



Minimizing the makespan on parallel machines with sequence dependent deteriorating effects

Olinto César Bassi de Araújo
Universidade Federal de Santa Maria
olinto@ctism.ufsm.br

Guilherme Dhein
Universidade Federal de Santa Maria
gdhein@redes.ufsm.br

Marcia Fampa
Universidade Federal do Rio de Janeiro
fampa@cos.ufrj.br

ABSTRACT

We address in this paper a minimum makespan problem, where jobs are scheduled in parallel machines with deteriorating effects. The problem has been modeled in the literature by a mixed integer nonlinear programming formulation and heuristic approaches were applied to obtain feasible solutions for it. We first apply a linearization strategy to build a mixed integer linear programming formulation for the problem, and then consider proved results about its optimal solution to enhance this formulation. Finally, valid inequalities for the problem are proposed. Numerical experiments were conducted, showing the strength of our formulations, and solving benchmark instances to optimality for the first time.

KEYWORDS. Job scheduling, parallel machines, machine and job deterioration, MINLP, linearization, valid inequalities.

Mathematical Programming, Combinatorial optimization



1. Introduction

This paper addresses a job scheduling problem with sequence deterioration effects on unrelated parallel machines. In real world situations, it is expected that jobs scheduled later in a sequence will require more time to be processed. In some cases, the deterioration effect is slow, so it can be neglected when a short horizon planning is used, but in other cases, the deterioration is more evident and results in a significant deterioration in the performance of the machines. In such cases, neglecting the deterioration may certainly lead to an infeasible scheduling.

In this paper, we address the specific scheduling problem that was proposed in [Ruiz-Torres et al., 2013], where the authors present a mixed integer nonlinear programming (MINLP) formulation and design simulated annealing heuristics for the problem. Both formulation and heuristic procedures are evaluated on benchmark instances. Here, we apply linearization techniques to formulate the problem as a mixed integer linear program and then consider specific characteristics of the optimal job scheduling sequence on a single machine, to enhance this formulation. Valid inequalities are also proposed to produce a tighter formulation that presents a much lower computational burden. The computational experiments reveal that the models proposed can be solved by commercial solvers providing optimal solutions and best known solutions for the set of benchmark instances.

The seminar publications on scheduling problems with deterioration are considered to be [Gupta et al., 1987] and [Browne and Yechiali, 1990]. In both papers, the minimization of the makespan on a single machine is addressed and the job processing time is an increasing function of its waiting time for processing. In some other papers the deterioration is defined as a function of the position that the job occupies in the sequence on a machine, as in [Mosheiov, 2012], where a problem with multiple identical parallel machines is considered, aiming to minimize the total load.

In [Yang et al., 2012] and [Yang, 2011], maintenance activities on the machines are taken into account, which restore their performance to the initial level. The deterioration of a job depends on its position in the sequence and also on the most recent maintenance. The first work considers identical machines, while the second work considers unrelated machines. In [Biskup, 1999], the concept of learning in single-machine scheduling problems was introduced. According to this concept, the repeated execution of similar jobs results in learning that decreases the time required to execute subsequent jobs. More recently, some works have addressed problems in which learning and deteriorating effects coexist, as, for example, [Wang and Wang, 2014], [Niu et al., 2015], and [Rostami et al., 2015].

In this work, we admit that the deterioration occurs in the machine and not in the jobs. Processing a job reduces the efficiency of the machine and consequently increases the processing time of subsequent jobs. As the machines are unrelated, the deterioration depends on the sequence of processed jobs and also on the machine. A real world example for this type of deterioration occurs in the operation of cutting or drilling resistant materials. The wear of the saw blades and drill bits that results from one job execution directly impacts the processing time of the next job. It is interesting to note that the same deterioration in the level of performance can happen to people or teams, due to fatigue caused by performing previous tasks.

The machine deterioration as a function of the sequencing of jobs is considered in different works. In [Santos and Arroyo, 2015], Iterated Local Search (ILS) and ILS with random variable neighborhood descent are applied to the problem proposed in [Ruiz-Torres et al., 2013]. In [Ruiz-Torres et al., 2014], the objective is to minimize the total tardiness, in [Ruiz-Torres et al., 2015], the objective is to maximize the percentage of jobs completed on time, and in [Ruiz-Torres et al., 2017], the objective is to minimize the makespan in a problem where machines are subject to maintenance. In all cases, heuristic approaches are proposed to the problems.

