Logistics scheduling with batching and transportation

Bo Chen a,*, Chung-Yee Lee b

a Warwick Business School, University of Warwick, Coventry CV4 7AL, UK
b Department of Industrial Engineering and Engineering Management, Hong Kong University of Science and Technology, Kowloon, Hong Kong

Received 1 October 2005; accepted 1 November 2006
Available online 17 July 2007

Abstract

This paper studies a general two-stage scheduling problem, in which jobs of different importance are processed by one first-stage processor and then, in the second stage, the completed jobs need to be batch delivered to various pre-specified destinations in one of a number of available transportation modes. Our objective is to minimize the sum of weighted job delivery time and total transportation cost. Since this problem involves not only the traditional performance measurement, such as weighted completion time, but also transportation arrangement and cost, key factors in logistics management, we thus call this problem logistics scheduling with batching and transportation (LSBT) problem.

We draw an overall picture of the problem complexity for various cases of problem parameters accompanied by polynomial algorithms for solvable cases. On the other hand, we provide for the most general case an approximation algorithm of performance guarantee.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Sequencing; Batching; Transportation; Performance guarantee

1. Introduction

Most of classical scheduling literature considers job processing without taking into account the delivery issue (see, for example, Blazewicz et al. [2], Brucker [3] and Pinedo [14]) or with the assumption that job delivery can be done instantaneously (see, for example, Herrmann and Lee [9], Chen [4], Cheng et al. [6] and Hall and Potts [8]).

Lee and Chen [10] consider transportation issue in machine scheduling problems, in which there are either a certain number of transporters to transfer jobs from one-machine to another (Type-I), or transfer finished jobs to the customers or warehouses (Type-II). Both transportation time and transporter capacity were considered. Hall et al. [7] consider extensively scheduling problems in an environment in which the finished jobs can be delivered only on fixed dates. Matsuo [12] considers a similar problem in the single machine environment. Li et al. [11] consider a scheduling problem similar to Type-II in [10] yet with two customer locations and hence involving routing issue. None of the above four papers considers the transportation cost. Chen and Vairaktarakis [5] consider a two-stage scheduling problem in which the first stage is manufacturing facility and the...
second is the delivery to customers, i.e., type-II problem in [10]. Their objective function is a combination of customer service level and total distribution cost, where customer service level is measured as a function of the job delivery times to customers. Wang and Lee [16] also consider a two-stage scheduling problem in which the second stage involves the transportation mode selection. There are two different transportation modes available and the one with shorter delivery time will incur a higher delivery cost.

In this paper, we consider a more general two-stage scheduling problem where in the second stage of transportation there are multiple (more than two) transportation modes to select and multiple destinations. We also incorporate job weights into our model to take job priority issue into account. More specifically, we consider the following problem of logistics scheduling: There are \( n \) independent jobs of respective weights \( w_1, \ldots, w_n \), which need to be processed by a single processor for respective uninterrupted \( p_1, p_2, \ldots, p_n \) time units. The processor can handle at most one job at a time. Once completed on the processor, each job \( j \) needs to be delivered to a pre-specified destination \( s_j \), \( s_j \in \{1, \ldots, k\} \). Up to \( B \) jobs of the same destination may be batched together for simultaneous delivery, which will incur a (batch) transportation time \( t_i^{(s)} \) and a (batch) transportation cost \( c_i^{(s)} \), where \( s \in \{1, \ldots, k\} \) and \( i \in \{1, \ldots, m\} \) denote, respectively, the pre-specified destination of the batch of jobs and the transportation mode for delivering the batch. Note that the batch transportation time and cost are independent of the batch size. Delivery of a batch takes place immediately after all jobs of the batch are completed on the processor, independent of other batches.

