ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
DOI: 10.17559/TV-20150312154507

THE SINGLE MACHINE SERIAL BATCH PRODUCTION SCEDULING WITH OUTSOURCING ALLOWED

Juan Zou, Yuzhong Zhang, Cuixia Miao

Original scientific paper

We consider several serial batch scheduling problems with outsourcing allowed. Each job can be either processed on the in-house machine or outsourced. The objective is to minimize the weighted sum of the scheduling cost and the total outsourcing cost. We discuss three classical scheduling objective functions. The first is the total completion time, the second is the maximum lateness, and the third is the weighted number of late jobs. For each problem, we derive an efficient dynamic programming algorithm that minimizes the total cost.

Keywords: *dynamic programming; outsourcing; scheduling; serial batching*

Planiranje serijske proizvodnje na jednom stroju uz dopušteni outsourcing

Izvorni znanstveni članak

Razmatramo nekoliko problema planiranja proizvodnje u serijama s dopuštenom mogućnošću ugovaranja proizvodnje s vanjskim proizvođačem (outsourcing). Postoji mogućnost da se svaki posao obavi na vlastitom stroju ili se ugovori s drugim poduzećem. Cilj je smanjiti ponderirani iznos cijene planirane proizvodnje i ukupne cijene outsourcinga. Razmatramo tri klasične objektivne funkcije programiranja. Prva je ukupno vrijeme izvršenja, druga je maksimalno kašnjenje, a treća je ponderirani broj kasnih poslova. Za svaki problem izvodima učinkovit dinamički algoritam programiranja koji minimalizira ukupni trošak.

Ključne riječi: *dinamičko programiranje; outsourcing; planiranje; serijska proizvodnja robe*

1 Introduction

In flexible manufacturing systems, batch scheduling is the decision about whether or not to process certain jobs consecutively. Batching problems that arise in modern manufacturing present a fruitful research direction and their solution enhances the ability to manage manufacturing operations efficiently. In the serial batch model (see Fig. 1), the machine can handle one job at a time. The jobs in a batch are processed serially and their completion times are defined to be the finishing time of the last job in the batch. For each batch, a machine setup time $s \geq 0$ is required before the first job of this batch is processed. Coffman et al. [1] provided an algorithm to solve the problem $1|s - batch|\Sigma Cj$ in $O(n \log n)$ time. Webster and Baker [2] proposed a dynamic programming algorithm to minimize the maximum lateness in $O(n^2)$ time. Brucker and Kovalyov [3] developed a pseudo-polynomial dynamic programming algorithm to minimize the weighted number of late jobs, which is converted into a fully polynomial approximation scheme, and also solved the problem with equal weights in $O(n^3)$ time. In the batch delivery model, it is generally assumed that jobs are delivered to customers in batches where the delivery cost will be incurred. Cheng et al. [4] considered a single machine batch scheduling problem, in which they minimize the sum of batch delivery cost plus job earliness penalties. Cheng et al. [5] studied the single machine batch delivery problem to minimize the sum of the total weighted earliness and mean batch delivery time. This problem is significant and relevant to logistics and supply chain management. In the context of supply chains, Hall and Potts [6] analyzed the complexity of some single machine scheduling problems with batch deliveries.

On the other hand, in the modern manufacturing industries, outsourcing is often used as a complementary source to its own in-house production which has a limited capacity. When the demand level is beyond the in-house production capacity, the manufacturer may have the option to outsource some orders to available subcontractors so that all orders can be completed as early as possible. In our model, the manufacturer needs to pay the subcontractor for what is outsourced, the outsourced jobs, upon completion, need to be transported back from the subcontractor to the manufacturer, and thus necessary transportation delay and transportation cost may be incurred. The problem is then how to coordinate the in-house production and outsourcing in an efficient way. For a given set of jobs, the decisions we need to make include the selection of a subset of jobs to be outsourced, the schedule of the jobs, the assignment of jobs to the batches and a transportation plan for the outsourced jobs.

