2016 年度重庆市出版专项基金资助栏目

Operations Research and Cybernetics

DOI:10.11721/cqnuj20170219

Scheduling with a Common Due Date on a Single Batch Processing Machine*

LIU Lili[1], REN Han[2,3], TANG Guochun[4]

(1. Department of Applied Mathematics, Shanghai Second Polytechnic University, Shanghai 201209;
2. Department of Mathematics, East China Normal University, Shanghai 200062;
3. Shanghai Key Laboratory of Pure Mathematics and Mathematical Practice, Shanghai 200062;
4. Institute of Management Engineering, Shanghai Second Polytechnic University, Shanghai 201209, China)

Abstract: [**Purposes**] Scheduling problems on a single batch processing machine have a wide range of practical applications, and most of them are NP-hard. Single machine batch scheduling problems with common due date is a very important research direction. [**Methods**] By using the combinatorial optimization methods, the problems of scheduling jobs with a common due date on a single batch processing machine to minimize the number of tardy jobs and the total tardiness are studied. [**Findings**] A polynomial time algorithm is presented for minimizing the number of tardy jobs and a pseudo-polynomial time dynamic programming algorithm is proposed for minimizing the total tardiness. [**Conclusions**] The research methods can be applied to solve other batch scheduling problems with a common due date.

Keywords: batch processing; common due date; dynamic programming

CLC number: O223　　**Document code**: A　　**Artide number**: 1672-6693(2017)02-0001-05

Batch scheduling is motivated by many industrial manufacturing processes in which the machine can process multiple jobs at the same time. Ikura and Gimple[1] present a polynomial time algorithm to determine whether a feasible schedule exists for a single machine batch scheduling problem with equal processing times, agreeable release times and due dates. Polynomial time algorithms are developed for problems with agreeable processing times and due dates to minimize maximum tardiness and number of tardy jobs in [2]. Li and Lee[3] prove that problem with agreeable release dates and deadlines is strongly NP-hard. They develop polynomial time algorithms for minimizing maximum tardiness and number of tardy jobs with jobs having agreeable release dates, due dates and processing times. Baptiste[4] present polynomial time algorithms for single machine batch scheduling problems of minimizing maximum tardiness, total tardiness and weighted number of tardy jobs when jobs have different release dates and equal processing times. Brucker et al.[5] show that a single machine batch scheduling problem with deadlines is strongly NP-hard even if the jobs are released at the same time.

Given the NP-hardness of the single machine batch scheduling problems with different job due dates, in this paper we study batch scheduling problems with a common due date to minimize the number of tardy jobs and the total tardiness. The problem with a common due date is interesting since in reality the jobs released at

* **Received**: 04-19-2016　**Accepted**: 02-18-2017　**Internet publishing time**: 2017-03-13 11:06

Fundation: National Natural Science Foundation of China (No. 11601316); The Discipline "Applied Mathematics" of Shanghai Second Polytechnic University (No. XXKPY1604)

The first author biography: LIU Lili, female, professor, doctor degree, major in combinatorial optimization, E-mail: 03902707@163.com

收稿日期:2016-04-19　**修回日期**:2017-02-18　**网络出版时间**:2017-03-13 11:06

资助项目:国家自然科学基金(No. 11601316);上海第二工业大学应用数学学科(No. XXKPY1604)

第一作者简介:刘丽丽,女,教授,博士,研究方向为组合最优化,E-mail: 03902707@163.com

网络出版地址:http://kns.cnki.net/kcms/detail/50.1165.N.20170313.1106.010.html

the same time are usually assigned a common due date. In Section 1 we present a description of the model under study, and introduce the notation that will be used in this paper. In Section 2 we provide a polynomial time algorithm for problem of minimizing the number of tardy jobs. In Section 3 we propose a pseudo-polynomial time dynamic programming algorithm for problem of minimizing the total tardiness. Finally, we draw some conclusions and suggest directions for future research.

1 Notation

We have n jobs to be processed on a single batch processing machine, all of which are available at time zero. The processing time of job J_j is denoted by p_j and the common due date by d. We assume that all the jobs have been ordered in increasing order of their processing times such that $p_1 \leqslant p_2 \leqslant \cdots \leqslant p_n$.