Without loss of generality, we assume that speedier delivery incurs higher cost, i.e., we assume that \( t_i^{(s)} > t_j^{(s)} \) \( \cdots \) \( t_m^{(s)} \) and \( c_i^{(s)} < c_j^{(s)} \) \( \cdots \) \( c_m^{(s)} \) for any destination \( s = 1, \ldots, k \). The delivery time \( D_j \) of job \( j \) is \( B_j + t_m^{(s_j)} \), the completion time \( C_j \) on the processor of the last job of the batch job \( j \) belongs to plus transportation time \( t_m^{(s_j)} \) of the batch of job \( j \), where \( m_j \) is the transportation mode in which the batch of job \( j \) is delivered. The problem is to construct a schedule \( S \) based on a three-fold decision: (a) the sequence these jobs are processed, (b) how completed jobs are batched, and (c) which transportation mode should be used for delivery of each batch. The objective is to minimize the sum \( \phi(S) \) of weighted job delivery time and total transportation cost in schedule \( S \), i.e.,

\[
\phi(S) = \sum_{j=1}^{n} w_j D_j + \sum_{i} c_{m_i}^{(s_i)},
\]

where \( i \) is the batch index, \( m_i \) and \( s_i \) are the transportation mode and destination of batch \( i \), respectively.

Since this problem involves not only the traditional performance measurement, such as weighted completion time, but also transportation arrangement and cost, key factors in logistics management, we thus call this problem logistics scheduling with batching and transportation (LSBT) problem.

In this paper, under a practical assumption that both the number \( k \) of delivery destinations and the number \( m \) of available transportation modes are fixed constants, we provide, on the one hand, an overall picture of the problem complexity for various cases of problem parameters, which is accompanied by polynomial algorithms for solvable cases. The picture is complete except for one minor case, which we leave as an open problem. On the other hand, we provide for the most general case an approximation algorithm of performance guarantee. Note that complexity issues are also considered by Soukhal et al. [15] for LSBT problem of flow shop (with equal job weights) and by Albers and Brucker [1] with batch setup times.

The rest of the paper is organized as follows. After establishing a preliminary result in Section 2, we deal with various special cases in Section 3 and general cases in Section 4. Two polynomial approximation algorithms will be provided in Section 5 for the general problem together with analysis of performance guarantee.

2. Preliminaries

We establish an observation in this section, which will be used in subsequent sections.

**Lemma 1.** Suppose jobs \( j_1 \) and \( j_2 \) are of the same destination. If \( p_{j_1} > p_{j_2} \) and \( w_{j_1} \leq w_{j_2} \), then job \( j_1 \) cannot be processed in a separate batch earlier than the one job \( j_2 \) belongs to in any optimal schedule.

**Proof.** Suppose to the contrary that jobs \( j_1 \) and \( j_2 \) are scheduled in two separate batches and \( j_1 \) is in an earlier batch. Let \( C_1 < C_2 \) be the respective completion times of the two batches and \( W_1, W_2 \) be their respective sums of weights of the jobs in the batches. Denote \( \Delta p = p_{j_1} - p_{j_2} \) and \( \Delta w = w_{j_1} - w_{j_2} \). Consider swapping the assignment of the two jobs to their batches while keeping all other schedul-
ing decisions the same. The completion time of the first batch becomes $C_1 - \Delta p$ and that of the second batch remains the same. On the other hand, the total weights of the two batches become $W_1 + \Delta w$ and $W_2 - \Delta w$, respectively. Note that the completion time of any other batch is not increased. Therefore, while the total transportation cost remains the same, the weighted arrival time is decreased by at least

\[
(W_1C_1 + W_2C_2) - ((W_1 + \Delta w)(C_1 - \Delta p)) + (W_2 - \Delta w)C_2
\]

\[
= W_1\Delta p + \Delta w(C_2 + \Delta p - C_1) > 0,
\]

which contradicts the optimality of the original schedule. □

3. Special cases

In this section we solve to optimality three categories of special cases: (a) the weights are agreeable, i.e., $p_j < p_k$ implies $w_j \geq w_k$; (b) the number of distinct processing times or distinct weights of jobs of each destination is bounded by a constant, i.e., either $|\{p_j^{(s)}\}, j = 1, \ldots, n_s| \leq N$ or $|\{w_j^{(s)}\}, j = 1, \ldots, n_s| \leq N$ for any $s = 1, \ldots, k$, where the superscript $(s)$ on the processing time and weight indicates the destination of the corresponding job, $n_s$ is the number of jobs of destination $s$, and $N$ is a constant; (c) batching is not allowed, i.e., $B = 1$. Note that category (a) includes the case where all jobs are of equal weight.