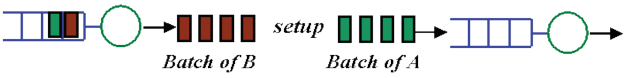

Figure 1 Serial process batching

The literature on scheduling with an outsourcing option has studied the problem of simultaneously deciding which jobs to be outsourced and how to schedule in-house jobs. Qi [7] considered a single machine scheduling problem that allows some jobs to be outsourced to a single outsourcing machine, with a transportation delay and transportation cost. Qi [8] considered the production scheduling problem for a two-stage flow shop in which there are options of outsourcing some operations to subcontractors. Ruiz-Torres et al. [9] considered the situation with parallel in-house machines and parallel outsourcing machines, where they dealt with a bi-criteria scheduling problem to minimize the outsourced production time, associated presenting the outsourcing cost and the number of late orders. Lee and Sung [10, 11] considered a single machine scheduling problem to minimize the weighted sum of the outsourcing cost and the associated scheduling cost with an outsourcing budget constraint. Chen and Li [12] dealt with a parallel machine scheduling problem with an outsourcing option to minimize the total production cost subject to a constraint on the makespan. For updated results on scheduling with an outsourcing option, the reader may refer to Mokhtari et al. [14] and Choi et al. [15]. In Qi's model, jobs are processed on a single machine at in-house production. Whereas in our model, we consider the jobs are processed in batches on a single machine at in-house production, and the objective is to minimize the weighted sum of the scheduling cost and the total outsourcing cost. To the best of our knowledge, this subject has not been studied in the literature.

This paper is organized as follows. In Section 2, we introduce some notation and describe the scheduling problems with outsourcing allowed. In Section 3, we provide three dynamic programming algorithms for the three scheduling problems with the option of outsourcing. In Section 4, we conclude the paper and suggest some topics for future research.

2 Problem formulation and notation

The proposed problem is stated in detail as follows (see Fig. 2): there are n jobs available at time zero. A manufacturer can either process each job at its own machine or outsource it to a subcontractor for processing. There is a single machine available at the manufacturer's plant. Each job j has a processing time p_j, a due date d_j and a weight w_j. The jobs are processed in batches on the in-house machine. The completion time of all jobs in a batch is defined to be equal to the finishing time of the last job in the batch, i.e., the processing time of a batch is equal to the sum of the processing times of all jobs in the batch. If a new batch is formed, a constant setup time s will be incurred. To make the jobs completed earlier, we can outsource some jobs to a subcontractor with a single machine. Outsourcing will cause some new concerns, the outsourcing cost, the transportation delay and the transportation cost. If job j is outsourced, its processing time at the outsourcing machine is αp_j, where α denotes a probably different machine speed. At the same time, an extra cost βp_j will be incurred which is assumed to be proportional to the processing time p_j. This is the cost for paying to the subcontractor. In many cases, the subcontractor is in a different location from the in-house plant, and an outsourced job is not regarded as completed until it is transported back from the subcontractor. Therefore, a transportation delay τ may be incurred after an outsourced job is completed by the subcontractor. Transportation costs may also be incurred for the outsourced jobs. To reduce transportation costs, the outsourced jobs may be transported in batches where each batch incurs a fixed transportation cost D. A group of jobs forms a batch if all of these jobs are dispatched to the in-house plant together in a single delivery. The batch delivery date is equal to the completion time of the last job in the batch.

For any schedule σ, we define the following notation:
1: a single machine;
s-batch: serial batching ;
$C_j(\sigma)$: the completion time of job j coincides with the completion time of the last job in its batch, if job j is processed on the in-house machine; the completion time of job j is the time when job j arrives at the in-house plant, if job j is outsourced;
$L_j(\sigma) = C_j(\sigma) - d_j$, the lateness of job j;
$U_j(\sigma) = 1$, if $C_j > d_j$;
$U_j(\sigma) = 0$, if $C_j \le d_j$;
G: the total outsourcing cost;

$P = \sum_{j=1}^{n} p_j$: sum of processing times of all jobs;

$P_j = \sum_{i=1}^{j} p_i$: sum of processing times of jobs 1,..., j.

