European Journal of Operational Research 255 (2016) 21–33

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

# Modeling the Parallel Machine Scheduling Problem with Step Deteriorating Jobs

Eduardo Lalla-Ruiz[a], Stefan Voß[a,b,*]

[a] Universität Hamburg, Institute of Information Systems (IWI), Von-Melle-Park 5, 20146 Hamburg, Germany
[b] Escuela de Ingeniería Industrial, Pontificia Universidad Católica de Valparaíso, Avenida Brasil 2241, Valparaíso 2362807, Chile

ARTICLE INFO

*Article history:*
Received 20 May 2015
Accepted 6 April 2016
Available online 13 April 2016

*Keywords:*
Machine scheduling
Step deterioration
Mathematical programming
Parallel machine scheduling with step deteriorating jobs
Set-partitioning formulations

ABSTRACT

This paper addresses the Parallel Machine Scheduling Problem with Step Deteriorating Jobs. This problem arises from real environments in which processing a job later than at a specific time may require an extra processing time. This time-dependent variation is known in the literature as step deterioration and has several practical applications (production planning, computer programming, medicine treatment, equipment maintenance, etc.). In the problem tackled in this work we aim to minimize the total completion time on identical parallel machines where each job has a deteriorating date and observes a step function for the processing time. For solving it, we propose two novel mathematical models based on the Set Partitioning Problem (SPP) that improve the unique model proposed in the literature. The computational performance of these models implemented in a general purpose solver allows to compete with the best algorithms proposed in the literature. Finally, we provide some insights for managing similar SPP formulations when large-sized instances have to be addressed.



## 1. Introduction

The problem of scheduling $n$ independent jobs with different processing times on $m$ identical parallel machines considering job processing times following a step function arises to address a variety of practical situations. Some applications, as indicated in (Cheng & Ding, 2001; Jeng & Lin, 2005; Mosheiov, 1995; Sundararaghavan & Kunnathur, 1994), can be production planning, computer programming, medicine treatment, equipment maintenance, and risk management, among others. The way this problem differs from the traditional parallel machine scheduling problem is based on the consideration of the processing times. In this problem, the processing time of each job may vary depending on its starting time. Namely, if a job is not processed during a specific interval, the processing time is increased by a specific amount following a step function. This allows to tackle scenarios where, for example, a product associated with a determined job requires to be started prior to a certain pre-specified deadline, otherwise additional actions have to be performed which require an extra processing time. An illustrative example can be taken from (Sundararaghavan & Kunnathur, 1994) where a given job or class of jobs can be processed during the time-frame where the machine is warmed-up. After that period (*i.e.*, deteriorating date), the machine has to be prepared for processing the job, which incurs in an extra time. This means that each job has a starting time-dependent setup time where it depends on the job itself and the time when it starts to be processed.

In this paper, we consider the Parallel Machine Scheduling Problem with Step Deteriorating Jobs (denoted as $Pm|p_i = a_i$ or $a_i + b_i|\sum C_i$) that seeks to minimize the total completion time proposed by Cheng, Guo, Zhang, Zeng, and Liang (2012). The main goals of the present work can be summarized as follows:

(i) We provide two mathematical formulations that allow to solve, on the one hand, in less computational time those instances where the optimal solution was already known and, on the other hand, solve to optimality other instances that have not been solved before. That is, the mathematical formulations proposed, due to their performance in terms of computational time, are suitable when addressing real-world scenarios. For assessing the performance of our formulations we employ the most common benchmark instances for this problem proposed by Cheng et al. (2012) and generated new problem instances in the same way as done in (Cheng et al., 2012). It should be noted that the computational results reported in this work allow to appropriately assess the

* Corresponding author at: Institute of Information Systems (IWI), University of Hamburg, Germany. Tel.: +49 40 42838 3062; fax: +49 40 42838 5535.
*E-mail addresses:* eduardo.lalla-ruiz@uni-hamburg.de (E. Lalla-Ruiz), stefan.voss@uni-hamburg.de (S. Voß).

http://dx.doi.org/10.1016/j.ejor.2016.04.010

performance of the best exact solution approach for this problem (Cheng et al., 2012) in large-sized instances.

(ii) We present some new insights with respect to Set Partitioning Problem (SPP) formulations. In this regard, one of the major drawbacks of SPP formulations is the difficulty to directly handle large-sized instances since they may require large amounts of memory. This, as discussed in Section 4.3, happens at several problem instances of $Pm|p_i = a_i$ or $a_i + b_i|\sum C_i$ provided in (Cheng et al., 2012) as well as some of the instances proposed in this work. Hence, in order to tackle larger instances we propose to implicitly treat those parts of the problem structure that are redundant. In this sense, through our approach we aim to reduce the size of the problem structure and, in this way solve larger instances. Therefore, in our work, we propose a new way to address some SPP mathematical formulations that are based on enumerating all possible cases (*e.g.*, each column represents a single assignment or scheduling) that can be translated to related problems.

(iii) Finally, as a side conclusion from the extensive computational experiments done in this work, we have detected some erraticism among the reformulations. To gain some related insights, we compare both reformulations in terms of variables, constraints and non-zeros, in order to assess and detect the effect of reformulations within a commercial solver like, e.g., CPLEX. In this sense, while Fischetti and Monaci (2014) investigate the variability associated to initial conditions, we aim to investigate the effect of handling different formulations of the same dimensions for the same problem.

