

A Batching Machine Model for Lot Scheduling on a Single Machine

Mikhail Y. Kovalyov *

Abstract. A recently introduced lot scheduling problem is considered. It is to find a partition of jobs of n orders into lots and to sequence these lots on a single machine so that the total average completion time of the orders is minimized. A simple $O(n \log n)$ time algorithm is presented for this problem in the literature, with a relatively sophisticated proof of its optimality. We show that modeling this problem as a classic batching machine problem makes its optimal solution obvious.

Keywords: scheduling, lot-sizing, single machine, batching machine, polynomial time algorithm

1. Introduction

Hou et al. [7] have recently introduced the following lot scheduling problem. There are n orders to be scheduled on a single machine. Order i includes σ_i jobs, where σ_i is called *size* of the order i , $i = 1, \dots, n$. The machine can process up to k jobs simultaneously in the same lot. A lot including k jobs is called *full*. It is assumed that $\sigma_i \leq k$, $i = 1, \dots, n$. The processing time of any lot is the same value u . There is no machine idle time between the lots. A schedule is determined by the assignment of jobs into lots and the sequence of lots. Consider a schedule and assume that jobs of order i are assigned to lots of a set S . The average completion time of the jobs of order i is defined as $C_i^{(A)} = (\sum_{L \in S} T_L x_{iL}) / \sigma_i$, where T_L is the completion time of lot L and x_{iL} is the number of jobs of order i in lot L . The objective is to find a schedule such that jobs of the same order are assigned to consecutive lots and the total average order completion time $\sum_{i=1}^n C_i^{(A)}$ is minimized. We denote this problem as $1/Lot, split / \sum C_i^{(A)}$, where the only difference with the notation in [7] is $C_i^{(A)}$ instead

*United Institute of Informatics Problems, National Academy of Sciences of Belarus, Surganova 6, 220012, Minsk, Belarus, e-mail: kovalyov_my@newman.bas-net.by

of C_i , because the latter notation is traditionally used in the scheduling literature to denote a completion time and not an average completion time.

Hou et al. [7] write that motivation for this problem comes from planning operations of simultaneous testing of integrated circuits in a burn-in oven, or planning the production of adhesives and glues, which is to mix raw materials in a heated container for a time period. We can add that the objective of the problem is related to minimizing the average testing or production time of orders received from different customers.

The problem $1/Lot, split / \sum C_i^{(A)}$ falls into the categories of batch scheduling problems and lot-sizing and scheduling problems. Both batching and lot-sizing terminologies can be used to describe the same planning situation, in which jobs of the same order have to be grouped into batches (batching terminology) or each job order has to be partitioned into lots (lot-sizing terminology). Reviews of the results for these problems can be found in Potts and Van Wassenhove [10], Webster and Baker [11], Brucker et al. [4], Potts and Kovalyov [9], Brucker [3] and Allahverdi et al. [1]. Specific batch scheduling problems have been studied by Baptiste [2] and Gerstl and Mosheiov [6]. The paper of Brucker et al. [4] is devoted to scheduling a batching machine. This machine can process several jobs simultaneously in a batch such that they have a common start time, a common completion time and a common batch processing time which is equal to the maximum processing time of its jobs.

The above mentioned batch scheduling problems and lot-sizing and scheduling problems differ from the so-called economic lot-sizing and scheduling problems, in which the orders are replaced by the time dependent product demands, the lot-sizing decisions are associated with unit time production intervals, and the objective function includes product holding and backlogging costs, see Drexel and Kimms [5] and Karimi et al. [8].

In [7], an $O(n \log n)$ time solution algorithm for the problem $1/Lot, split / \sum C_i^{(A)}$ is presented. While the algorithm is simple, its justification is relatively complicated. We show that modeling this problem as a classic batching machine problem makes its optimal solution obvious. Our main result is presented in the next section. The note concludes with a short summary and suggestions for future research.

2. Main result

A batching machine scheduling problem, denoted as $1 \mid p_j = p, p - batch, b < N \mid \sum w_j C_j$, is most relevant for $1/Lot, split / \sum C_i^{(A)}$. In this former problem, N jobs with the same processing times $p_j = p$ and distinct weights w_j are processed on a single machine in batches. Notation $p - batch$ abbreviates parallel (simultaneous) processing of jobs in the same batch. Recall that the processing time of a batch is equal to the maximum processing time of its jobs and that the completion time of a job is equal to the completion time of its batch. The completion time of a batch is equal to the sum of processing times of the batches preceding and including this batch. The batch sizes are upper bounded by b , $b < N$. The objective is to partition the jobs

into batches and to sequence the batches so that the total weighted job completion time, $\sum_{j=1}^N w_j C_j$, is minimized.

Observe that an instance of the problem $1/Lot, split / \sum C_i^{(A)}$ can be viewed as an equivalent instance of the problem $1 \mid p_j = p, p - batch, b < N \mid \sum w_j C_j$, in which $N = \sum_{i=1}^n \sigma_i$, $b = k$ and $p = u$. The weights w_j are defined such that if job j belongs to order i , then $w_j = 1/\sigma_i$, $j = 1, \dots, N$, $i = 1, \dots, n$.