Using the conventional three-field notation of Graham et al. [13]. The scheduling problems with outsourcing allowed is denoted by

$$1+1\left|s-batch\right|\lambda\sum C_j+(1-\lambda)G,$$

$$1+1\left|s-batch\right|\lambda L_{\max}+(1-\lambda)G,$$

and $1+1\left|s-batch\right|\lambda\sum w_jU_j+(1-\lambda)G$, where λ denotes a constant cost parameter and $0 < \lambda < 1$.

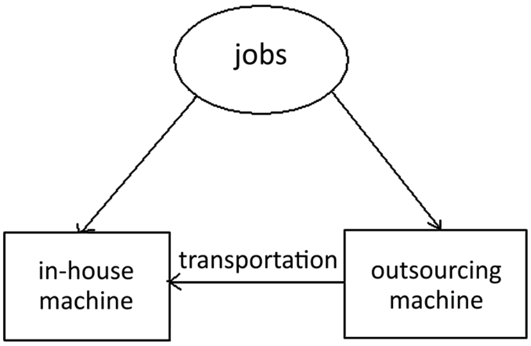

Figure 2 Problem model

3 Main results

In this section, we first present the complexity of discussed problems.

Lemma 1 ([7]): The problems $1+1\,||\,L_{\max}+G$ and $1+1\,||\sum U_j+G$ are NP-hard in the ordinary sense.

Based on **Lemma 1**, we can obtain that the problems $1+1\left|s-batch\right|\lambda L_{\max}+(1-\lambda)G$ and $1+1\left|s-batch\right|\lambda\sum w_jU_j+(1-\lambda)G$ are at least NP-hard in the ordinary sense. For the problem $1+1\left|s-batch\right|\lambda\sum C_j+(1-\lambda)G$, we show that it is solvable in polynomial time.

3.1 Sum of completion times

We first study $1+1\left|s-batch\right|\lambda\sum C_j+(1-\lambda)G$ (i.e., minimizing the weighted sum of the total completion time and the total outsourcing cost).

Lemma 2: For $1+1|s-batch|\lambda\sum C_j+(1-\lambda)G$, there exists an optimal solution in which jobs are sequenced in SPT order at both the in-house machine and the outsourcing machine (i.e., jobs are sequenced in the non-decreasing order of processing times on both machines).

Proof: The claim is clearly true for the jobs on the in-house machine as a conventional serial batch machine scheduling problem. Now assume that the outsourcing machine contains jobs that are not sequenced in SPT order. Consider an optimal schedule *σ** on the outsourcing machine. Since all jobs within a delivery batch can be sequenced in SPT order without affecting the cost, we assume that this SPT property for jobs within a batch holds for *σ**. If *σ** contains jobs that are not sequenced in SPT order, then there is a job *j* that is the last job to be processed in some batch *B*, and another job *i* that is the first job to be processed in the next batch *B′* where $p_j > p_i$. Consider another schedule *σ* that is created by interchanging jobs *j* and *i*, and forming delivery batches containing jobs $B\cup\{i\}\backslash\{j\}$ and $B'\cup\{j\}\backslash\{i\}$ in *σ*, where the first of these batches is delivered at the earlier time than batch *B* in *σ** and the second of these batches is delivered at the same time as batch *B′* in *σ**. All other delivery batches are identical and delivered at the same time in *σ* as in *σ**. A finite number of repetitions of this argument establishes that there exists an optimal schedule in which the jobs are sequenced in SPT order. This completes the proof.

As a result of **Lemma 2**, the jobs are indexed in SPT order such that $p_1 \le p_2 \le \cdots \le p_n$. Define $F(j, m, k, k', h, h')$ as the minimum weighted cost of the objective function for a sub-schedule satisfying the following conditions:

(1) The jobs in consideration are $j, \ldots, n$.

(2) There are *m* jobs scheduled at in-house plant.

(3) The earliest batch has totally *k* jobs to be scheduled on the in-house machine, there are *k′* jobs already being scheduled in the earliest batch, where $k-k'$ is the number of jobs to be scheduled in the earliest batch from jobs $1, \ldots, j-1$.