In Section 2, we define the specific problem addressed in this paper, present the MINLP formulation proposed for the problem in [Ruiz-Torres et al., 2013], and propose our new formulations based on characteristics of the optimal scheduling and on the valid inequalities developed. In



Section 3, we present results from an extensive numerical experiment that shows the effectiveness of the valid inequalities proposed and tightness of the formulation presented. Concluding remarks are made in Section 4.

2. The minimum makespan problem addressed

The minimum makespan problem addressed in this work can be specified as follows.

There are n independent jobs, $N = \{1, \dots, n\}$, to be scheduled and processed on m parallel machines, $M = \{1, \dots, m\}$. All the jobs are non-preemptive and available for processing at time zero. Each machine can process only one job at a time and cannot stand idle until the last job assigned to it has been finished.

For all $j \in N$ and $k \in M$, we denote by p_{jk} , the given baseline processing time of job j on machine k , if no deterioration is considered. We denote by $d_{jk} \in [0, 1)$, the deteriorating effect of job j on machine k . Considering then the set $H = \{1, \dots, n\}$ as the set of all possible n positions for a job in each machine, let $\tau(h, k)$ be the job assigned to position h of machine k , for all $h \in H$ and $k \in M$. Let q_{kh} be the performance level of machine k for the job in position h , which is defined by

$$q_{kh} := (1 - d_{\tau(h-1, k)k}) \times q_{k(h-1)}, \quad (1)$$

for each machine $k \in M$ and each position h greater than 1. It is assumed that the machines start with no deterioration, i.e. $q_{k1} = 1$ for all $k \in M$. The actual processing time of the job $\tau(h, k)$ is

$$p'_{\tau(h, k)k} = \frac{p_{\tau(h, k)k}}{q_{kh}}.$$

The objective of the problem is to schedule the jobs on the machines so that the makespan is minimized, where the makespan is defined as the time to complete all jobs, and the scheduling of the jobs is defined by their assignment to a machine and by their processing sequence on the machines. In the next subsection, we present the mixed integer nonlinear programming (MINLP) formulation proposed in [Ruiz-Torres et al., 2013] for this problem.

2.1 A MINLP formulation from the literature

In [Ruiz-Torres et al., 2013], the authors define the binary variables

$$x_{jkh} := \begin{cases} 1, & \text{if job } j \text{ is assigned to machine } k \text{ in position } h, \\ 0, & \text{otherwise,} \end{cases}$$

for all $j \in N$, $k \in M$, and $h \in H$. They also define the nonnegative real variable C_{\max} , which represents the makespan, and the nonnegative real variables q_{kh} , which represent the performance level of machine k for the job in position h , for all $k \in M$, and $h \in H$.

The problem is then formulated as the following MINLP:

$$(P_1) \quad \min C_{\max}, \quad (2)$$

$$\sum_{j \in N} x_{jkh} \leq 1, \quad \forall k \in M, h \in H, \quad (3)$$

$$\sum_{k \in M} \sum_{h \in H} x_{jkh} = 1, \quad \forall j \in N, \quad (4)$$

$$\sum_{j \in N} \sum_{h \in H} \frac{p_{jk}}{q_{kh}} \times x_{jkh} \leq C_{\max}, \quad \forall k \in M, \quad (5)$$

$$x_{jkh} \leq \sum_{\ell \in N} x_{\ell k(h-1)}, \quad \forall j \in N, k \in M, h \in H \setminus \{1\}, \quad (6)$$

$$q_{kh} = \sum_{j \in N} (1 - d_{jk}) \times q_{k(h-1)} \times x_{jk(h-1)}, \quad \forall k \in M, h \in H \setminus \{1\}, \quad (7)$$

$$q_{k1} = 1, \quad \forall k \in M, \quad (8)$$

$$x_{jkh} \in \{0, 1\}, \quad \forall j \in N, k \in M, h \in H, \quad (9)$$



where (2) is the objective function of the problem, (3) enforces that at most one job is assigned to each position in each machine, (4) enforces that all jobs are assigned to exactly one position in one machine, (5) defines the makespan as the maximum total processing time among all machines, (6) enforces continuous assignments, and (7) and (8) define the performance level of each machine for each job position.