3.1. Agreeable weights

In this subsection, we assume the job weights are agreeable. Based on Lemma 1, we reindex all jobs so that jobs of the same destination are in SPT order, i.e., in order of non-decreasing processing times. As a result, we have one SPT sequence for each destination. Denote the processing time sequence for destination $s$ by $p_j^{(s)} \leq p_2^{(s)} \leq \cdots \leq p_{n_s}^{(s)}$ and let

\[
P_j^{(s)} = \sum_{i=1}^{j} p_i^{(s)}
\]

for $\ell = 1, \ldots, n_s$, $s = 1, \ldots, k$. Denote by $f(\ell_1, \ell_2, \ldots, \ell_k, b, s, i)$ the optimal objective function value (i.e., minimum sum of weighted delivery time and total transportation cost) of schedules that satisfy the following conditions: (i) the total number of scheduled jobs is $\ell_1 + \cdots + \ell_k$, of which $\ell_i$ jobs are from the top of the SPT sequence for destination $t_i$, $t = 1, \ldots, k$; (ii) the number of jobs in the last batch is $b$; (iii) the last scheduled job is for destination $s$ with transportation mode $i$. Then we have the following recursive relationship:

\[
f(\ell_1, \ldots, \ell_k, b, s, i) = w_{\ell_1}^{(i)} \left( \sum_{i=1}^{k} P_{\ell_i}^{(i)} + t_{\ell_i}^{(i)} \right) + \tilde{f}(\ell_1, \ldots, \ell_k, b, s, i),
\]

where

\[
\tilde{f}(\ell_1, \ldots, \ell_k, b, s, i) \begin{cases} 
  f(\ell_1, \ldots, \ell_{s-1}, \ell_s - 1, \ell_{s+1}, \ldots, \ell_k, b - 1, s, i) + \tau^{(s)} & \text{if } b > 1, \\
  \min_{b', s', i'} f(\ell_1, \ldots, \ell_{s-1}, \ell_s - 1, \ell_{s+1}, \ldots, \ell_k, b', s', i') + c_{i}^{(i)} & \text{if } b = 1,
\end{cases}
\]

and $\tau^{(s)} = p_{\ell_i}^{(i)} \sum_{j=1}^{b-1} w_{\ell_j}^{(s)}$. The equation for $\tilde{f}$ in the case of $b > 1$ reflects the fact that the last job in the schedule delays by $\tau^{(s)}$ the delivery time of each of the other $b - 1$ jobs in the last batch, while it does not incur additional transportation cost. On the other hand, the above equation for $\tilde{f}$ in the case of $b = 1$ reflects the fact that the last job in the schedule only incurs an additional transportation cost $c_{i}^{(i)}$ on top of its own weighted delivery time.

Therefore, the overall optimal objective function value is obtained by minimizing $f(n_1, \ldots, n_k, b, s, i)$ over all $b = 1, \ldots, B$, all $s = 1, \ldots, k$ and all $i = 1, \ldots, m$, which can be accomplished in time $O(m^3k^2B^2n^k) = O(B^2n^k)$. (Recall that both $m$ and $k$ are constant.)

Theorem 2. If the weights of all jobs of each destination are agreeable, then the LSBT problem is solved to optimality with dynamic programming (1) in polynomial time $O(B^2n^k)$.

As we know, polynomial solvability of the problem is based on the fact that, in any optimal schedule, jobs of each destination must be in the SPT order. In other words, addition of precedence constraints of $k$ chains based on the $k$ SPT sequences will not change the optimal schedules. This observation leads us to another category of solvable cases as discussed in the following subsection.

3.2. Bounded number of distinct processing times or weights

Weight agreeability is a partial order, which is transitive. Consider jobs of the same destination. Observe that all jobs of the same processing time satisfy weight agreeability and so do all jobs of the
same weight. Therefore, in any optimal schedule, jobs of the same destination form a chain precedence relation if they have the same processing time or the same weight. In other words, optimal schedules will not change if we add precedence constraints of up to $N$ chains for each destination, where $N$ is the upper bound of the number of distinct processing times or weights in each destination. Based on this observation, we can adapt dynamic programming (1) with the modification that one single SPT chain of jobs of each destination is involved, i.e., $B = 1$. Similar to Lemma 1, we have the following.