To summarize, we aim to gain new insights on modeling the problem. While doing so, we show that we can solve considerably larger instances to optimality than it was possible before in literature. The remainder of this paper is organized as follows. After a concise literature review in Section 2, the proposed mathematical formulations are presented in Section 3. Our computational experiments and a comparison summary are presented in Section 4. Finally, some conclusions and lines for further research are drawn in Section 5.

## 2. Literature review

The Parallel Machine Scheduling Problem with Step Deteriorating Jobs denoted as $Pm|p_i = a_i$ or $a_i + b_i|\sum C_i$ is a recent problem proposed by Cheng et al. (2012) aimed at capturing real contexts where the job processing times are subject to a step function, with non-common deadlines and varying processing times on $m$ identical parallel machines. Due to its novelty there are only few works addressing this problem. The majority of works address the single machine case. In this literature review we mainly include papers dealing directly with related problems. For a general overview about parallel machine scheduling problems, the reader is referred to Blazewicz, Dror, and Weglarz (1991); Cheng and Sin (1990); Leung (2004), and Pinedo (2012).

Some of the first works studying step deteriorating models are from Mosheiov (1995); Sundararaghavan and Kunnathur (1994), and Cheng and Ding (2001). In (Mosheiov, 1995) the step deterioration in single and multi-machine problems seeks to minimize the makespan. He demonstrates that this problem is $\mathcal{NP}$-hard and proposes some heuristics for solving it. Sundararaghavan and Kunnathur (1994) study the single machine case where the jobs have a penalty for starting after the due date. In their work they aim to minimize the total weighted completion time and a common normal processing time. They propose four algorithms, where for the first algorithm they conjecture that it is optimal when the machines have a common due date. Cheng and Ding (2001) study the single machine scheduling case where all the jobs share a common deterioration-date and propose a pseudo-polynomial algorithm for minimizing the makespan. Moreover, they provide a counter-example for the conjecture of optimality of the switching algorithm proposed by Sundararaghavan and Kunnathur (1994).

Further relevant works tackling single machine cases are the following. Jeng and Lin (2004) address the single machine scheduling with step deteriorating processing times for minimizing the maximum completion time. They propose two dominance rules and a lower bound to design a branch-and-bound algorithm. The computational results indicate that the contribution of the dominance rules and the lower bound proposed is significant. Jeng and Lin (2005) aim to minimize the total completion time on a single machine with jobs considering a step deteriorating function; they propose two elimination rules and a lower bound. Branch-and-bound algorithms incorporating those properties are described. The results show that the related properties are effective to avoid unnecessary branches. Layegh, Jolai, and Amalnik (2009) investigate the total weighted completion time minimization for a single machine under step deterioration. They present new lemmas and dominance properties for this problem. Additionally, they develop a memetic algorithm (MA), an evolutionary population-based metaheuristic search similar to genetic algorithms that is combined with problem-specific solvers. In this case, the authors combine it with neighborhood search methods that consider some of the properties and lemmas proposed in their work. He, Wu, and Lee (2009) derive some dominance properties and a lower bound for minimizing the sum of completion times. They include them in a branch-and-bound method to exalt its efficiency. Moreover, they propose a heuristic algorithm termed as Weight Combination Search Approach (WCSA). The computational results indicate that the branch-and-bound approach can solve problem instances up to 24 jobs in reasonable computational time and their proposed heuristic yields good quality solutions for those instances exhibiting a deviation from the optimum value of around 0.3 percent, on average. Guo, Cheng, and Wang (2014) address the single machine scheduling problem with step function processing times and aim to minimize the total tardiness function. For solving this problem they propose a heuristic termed as Simple Weighted Search Procedure (SWSP) and they also develop a General Variable Neighborhood Search algorithm (GVNS). The computational results reported in that work indicate that the GVNS presents a competitive performance in terms of percentage deviation from the best solution known for the instances addressed.

Leung, Ng, and Cheng (2008) consider the batch scheduling of jobs where the processing time of each job is under a step function based on its waiting time. The objective in this work is to minimize the sum of completion times. They study the $\mathcal{NP}$-hardness of this problem and provide an approximation algorithm. Mor and Mosheiov (2012) combine the model from Cheng and Ding (2001) where all the jobs share a common deterioration date with the batch setting proposed in (Santos & Magazine, 1985) which assumes both identical job processing times and identical (batch-independent) setup times. The objective function tackled is to minimize the total flow time assuming a single machine, unit jobs and batch-independent setup times. As future work they aim to extend this problem to parallel identical machines, which, as explained below, is done in (Cheng et al., 2012) as well as in this paper.

Cheng et al. (2012) study the total completion time minimization on identical parallel machines and propose a Mixed Integer Programming (MIP) formulation; they indicate that this problem is $\mathcal{NP}$-hard because the single machine problem has been proven to be $\mathcal{NP}$-hard by Mosheiov (1995). Moreover, since there does not exist a polynomial time algorithm for the exact solution of this problem they propose a modified heuristic, MWCSA, based on