J Sched
DOI 10.1007/s10951-009-0134-8

Polynomial-time approximation schemes for scheduling problems with time lags

Xiandong Zhang · Steef van de Velde

Abstract We identify two classes of machine scheduling problems with time lags that possess Polynomial-Time Approximation Schemes (PTASs). These classes together, one for minimizing makespan and one for minimizing total completion time, include many well-studied time lag scheduling problems. The running times of these approximation schemes are polynomial in the number of jobs, but exponential in the number of machines and the ratio between the largest time lag and the smallest positive operation time.

These classes constitute the first PTAS results for scheduling problems with time lags.

Keywords Machine scheduling · Time lags · Approximability · Polynomial-Time Approximation Scheme (PTAS)

1 Introduction

1.1 Problem description

Machine scheduling problems with time lags can arise both in multi-stage and single-stage processing environments as long as the jobs to be processed consist of multiple operations. A time lag, after all, specifies a minimum delay between the execution of two consecutive operations of the same job. Time lags can model the transportation times between machines when the number of vehicles is not restrictive or when the jobs travel by themselves, for example, like barges sailing between container terminals along the river bank and trucks traveling between pick up and delivery depots. Time lags can also model activities that require no limited resources, for instance, cooling-down times. These practical justifications explain why time lags are sometimes also referred to in the literature as *transportation times* or *delays*.

We assume that there are m machines $M_1, \ldots, M_m$ available from time zero onwards for processing a set of n jobs $J_1, \ldots, J_n$, each consisting of o operations $(O_{1j}, \ldots, O_{oj})$. Every operation O_{ij} $(i = 1, \ldots, o;\ j = 1, \ldots, n)$ needs to be processed during an uninterrupted processing time $p_{ij} \geq 0$ on a dedicated machine $\mu_{ij} \in \{M_1, \ldots, M_m\}$ and the operations of the same job cannot be processed simultaneously. Each machine can handle only one operation at a time. Each job J_j $(j = 1, \ldots, n)$ may have a release time r_j before which no operation of J_j can be started.

We consider only the basic (and most common) time lag scheduling models with exactly one machine per stage. We consider three such multi-stage scheduling environments with $o = m$: an *open shop* problem, where the operations of a job can be processed in any order; a *job shop* problem, where the operations of every job J_j $(j = 1, \ldots, n)$ need to processed in the order $O_{1j} \rightarrow O_{2j} \rightarrow \cdots \rightarrow O_{oj}$; and a *flow shop*, which is essentially a job shop with the special condition that $\mu_{ij} = M_i$ for each operation O_{ij} $(i = 1, \ldots, m;\ j = 1, \ldots, n)$. Hence in a flow shop, all jobs pass through the machines in the same order $M_1 \rightarrow M_2 \rightarrow \cdots \rightarrow M_m$.

In the single-stage scheduling environment with time lags that we consider, there is only a single machine M_1 available for processing jobs $J_1, \ldots, J_n$, each of which consists of a chain of two operations (O_{1j}, O_{2j}) that need to

X. Zhang (✉)
Department of Management Science, School of Management, Fudan University, Shanghai 200433, China
e-mail: xiandongzhang@fudan.edu.cn

S. van de Velde
Rotterdam School of Management, Erasmus University, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands
e-mail: svelde@rsm.nl

be processed in the order $O_{1j} \to O_{2j}$ for each j ($j = 1, \ldots, n$). Hence, in this environment we have $m = 1$, $o = 2$, and $\mu_{1j} = \mu_{2j} = M_1$ for each j ($j = 1, \ldots, n$). These problems are commonly referred to as single machine *coupled operations* scheduling problems (Potts and Whitehead 2007).

In an open shop environment, time lags take the form l_{hij} ($j = 1, \ldots, n; h, i = 1, \ldots, m$), specifying the minimum required delay between the completion of operation O_{hj} and the start of operation O_{ij}. If $l_{hij} = l_{ihj}$ for all jobs J_j ($j = 1, \ldots, n$) and all operations O_{hj} and O_{ij} ($h, i = 1, \ldots, m; h \neq i$), then the time lags are called symmetrical. In flow shops, job shops, and single machine shops with coupled operations, the order of the operations is fixed for every job, and hence the given time lags are of the form l_{ij}, specifying the required minimum time lag between operation O_{ij} and $O_{i+1,j}$. Special time lag cases include the situation where the time lags between operations of job J_j are of the form l_j for each $j = 1, \ldots, n$ and the situation where all time lags equal a given positive constant l.