**Theorem 3.** If the number of distinct processing times or weights is bounded by a constant $N$ in each destination, then the LSBT problem can be solved in polynomial time $O(B^2n^N)$.  

3.3. No batching

Here we deal with the case where no batching is involved, i.e., $B = 1$. Similar to Lemma 1, we have the following.

**Theorem 4.** If batching is not allowed, then the sequence of all jobs in any optimal schedule is in WSPT order, i.e., in order of non-decreasing $p_j/w_j$. The problem can be solved to optimality in $O(n\log n)$ time.

**Proof.** Given an optimal schedule. Consider any two consecutive jobs $j_1$ and $j_2$ in this order with processing times $p_{j_1}$ and $p_{j_2}$, and weights $w_{j_1}$ and $w_{j_2}$, respectively. These two jobs are of destinations $k_1$ and $k_2$, and are sent in transportation modes $m_1$ and $m_2$, respectively. Suppose to the contrary that $p_{j_1}/w_{j_1} > p_{j_2}/w_{j_2}$. Now we swap the positions of these two jobs while keeping their transportation modes the same. While the total transportation cost remains the same, the objective function is changed only in two terms: (a) the delivery time of job $j_1$ is increased by $p_{j_1}$ and (b) the delivery time of job $j_2$ is decreased by $p_{j_1}$. Therefore, the overall change of the value of the objective function is $w_{j_1}p_{j_2} - w_{j_2}p_{j_1}$, which is negative, implying that the schedule is improved and contradicting the optimality of the original schedule.

Based on the above argument, we find an optimal schedule in the following recursive way. First reindex all jobs such that they are in WSPT order: $p_1/w_1 \leq \cdots \leq p_n/w_n$. Denote by $t_i$ and $c_i$ the transportation time and cost, respectively, of job $\ell$ under the transportation mode $i, \ell = 1, \ldots, n, i = 1, \ldots, m$. Let $f_\ell$ be the optimal objective value of scheduling the first $\ell$ jobs, $\ell = 1, \ldots, n$. Then we have

$$f_\ell = f_{\ell - 1} + \min_{1 \leq i \leq m} \left( w_{\ell} \left( \sum_{j=1}^{\ell} p_j + t_i \right) + c_i \right),$$

for $\ell = 1, \ldots, n$ and $f_0 = 0$.

It is easily seen that the schedule found above is optimal, which can be completed in time $O(n\log n + nm) = O(n\log n)$. (Recall that $m$ is constant.) \hfill \Box

If we regard a job batch as a composite job, whose processing time and weight are the sums of processing times and weights, respectively, of the jobs in the batch, then the above arguments carry over verbatim, leading to the following.

**Corollary 5.** If the job batches are given and fixed, then the LSBT problem can be solved to optimality in $O(n\log n)$ time.

4. General case

In this section, we do not assume any special condition for the job processing times and weights. We will see our scheduling problem LSBT becomes extremely hard.

**Theorem 6.** For any fixed $B \geq 3$, the LSBT problem with batch size bounded by $B$ is strongly NP-hard, even if there is only one transportation mode ($m = 1$) and one delivery destination ($k = 1$).

**Proof.** Let us first assume $B = 3$. We will transform the strongly NP-complete 3-Partition to the decision version of our scheduling problem. Given any instance of 3-Partition, $\{a_1, \ldots, a_{3n}\}$, where $\sum_{j=1}^{3n} a_j = nb$ and $\frac{1}{2}b < a_j < \frac{3}{2}b$ for any $j (1 \leq j \leq 3n)$. We define an instance of our scheduling problem as follows: There are $3n$ jobs of processing times and weights defined by $p_j = w_j = a_j$ for $j = 1, \ldots, 3n$. The maximum batch size is 3 and the transportation time is negligible (zero) and transportation cost of each batch is $b^2$. We prove below that there is a partition $I_1, \ldots, I_n$ of indices $\{1, \ldots, 3n\}$ such that $|I_1| = \cdots = |I_n| = 3$ and $\sum_{j \in I_i} a_j = b$ for any $i = 1, \ldots, n$ if and only if there is a schedule with sum of weighted arrival time and total transportation costs no more than $n(n + 1)b^2/2 + nb^2$. 