A batch processing machine can process at most B jobs simultaneously. We call a batch is full if it comprise exactly B jobs; otherwise, we call it a partial batch. Let $p(B_i)$ denote the processing time of batch B_i, which is equal to the longest processing time of the jobs in this batch. Let $|B_i|$ denote the number of jobs in batch B_i.

Given a schedule, for job J_j, we use C_j to denote its completion time, its tardiness is defined as $T_j = \max\{0, C_j - d_j\}$, and its unit penalty is defined as $U_j = 1$ if $C_j > d_j$ and zero otherwise. In this paper we study two scheduling objectives: $\sum U_j$ and $\sum T_j$.

We denote the scheduling problems by the three-field notation. For example, $1|B, d_j = d|\sum T_j$ denotes the problem of minimizing the total tardiness on a single batch processing machine with jobs having a common due date.

2 Minimizing the number of tardy jobs

For single machine batch scheduling problem with a common due date, minimizing the maximum tardiness is equivalent to minimizing the makespan, the completion time of the last batch, which can be solved by FB-LPT (Full batch largest processing time) proposed by Bartholdi[6] in $O(n\log n)$ time. The FB-LPT algorithm first indexes the jobs in decreasing order of their processing times, then from the beginning assigns adjacent B jobs into a batch until all the jobs have been assigned, and finally arranges the batches in any arbitrary order.

Problem $1 \mid B \mid \sum U_j$ is strongly NP-hard for given B[5]. If jobs have a common due date, we derive a polynomial time algorithm for this problem. A schedule is called in the batch-SPT order[5] if in this schedule for any two batches B_1 and B_2, batch B_1 is assigned before batch B_2 and there does not exist two jobs J_k, J_l satisfy that $J_k \in B_1, J_l \in B_2$ and $p_k > p_l$. The following lemma establishes a very useful property of an optimal schedule.

Lemma 1 For problem $1 \mid B, d_j = d \mid \sum U_j$, there exists an optimal schedule in which all the jobs are in batch-SPT order and all the on time batch(es) are full except possibly the first one.

Proof Suppose that there exists an optimal schedule such that there are two batches P and Q, where P is finished before d and Q is finished after d and there are two jobs J_i and J_j such that $J_j \in P, J_i \in Q$ and $p_i < p_j$. We exchange jobs J_i and J_j by moving J_i to batch P and J_j to batch Q. Since $p_i < p_j$, the completion times of the on time batches will not increase after the exchange. Hence, the number of the tardy jobs will not change. Repeating applying this argument, we construct an optimal schedule in which the on time jobs have smaller processing times than the tardy jobs.

We know that FB-LPT algorithm gives an optimal schedule for problem $1|B|C_{\max}$. Apply FB-LPT algorithm to the on time jobs and the tardy jobs, respectively, then arrange the batches in increasing order of their processing times. The completion time of the last on time batch will not increase, as well as the number of the tardy jobs. Hence, we can obtain an optimal schedule with the desired property.

Based on Lemma 1, we can now present the following algorithm for problem $1 \mid B, d_j = d \mid \sum U_j$. Let l denote the number of jobs in the first on time batch.

Step 1, For each $l=1,2,\cdots,B$, assign l jobs with smallest processing times to batch B_1.

Step 2, Starting from job J_{l+1}, group successive B jobs into full batches until the completion time of the batch B_k is larger than d. Arrange the remaining jobs in any arbitrary order.

Step 3, Among the B feasible schedules obtained from Step 1 and Step 2, select the one with the smallest number of tardy jobs.

This algorithm can provide an optimal schedule for problem $1 \mid B, d_j=d \mid \sum U_j$ according to Lemma 1. The time complexity of this algorithm is $O(n\log n)$, which is polynomial.

3 Minimizing the total tardiness

In this section, we examine the problem of minimizing the total tardiness with a common due date. Given an instance, apply algorithm FB-LPT to obtain the minimum makesapn, denote this makespan as C. We only need to investigate the case of $C>d$.

In order to describe a feasible schedule for problem $1 \mid B, d_j=d \mid \sum T_j$, we partition all the jobs in this schedule into two subsets.