Finally, in [Ruiz-Torres et al., 2013] the following lemma concerning the minimum makespan is proved. We include it in this paper because it is an important result for the development of our work.

Lemma 1 *Let $n_k \leq n$ be the number of jobs assigned to machine k . Let $\tau(h, k)$ be the job assigned to position h of machine k , for all $h \in H$ and $k \in M$. Then, if the job scheduling satisfies:*

$$\frac{p_{\tau(1,k)k}(1 - d_{\tau(1,k)k})}{d_{\tau(1,k)k}} \geq \frac{p_{\tau(2,k)k}(1 - d_{\tau(2,k)k})}{d_{\tau(2,k)k}} \geq \dots \geq \frac{p_{\tau(n_k,k)k}(1 - d_{\tau(n_k,k)k})}{d_{\tau(n_k,k)k}}, \quad (10)$$

the completion time of all the jobs assigned to machine k is minimum.

2.2 Linearizing the formulation from the literature

Let's consider in this subsection the same notation previously introduced, and also define:

$$\tilde{q}_{kh} := \frac{1}{q_{kh}},$$

for all $k \in M$ and $h \in H$, and

$$\tilde{d}_{jk} := \frac{d_{jk}}{1 - d_{jk}},$$

for all $j \in N$ and $k \in M$. Note that

$$\frac{1}{1 - d_{jk}} = 1 + \tilde{d}_{jk},$$

therefore, following (1), we set

$$\tilde{q}_{kh} := (1 + \tilde{d}_{\tau(h-1,k)k}) \times \tilde{q}_{k(h-1)}, \quad (11)$$

for each machine $k \in M$ and each position h greater than 1, and $\tilde{q}_{k1} = 1$ for all $k \in M$.

We now rewrite (7) as

$$\tilde{q}_{kh} = \tilde{q}_{k(h-1)} \times \sum_{j \in N} (1 + \tilde{d}_{jk}) \times x_{jk(h-1)}, \quad (12)$$

for all $k \in M$ and $h \in H \setminus \{1\}$.

Using a parameter \bar{M} large enough, and considering that

$$\sum_{j \in N} x_{jk(h-1)} \leq 1,$$

for all $k \in M$ and $h \in H \setminus \{1\}$. it is straightforward to see that (12) can be linearized, and thereby (7) and (8) can be substituted in (P_1) by

$$\begin{aligned} \tilde{q}_{kh} &\geq (1 + \tilde{d}_{jk}) \times \tilde{q}_{k(h-1)} - \bar{M}(1 - x_{jk(h-1)}), & \forall j \in N, k \in M, h \in H \setminus \{1\}, \\ \tilde{q}_{kh} &\geq 0, & \forall k \in M, h \in H \setminus \{1\}, \\ \tilde{q}_{k1} &= 1, & \forall k \in M. \end{aligned}$$



Based on the same ideas, considering new nonnegative real variables u_{jk} , for all $j \in N$, and $k \in M$, and a suitable parameter \tilde{M} , it is possible to substitute (5) in (P_1) by

$$\begin{aligned} \sum_{j \in N} u_{jk} &\leq C_{max}, & \forall k \in M, \\ u_{jk} &\geq p_{jk} \tilde{q}_{kh} - \tilde{M}(1 - x_{jkh}), & \forall j \in N, k \in M, h \in H, \\ u_{jk} &\geq 0, & \forall j \in N, k \in M. \end{aligned}$$

