European Journal of Operational Research 233 (2014) 64–74

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

The single machine serial batch scheduling problem with rejection to minimize total completion time and total rejection cost

Dvir Shabtay *

Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beer-Sheva, Israel

ARTICLE INFO

Article history:
Received 30 August 2012
Accepted 8 August 2013
Available online 17 August 2013

Keywords:
Batch scheduling
Bicriteria scheduling
Rejection
Total completion time

ABSTRACT

We study a scheduling problem with rejection on a single serial batching machine, where the objectives are to minimize the total completion time and the total rejection cost. We consider four different problem variations. The first is to minimize the sum of the two objectives. The second and the third are to minimize one objective, given an upper bound on the value of the other objective and the last is to find a Pareto-optimal solution for each Pareto-optimal point. We provide a polynomial time procedure to solve the first variation and show that the three other variations are $\mathcal{NP}$-hard. For solving the three $\mathcal{NP}$-hard problems, we construct a pseudo-polynomial time algorithm. Finally, for one of the $\mathcal{NP}$-hard variants of the problem we propose an FPTAS, provided some conditions hold.

1. Introduction and problem definition

Batch scheduling problems have been traditionally studied under the assumption that all jobs have to be processed within the shop (see Potts and Van Wassenhove, 1991; Webster and Baker, 1995; Potts and Kovalyov, 2000 for survey papers on this field). However, in many cases, especially in highly loaded manufacturing systems, the scheduler may not be able to process all jobs in the shop while maintaining acceptable quality of service or reasonable (economic) inventory level. In such cases, the scheduler may have to make a higher level decision as to which job to process in the shop and which to reject. Each rejected job may either be outsourced or not produced at all which will result in a rejection cost. In order to arrive at the best decision, the scheduler has to carefully coordinate the higher level decisions about the sourcing strategy for each job with the lower level decisions of how to schedule the set of accepted jobs. In this paper we study the problem of coordinating these two critical decisions for the case when the scheduling is done on a single serial batching machine and the scheduling criterion is the total completion time. The importance of the total completion time criterion is well recognized in the literature. According to Pinedo (1995), this criterion is usually used as a surrogate criterion for minimizing the Work-In-Process (WIP) inventory. WIP ties up capital and a large amount of it can clog up operation and increases handling cost.

Our batch scheduling problem with rejection can be formally stated as follows. We are given a set of n jobs, $J = \{J_1, J_2, \ldots, J_n\}$, that is available for processing at time zero. For $j = 1, \ldots, n$, the scheduler can either reject to process job J_j in the shop, at a cost of e_j, or can accept and process it non-preemptively during p_j units of time on a single batching machine. The accepted jobs are to be processed in batches and a setup time s is required before the production of each batch. We use the *batch availability* assumption according to which all jobs in a batch are considered to have been completed together at the completion time of the last job in their batch, i.e., a batch of jobs is removed from the system at this common completion time. We assume that the number of jobs to be included in each batch is not restricted. However, our solution method can be easily modified to deal also with the restricted version of the problem. We consider the case of a *serial batching machine* for which the batch processing time equals the total processing time of the jobs assigned to the batch. A solution S to our problem is defined by

1. Partitioning set J into two subsets, A and $\bar{A}$, each of which correspond to the set of accepted and rejected jobs;
2. Determining the sequence $\pi(A) = \{J_{[1]}, J_{[2]}, \ldots, J_{[|A|]}\}$ in which the jobs in set A are to be processed on a single batching machine, where $[j]$ represents the index of the jth job to be processed on the batching machine ($J_{[j]} \in A$ for $j = 1, \ldots, |A|$); and
3. Partitioning $\pi(A)$ into m batches (where m is a decision variable), $\mathbf{B} = (B_1, B_2, \ldots, B_m)$ where $B_1 = \{J_{[l_0+1]}, \ldots, J_{[l_1]}\}$, $B_2 = \{J_{[l_1+1]}, \ldots, J_{[l_2]}\}$, $\ldots$, $B_m = \{J_{[l_{m-1}+1]}, \ldots, J_{[l_m]}\}$, and l_i counts the number of jobs in the first i batches for $i = 1, 2, \ldots, m$, with $l_0 = 0$ and $l_m = |A|$.