The first subset contains the on time jobs (completed before or at time d). The second one contains the tardy jobs (completed after time d). Hence, the total tardiness of the jobs in the first subset is zero and these jobs are finished on time. In order to minimize the total tardiness, we only need to minimize the total tardiness of the jobs in the second subset. Since all the jobs have a common due date, minimizing the total tardiness is equivalent to minimizing the total completion time of the tardy jobs. We now present several properties of an optimal schedule for problem $1 \mid B, d_j=d \mid \sum T_j$ based on those of problem $1 \mid B \mid \sum C_j$ [5,7-8].

Lemma 2 For problem $1 \mid B, d_j=d \mid \sum T_j$ there exists an optimal schedule such that all the batches contain consecutively indexed jobs.

Proof Consider an optimal schedule with jobs J_i and J_k being assigned in the same batch P and job J_j ($i<j<k$) being assigned in another batch Q. Exchange jobs J_i and J_j by moving J_i to Q and J_j to P. Because $p_i \leqslant p_j \leqslant p_k$, exchanging jobs does not increase the completion times of all the batches. Since the jobs have a common due date, the total tardiness does not increase either after the job interchange. Repeating the above procedures, we can obtain an optimal schedule such that all the batches contain consecutively indexed jobs.

Lemma 3 For problem $1 \mid B, d_j=d \mid \sum T_j$, suppose there are l tardy batches $B_1, B_2, \cdots, B_l$, then it is optimal to order the batches in $p(B_1)/|B_1|, p(B_2)/|B_2|, \cdots, p(B_l)/|B_l|$ order.

Lemma 4 For problem $1 \mid B, d_j=d \mid \sum T_j$, there exists an optimal schedule such that all the on time batches are in batch-SPT order and the on time batch(es) are full except possibly the first one.

Proof Similar to the proof of Lemma 1.

Batch B_l is called a deferred batch of B_k (or batch B_l is deferred by batch B_k) if batch B_l is assigned after batch B_k and $p(B_k)>p(B_l)$, and batch B_k is called a deferring batch of B_l.

Lemma 5 For problem $1 \mid B, d_j=d \mid \sum T_j$, there exists an optimal schedule with all the deferring batches being full, all the deferred batches being partial, all the partial batches and full batches being scheduled in increasing order of their processing times, respectively.

Lemma 6 In any optimal schedule, the number of deferred batches by any full batch is no larger than B^2-B-1.

We now provide a backward dynamic programming algorithm by using state variables to define the deferred tardy batches by a full deferring batch, which is based on the dynamic programming algorithm for problem $1\mid B\mid \sum C_j$ [5]. By enumeration, we construct the first possibly partial batch B_1 that is finished on time in $O(B)$ time, and batch B_s that is started before time d and finished at or after time d in $O(nB)$ time, respectively. Denote the number of the remaining jobs as m, order the remaining jobs in increasing order of their processing times. Let σ be a partial schedule that contains jobs $J_j,\cdots,J_m$, but does not contain job J_{j-1}, and also contains partial tardy batches $B_1,B_2,\cdots,B_r$, in which $p(B_1)\leqslant p(B_2)\leqslant\cdots\leqslant p(B_r)$ from Lemma 5, and batches $B_1,B_2,\cdots,B_r$ are deferred by the batch containing job J_{j-1} that remains to be scheduled. According to Lemma 2, we assume that $B_l=\{J_{j_l},J_{j_l+1},\cdots,J_{j'_l}\}$, where $j'_l<j_{l+1}$ for $l=1,\cdots,r$ and $j_{r+1}=j$. We also assume that $r\leqslant B^2-B-1$ according to Lemma 6. Denote the total processing time of the on time batches in partial schedule σ as t. Define σ to be in state $(t,j,j_1,\cdots,j_r)$ and (t,j,o) for $r>0$ and $r=0$, respectively. Let $F_j(t,j_1,\cdots,j_r)$ and $F_j(t,o)$ denote the minimum total tardiness of partial schedule σ in state $(t,j,j_1,\cdots,j_r)$ and (t,j,o), respectively. The initialization of the dynamic programming algorithm is as follows:

$$F_{m+1}(t,o)=\begin{cases}0,\text{if } t=0,\cdots,d-1,\\ \infty,\text{otherwise.}\end{cases}$$

for $j=m,\cdots,1$, and $t=0,\cdots,d-1$, the recursion equations are

$$F_j(t,o)=\min\begin{cases}\min\limits_{j+1\leqslant k\leqslant j+B}\{F_k(t,o)+m'p_{k-1}\},\\ \min\limits_{j+1\leqslant k\leqslant m-B+1,j'_1,\cdots,j'_{r'}}\{F_{k+B}(t,j'_1,\cdots,j'_{r'})+m'p_{k+B-1}\},\\ \min\limits_{j+1\leqslant k\leqslant m-B+1,j'_1,\cdots,j'_{r'}}\{F_{k+B}(t-p_{k+B-1},j'_1,\cdots,j'_{r'})\},\\ F_{j+B}(t-p_{j+B-1},o).\end{cases}$$

Where m' is the number of tardy jobs in partial schedule σ, which can be computed in every state. The first term in the minimization corresponds to the case where a full or partial batch $\{J_j,\cdots,J_{k-1}\}$ is inserted at the beginning of the tardy batches in state (t,k,o). The second term corresponds to the case where a full tardy batch $\{J_k,\cdots,J_{k+B-1}\}$ that does not contain job J_j is inserted at the beginning of the tardy batches in state $(t,k+B,j'_1,\cdots,j'_r)$ and batches $B'_1,B'_2,\cdots,B'_{r'}$ satisfy $B'_1\cup B'_2\cup\cdots\cup B'_r=\{J_j,\cdots,J_{k-1}\}$ and are deferred by the inserted full batch. The third term corresponds to the case where a full batch $\{J_k,\cdots,J_{k+B-1}\}$ that does not contain job J_j is assigned to the end of the on time batches in state $(t-p_{k+B-1},k+B,j'_1,\cdots,j'_r)$ and batches $B'_1,B'_2,\cdots,B'_r$ satisfy $B'_1\cup B'_2\cup\cdots\cup B'_r=\{J_j,\cdots,J_{k-1}\}$ and are deferred by the inserted full batch. The forth term corresponds to the case where a full batch $\{J_j,\cdots,J_{j+B-1}\}$ is assigned to the end of the on time batches in state $(t-p_{k+B-1},j+B,o)$.

Also, for $j=m+1,m,\cdots,1,r=1,\cdots,B^2-B-1$ and $j_1<j_2<\cdots<j_r<j-B$ and $(j_{l+1}-j_l)\bmod B\neq 0$ for $l=1,\cdots,r$, the recursion equations are

$$F_j(t,j_1,\cdots,j_r)=\min\begin{cases}F_j(t,j_2,\cdots,j_r)+m'p_{j'_1},\\ F_{j+B}(t-p_{j+B-1},j_1,\cdots,j_r),\\ F_{j+B}(t,j_1,\cdots,j_r)+m'p_{j+B-1},\\ \min\limits_{j+1\leqslant k\leqslant m-B+1,j'_1,\cdots,j'_{r'}}\{F_{k+B}(t-p_{k+B-1},j_1,\cdots,j_r,j'_1,\cdots,j'_{r'})\},\\ \min\limits_{j+1\leqslant k\leqslant m-B+1,j'_1,\cdots,j'_{r'}}\{F_{k+B}(t,j_1,\cdots,j_r,j'_1,\cdots,j'_{r'})+m'p_{j+B-1}\}.\end{cases}$$

The first term in the minimization corresponds to the case where a partial tardy batch is inserted at the beginning of the tardy batches in state $(t,j,j_2,\cdots,j_r)$. The second term corresponds to the case where a full batch $\{J_j,\cdots,J_{j+B-1}\}$ is assigned to the end of the on time batches in state $(t-p_{j+B-1},j+B,j_1,\cdots,j_r)$. The third term corresponds to the case where a full batch $\{J_j,\cdots,J_{j+B-1}\}$ is inserted at the beginning of the tardy batches in state $(t,j+B,j_1,\cdots,j_r)$. The forth term corresponds to the case where a full batch $\{J_k,\cdots,J_{k+B-1}\}$ that does not contain job J_j is assigned to the end of the on time batches in state $(t-p_{k+B-1},k+B,j_1,\cdots,j_r,j'_1,$

