European Journal of Operational Research 239 (2014) 214–226

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Innovative Applications of O.R.

A branch and bound based heuristic for makespan minimization of washing operations in hospital sterilization services

Onur Ozturk, Mehmet A. Begen *, Gregory S. Zaric

Ivey Business School, Western University, 1255 Western Road, London, Ontario N6G 0N1, Canada

ARTICLE INFO

Article history:
Received 12 June 2013
Accepted 7 May 2014
Available online 24 May 2014

Keywords:
OR in health services
Parallel batch scheduling
Makespan
Branch and bound heuristic

ABSTRACT

In this paper, we address the problem of parallel batching of jobs on identical machines to minimize makespan. The problem is motivated from the washing step of hospital sterilization services where jobs have different sizes, different release dates and equal processing times. Machines can process more than one job at the same time as long as the total size of jobs in a batch does not exceed the machine capacity. We present a branch and bound based heuristic method and compare it to a linear model and two other heuristics from the literature. Computational experiments show that our method can find high quality solutions within short computation time.

1. Introduction

Sterilization services are hospital departments where medical devices (MDs) are sterilized. There are two types of MDs: single use MDs and reusable MDs. Reusable MDs (RMDs) are used in surgeries, sterilized, and then reused in other surgeries. We consider the sterilization process of RMDs in this study.

All RMDs used in a surgery constitute the RMD set of the surgery. After a surgery, all RMDs used are sent to the sterilization service. Due to surgery characteristics and surgeons needs, RMD sets may contain different numbers and types of instruments. Hence, they may have different sizes (or volumes). Moreover, they are sent to the sterilization service at different times within a day since each surgery may have a different starting and ending time.

A typical sterilization service is composed of the following steps (Di Mascolo & Gouin, 2013): pre-disinfection, washing, packing and sterilization. Pre-disinfection is a manual step during which RMDs are submerged in a chemical substance. Then, they are washed in an automatic washer. Afterwards, they are packed and sterilized with steam in autoclaves.

We are interested in the washing step which is a bottleneck for sterilization services. More than one RMD set can be washed in an automatic washer at the same as long as the machine capacity is not exceeded. All RMD sets washed at the same time constitute a single batch. Depending on the organization between operating theatres and the sterilization service, RMD arrivals can be known in advance. For instance, RMD arrivals can be known accurately for operating theatres where ambulatory surgeries take place. Another example is sterilization services that accept RMD arrivals only at specific times within a day. However, although RMD arrival times and sizes are known in advance, the decision of how to load the machines, i.e., how to batch RMD sets and launch washing cycles is not trivial. We model this problem using a parallel batch scheduling approach. Jobs may have different sizes (or volumes), different release dates and equal processing times. All jobs processed at the same time constitute a single batch which is processed on a single machine. The processing time of batches are the same and equal to the processing time of jobs. Hence, our problem becomes a parallel batching problem where RMD sets are treated as jobs having different sizes, different release dates and equal processing times.

The remainder of this paper is organized as follows. In Section 2, we provide a literature review about batch scheduling problems and summarize the contributions of this paper. In Section 3, we give a formal description of our problem. Section 4 is dedicated to the solution methodology. Section 5 presents computational tests. Finally, we conclude the study and propose some further research directions.

2. Literature review

2.1. Batch scheduling

We review only batch scheduling literature regarding jobs with different sizes. For more information about batch scheduling, we

* Corresponding author. Tel.: +1 519 661 4146.
E-mail addresses: oozturk@uwo.ca (O. Ozturk), mbegen@ivey.uwo.ca (M.A. Begen), gzaric@ivey.uwo.ca (G.S. Zaric).

http://dx.doi.org/10.1016/j.ejor.2014.05.014
0377-2217/

refer the reader to Potts and Kovalyov (2000) and Mathirajan and Sivakumar (2006). There are two types of batch scheduling: serial and parallel. In serial batch scheduling, jobs in the same batch are processed sequentially on one or more machines. The processing of a batch is completed when the last job of the batch is processed. A typical example is confection workshops where many types of clothes are sewed. For instance, sewing of t-shirts constitutes a batch while shirts, trousers, etc. may constitute a second batch. In parallel batching however, all jobs are processed simultaneously in the same machine. In this paper, we study a parallel batch scheduling problem.