* Tel.: +972 8 6461389; fax: +972 8 6472958.
E-mail address: dvirs@bgu.ac.il

0377-2217/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2013.08.013

For a given solution S, let C_j be the completion time of job $J_j \in A$. We evaluate the quality of a solution by two different criteria. The first, $F_1(S) = \sum_{J_j \in A} C_j$, is the *total completion time* (scheduling) criterion, and the second, $F_2(S) = \sum_{J_j \in \bar{A}} e_j$, is the *total rejection cost*. Due to the batch availability assumption the total completion time can be given by

$$F_1(S) = \sum_{i=1}^{m} (l_m - l_{i-1}) \times \left[s + \sum_{j=l_{i-1}+1}^{l_i} p_{[j]} \right]. \quad (1)$$

Since we are dealing with a bicriteria problem, several different problem variations can be defined (see, e.g., T'kindt and Billaut, 2006, pp. 121–122). We focus on the following four different variations of the problem (see Shabtay et al., 2013 for a more detailed description of these variations).

- The first, P1, is to minimize the total integrated cost, i.e., to find a solution S which minimizes $F_1(S) + F_2(S)$.
- The second, P2, is to find a solution S that minimizes $F_1(S)$ subject to $F_2(S) \leqslant \bar{E}$, where $\bar{E}$ is a given upper bound on the total rejection cost.
- The third, P3, is to find a solution S that minimizes $F_2(S)$ subject to $F_1(S) \leqslant \bar{K}$, where $\bar{K}$ is a given upper bound on the value of the scheduling criterion.
- The fourth, P4, is to identify a *Pareto-optimal* solution for each Pareto-optimal point, where a solution S is called Pareto-optimal (or efficient) if there does not exist another solution S' with $F_1(S') \leqslant F_1(S)$ and $F_2(S') \leqslant F_2(S)$, with at least one of these inequalities being strict. The point (K, E), where $K = F_1(S)$ and $E = F_2(S)$, is Pareto-optimal point corresponding to this Pareto-optimal solution.

Following the classification of bicriteria problems introduced in T'kindt and Billaut (2006), our problem can be referred to by $1|rej, s-batch|X$, where rej implies that rejection is allowed and $s-batch$ implies that the set of accepted jobs are to be processed on a serial batching machine. Moreover, X is replaced by (F_1, F_2) for the general problem and with $F_1 + F_2$, $\epsilon(F_1/F_2)$, $\epsilon(F_2/F_1)$ and $\#(F_1, F_2)$ for variants P1, P2, P3 and P4, respectively. We note that problems P2 and P3 are also known as the ϵ-constraint problems with respect to the total rejection cost (F_2) and the total completion time (F_1), respectively. We also remark here that solving problem P4, also solves problems P1–P3 as a by-product, and that the decision versions of P2 and P3 are identical as they both ask if, given an instance of the $1|rej, s-batch|(F_1, F_2)$ problem and parameters K and E, is there a solution S with $F_1(S) \leqslant K$ and $F_2(S) \leqslant E$? The fact that both P2 and P3 share the same decision version implies that either both or none of them is $\mathcal{NP}$-hard.

It is easy to observe that when $s = 0$ each Pareto-optimal solution is such that each accepted job is included in a different batch (there is no motivation to include several jobs in a batch as the total completion time will obviously increase). Thus, when $s = 0$ the number of batches equals to the number of accepted jobs ($m = |E|$) and we have that $l_i = i$ for $i = 1, \ldots, |E|$. The total completion time in (1) is thus simply reduces to $F_1(S) = \sum_{i=1}^{|E|} (|E| - (i-1)) p_{[i]} = \sum_{i=1}^{|E|} (|E| - i + 1) p_{[i]}$. This last term simply identical to the total completion time of any set E in a $1|rej|(F_1, F_2)$ problem with $F_1 = \sum_{J_j \in A} C_j$. This implies that the $1|rej|(F_1, F_2)$ problem with $F_1 = \sum_{J_j \in A} C_j$ reduces to the $1|rej, s-batch|(F_1, F_2)$ problem. The fact that the decision versions of P2 and P3 are both $\mathcal{NP}$-complete is now directly follows from the fact that Shabtay et al. (2012) proved that the decision version of problems $1|rej|\ \epsilon(F_1/F_2)$ and $1|rej|\ \epsilon(F_2/F_1)$ with $F_1 = \sum_{J_j \in A} C_j$ is $\mathcal{NP}$-complete. Accordingly, our main objectives in the paper is to find out whether the P1 problem is solvable in polynomial time, and whether problems P2–P4 are ordinary or strongly $\mathcal{NP}$-hard.