(4) The earliest batch has totally *h* jobs to be scheduled on the outsourcing machine, there are *h′* jobs already being scheduled in the earliest batch, where $h-h'$ is the number of jobs to be scheduled in the earliest batch from jobs $1, \ldots, j-1$.

(5) Where $1 \le j \le n$, $0 \le m \le n-j+1$, $0 \le k \le m$, $0 \le k' \le k$, $0 \le h \le n-m$, $0 \le h' \le h$.

In any such schedule, there are two possible cases: either job *j* is the first job on the in-house machine or the first job on the outsourcing machine. Now, we propose the following dynamic programming algorithm to solve $1+1|s-batch|\lambda\sum C_j+(1-\lambda)G$.

Algorithm DP1

The boundary conditions: If $h = h' = 0$, it implies that all jobs $j, \ldots, n$ are to be processed on the in-house machine. So we have

$F(j, m, k, k', 0, 0) =$

$$=\min\begin{cases} F(j+1,m-1,k,k'-1,0,0) & \\ +\lambda(m+k-k')p_j & m=n-j+1, k'>1 \\ \min\limits_{k''\le m-1}\{F(j+1,m-1,k'',k'',0,0)\} & \\ +\lambda(m+k-1)(s+p_j) & m=n-j+1, k'=1 \\ +\infty & \text{otherwise} \end{cases}$$

If $k = k' = 0$, then $m = 0$, it implies that all jobs are outsourced. So we have

$F(j, 0, 0, 0, h, h',) =$

$$=\min\begin{cases} F(j+1,0,0,0,h,h'-1,) & \\ +\lambda((n-j+1+h-h')\alpha p_j+\tau)+(1-\lambda)\beta p_j & h'>1 \\ \min\limits_{h''\le n-j+1}\{F(j+1,0,0,0,h'',h'')\} & \\ +\lambda((n-j+h)\alpha p_j+\tau)+(1-\lambda)(\beta p_j+D) & h'=1 \\ +\infty & \text{otherwise} \end{cases}$$

For $j = n$, if job *j* is outsourced, we have

$$F(n,0,0,0,h,1)=\lambda(h\alpha p_n+\tau)+(1-\lambda)(D+\beta p_n).$$

For $j = n$, if job *j* is processed on the in-house machine, we have $F(n,1,k,0,0,)=\lambda k(s+p_n)$.

The recursive function: If $k' > 1$, $h' > 1$,

$F(j, m, k, k', h, h') =$

$$=\min\begin{cases} F(j+1,m-1,k,k'-1,h,h') & \\ +\lambda(m+k-k')p_j & (1) \\ F(j+1,m,k,k',h,h'-1) & \\ +\lambda((n-j+1+h-h')\alpha p_j+\tau)+(1-\lambda)\beta p_j & (2) \end{cases}$$

If $k' = 1$ and job *j* is processed on the in-house machine, job *j* must be in a new batch. If $h' = 1$ and job *j* is outsourced, job *j* must be in a new batch different from the jobs $j+1, \ldots, n$. Thus,

$F(j, m, k, 1, h, 1) =$

$$=\min\begin{cases} \min\limits_{k''\le m-1}\{F(j+1,m-1,k'',k'',h,1)\} & \\ +\lambda(m+k-1)(s+p_j) & (3) \\ \min\limits_{h''\le n-j-m}\{F(j+1,m,k,1,h'',h'')\} & \\ +\lambda((n-j-m+h)\alpha p_j+\tau)+(1-\lambda)(D+\beta p_j) & (4) \end{cases}$$

If $k' > 1$, $h' = 1$,

$F(j, m, k, k', h, 1) =$

$$=\min\begin{cases} F(j+1,m-1,k,k'-1,h,1) & \\ +\lambda(m+k-k')p_j & (5) \\ \min\limits_{h''\le n-j-m}\{F(j+1,m,k,k',h'',h'')\} & \\ +\lambda((n-j-m+h)\alpha p_j+\tau)+(1-\lambda)(D+\beta p_j) & (6) \end{cases}$$