Consider the problem $1 \mid p_j = p, p - batch, b < N \mid \sum w_j C_j$. To the best of our knowledge, this problem has not been studied in the literature. Its generalization to the case of non-zero job release dates has been studied by Baptiste [2], where a high power polynomial time algorithm was developed. We will show that the problem $1 \mid p_j = p, p - batch, b < N \mid \sum w_j C_j$ admits a simple solution.

A batch is called *full* if it includes b jobs. We call *Largest Weight Full Batch (LWFB) sequence* a sequence of batches such that all batches are full if N is a multiple of b , all batches but the last batch are full if N is not a multiple of b , and the jobs are processed in the largest weight first sequence. The following statement is a basis for solving the problem $1 \mid p_j = p, p - batch, b < N \mid \sum w_j C_j$.

Statement 1 *Any LWFB sequence is optimal for an instance of the problem $1 \mid p_j = p, p - batch, b < N \mid \sum w_j C_j$, and hence, for the corresponding instance of the problem $1/Lot, split / \sum C_i^{(A)}$.*

Proof. Consider an optimal batch sequence for an instance of the problem $1 \mid p_j = p, p - batch, b < N \mid \sum w_j C_j$ and assume that it is not an LWFB sequence. If a non-full batch is followed by a full batch, then moving a job from the later batch to the earlier batch decreases the value of $\sum w_j C_j$. Hence, all the batches, with a possible exception for the last batch, are full. If any batches A and B are sequenced in this order, $i \in A$, $j \in B$ and $w_i < w_j$, then interchanging i and j decreases the value of $\sum w_j C_j$. Hence, the optimal batch sequence must be an LWFB sequence. ■

Observe that the chain of relations $w_1 \geq \dots \geq w_N$ for the jobs in an instance of the problem $1 \mid p_j = p, p - batch, b < N \mid \sum w_j C_j$ with weights $w_j = 1/\sigma_i$ is fully determined by the chain of relations $\sigma_1 \leq \dots \leq \sigma_n$ for the orders in the corresponding instance of the problem $1/Lot, split / \sum C_i^{(A)}$. With respect to the latter problem, this observation and the fact that any LWFB sequence includes $\lceil N/b \rceil \leq n + 1$ batches due to the assumption $\sigma_i \leq k$, $i = 1, \dots, n$, implies that an LWFB sequence for the problem $1/Lot, split / \sum C_i^{(A)}$ can be constructed in $O(n \log n)$ time. In this sequence, each order will belong to at most two consecutive batches (lots). The corresponding algorithm is the algorithm given in [7].

The difference between the two algorithms is in the used terminology: our algorithm realizes an LWFB sequence and the algorithm in [7] realizes a sequence that, in the lot scheduling terminology, can be called *Least Order Size Full Lot sequence*, in which all lots are full if $\sum_{i=1}^n \sigma_i$ is a multiple of k , all lots but the last lot are full if $\sum_{i=1}^n \sigma_i$ is not a multiple of k , and the orders are assigned to the lots in the least order size first sequence.

3. Conclusions

We have studied a lot scheduling problem introduced by Hou et al. [7]. The objective function of this problem includes average completion times of the product orders. Functions of this type can be adequate for certain practical planning applications. We have shown that the problem in [7] can be modeled as a classic batching machine problem, from where a simple sorting algorithm for its optimal solution follows. Further research can be undertaken to extend the studied model and the solution approach for other planning environments, such as multiple parallel or sequential machines.

References

- [1] Allahverdi A., Ng C.T., Cheng T.C.E., Kovalyov M.Y. A survey of scheduling problems with setup times or costs, *European Journal of Operational Research*, **187**, 2008, 985–1032.
- [2] Baptiste P., Batching identical jobs, *Mathematical Methods of Operations Research*, **53**, 2000, 355–367.
- [3] Brucker P., *Scheduling algorithms*, Fifth edition, Springer, 2007.
- [4] Brucker P., Gladky A., Hoogeveen H., Kovalyov M.Y., Potts C.N., Tautenhahn T., van de Velde S., Scheduling a batching machine, *Journal of Scheduling*, **1**, 1998, 31–54.
- [5] Drexel A., Kimms A. Lot sizing and scheduling - survey and extensions, *European Journal of Operational Research*, **99**, 1997, 221–235.
- [6] Gerstl E., Mosheiov G., A two-stage flow shop scheduling with a critical machine and batch availability, *Foundations of Computing And Decision Sciences*, **37**, 1, 2012, 39–56.
- [7] Hou Y.-T., Yang D.-L., Kuo W.-H., Lot scheduling on a single machine, *Information Processing Letters*, **114**, 2014, 718–722.
- [8] Karimi B., Fatemi Ghomi S.M.T., Wilson J.M. The capacitated lot sizing problem: a review of models and algorithms, *Omega*, **31**, 2003, 365–378.
- [9] Potts C.N., Kovalyov M.Y., Scheduling with batching: a review, *European Journal of Operational Research*, **120**, 2000, 228–249.
- [10] Potts C.N., Van Wassenhove L.N., Integrating scheduling with batching and lot-sizing: A review of algorithms and complexity, *Journal of the Operational Research Society*, **43**, 1992, 395–406.
- [11] Webster S.T., Baker K.R., Scheduling groups of jobs on a single machine, *Operations Research*, **43**, 1995, 692–703.

Received 15.09.2015, Accepted 11.12.2017