The substitutions mentioned above, lead to the following mixed linear integer programming (MILP) formulation for the problem

$$\begin{aligned} (P_2) \quad \min \quad & C_{max}, \\ & \sum_{j \in N} x_{jkh} \leq 1, & \forall k \in M, h \in H, \\ & \sum_{k \in M} \sum_{h \in H} x_{jkh} = 1, & \forall j \in N, \\ & \sum_{j \in N} u_{jk} \leq C_{max}, & \forall k \in M, \\ & u_{jk} \geq p_{jk} \tilde{q}_{kh} - \tilde{M}(1 - x_{jkh}), & \forall j \in N, k \in M, h \in H, \\ & u_{jk} \geq 0, & \forall j \in N, k \in M, \\ & x_{jkh} \leq \sum_{\ell \in N} x_{\ell k(h-1)}, & \forall j \in N, k \in M, h \in H \setminus \{1\}, \\ & \tilde{q}_{kh} \geq (1 + \tilde{d}_{jk}) \times \tilde{q}_{k(h-1)} - \tilde{M}(1 - x_{jk(h-1)}), & \forall j \in N, k \in M, h \in H \setminus \{1\}, \\ & \tilde{q}_{kh} \geq 0, & \forall j \in N, k \in M, h \in H \setminus \{1\}, \\ & \tilde{q}_{k1} = 1, & \forall k \in M, \\ & x_{jkh} \in \{0, 1\}, & \forall j \in N, k \in M, h \in H. \end{aligned}$$

2.3 An improved MILP formulation

We now propose a new MILP formulation for our minimum makespan problem, that represents an improved version of (P_2) . The improvement is based on the observation that, once the assignment of jobs to machines is decided, an optimized processing sequence of the jobs on each machine can be easily determined from the result in Lemma 1.

Following this idea, for each machine $k \in M$, we let $(j_k(1), j_k(2), \dots, j_k(n))$ be a permutation of the elements in N , such that

$$\frac{p_{j_k(1)k}(1 - d_{j_k(1)k})}{d_{j_k(1)k}} \geq \frac{p_{j_k(2)k}(1 - d_{j_k(2)k})}{d_{j_k(2)k}} \geq \dots \geq \frac{p_{j_k(n)k}(1 - d_{j_k(n)k})}{d_{j_k(n)k}}.$$

We then propose a new formulation for the problem, where the $n^2 \times m$ binary decisions variables in (P_2) , namely x_{jkh} , for all $j \in N$, $k \in M$, and $h \in H$, are replaced by only $n \times m$ binary variables. These variables are denoted by x_{jk} , for all $j \in N$ and $k \in M$, and indicate whether or not job j is assigned to machine k . If $x_{jk} = 1$, we assume that job j is scheduled to be processed in position $j_k(i)$ of the machine k , for i such that $j_k(i) = j$.

Note that by defining the positions of the jobs in the machines using this procedure, the relation (10) in Lemma 1 will be certainly satisfied, and therefore, the completion time of the jobs in each machine will be minimized. On the other hand, the jobs may not be continuously assigned to the positions of the machines, as in formulations (P_1) and (P_2) , and this fact should be considered



in order to correctly model in the improved formulation, the performance level of the machines for the jobs in each position. We deal with this particularity of the model, by updating our performance level \tilde{q}_{kh} for $h > 1$ according to (11), only if a job is assigned to position $h - 1$ in machine k , otherwise, we simply consider \tilde{q}_{kh} to be equal to $\tilde{q}_{k(h-1)}$.

We next present the new MILP formulation for the problem addressed. The basic idea of the formulation is then, to identify $j_k(i)$ as the only job that can possibly be assigned to position i of machine k , for $i = 1, \dots, n$, and $k \in M$, and to update the performance level \tilde{q}_{kh} , for all $k \in M$ and $h \in H \setminus \{1\}$, as explained above.

$$\begin{aligned}
 (P_3) \quad & \min C_{\max} , \\
 & \sum_{k \in M} x_{jk} = 1 , & \forall j \in N , \\
 & \sum_{j \in N} u_{jk} \leq C_{\max} , & \forall k \in M , \\
 & u_{j_k(h)k} \geq p_{j_k(h)k} \tilde{q}_{kh} - \mathbb{M}_1(k, h) (1 - x_{j_k(h)k}) , & \forall k \in M, h \in H , \\
 & u_{jk} \geq 0 , & \forall j \in N, k \in M , \\
 & \tilde{q}_{kh} \geq (1 + \tilde{d}_{j_k(h-1)k}) \times \tilde{q}_{k(h-1)} - \mathbb{M}_2(k, h) (1 - x_{j_k(h-1)k}) , & \forall k \in M, h \in H \setminus \{1\} , \\
 & \tilde{q}_{kh} \geq \tilde{q}_{k(h-1)} , & \forall k \in M, h \in H \setminus \{1\} , \\
 & \tilde{q}_{k1} = 1 , & \forall k \in M , \\
 & x_{jk} \in \{0, 1\} , & \forall j \in N, k \in M .
 \end{aligned}$$