A schedule specifies for every operation when it is executed such that all constraints are satisfied; in other words, it specifies for every operation O_{ij} a starting time S_{ij} and a completion time C_{ij} such that all conditions are met ($i = 1, \ldots, o; j = 1, \ldots, n$). In this paper, we consider two scheduling objectives; the minimization of the *makespan*, or length, $C_{\max}$ of the schedule, and the minimization of the sum of the job completion times or *total flow time* $\sum C_j$, where C_j denotes the completion time of the last processed operation of job J_j ($j = 1, \ldots, n$). Throughout this paper, we follow the standard three-field $\alpha|\beta|\gamma|$ scheduling notation (Graham et al. 1979). For instance, $F2|l_j|C_{\max}$ denotes the problem of minimizing makespan in a two-machine flow shop with job-dependent time lags; $J2|o_j \leq 2|\sum C_j$ denotes the problem of minimizing total completion time in the two-machine job shop where $o_j \leq 2$ denotes that each job consists of no more than two operations; $O3|l_{hij} = l|C_{\max}$ denotes the problem of minimizing makespan in the three-machine open shop with equal time lags; and $1|o_j = 2, l_j|C_{\max}$ denotes the single-machine problem of scheduling jobs with exactly two coupled operations to minimize makespan subject to job-dependent time lags.

1.2 Complexity

Scheduling problems with time lags are strongly $\mathcal{NP}$-hard in their most general forms, and hence it is very unlikely that they can be solved to optimality in polynomial time. For makespan minimization problems, for instance, Yu et al. (2004) showed that $F2|l_j|C_{\max}$ and $O2|l_j|C_{\max}$ are $\mathcal{NP}$-hard in the strong sense, even if all processing times are equal to 1. From the first result, it follows, of course, that $J2|l_j|C_{\max}$ is $\mathcal{NP}$-hard, too. Kern and Nawijn (1991) showed that the problem $1|o_j \leq 2, l_j|C_{\max}$ is $\mathcal{NP}$-hard; Gupta (1996) strengthened the result by proving that the problem is $\mathcal{NP}$-hard in the strong sense. Further $\mathcal{NP}$-hardness results are known for even more restrictive problems such as $O2|l_{hij} = l|C_{\max}$ and $O|p_{ij} = 1, l_{hij} = l_{ihj}|C_{\max}$ (Rayward-Smith and Rebaine 1992), and $F2|p_{1j} = p_{2j}, l_j \in \{l_1, l_2\}|C_{\max}$, which is $\mathcal{NP}$-hard in the strong sense (Yu 1996).

As far as the total flow time objective is concerned, Garey et al. (1976) and Achugbue and Chin (1982) proved that $F2 \parallel \sum C_j$ and $O2 \parallel \sum C_j$, respectively, are strongly $\mathcal{NP}$-hard. The implication is that two-machine flow shop, job shop and open shop problems to minimize total flow subject to time lags are $\mathcal{NP}$-hard in the strong sense, too, even if all time lags are the same. Brucker et al. (2004) have shown that the problem $F2|p_{ij} = p, r_j, l_j|\sum C_j$ is $\mathcal{NP}$-hard, also.

The best well-known polynomially solvable problem is undoubtedly $F2|l_j|C_{\max}$ if the solution space is restricted to permutation schedules (Mitten 1959). Other polynomially solvable cases are more restrictive, such as $F2|l_j = l|C_{\max}$, $O2|p_{ij} = p, l_{ij}|C_{\max}$ and $F|p_{ij} = p, l_j|\sum C_j$ (Brucker et al. 2004).

1.3 Approximability

In this paper, we are concerned with the approximability of scheduling models with time lags. We define an approximation algorithm to have *performance ratio* or *worst-case ratio* ρ, with $\rho > 1$, if it always produces a solution with the objective value at most ρ times the optimal solution value. If such an algorithm runs in polynomial time, we call it a ρ-approximation algorithm. A *Polynomial-Time Approximation Scheme* (PTAS) is a family of polynomial time $(1 + \epsilon)$-approximation algorithms over all $\epsilon > 0$.

The approximability of scheduling problems with time lags, and in particular the design of PTASs, is largely uncharted territory. Approximability results are limited to makespan minimization and concern ρ-approximation results only. For the two-machine flow shop environment, Dell'Amico (1996) provided a 2-approximation algorithm for $F2|l_j|C_{\max}$. Karuno and Nagamochi (2003) improved on this and gave an $\frac{11}{6}$-approximation algorithm. Ageev (2008) showed that the worst case ratio could be improved to $\frac{3}{2}$ if $p_{1j} = p_{2j}$ for each job J_j ($j = 1, \ldots, n$).

For the two-machine open shop environment, Strusevich and Rebaine (1995) presented a $\frac{7}{4}$-approximation algorithm for $O2|l_{hij} = l_{ihj}|C_{\max}$. This bound was improved to $\frac{3}{2}$ by Strusevich (1999). Rebaine and Strusevich (1999) presented an $\frac{8}{5}$-approximation algorithm for $O2|l_{ij}|C_{\max}$. Rebaine (2004) presented a 2-approximation algorithm for $O2|l_{hij}|C_{\max}$ and a $(\frac{7}{4} - \frac{1}{2n})$-approximation algorithm for $O2|p_{ij} = p, l_{hij}|C_{\max}$.