$\cdots,j_r$) and batches $B_1',B_2',\cdots,B_{r'}'$ satisfy $B_1'\cup B_2'\cup\cdots\cup B_r'=\{J_j,\cdots,J_{k-1}\}$ and are deferred by the inserted full batch. The fifth term corresponds to the case where a full batch $\{J_k,\cdots,J_{k+B-1}\}$ that does not contain job J_j is inserted at the beginning of the tardy batches in state $(t,k+B,j_1,\cdots,j_r,j_1',\cdots,j_r)$ and batches $B_1',B_2',\cdots,B_{r'}'$ satisfy $B_1'\cup B_2'\cup\cdots\cup B_r'=\{J_j,\cdots,J_{k-1}\}$ and are deferred by the inserted full batch.

The optimal value is equal to $\min_{B_1,B_s,d-p(B_s)\leqslant t<d}\{F_1(t,o)+(t+p(B_s)-d)m'\}$, where m' is the number of tardy jobs in state $(t,1,o)$. We can obtain the optimal schedule by backtracking. The time complexity of this algorithm is $O(B^2dn^{B^2-B+2})$, which is pseudo-polynomial time.

4 Conclusion

In this paper we investigate scheduling problems of minimizing the number of tardy jobs and the total tardiness on a single batch processing machine with jobs having a common due date, polynomial time algorithm and pseudo-polynomial time dynamic programming algorithm are developed, respectively.

Batch scheduling with a common due date is a very challenging topic for future research, it will be very interesting to explore the computational complexity of minimizing the total tardiness and the other batch scheduling problems with a common due date.

References:

[1] IKURA Y, GIMPLE M. Efficient scheduling algorithms for a single batch processing machine[J]. Operations Research Letters, 1986, 5: 61-65.

[2] LEE C Y, UZSOY R, MARTIN-VEGA L A. Efficient algorithms for scheduling semiconductor burn-in operations [J]. Operation Research, 1992, 40: 764-775.

[3] LI C L, LEE C Y. Scheduling with agreeable release times and due dates on a batch processing machine[J]. European Journal of Operational Research, 1997, 96: 564-569.

[4] BAPTISTE P. Batching identical jobs[J]. Mathematical Methods of Operations Research, 2000, 52: 355-367.

[5] BRUCKER P, GLADKY A, HOOGEVREEN H, et al. Scheduling a batching machine[J]. Journal of Scheduling, 1998, 1: 31-54.

[6] LEE C Y, UZSOY R. Minimizing makespan on a single batch processing machine with dynamic job arrivals[J]. International Journal of Production Research, 1999, 37: 219-236.

[7] CHANDRU V, LEE C Y, UZSOY R. Minimizing total completion time on a batch processing machine with job families[J]. Operations Research Letters, 1993, 13: 61-65.

[8] HOCHBAUM D S, LANDY D. Scheduling semiconductor burn-in operations to minimize total flow time[J]. Operations Research, 1997, 45: 874-885.

运筹学与控制论

有公共交货期的单机分批排序问题

刘丽丽[1]，任　韩[2,3]，唐国春[4]

（1. 上海第二工业大学 应用数学系，上海 201209；2. 华东师范大学 数学系，上海 200062；

3. 上海市核心数学与实践重点实验室，上海 200062；4. 上海第二工业大学 经济管理学院，上海 201209）

摘要：【目的】单机分批排序问题有着广泛的应用背景，很多问题是NP-困难的。有公共交货期的单机分批排序问题是一个非常重要的研究方向。【方法】利用组合最优化的方法，研究工件有公共交货期的最小化误工工件个数和总延误的单机分批排序问题。【结果】对于最小化误工工件个数问题提出了一个多项式时间算法，对于最小化总延误提出了一个伪多项式时间的动态规划算法。【结论】对于其他有公共交货期的分批排序问题的研究提供了重要的研究方法。

关键词：批处理；公共交货期；动态规划

（责任编辑　黄　颖）