If $k' = 1$, $h' > 1$,

$$F(j, m, k, 1, h, h') = \\ = \min\begin{cases} \min\limits_{k''\le m-1}\{F(j+1, m-1, k'', k'', h, h')\} \\ +\lambda(m+k-1)(s+p_j) & (7) \\ F(j+1, m, k, 1, h, h'-1) \\ +\lambda((n-j+1-m+h-h')\alpha p_j+\tau)+(1-\lambda)\beta p_j & (8) \end{cases}$$

Optimal solution value:

$\min\{F(1,m,k,k,h,h) \mid 0\le m\le n, 0\le k\le m, 0\le h\le n-m\}$

Some remarks should be made about algorithm DP1. In the recursive function, formulae (1) and (5) denote job j is processed on the in-house machine and append job j to the earliest batch, formulae (2) and (8) denote job j is processed on the outsourced machine and append job j to the earliest batch, formulae (3) and (7) denote job j is processed on the in-house machine and form a new batch of job j only, formulae (4) and (6) denote job j is processed on the outsourced machine and form a new batch of job j only.

Theorem 1. Algorithm DP1 gives an optimal schedule for $1+1|s-batch|\lambda\sum C_j+(1-\lambda)G$ in $O(n^6)$ time.

Proof: There are at most $O(n^6)$ states in the dynamic program. If $k' > 1$, $h' > 1$, each iteration costs a constant time; For other cases, there are $O(n^5)$ possible states, each of which takes $O(n)$ time. Therefore, the overall time complexity of Algorithm DP1 is $O(n^6)$.

3.2 Maximum Lateness

Lemma 3: For $1+1|s-batch|\lambda L_{\max}+(1-\lambda)G$, there exists an optimal schedule in which the jobs are sequenced according to an earliest due date (EDD) rule at the in-house machine and the outsourcing machine.

Proof: The claim is clearly true for the jobs on the in-house machine as a conventional single serial batch machine scheduling problem. Now assume that the outsourcing machine contains jobs that are not sequenced in EDD order. Consider an optimal schedule σ* on the outsourcing machine. Since all jobs within a delivery batch can be sequenced in EDD order without affecting the cost, we assume that this EDD property for jobs within a batch holds for σ*. If σ* contains jobs that are not sequenced in EDD order, then we must have a job j that is the last job to be processed in some batch B, and another job i that is the first job to be processed in the next batch B', where $d_j > d_i$. Consider another schedule σ that is created by interchanging jobs j and i, and forming delivery batches containing jobs $B\backslash\{j\}$ and $B'\cup\{j\}$ in σ, where the first of these batches is delivered at the same time as batch B in σ* and the second of these batches is delivered at the same time as batch B' in σ*. All other delivery batches are identical and delivered at the same time in σ as in σ*. Only job j is delivered later in σ than in σ*. However, since job j and i are delivered at the same time in σ and $d_j > d_i$, we have $L_j(\sigma) < L_i(\sigma) = L_i(\sigma^*)$, which establishes that σ is another optimal schedule. A finite number of repetitions of this argument establishes that there exists an optimal schedule in which the jobs are sequenced in EDD order. This completes the proof.

As a result of **Lemma 3**, the jobs are indexed in EDD order such that $d_1 \le d_2 \le \cdots \le d_n$.

Define $L(j, t_1, d, x, h, t)$ as the minimum value of the maximum lateness of the partial schedule satisfying the following conditions:

(1) The jobs in consideration are 1, …, j.
(2) The completion time of the last batch on the in-house machine is t_1.
(3) d is the due date of the first job in the last batch of the current partial schedule at in-house plant.
(4) The sum of processing times of the outsourced jobs is αx.
(5) The last outsourced one is in the h-th batch on the outsourcing machine.
(6) The h-th batch is delivered at time αt.
(7) Where $1 \le j \le n$, $0 \le t_1 \le ns + P$, $d \in \{d_1, …, d_n\}$, $0 \le x \le P_j$, $0 \le h \le j$, $x \le t \le x + P$.