Table 1 Some problems in class $\mathcal{M}$

Problem (P)	Complexity (P)	Complexity ($\bar{\text{P}}$)
$F2\|l_j\|C_{\max}$	! (Yu 1996)	$\mathcal{P}$ (Johnson 1954)
$F2\|r_j, l_j\|C_{\max}$	! (Yu 1996)	PTAS (Hall 1994; Kovalyov and Werner 1997)
$Fm\|p_{ij} = 1, tree, l_{jkl}\|C_{\max}$	! (Yu 1996)	$\mathcal{P}$ (Bruno and Jones 1980)
$O2\|l_{hij}\|C_{\max}$	! (Dell'Amico and Vaessens 1995)	$\mathcal{P}$ (Gonzalez and Sahni 1976)
$J2\|o_j \le 2, l_j\|C_{\max}$	! (Yu 1996)	$\mathcal{P}$ (Jackson 1956)
$J2\|p_{ij} = 1, r_j, l_j\|C_{\max}$	! (Yu 1996)	$\mathcal{P}$ (Timkovsky 1997)
$1\|o_j = 2, l_j\|C_{\max}$	! (Gupta 1996)	$\mathcal{P}$ (trivial)

Table 2 Two earlier studied problems in class $\mathcal{F}$

Problem (P)	Complexity (P)	Complexity ($\bar{\text{P}}$)
$F2\|p_{ij} = 1, r_j, l_{ij}\|\sum C_j$	! (Brucker et al. 2004)	$\mathcal{P}$ (Baptiste and Timkovsky 2004)
$Fm\|l_{ij}\|\sum C_j$	! (Garey et al. 1976)	PTAS (Fishkin et al. 2002)

2 Our contribution

As pointed out in the previous section, approximability of scheduling problems with time lags is a largely unexplored area. In this paper, we present the first PTASs for scheduling problems with time lags, both for minimizing makespan and minimizing total flow time. Specifically, we identify a class $\mathcal{M}$ for makespan minimization problems with time lags and a class for $\mathcal{F}$ for total flow time minimization problems with time lags with the property that every scheduling problem (P) in those classes has a PTAS.

Every problem (P) in class $\mathcal{M}$ and class $\mathcal{F}$ has the following properties:

(i) Problem (P) is a deterministic scheduling problem with time lags, with one machine per manufacturing stage.
(ii) (P) is $\mathcal{NP}$-hard or strongly $\mathcal{NP}$-hard.
(iii) The counterpart problem without time lags, referred to as Problem ($\bar{\text{P}}$), is polynomially solvable or has a PTAS.
(iv) All time lags are finite, that is, there exists a real $\mu > 0$ such that

$$l_{\max} \le \mu p_{\min}, \tag{1}$$

where $l_{\max}$ is the maximum time lag and $p_{\min}$ is the smallest positive processing time of any operation. The value μ is fixed, that is, it is not part of the problem instance.

In addition to Properties (i)–(iv), problems in class $\mathcal{M}$ have the further properties:

(v) The objective is to minimize makespan.
(vi) The scheduling environment is an m-machine flow shop (with m fixed, that is, m is not part of the problem instance but given a priori), a two-machine open shop, a two-machine job shop, or a single-machine shop with at most two coupled operations.
(vii) If the scheduling environment is a flow shop, then positive release times are allowed; otherwise, $r_j = 0$ for each job J_j ($j = 1, \ldots, n$).
(viii) If the scheduling environment is a flow shop with $m > 2$ or positive release times, then every job has at least one operation with a positive processing time.

Table 1 lists some problems that belong to this class $\mathcal{M}$; the sign '!' indicates that the problem is $\mathcal{NP}$-hard in the strong sense, whereas $\mathcal{P}$ indicates that the problem is solvable in polynomial time.

In addition to Properties (i)–(iv), problems in class $\mathcal{F}$ have the following further properties:

(ix) The scheduling objective is to minimize total flow time.
(x) The scheduling environment is an m-machine flow shop where m is fixed.
(xi) Every job has at least one operation with a positive processing time.

Table 2 lists two earlier studied problems belonging to class $\mathcal{F}$.

3 PTAS for makespan problems in class $\mathcal{M}$

Consider any makespan minimization scheduling problem (P) belonging to class $\mathcal{M}$. We start by introducing some notation. Let $C^*_{\max}$ denote the optimal makespan for Problem (P), and let $C_{\max}(\pi)$ denote the minimum makespan of a feasible schedule π for Problem (P). Finally, let $\varepsilon > 0$ be any positive real number.