There are dozens of papers on scheduling with rejection (see a recent survey by Shabtay et al., 2013) with some of them focusing on batch scheduling on parallel batch machines (see, e.g., Lu et al., 2008; Cao and Yang, 2009; Miao et al., 2010; Li and Feng, 2010). However, although scheduling problems on serial batching machines have been extensively studied in the literature (see, e.g., Coffman et al., 1990; Albers and Brucker, 1993; Cheng et al., 2001; Ng et al., 2003; Yuan et al., 2004; Mosheiov et al., 2005; Shabtay and Steiner, 2007; Mosheiov and Oron, 2008), to the best of our knowledge this subject has not been approached in the context where rejection is allowed. Thus, our main aim is to shift the focus of the research area to the case where the set of accepted jobs are scheduled on a serial (rather than parallel) batching machine.

Our paper proceeds as follows. In Section 2 we develop a polynomial time procedure for solving the P1 problem variation in $O(n^5)$ time. In Section 3, we provide a pseudo-polynomial time algorithm to solve the P2 problem, which can be modified to solve problems P3–P4 in pseudo-polynomial time as well. Moreover, we show how we can construct an ε-approximation algorithm that, given the existence of a Pareto-optimal solution with a total rejection cost of at most R and a total completion time of at most K, finds in $O(n^6/\varepsilon^2)$ time a solution with a total rejection cost of at most $(1 + \varepsilon)R$ and a total completion time value of at most $(1 + \varepsilon)K$. A summary section concludes our paper.

2. Polynomial algorithm for problem P1

Coffman et al. (1990) study the $1|s-batch|\sum_{J_j \in J} C_j$ problem and prove the following lemma.

Lemma 1. *There exists an optimal schedule for which the jobs are scheduled in a non-decreasing order of processing times, i.e., according to the SPT rule.*

It is easy to observe that Lemma 1 holds for the set of accepted jobs (set A) in our $1|rej, s-batch|\sum_{J_j \in A} C_j + \sum_{J_j \in \bar{A}} e_j$ problem, i.e., there exists an optimal solution where $\pi(A)$ follows the SPT rule. Thus, for ease of presentation, hereafter we assume that the jobs are indexed according to the SPT rule such that $p_1 \leqslant p_2 \leqslant \cdots \leqslant p_n$.

Next, we show how we can solve the $1|rej, s-batch|\sum_{J_j \in A} C_j + \sum_{J_j \in \bar{A}} e_j$ problem by using a dynamic programming-based optimization procedure. To do this, we let $F_j(l)$ be the minimal total cost for a partial schedule that includes jobs $J_j, \ldots, J_n$, where the number of accepted jobs is l (thus there are $n - j - l + 1$ rejected jobs among jobs $J_j, \ldots, J_n$), and subject to the conditions that job J_j is accepted and that it starts the first batch in the partial schedule. Then, if the next batch starts with job J_k ($j < k \leqslant n + 1$), the minimal cost for jobs $J_j, \ldots, J_n$ can be given by

$$\min_{r_{\min}(j,k,l) \leqslant r \leqslant r_{\max}(j,k,l)} \{ f(j,k,l,r) + F_k(l-r) \},$$

where r represents the number of accepted jobs among jobs $J_j, \ldots, J_{k-1}$ and $f(j, k, l, r)$ is the minimum additional cost that results from selecting a set $A(j, k, l, r)$ of r jobs out of jobs $J_j, \ldots, J_{k-1}$ to be included in set A and to be processed in the first batch. Note that if $k = n + 1$, all accepted jobs among jobs $J_j, \ldots, J_n$ are scheduled in a single batch that starts with job J_j.

Since $J_j \in A(j, k, l, r)$, we have that $r \geqslant 1$. In addition, due to the fact that there are at most $n - k + 1$ accepted jobs among jobs $J_k, \ldots, J_n$, in order to have l accepted jobs among jobs $J_j, \ldots, J_n$, there has to be at least $l - (n - k + 1)$ accepted jobs among jobs $J_j, \ldots, J_{k-1}$. Thus, we have that