2.1.1. Exact methods

To the best of our knowledge, exact methods for parallel batching with jobs having different processing times are only applied to the case when all jobs are available at the same time. Uzsoy (1994) proposes a branch and bound algorithm to minimize the sum of job completion times on a single machine in which jobs have different processing times and sizes. For the same problem but with the objective of minimizing makespan, Dupont and Dhaenens-Flipo (2002) develop a branch and bound algorithm. Later on, Parsa, Karimi, and Kashan (2010) propose a branch and price method for the same problem. They report that their method is more efficient in terms of solution time than the one proposed by Dupont and Dhaenens-Flipo (2002). Malapert, Gueret, and Rousseau (2012) study the minimization of maximum lateness on a single machine for which they propose a constraint programming approach. Other than these studies, there are many other studies where the case of unit size jobs is tackled. For instance, Yuan, Liu, Ng, and Cheng (2004) study the case where jobs have unit sizes but different processing times and release dates in the presence of job families. They provide dynamic programming algorithms when the number of jobs, number of job families and number of release dates are bounded. For the general case, they propose a 2-approximation algorithm. Cheng, Yuan, and Yang (2005) propose polynomial time dynamic programming algorithms for a set of regular objective functions when jobs have unit sizes, unit processing times, release dates and precedence constraints in the presence of a single machine.

Regardless of processing times, all problems considering different job sizes are in the class of NP-hard. The additional difficulty in our problem is due to different release dates.

2.1.2. Heuristic and approximation methods

Most studies on batch scheduling with different job sizes focus on heuristic, meta-heuristic methods and approximation algorithms. Zhang, Cai, Lee, and Wong (2001) consider the case where jobs are available at the same time while having different sizes and processing times. They develop an approximation algorithm with a worst case performance ratio equal to 7/4 for makespan minimization on a single machine. Cheng, Yang, Hu, and Chen (2012) propose an approximation algorithm with a worst case ratio of 2 and $(8/3 - 2/3^{*}m)$ for makespan and total completion time criteria, respectively, in the presence of m identical machines. Li, Li, Wang, and Liu (2005) extend the problem studied in Zhang et al. (2001) by considering job release dates. They present a $2+\epsilon$ approximation algorithm which is derived from a polynomial time approximation scheme that they propose for the case where jobs have unit sizes. Lu, Feng, and Li (2010) use a similar approach and provide a $2+\epsilon$ approximation algorithm for bi-objective minimization of makespan and penalization of unscheduled jobs. Liu, Ng, and Cheng (2014) present heuristics and approximation algorithms for makespan minimization in the presence of unit size jobs with release dates and different processing times. Their work is later generalized to the case of different job sizes by Li (2012). Chou (2007) studies the same problem as in Li et al. (2005) and proposes a genetic algorithm using a dynamic programming procedure to find the makespan of a given chromosome.

Because in our problem we have release dates, different job sizes and parallel machines, the articles cited in this paragraph are more related to our problem. Li (2012) presents the only approximation algorithm with a worst case performance ratio equal to $2+\epsilon$ when jobs have different sizes, different processing times, release dates. There are, however, mostly heuristic/meta-heuristic methods in the literature for the batch scheduling problem studied by Li (2012). For the same problem, Chung, Tai, and Pearn (2009) propose a mixed integer linear programming model (MILP) and heuristics. Many other authors use the heuristics of Chung et al. (2009) for benchmarking. Wang and Chou (2010), Damodaran and Velez Gallego (2010) and Damodaran, Velez-Gallego, and Maya (2011) consider the same problem for which they develop a genetic algorithm, a greedy randomized adaptive search procedure (GRASP) meta-heuristic and a constructive heuristic, respectively. All report that their approaches outperform the heuristics proposed in Chung et al. (2009). In another work, Damodaran and Velez-Gallego (2012) propose a simulated annealing algorithm which is able compete with the GRASP approach. Ozturk, Espinouse, Di Mascolo, and Gouin (2012) develop a MILP model that runs faster than that proposed by Chung et al. (2009) for the case with equal job processing times. They also treat some special cases and provide optimal greedy algorithms. Recently, Pearn, Hong, and Tai (2013) enlarge the broad of the problem considering job families, due dates and set-up times between the processing of batches from different families.

2.2. Contribution of this paper

The method we propose exploits the structural properties of the problem under study. It is based on constructing a search tree where each node represents a job release date or the starting time of batch processing thanks to equal job processing time property. Numerical tests show that our branch bound based heuristic method ($B\&B_H$) can solve problem instances containing up to 40 jobs in short computational time and can solve larger instances in reasonable time. MILP model of Ozturk et al. (2012) can find the optimal solution for small and medium size instances but it requires too much computational time. Regarding other methods from the literature, benchmarking results show that our method's solution quality is higher than two other heuristics from the literature. Our method is applicable in sterilization services since it can quickly solve real size instances.

3. Problem description, notation and complexity

We begin with definitions and notation:

- There are m identical parallel machines with a limited capacity B.
- There are n jobs to be processed. A job is a task that is characterized by a release date, r_j, a size, w_j, and a processing time, p.
- The size of a job cannot be greater than the machine capacity.
- Since washing times are the same for all RMD sets, job processing times are the same for all jobs.
- A batch is composed of jobs processed at the same time on the same machine. Several jobs can be batched together, complying with the machine capacity constraint.
- Once the processing of a batch is started, it cannot be interrupted (i.e. pre-emption is not allowed). Jobs cannot be split into multiple batches.
- The objective is to minimize makespan.