Define $P_k = \sum_{1\le j\le n, 1\le i\le o \mid \mu_{ij}=M_k} p_{ij}$ as the work content for machine M_k, for $k = 1, \ldots, m$. Clearly, we must have that

$$P_k \leq C^*_{\max}, \quad \text{for } k = 1, \ldots, m. \tag{2}$$

Next, we divide the $n \times o$ operations O_{ij} ($i = 1, \ldots, o$; $j = 1, \ldots, n$) into $3m$ subsets in the following way:

- $\mathcal{Z}_k = \{O_{ij} | \mu_{ij} = M_k \text{ and } p_{ij} = 0\}$ for $k = 1, \ldots, m$; these are the *zero* operations.
- $\mathcal{S}_k = \{O_{ij} | \mu_{ij} = M_k \text{ and } 0 < p_{ij} \leq \frac{\varepsilon}{\mu(o-1)} P_k\}$ for $k = 1, \ldots, m$; these are the *small* operations.
- $\mathcal{L}_k = \{O_{ij} | \mu_{ij} = M_k \text{ and } p_{ij} > \frac{\varepsilon}{\mu(o-1)} P_k\}$ for $k = 1, \ldots, m$; these are the *large* operations.

Note that all zero operations need to be scheduled and executed, although the duration of their processing is zero. The number of large operations per machine is bounded from above; indeed, we have that

$$|\mathcal{L}_k| \leq \frac{\mu(o-1)}{\varepsilon}, \quad \text{for each } k = 1, \ldots, m. \tag{3}$$

To see that this is true, assume the opposite, that is, $|\mathcal{L}_k| > \frac{\mu(o-1)}{\varepsilon}$ for some k ($k = 1, \ldots, m$). The work content induced by the large operations on machine M_k would then be at least

$$\sum_{i,j | O_{ij} \in \mathcal{L}_k} p_{ij} \geq |\mathcal{L}_k| \frac{\varepsilon}{\mu(o-1)} P_k > P_k,$$

which would imply that the total processing time of the large operations on M_k exceeds the total processing time of all operations on M_k, which is a contradiction. We now differentiate between two cases: $\sum_{k=1}^m |\mathcal{S}_k| = 0$ and $\sum_{k=1}^m |\mathcal{S}_k| > 0$.

If $\sum_{k=1}^m |\mathcal{S}_k| = 0$, then we have jobs with large and zero operations only, and we can find an optimal solution for this case in time polynomial in n but exponential in $\frac{\mu(o-1)m}{\varepsilon}$ in the following way. First, we use explicit enumeration to schedule all the jobs with at least one large operation, of which there are at most $\sum_{k=1}^m |\mathcal{L}_k| \leq \frac{\mu(o-1)m}{\varepsilon}$ by virtue of Inequality (3). For m-machine flow shop problems with $m > 2$ or flow shop problems where jobs have positive release times, we are then done since, by definition of class $\mathcal{M}$ and in particular by Property (viii), no job has zero operations only. For any other type of problem in class $\mathcal{M}$, there may be jobs with zero operations only, and we insert those jobs into the intermediate schedule in such a way that feasibility and optimality of the schedule is maintained; this can easily be achieved, for instance, by scheduling all first operations as early as possible and all second operations as late as possible.

Alternatively, if $\sum_{k=1}^m |\mathcal{S}_k| > 0$, then we have at least one operation O_{ij} with $\mu_{ij} = M_k$ and $0 < p_{ij} \leq \frac{\varepsilon}{\mu(o-1)} P_k$, for some i, j, and k. Using (1) and (2), we have

$$l_{\max} \leq \mu \left(\frac{\varepsilon}{\mu(o-1)} P_k \right) \leq \frac{\varepsilon}{o-1} P_k \leq \frac{\varepsilon}{o-1} C^*_{\max}. \tag{4}$$

Let now Problem ($\bar{\text{P}}$) be the counterpart problem of (P) *without* given time lags; by definition of class $\mathcal{M}$, ($\bar{\text{P}}$) is either solvable in polynomial time or has a PTAS. If ($\bar{\text{P}}$) is solvable in polynomial time, then let $\bar{\sigma}^*$ denote an optimal schedule for this problem. Otherwise, that is, if ($\bar{\text{P}}$) is $\mathcal{N}P$-hard but has a PTAS, let $\bar{\sigma}^\varepsilon$ denote a feasible schedule for ($\bar{\text{P}}$) with makespan at most $(1+\epsilon)\bar{C}^*_{\max}$, where $\bar{C}^*_{\max}$ is the optimal makespan for problem ($\bar{\text{P}}$). Of course, we have that $\bar{C}^*_{\max} \leq C^*_{\max}$.