If there is such a partition, then we schedule all jobs \( j \in I_i \) into one batch \( i \) with batch processing time and batch weight both equal to \( b \). Schedule these batches in any sequence. Then the objective value is easily seen to be \( n(n + 1)b^2/2 + nb^2 \). Conversely, suppose we have a schedule of objective value no more than \( n(n + 1)b^2/2 + nb^2 \). Apparently, the total number \( n' \) of batches in the schedule is at least \( n \) due to the restriction on the batch sizes. We also claim \( n' \leq n \). To see this, let us assume to the contrary that \( n' > n \). Denote by \( x_i \) the total processing time (and hence total weight) of batch \( i \), \( i = 1, \ldots, n' \). Then the objective value of the schedule is equal to

\[
h(x_1, \ldots, x_{n'}) = \sum_{i=1}^{n'} x_i \sum_{j=1}^{i} x_j + n'b^2.
\]

However, since the optimal value of convex program \( \min h(x_1, \ldots, x_{n'}) \) subject to \( \sum_{i=1}^{n'} x_i = nb \) and \( x_i \geq 0 \) for \( i = 1, \ldots, n' \) is \((n' + 1)/(2n') (nb)^2 + n' b^2 > n(n + 1)b^2/2 + nb^2 \), we have a contradiction!

Therefore, \( n' = n \). According to our assumption on the objective value of the schedule, we conclude that \((x_1, \ldots, x_n)\) is an optimal solution of the aforementioned convex program with \( n' = n \). Since the convex program has a unique optimal solution \( x_1 = \cdots = x_n = b \), we conclude that the original 3-Partition instance has a required partition that is based on the batching. This completes our proof for the problem complexity if \( B = 3 \).

For any fixed \( B > 3 \), we modify the above proof by adding \( n(B-3) \) dummy jobs of negligible processing times and weights. Formally, we define \( 3n \) regular jobs with \( p_j = w_j = 4(B-3)a_j \) for \( j = 1, \ldots, 3n \) and \( n(B-3) \) dummy jobs with \( p_j = w_j = 1 \) for \( j = 3n+1, \ldots, nB \). Define transportation cost as \( 16(B-3)^2b^2 \). Then, we can carry over the whole proof for the case of \( B = 3 \) with straightforward adjustment by noticing that, in the optimal solution of the convex program, each \( x_i = (B-3)(4b+1) \), which is strictly smaller than the sum of the processing times of any four regular jobs. The uniqueness of the optimal solution to the convex program together with the pigeonhole principle forces any such batch to contain three regular jobs and \( B-3 \) dummy jobs. \( \square \)

Note that for the remaining case where \( B = 2 \), the problem complexity is still open. In this case, according to Corollary 5, the problem is interestingly reduced to how to (partially) pair together jobs of the same destinations, which is distinctly different from the traditional matching problem in that the cost (the objective value to be minimized) of each matching is determined by an algorithm run in \( O(n \log n) \) time. Note that using an algorithm to define the objective function is another (indeed more general) way to specify a combinatorial optimization problem (see [13] for details).

5. Approximation

Given the high difficulty of our problem, we establish some approximation results in this section. Recall that \( \phi(S) \) denotes the objective value of schedule \( S \), i.e., the sum of weighted delivery time and total transportation cost of schedule \( S \). The first result is derived easily from Theorem 4.

Theorem 7. The general LSBT problem can be approximated within a factor of \( B \) in \( O(n \log n) \) time.

Proof. According to Theorem 4, we find in \( O(n \log n) \) time schedule \( S_1 \) that is optimal among all schedules without batching. Let \( S^* \) be an optimal schedule of the original LSBT problem (with batching). Consider schedule \( S \) that is obtained from \( S^* \) by breaking all batches and deliver the jobs individually using the same transportation modes as for the batches they belong to in schedule \( S^* \).