Let $L_O(j, t_1, d, x, h, t)$ be the minimum maximum lateness if job j is outsourced , let $L_I(j, t_1, d, x, h, t)$ be the minimum maximum lateness if job j is processed at in-house plant. Obviously, we have $L(j, t_1, d, x, h, t) = \min\{L_O(j, t_1, d, x, h, t), L_I(j, t_1, d, x, h, t)\}$. Now, we propose the following dynamic programming algorithm to solve the problem $1+1|s-batch|\lambda L_{\max}+(1-\lambda)G$.

Algorithm DP2
The boundary conditions:

$$L(1, t_1, d, x, h, t) = \\ = \begin{cases} s+p_1-d_1 & t_1 = s+p_1, x=0, h=0, t=0 \\ \alpha t+\tau-d_1 & t_1=0, x=p_1, h=1, t\ge p_1 \\ +\infty & \text{otherwise} \end{cases}$$

The recursive function: If job j is outsourced,

$$L_O(j, t_1, d, x, h, t) = \\ = \min\begin{cases} \max\{L(j-1, t_1, d, x-p_j, h-1, x-p_j), \alpha t+\tau-d_j\} & (9) \\ L(j-1, t_1, d, x-p_j, h, t) & (10) \end{cases}$$

If job j is processed at in-house production,

$$L_I(j, t_1, d, x, h, t) = \\ = \min\begin{cases} \max\{L(j-1, t_1-p_j, d, x, h, t), t_1-d\} & (11) \\ \max\left\{\min\limits_{d'\in\{d_1,\ldots,d_{j-1}\}}\{L(j-1, t_1-p_j-s, d', x, h, t), t_1-d_j\}\right\} & (10) \end{cases}$$

Optimal solution value:

$\min\{\lambda L(n,t_1,d,x,h,x)+(1-\lambda)(\beta x+hD)\}$ where

$0 \le t_1 \le ns + P$, $0 \le x \le P$, $0 \le h \le n$, $d \in \{d_1, …, d_n\}$.

Some remarks should be made about algorithm DP2. In the recursive function, if job j is outsourced, formula (9) denotes job j is the first job in the h-batch, formula (10) denotes job j is not the first job in the h-batch; If job j is processed at in-house production, formula (11) denotes job *j* is appended to the last batch, formula (12) denotes job *j* belongs to a new batch.

Theorem 2: Algorithm DP2 gives an optimal schedule for $1+1|s-batch|\lambda L_{\max}+(1-\lambda)G$ in $O(n^3P^3)$ time.

Proof: There are at most $O(n^3P^3)$ states in the dynamic program. Each iteration costs a constant time. Therefore, the overall time complexity of Algorithm DP2 is $O(n^3P^3)$.

3.3 Weighted number of late jobs

Lemma 4: For $1+1|s-batch|\lambda\sum w_jU_j+(1-\lambda)G$, there exists an optimal schedule in which the on-time jobs are sequenced according to an earliest due date (EDD) rule on both the in-house and the outsourcing machines, and all tardy jobs are processed on the in-house machine following the on-time jobs.

Proof: The EDD property can be built similarly to **Lemma 3**. The claim of no outsourced jobs being tardy is due to the fact that if an outsourced job is tardy, we can always make it in-house to save outsourcing costs.

As a result of **Lemma 4**, the on-time jobs are indexed in EDD order. For convenience in presentation, here we instead consider the maximization of the weighted number of non-tardy jobs. A job is non-tardy if the batch containing it is fully finished no later than its due date. Define $V(j, t_1, d, x, h, t)$ as the maximum weighted number of non-tardy jobs for a sub-schedule satisfying the following conditions:

(1) The jobs in consideration are $1, \ldots, j$.
(2) The completion time of the last batch on the in-house machine is t_1.
(3) d is the due date of the first job in the last batch of the current partial schedule at in-house production.
(4) The sum of processing times of the outsourced jobs is αx.
(5) The last outsourced one is in the h-th batch on the outsourcing machine.
(6) The h-th batch is delivered at time αt.
(7) Where $1 \le j \le n$, $0 \le t_1 \le ns + P$, $d \in \{d_1, \ldots, d_n\}$, $0 \le x \le P_j$, $0 \le h \le j$, $x \le t \le x + P$.