Let now σ be the schedule obtained from either $\bar{\sigma}^*$ or $\bar{\sigma}^\varepsilon$, whichever is appropriate, by inserting as little idle time as possible between the different operations to ensure that σ is a feasible schedule for Problem (P). How much idle time needs to be inserted, and subsequently, how good is $C_{\max}(\sigma)$?

First, suppose that ($\bar{\text{P}}$) is an m-machine flow shop problem. Then we need to insert no more than $l_{\max}$ idle time on each machine $M_2, \ldots, M_m$ to transform either $\bar{\sigma}^*$ or $\bar{\sigma}^\varepsilon$ into a feasible solution σ for Problem (P). Accordingly, we have

$$C_{\max}(\sigma) \leq C_{\max}(\bar{\sigma}^*) + (m-1)l_{\max} \leq (1+\varepsilon)C^*_{\max},$$

if σ has been obtained from $\bar{\sigma}^*$,

and

$$C_{\max}(\sigma) \leq C_{\max}(\bar{\sigma}^\varepsilon) + (m-1)l_{\max} \leq (1+2\varepsilon)C^*_{\max},$$

if σ has been obtained from $\bar{\sigma}^\varepsilon$.

Now suppose that ($\bar{\text{P}}$) is any other type of problem than a flow shop problem, that is, suppose it is a two-machine job shop, two-machine open shop, or a single machine shop with at most two coupled operations. We will first show that no operation in schedule $\bar{\sigma}^*$ or $\bar{\sigma}^\varepsilon$, whichever is appropriate, needs to be delayed by more than $l_{\max}$ time units to obtain a feasible schedule σ for Problem (P). First, suppose that ($\bar{\text{P}}$) is a two-machine job shop problem. Then we may assume, without loss of generality, that in $\bar{\sigma}^*$ or $\bar{\sigma}^\varepsilon$, whichever is appropriate, either machine processes all first operations O_{1j} before any second operation O_{2k}; if $\bar{\sigma}^*$ or $\bar{\sigma}^\varepsilon$ is no such schedule we can easily transform it into an equivalent schedule with the stated property. This property implies that we need to insert no more than $l_{\max}$ idle time on either machine to transform $\bar{\sigma}^*$ or $\bar{\sigma}^\varepsilon$ into a feasible solution σ for Problem (P). Second, suppose that ($\bar{\text{P}}$) is a two-machine open shop problem. An argument similar to the one used for the two-machine job shop problem applies; we may assume without loss of generality that in $\bar{\sigma}^*$ or $\bar{\sigma}^\varepsilon$ all first operations precede all second operations on either machine. Accordingly, the second operations need to be delayed by at most $l_{\max}$ time to transform $\bar{\sigma}^*$ or $\bar{\sigma}^\varepsilon$ into a feasible solution σ for Problem (P). Finally, let ($\bar{\text{P}}$) be a single-machine problem where jobs have at most two coupled operations. Again, we may assume that all first operations precede all second operations, and hence we need to insert at most $l_{\max}$ time in between to guarantee that the resulting schedule σ is feasible for Problem (P). So indeed, if ($\bar{\text{P}}$) is any other problem than an m-machine flow shop problem, we need to insert no more than $l_{\max}$ idle time per machine to obtain a feasible schedule σ for Problem (P). This means that

$$C_{\max}(\sigma) \le C_{\max}(\bar{\sigma}^*) + l_{\max} \le (1+\varepsilon)C^*_{\max},$$

if σ has been obtained from $\bar{\sigma}^*$,

and

$$C_{\max}(\sigma) \le C_{\max}(\bar{\sigma}^\varepsilon) + l_{\max} \le (1+2\varepsilon)C^*_{\max},$$

if σ has been obtained from $\bar{\sigma}^\varepsilon$.

In conclusion, if $\sum_{k=1}^m |\mathcal{S}_k| > 0$, then any makespan minimization Problem (P) in class $\mathcal{M}$ has a PTAS, also. Now, we are ready to give a description of our algorithm.

Algorithm I PTAS for any Problem (P) in class $\mathcal{M}$

Step 1. If $\sum_{k=1}^m |\mathcal{S}_k| > 0$, go to Step 2; otherwise go to Step 3.
Step 2. Find a schedule σ, to be obtained either from $\bar{\sigma}^*$, if ($\bar{\text{P}}$) is polynomially solvable, or from $\bar{\sigma}^\varepsilon$, if ($\bar{\text{P}}$) has a PTAS. Stop.
Step 3. Explicitly enumerate all possible sequences for the jobs with at least one large operation. For jobs with zero operations only, we schedule those jobs by scheduling all first operations as early as possible and all second operations as late as possible. Find thus an optimal schedule σ^* for Problem (P). Stop.

Theorem 1 *Algorithm I is a Polynomial-Time Approximation Scheme for any makespan minimization scheduling problem with time lags in class $\mathcal{M}$. The running time of the algorithm is polynomial in n but exponential in* $\frac{\mu(o-1)m}{\varepsilon}$.