On the one hand, for each job \( j \), its delivery time \( D_j \) in \( S \) satisfies \( D_j \leq D_j^* \), its delivery time in schedule \( S^* \). On the other hand, the total transportation cost \( T \) in \( S \) satisfies \( T \leq BT^* \), where \( T^* \) is the total transportation cost in \( S^* \). Therefore,

\[
\phi(S^*) = \sum_{j=1}^{n} w_j D_j^* + T^* \geq \sum_{j=1}^{n} w_j D_j + T/B \\
\geq \left( \sum_{j=1}^{n} w_j D_j + T \right)/B \geq \phi(S_1)/B,
\]

where the last inequality is due to the optimality of \( S_1 \) among all schedules without batching. This completes our proof. \( \square \)

Note that the performance bound of \( B \) in the above theorem is tight if the approximation is done according to the way suggested in the proof, which can be easily seen from the following sketch worst-case instance: There are \( tB \) jobs of equal weights and of the same destination. Both the job processing times and batch transportation time are negligible (zero), and there is only one transportation mode. Let the (only) batch transportation cost be \( c \). Then processing the jobs in any order and batching any
$B$ jobs together result in the optimal objective value of $\ell c$, while breaking the batches into individual jobs will increase the value to $B\ell c$.

Although the complexity of the LSBT problem with $B = 2$ is still open, it is interesting to notice that the above approximation of the general LSBT problem can be improved if improved approximation can be made on the special case of $B = 2$, which currently has a performance guarantee of 2 according to the above theorem. In particular, if the LSBT problem with $B = 2$ is polynomially solvable, then the approximation factor for the general LSBT problem can be reduced from $B$ to $B/2$. More specifically, we have the following.

**Theorem 8.** Suppose the LSBT problem with $B = 2$ can be polynomially approximated within a factor of $\rho$ ($\leq 2$), then the general LSBT problem can be polynomially approximated within a factor of $(\rho/2)B$. In particular, if $\rho = 1$ (i.e., the special case of $B = 2$ is polynomial solvable), then the approximation factor becomes $B/2$.

**Proof.** Without loss of generality, we assume $B \geq 2$ according to Theorem 4. Let $S^*$ be an optimal schedule for the general problem. Let us temporarily regard the problem as one with $B = 2$ and construct a schedule $S$ in polynomial time such that $\phi(S) \leq \rho \phi(S^*)$, where both schedules $S$ and $S^*$ contain at most two jobs in each batch and $S^*$ is optimal among all such schedules. We show that $S$ is a qualified approximate schedule for the original general problem. In fact, consider schedule $S'$ that is obtained from $S^*$ by breaking each batch into at most $B/2$ sub-batches of size at most 2 and deliver the sub-batches separately using the same transportation mode as for the original batch. Then it is clear that (a) the total transportation cost $T'$ in schedule $S'$ satisfies $T' \leq (B/2)T^*$, where $T^*$ is the total transportation cost in schedule $S^*$, and (b) the delivery time $D'_j$ in $S'$ for each job $j$ satisfies $D'_j \leq D_j^*$, its delivery time in schedule $S^*$. Therefore,

$$\phi(S^*) = \sum_{j=1}^{n} w_j D'_j + T^* \geq \sum_{j=1}^{n} w_j D'_j + T'/(B/2)$$

$$\geq \left( \sum_{j=1}^{n} w_j D'_j + T' \right) / (B/2) \geq \phi(S^*) / (B/2)$$

$$\geq \phi(S)/(\rho B/2),$$

i.e., $\phi(S) \leq (\rho/2)B\phi(S^*)$. $\square$

Note that the approximation factor $(\rho/2)B$ in the above theorem may not be tight, since the error in approximating the special case of $B = 2$ and that in breaking all batches into individual jobs is likely to cancel out. However, a definitive conclusion on this is impossible without the knowledge of how the approximation factor $\rho$ is achieved for the special case of $B = 2$.

**Acknowledgement**

This research is supported in part by Hong Kong RGC Earmark Grant: HKUST 6010/02E.

**References**