We note that besides the decrease on the number of binary variables in (P_3) , when compared to (P_2) , there is also a significant decrease on the number of constraints. While we have $n + 2m + 2mn + 3mn(n - 1) + mn^2$ constraints in (P_2) , we have only $n + 2m + 2mn + 2m(n - 1)$ constraints in (P_3) .

The following proposition determines values for the parameters $\mathbb{M}_1(k, h)$ and $\mathbb{M}_2(k, h)$ in (P_3) , which guarantee the identity between the optimal makespan given by its solution and the solution of (P_1) . The proof of the proposition is straightforward.

Proposition 1 *Let us set the parameters $\mathbb{M}_1(k, h)$ and $\mathbb{M}_2(k, h)$ in (P_3) as*

$$\mathbb{M}_2(k, h) := \prod_{\ell=1}^{h-1} (1 + \tilde{d}_{j_k(\ell)k}) , \quad \mathbb{M}_1(k, h) := p_{j_k(h)k} \times \mathbb{M}_2(k, h) .$$

Then the optimal solution values of (P_3) and (P_1) are the same.

2.4 Valid inequalities

In the following propositions we present valid inequalities to strengthen (P_3) .

Proposition 2 *If job j is assigned to machine k , its actual processing time u_{jk} is no less than the baseline processing time p_{jk} . This observation can be modeled by the valid inequalities:*

$$p_{jk} x_{jk} \leq u_{jk} , \tag{13}$$

for all $j \in N$ and $k \in M$.

Proposition 3 *The following inequalities are satisfied by any feasible solution of (P_3) :*

$$\tilde{q}_{kh} \geq 1 + \sum_{\ell=1}^{h-1} \tilde{d}_{j_k(\ell)k} x_{j_k(\ell)k} , \tag{14}$$



for all $k \in M$ and $h \in H$. Moreover, defining $y_k(i, j) := x_{ik}x_{jk}$, for $k \in M$, and $i, j \in N$, we also have

$$\begin{aligned} \tilde{q}_{kh} &\geq 1 + \sum_{\ell=1}^{h-1} \tilde{d}_{j_k(\ell)k} x_{j_k(\ell)k} + \sum_{\ell=1}^{h-1} \sum_{s=\ell+1}^{h-1} \tilde{d}_{j_k(\ell)k} \tilde{d}_{j_k(s)k} y_k(j_k(\ell), j_k(s)), \\ y_k(j_k(\ell), j_k(s)) &\leq x_{j_k(\ell)k}, & \forall \ell, s \in H, \ell < s, \\ y_k(j_k(\ell), j_k(s)) &\leq x_{j_k(s)k}, & \forall \ell, s \in H, \ell < s, \\ y_k(j_k(\ell), j_k(s)) &\geq x_{j_k(\ell)k} + x_{j_k(s)k} - 1, & \forall \ell, s \in H, \ell < s, \\ y_k(j_k(\ell), j_k(s)) &\geq 0, & \forall \ell, s \in H, \ell < s, \end{aligned} \quad (15)$$

for all $k \in M$ and $h \in H$.

Proof The case $h = 1$ is trivial, as $\tilde{q}_{k1} = 1$, for all $k \in M$. Therefore, in the following we assume $h > 1$. From (12), for all $h \in H \setminus \{1\}$, we have

$$\tilde{q}_{kh} = \prod_{\ell \in \mathcal{L}_{kh}} \left(1 + \tilde{d}_{j_k(\ell)k} \right),$$

where $\mathcal{L}_{kh} \subseteq \{1, \dots, h-1\}$ is the set of indexes such that $\ell \in \mathcal{L}_{kh} \iff x_{j_k(\ell)k} = 1$, or, equivalently:

$$\tilde{q}_{kh} = \prod_{\ell=1}^{h-1} \left(1 + \tilde{d}_{j_k(\ell)k} x_{j_k(\ell)k} \right),$$

as $\tilde{q}_{k1} = 1$, for all $k \in M$.