4 PTAS for problems in class $\mathcal{F}$

Let now (P) be any problem belonging to class $\mathcal{F}$, which consists of m-machine flow shop total flow time minimization problems with time lags; see Sect. 2. Let F^* denote the optimal solution value for Problem (P), and let $\varepsilon > 0$ be any real positive number. Let $p_{i[j]}$ denote the jth smallest processing time on machine M_i ($i = 1, \dots, m$; $j = 1, \dots, n$) and define $P_{ik} = \sum_{j=1}^k p_{i[j]}$. We then have that

$$F^* \ge P_{in}, \quad \text{for each } i = 1, \dots, m. \tag{5}$$

Similar to the previous subsection, we divide the $n \times m$ operations into $3m$ subsets:

- $\mathcal{Z}_i = \{O_{ij} | p_{ij} = 0\}$ for $i = 1, \dots, m$; these are the *zero* operations.
- $\mathcal{S}_i = \{O_{ij} | 0 < p_{ij} \le \frac{\varepsilon}{\mu(m-1)} \frac{P_{in}}{n}\}$ for $i = 1, \dots, m$; these are the *small* operations.
- $\mathcal{L}_i = \{O_{ij} | p_{ij} > \frac{\varepsilon}{\mu(m-1)} \frac{P_{in}}{n}\}$, for $i = 1, \dots, m$; these are the *large* operations.

If $\sum_{i=1}^m |\mathcal{S}_i| > 0$, we have at least one operation O_{ij} with $0 < p_{ij} < \frac{\varepsilon}{\mu(m-1)} \frac{P_{in}}{n}$. With (1) and (5), we have

$$l_{\max} \le \mu\left(\frac{\varepsilon}{\mu(m-1)} \frac{P_{in}}{n}\right) \le \frac{\varepsilon}{m-1} \frac{P_{in}}{n} \le \frac{\varepsilon}{m-1} \frac{F^*}{n}. \tag{6}$$

Let now Problem ($\bar{\text{P}}$) be the counterpart problem of (P) *without* given time lags; accordingly, by definition of class $\mathcal{F}$, ($\bar{\text{P}}$) either is solvable in polynomial time or has a PTAS. If ($\bar{\text{P}}$) is solvable in polynomial time, then let $\bar{\sigma}^*$ denote an optimal schedule for this problem. Otherwise, that is, if ($\bar{\text{P}}$) is $\mathcal{NP}$-hard but has a PTAS, let $\bar{\sigma}^\varepsilon$ denote a feasible schedule for ($\bar{\text{P}}$) with solution value at most $(1+\epsilon)\bar{F}^*$, where $\bar{F}^*$ is the optimal solution value for Problem ($\bar{\text{P}}$). Of course, we have that $\bar{F}^* \le F^*$.

Let σ be the schedule obtained from either $\bar{\sigma}^*$ or $\bar{\sigma}^\varepsilon$, whichever is appropriate, by inserting as little idle time as possible between the different operations to ensure that σ is a feasible schedule for Problem (P). Then we need to insert no more than $l_{\max}$ idle time before each operation on each machine $M_2, \dots, M_m$ to transform either $\bar{\sigma}^*$ or $\bar{\sigma}^\varepsilon$ into a feasible solution σ for Problem (P). Accordingly, the completion time of each job J_j in σ is at most $(m-1)l_{\max}$ time later than the completion time of J_j in $\bar{\sigma}^*$ or $\bar{\sigma}^\varepsilon$. Hence, using (6), we have

$$F(\sigma) \le F(\bar{\sigma}^*) + n(m-1)l_{\max} \le (1+\varepsilon)F^*,$$

if σ has been obtained from $\bar{\sigma}^*$,

and

$$F(\sigma) \le F(\bar{\sigma}^\varepsilon) + n(m-1)l_{\max} \le (1+2\varepsilon)F^*,$$

if σ has been obtained from $\bar{\sigma}^\varepsilon$.

So, if $\sum_{i=1}^m |\mathcal{S}_i| > 0$, then Problem (P) has a PTAS.

Now consider the case that $\mathcal{S}_i = \emptyset$ for some i ($i = 1, \dots, m$). We have then the following lemma.

Lemma 1 *If $\mathcal{S}_i = \emptyset$, then $|\mathcal{L}_i| \le \frac{2\mu(m-1)}{\varepsilon} - 1$; that is, if there are no small operations on machine M_i, the number of large operations on machine M_i is bounded from above by* $\frac{2\mu(m-1)}{\varepsilon} - 1$.