Let $V_O(j, t_1, d, x, h, t)$ be the maximum weighted number of non-tardy jobs if job J_j is outsourced, $V_I(j, t_1, d, x, h, t)$ be the maximum weighted number of non-tardy jobs if job j is not outsourced. Obviously, we have $V(j, t_1, d, x, h, t) = \max\{V_O(j, t_1, d, x, h, t), V_I(j, t_1, d, x, h, t)\}$.

When considering job *j*, we may encounter three possible situations: job *j* is discarded and treated to be tardy; job *j* is scheduled early or on time and solely forms a batch; or job *j* is appended to an existing batch without making other jobs in the same batch tardy. With the observations, we can formulate the following recursive program for composing a schedule with a maximum weighted number of non-tardy jobs.

Algorithm DP3
The boundary conditions:
$V(1, t_1, d, x, h, t) =$

$$=\begin{cases} w_1 & x=0, h=0, t=0, t_1=s+p_1\le d_1 \\ w_1 & t_1=0, x=p_1, h=1, \alpha t+\tau\le d_1 \\ 0 & x=0, h=0, t=0, t_1=0 \\ -\infty & \text{otherwise} \end{cases}$$

The recursive function: If job *j* is processed in batches on the in-house machine,

$V_I(j, t_1, d, x, h, t) =$

$$=\max\begin{cases} V(j-1, t_1, d, x, h, t) & \\ \max\limits_{d'\in\{d_1,\ldots,d_{j-1}\}}\{V(j-1, t_1-s-p_j, d', x, h, t)\}+w_j & t_1\le d_j \\ V(j-1, t_1-p_j, d, x, h, t)+w_j & t_1\le d_j \\ -\infty & \text{otherwise} \end{cases}$$

If job *j* is outsourced,

$V_O(j, t_1, d, x, h, t) =$

$$=\max\begin{cases} V(j-1, t_1, d, x-p_j, h, t)+w_j & \alpha t+\tau\le d_j \\ V(j-1, t_1, d, x-p_j, h-1, x-p_j)+w_j & \alpha t+\tau\le d_j \\ -\infty & \text{otherwise} \end{cases}$$

Optimal solution value:

$$\min\left\{\lambda(\sum_{j=1}^{n} w_j - V(n, t_1, d, x, h, x)) + (1-\lambda)(\beta x + hD)\right\}$$

where $0 \le t_1 \le ns + P$, $0 \le x \le P$, $0 \le h \le n$, $d \in \{d_1, \ldots, d_n\}$.

Theorem 3: Algorithm DP3 gives an optimal schedule for $1+1|s-batch|\lambda\sum w_jU_j+(1-\lambda)G$ in $O(n^3P^3)$ time.

Proof: There are at most $O(n^3P^3)$ states in the dynamic program. Each iteration costs a constant time. Therefore, the overall time complexity of Algorithm DP3 is $O(n^3P^3)$.

4 Conclusion

We have studied a single machine serial batch scheduling problems with outsourcing allowed. We proposed three dynamic programming algorithms for the three scheduling problems with the option of outsourcing.

For the first problem, it is optimally solved in polynomial time, for the other two problems, they are solved in pseudo-polynomial time. For future research, it would be interesting to focus on the scheduling problems with multiple available subcontractors, other objective functions and more complex outsourcing models.

Acknowledgements

We thank two anonymous reviewers for their helpful and detailed comments on an earlier version of our paper. This paper is supported by the National Natural Science Foundation of China (11201259), Promotive research fund for young and middle-aged scientists of Shandong Province (BS2013SF016) and the Doctoral Fund of the Ministry of Education (20123705120001, 20123705110003).