Now, let's consider the following expression for the product of binomials [Abramowitz and Stegun, 1972, p.10]

$$\prod_{i=1}^r (1 + w_i) = \sum_{k_1=0}^1 \dots \sum_{k_r=0}^1 w_1^{1-k_1} \dots w_r^{1-k_r}, \quad (16)$$

or equivalently,

$$\begin{aligned} \prod_{i=1}^r (1 + w_i) &= 1 + \sum_{k_1=1}^r w_{k_1} + \sum_{k_1=1}^r \sum_{k_2=k_1+1}^r w_{k_1} w_{k_2} + \\ &+ \sum_{k_1=1}^r \sum_{k_2=k_1+1}^r \sum_{k_3=k_2+1}^r w_{k_1} w_{k_2} w_{k_3} + \dots + w_1 w_2 \dots w_r. \end{aligned} \quad (17)$$

Then, considering $w_i \geq 0$, for all $i = 1, \dots, r$, and applying the standard linearization strategy to enforce $y_k(i, j) := x_{ik}x_{jk}$, the results of the proposition are easily obtained. \square

Remark 1 Note that, besides (14) and (15), other linear valid inequalities could be generated from (17), by considering the linearization of the monomials of degree greater than 2.

Proposition 4 Let $y_k(i, j) := x_{ik}x_{jk}$, for $k \in M$, and $i, j \in N$. Then, for all $k \in M$, we have

$$\begin{aligned} \sum_{s \in H} p_{j_k(s)k} \left(x_{j_k(s)k} + \sum_{\ell=1}^{s-1} \tilde{d}_{j_k(\ell)k} y_k(j_k(\ell), j_k(s)) \right) &\leq C_{max}, \quad \forall k \in M, \\ y_k(j_k(\ell), j_k(s)) &\leq x_{j_k(\ell)k}, & \forall \ell, s \in H, \ell < s, \\ y_k(j_k(\ell), j_k(s)) &\leq x_{j_k(s)k}, & \forall \ell, s \in H, \ell < s, \\ y_k(j_k(\ell), j_k(s)) &\geq x_{j_k(\ell)k} + x_{j_k(s)k} - 1, & \forall \ell, s \in H, \ell < s, \\ y_k(j_k(\ell), j_k(s)) &\geq 0, & \forall \ell, s \in H, \ell < s, \end{aligned} \quad (18)$$



Remark 2 Note that, besides (18) other linear valid inequalities could also be generated, by considering the linearization of the monomials of degree greater than 2. As an example, for the case of monomials of degree 3, if we define $t_k(j_1, j_2, j_3) := x_{j_1 k} x_{j_2 k} x_{j_3 k}$, we could add to (P_3) , for all $k \in M$, the valid inequalities

$$\sum_{s \in H} p_{j_k(s)k} \left(x_{j_k(s)k} + \sum_{\ell=1}^{s-1} \sum_{r=1}^{\ell-1} \left(\tilde{d}_{j_k(r)} + \tilde{d}_{j_k(\ell)} + \tilde{d}_{j_k(r)} \tilde{d}_{j_k(\ell)} \right) t_k(j_k(r), j_k(\ell), j_k(s)) \right) \leq C_{max} ,$$

Remark 3 The inequalities

$$\begin{aligned} y_k(j_k(\ell), j_k(s)) &\leq x_{j_k(\ell)k} , \quad \forall \ell, s \in H \setminus \{1\}, \ell < s , \\ y_k(j_k(\ell), j_k(s)) &\leq x_{j_k(s)k} , \quad \forall \ell, s \in H \setminus \{1\}, \ell < s , \end{aligned}$$

for all $k \in M$, in the set of valid inequalities (15) and (18), could be omitted, due to the minimization of the objective function of the problem addressed.