Proof Suppose k is the smallest index such that $p_{i[k]} > \frac{\varepsilon}{\mu(m-1)} \frac{P_{in}}{n}$, and hence suppose there are $K = n - k + 1$ large operations to be scheduled on M_i. This implies that

$$\frac{(K)(K+1)}{2} p_{i[k]} \le P_{in}.$$

We also have that

$$\frac{K(K+1)}{2}p_{i[k]} > \left(\frac{K(K+1)}{2}\right)\left(\frac{\varepsilon}{\mu(m-1)}\right)\left(\frac{P_{in}}{n}\right).$$

Hence, we must have that

$$\frac{K(K+1)}{2}\frac{\varepsilon}{\mu(m-1)}\frac{P_{in}}{n} < P_{in},$$

which implies that

$$\frac{K(K+1)}{2n} \le \frac{\mu(m-1)}{\varepsilon}.$$

Since $K \le n$, we have $K < \frac{2\mu(m-1)}{\varepsilon} - 1$. □

Lemma 1 implies that $n < (\frac{2\mu(m-1)}{\varepsilon} - 1)m$, since each has job at least contains one large one operation (see Property (xi) of Problems (P) in class $\mathcal{F}$; see Sect. 2). Hence, for fixed m and μ, we can find the optimal schedule in polynomial time by complete enumeration.

Now, we give a summary of our algorithm.

Algorithm II PTAS for any problem (P) in class $\mathcal{F}$

Step 1. If $\sum_{i=1}^{m}|\mathcal{S}_i| > 0$, go to Step 2, otherwise go to Step 3.

Step 2. Find schedule σ, to be obtained either from $\bar{\sigma}^*$, if ($\bar{\text{P}}$) is polynomially solvable, or from $\bar{\sigma}^{\varepsilon}$, if ($\bar{\text{P}}$) has a PTAS. Stop.

Step 3. Enumerate all possible sequences explicitly and find thus a schedule σ^* with minimum total flow time. Stop.

Theorem 2 *Algorithm II is a Polynomial-Time Approximation Scheme for any total flow time minimization scheduling problem with time lags in class $\mathcal{F}$. The running time of the algorithm is polynomial in n but exponential in* $(\frac{2\mu(m-1)m}{\varepsilon} - m)$.

5 Conclusions

In this paper, we have presented the first PTASs for machine scheduling problems with time lags. Specifically, we have defined two classes of scheduling problems with time lags, one for minimizing makespan and one for minimizing total completion time, such that each problem in those classes possesses a PTAS. Our algorithms mark a step forward for time lag problems without earlier known approximability results, such as $F2|r_j, l_j|C_{\max}$ and $Fm|l_{ij}|\sum C_j$, as well as for problems with known approximability results, such as $F2|l_j|C_{\max}$, if the time lags are relatively restricted in size. For example, the best approximation algorithm for the problem $F2|l_j|C_{\max}$ has a worst-case ratio of $\frac{11}{6}$ (Karuno and Nagamochi 2003). For $\mu \le 5$, our algorithm either improves the ratio to $\frac{3}{2}$ in $O(n\log n)$ time, or finds the optimal solution by enumerating at most 10 large jobs. For $\mu \le 10$, our algorithm either improves the ratio to $\frac{5}{3}$ in $O(n\log n)$ time, or finds the optimal solution by enumerating at most 15 large jobs.

Remember that no explicit enumeration is required if there is at least one job with a small operation. This implies that, for any Problem (P) whose counterpart ($\bar{\text{P}}$) is polynomially solvable, we can find an $n_0 > 0$ for any given $\mu > 0$ and $\varepsilon > 0$ such that our algorithm requires no explicit enumeration if $n > n_0$. This n_0 is defined by the maximum number of jobs with large operations; accordingly, for problems in class $\mathcal{M}$, $n_0 = \mu(o-1)m/\varepsilon$ (see Inequality (3)), and for problems in class $\mathcal{F}$, $n_0 = (\frac{2\mu(m-1)}{\varepsilon} - 1)m$ (see Lemma 1). Take, for example, again the problem $F2|l_j|C_{\max}$ or the problem $O2|l_{hij}|C_{\max}$, which has a known worst-case ratio of 2 (Rebaine 2004). For $\mu \le 10$, our algorithm improves the worst-case ratio to $\frac{6}{5}$ in $O(n\log n)$ time for any $n > n_0 = 100$ with no enumeration required. For $\mu \le 25$, our PTAS improves the worst-case ratio to $\frac{3}{2}$ in $O(n\log n)$ time for any $n > n_0 = 100$, also with no enumeration required.

Acknowledgements The authors thank the anonymous reviewers for their valuable comments on earlier versions of this paper. Research of the first author was partially supported by projects from National Natural Science Foundation of China (No. 70432001 and No. 70771028).

References

Achugbue, J. O., & Chin, F. Y. (1982). Scheduling the open shop to minimize mean flow time. *SIAM Journal on Computing*, *11*, 709–720.

