry: A supply chain review. *International Journal of Production Economics*, 114, 571–593.
- Tang, L., & Gong, H. (2008). A hybrid two-stage transportation and batch scheduling problem. *Applied Mathematical Modelling*, 32, 2467–2479.
- Tang, L., & Gong, H. (2009). The coordination of transportation and batching scheduling. *Applied Mathematical Modelling*, 33, 3854–3862.
- Tang, L., Gong, H., Liu, J., & Li, F. (2014). Bicriteria scheduling on a single batching machine with job transportation and deterioration considerations. *Naval Research Logistics*, 61, 269–285.
- Thomas, D. J., & Griffin, P. M. (1996). Coordinated supply chain management. *European Journal of Operational Research*, 94, 1–15.
- Toptal, A., Koc, U., & Sabuncuoglu, I. (2014). A joint production and transportation planning problem with heterogeneous vehicles. *Journal of Operational Research Society*, 65, 180–196.
- Wang, X., & Cheng, T. C. E. (2006). Machine scheduling with an availability constraint and job delivery coordination. *Naval Research Logistics*, 54, 11–20.
- Wang, X., & Cheng, T. C. E. (2009). Production scheduling with supply and delivery considerations to minimize the makespan. *European Journal of Operational Research*, 194, 743–752.
- Wang, H., & Lee, C.-Y. (2005). Production and transport logistics scheduling with two transport mode choices. *Naval Research Logistics*, 52, 796–809.
- Zhong, W., Dósa, G., & Tan, Z. (2007). On the machine scheduling problem with job delivery coordination. *European Journal of Operational Research*, 182, 1057–1072.