5 References

[1] Coffman, E. G.; Yannakakis, M.; Magazine, M. J.; Santos, C. Batch sizing and sequencing on a single machine. // Operations Research. 51, 4(1990), pp. 566-584. DOI: 10.1007/BF02248589

[2] Webster, S.; Baker, K. R. Scheduling groups of jobs on a single machine. // Operations Research. 43, 4(1995), pp. 692-703. DOI: 10.1287/opre.43.4.692

[3] Brucker, P.; Kovalyov, M. Y. Single machine batch scheduling to minimize the weighted number of late jobs. // Mathematical Methods of Operations Research. 43, 1(1996), pp. 1-8. DOI: 10.1007/BF01303431

[4] Cheng, T. C. E.; Gordon, V. S.; Kovalyov, M. Y. Single machine scheduling with batch deliveries. // European Journal of Operational Research. 94, 2(1996), pp. 277-283. DOI: 10.1016/0377-2217(96)00127-0

[5] Cheng, T. C. E.; Kovalyov, M. Y.; Gordon, V. S.; Lin, B. M. T. Single machine scheduling to minimize batch delivery and job earliness penalties. // SIAM Journal on Optimization. 7, 2(1997), pp. 547-559. DOI: 10.1137/S1052623494269540

[6] Hall, N. G.; Potts, C. N. Supply Chain Scheduling: Batching and Delivery. // Operations Research. 51, 4(2003), pp. 566-584. DOI: 10.1287/opre.51.4.566.16106

[7] Qi, X. Coordinated logistics scheduling for in-house production and outsourcing. // IEEE Transactions on Automation Science and Technology. 5, 1(2008), pp. 188-192. DOI: 10.1109/TASE.2006.887159

[8] Qi, X. Outsourcing and production scheduling for a two-stage flow shop. // International Journal of Production Economics. 129, 1(2011), pp. 43-50. DOI: 10.1016/j.ijpe.2010.08.011

[9] Ruiz-Torres, A. J.; Ho, J. S.; Lopez, F. J. Generating Pareto schedules with outsource and internal parallel resources. // International Journal of Production Economics. 103, 2(2006), pp. 810-825. DOI: 10.1016/j.ijpe.2005.11.010

[10] Lee, I. S.; Sung, C. S. Minimizing due date related measures for a single machine scheduling problem with outsourcing allowed. // European Journal of Operational Research, 186, 3(2008), pp. 931-952. DOI: 10.1016/j.ejor.2007.02.015

[11] Lee, I. S.; Sung, C. S. Single machine scheduling with outsourcing allowed. // International Journal of Production Economics. 111, 2(2008), pp. 623-634. DOI: 10.1016/j.ijpe.2007.02.036

[12] Chen, Z. L.; Li, C. L. Scheduling with subcontracting options. // IIE Transactions. 40, 12(2008), pp. 1171-1184. DOI: 10.1080/07408170801975057

[13] Graham, R. L.; Lawler, E. L.; Lenstra, J. K.; Rinnooy Kan, A. H. G. Optimization and approximation indeterministic sequencing and scheduling: A survey. // Annals Discrete Mathematics. 5, (1979), pp. 287-326. DOI: 10.1016/S0167-5060(08)70356-X

[14] Chung, D. Y.; Choi, B. C. Outsourcing and scheduling for two-machine ordered flow shop scheduling problems. // European Journal of Operational Research, 226, 1(2013), pp. 46-52. DOI: 10.1016/j.ejor.2012.10.048

[15] Mokhtari, H.; Abadi, I. N. K. Scheduling with an outsourcing option on both manufacturer and subcontractors. // Computers & Operations Research. 40, 5(2013), pp. 1234-1242. DOI: 10.1016/j.cor.2012.12.003

Authors' addresses

Juan Zou
School of Management,
Qufu Normal University,
Rizhao, Shandong, 276826, China

School of Mathematical Sciences,
Qufu Normal University,
Qufu, Shandong, 273165, China
E-mail: zoujuanjn@163.com

Yuzhong Zhang, Professor
School of Management,
Qufu Normal University,
Rizhao, Shandong, 276826, China

Cuixia Miao, Associate Professor
School of Mathematical Sciences,
Qufu Normal University,
Qufu, Shandong, 273165, China