Ageev, A. A. (2008). A 3/2-approximation for the proportionate two-machine flow shop scheduling with minimum delays. *Lecture Notes in Computer Science*, *4927*, 55–66.

Baptiste, P., & Timkovsky, V. G. (2004). Shortest path to nonpreemptive schedules of unit-time jobs on two identical parallel machines with minimum total completion time. *Mathematical Methods of Operations Research*, *60*(1), 145–153.

Brucker, P., Knust, S., Cheng, T. C. E., & Shakhlevich, N. V. (2004). Complexity results for flow-shop and open-shop scheduling problems with transportation delays. *Annals of Operations Research*, *129*(1), 81–106.

Bruno, J., Jones III, J. W., & So, K. (1980). Deterministic scheduling with pipelined processors. *IEEE Transactions on Computers*, *29*(4), 308–316.

Dell'Amico, M. (1996). Shop problems with two machines and time lags. *Operations Research*, *44*(5), 777–787.

Dell'Amico, M., & Vaessens, R. J. M. (1995). Flow and open shop scheduling on two machines with transportation times and machine-independent processing times is NP-hard. *Materiali di discussione*, *1*, 141.

Fishkin, A. V., Jansen, K., & Mastrolilli, M. (2002). *A PTAS for shop scheduling with release dates to minimize the average weighted completion time* (Technical Report: IDSIA-19-02).

Garey, M. R., Johnson, D. S., & Sethi, R. (1976). The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research*, *1*(2), 117–129.

Gonzalez, T., & Sahni, S. (1976). Open shop scheduling to minimize finish time. *Journal of the ACM*, *23*(4), 665–679.

Graham, R. L., Lawler, E. L., Lenstra, J. K., & Kan, A. H. G. R. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, *5*(2), 287–326.

Gupta, J. N. D. (1996). Comparative evaluation of heuristic algorithms for the single machine scheduling problem with two operations per job and time-lags. *Journal of Global Optimization*, *9*(3), 239–253.

Hall, L. A. (1994). A polynomial approximation scheme for a constrained flow-shop scheduling problem. *Mathematics of Operations Research*, *19*(1), 68–85.

Jackson, J. R. (1956). An extension of Johnson's results on job lot scheduling. *Naval Research Logistics Quarterly*, *3*(3), 201–203.

Johnson, S. M. (1954). Optimal two-and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, *1*(1), 61–68.

Karuno, Y., & Nagamochi, H. (2003). A better approximation for the two-machine flowshop scheduling problem with time lags. In *Algorithms and computation: 14th international symposium, ISAAC 2003, Kyoto, Japan, December 15–17, 2003: Proceedings*.

Kern, W., & Nawijn, W. N. (1991). Scheduling multi-operation jobs with time lags on a single machine. In *2nd Twente workshop on graphs and combinatorial optimization pp. 81–86 (SEEN 92-24905 15–64)*.

Kovalyov, M. Y., & Werner, F. (1997). A polynomial approximation scheme for problem $F2/r_j/C_{\max}$. *Operations Research Letters*, *20*(2), 75–79.

Mitten, L. G. (1959). Sequencing n jobs on two machines with arbitrary time lags. *Management Science*, *5*(3), 293–298.

Potts, C. N., & Whitehead, J. D. (2007). Heuristics for a coupled-operation scheduling problem. *Journal of the Operational Research Society*, *58*(10), 1375–1388.

Rayward-Smith, V. J., & Rebaine, D. (1992). Open shop scheduling with delays. *Informatique Théorique et Applications (Imprimé)*, *26*(5), 439–448.

Rebaine, D. (2004). Scheduling the two-machine open shop problem with non-symmetric time delays. In *Congres ASAC 2004, Quebec, Canada*.

Rebaine, D., & Strusevich, V. A. (1999). Two-machine open shop scheduling with special transportation times. *The Journal of the Operational Research Society*, *50*(7), 756–764.

Strusevich, V. A. (1999). A heuristic for the two-machine open-shop scheduling problem with transportation times. *Discrete Applied Mathematics*, *93*(2–3), 287–304.

Strusevich, V. A., & Rebaine, D. (1995). Two-machine open shop with transportation times. In *ECCO VIII, May 8–10, Poznan, Poland*.

Timkovsky, V. G. (1997). A polynomial-time algorithm for the two-machine unit-time release-date job-shop schedule-length problem. *Discrete Applied Mathematics*, *77*(2), 185–200.

Yu, W. (1996). *The two-machine flow shop problem with delays and the one-machine total tardiness problem*. Eindhoven University of Technology.

Yu, W., Hoogeveen, H., & Lenstra, J. K. (2004). Minimizing makespan in a two-machine flow Shop with delays and unit-time operations is NP-hard. *Journal of Scheduling*, *7*(5), 333–348.