3. Numerical experiments

In this section, we evaluate through computational experiments, three formulations presented for the scheduling problem addressed in this paper. The first one is formulation (P_3) presented in Subsection 2.3. The second, denoted here as (P_3^+) is formulation (P_3) with the addition of the valid inequalities (13), defined in Proposition 2, and (14) and (15), defined in Proposition 3. The third formulation, denoted here as (P_3^{++}) is formulation (P_3^+) with the addition of the valid inequalities (18), defined in Proposition 4. Our computational experiments were performed on a computer with an Intel ES-2680 processor, a clock speed of 2.7 GHz, and 64 GB RAM. The tests were executed in a time limit of 1 hour per instance (CPU time). We used IBM ILOG CPLEX Optimization Studio v12.6.2.0 with default settings and a maximum of 4 threads for solving problems (P_3) , (P_3^+) and (P_3^{++}) .

The benchmark instances used in the experiments were proposed in [Ruiz-Torres et al., 2013] and are available at <http://ruiz-torres.uprrp.edu/dm/>. The instances were generated based on four parameters: number of machines m , number of jobs n , range of processing time p_{range} and range of deterioration effect d_{range} . Two sets of instances are defined: small instances with 2, 3, or 4 machines and 8, 11, or 14 jobs, and large instances with 4, 7 or 10 machines and 20, 35 or 50 jobs. The values for the processing time (p_{jk}) were randomly selected in the range p_{range} , which was set equal to [1, 100] or [100, 200]. The values for the deteriorating effect (d_{jk}) were randomly selected in the range d_{range} , which was set equal to [1%, 5%] or [5%, 10%]. In total, 1800 instances were generated, 900 small instances and 900 large instances.

The solver could obtain the optimal solution of all small instances with each of the three formulations. The total CPU times to solve these 900 instances were 167.90, 170.57, and 315.64 seconds, with formulations (P_3) , (P_3^+) and (P_3^{++}) , respectively. As could be expected for these easy instances, the addition of the valid inequalities increases the average CPU times in these tests, once the number of constraints of the problems solved increases with no significant decrease on the number of subproblems solved in the branch-and-bound algorithm executed in CPLEX.

Results for the more challenging instances, the 900 large instances, are shown in Table 1. Each line presents results concerning 25 instances that were generated with each configuration. The first four columns of the table define the configuration of the instances. The remaining columns present, for the three formulations evaluated, the average CPU time to obtain the optimal solution in seconds (T(sec)) for the 25 instances, the average percentage duality gap at the end of the execution (Gap(%)) for the 25 instances, and the number of instances for which the solver achieved optimality (Opt(#)). We can see from the results on the table that the difficulty of the problem increases with the processing time, the deterioration effect and with the number of jobs. The number of machines, on the other hand, does not impact the difficulty so clearly. Concerning all three measures: time,



gap, and number of optimal solutions obtained, formulation (P_3^{++}) presented the the best average results. It is clear from the results the effectiveness of the valid inequalities proposed, mainly on the most difficult problems.

Comparing (P_3^+) to (P_3), we see that the addition of the valid inequalities improved the results for the majority of the difficult configurations. In particular, note the significant decrease on the average gap of problems with 50 jobs. Comparing (P_3^{++}) to (P_3), we see further improvement. The average gap does not increase with the addition of all valid inequalities, for any configuration, and significantly decreases for some configurations. The average decrease is of 50%. Also, the number of instances solved only decreases for one configuration, going from 18 to 15, and on the other side, it significantly increases for several other configurations, going from 0 to 14 on another group. We also can note that the time to solve the problems can increase with the addition of valid inequalities, which again, is expected on the easiest instances. This increase is compensated by the decrease on the times for the most difficult instances.

We also did numerical experiments comparing the time spent by the solver to achieve a target solution, with the three formulations. This target solution is the best known solution in the literature for our test problems, also available at <http://ruiz-torres.uprrp.edu/dm/>. When the small instances are considered, the three formulations present very good results once more. The total CPU times to achieve the targets were 20.89, 24.98, and 76.24 seconds, for formulations (P_3), (P_3^+) and (P_3^{++}), respectively. Results for the large instances, are presented in Table 2. The first four columns of the table define the configuration of the instances. The remaining columns present, for the three formulations evaluated, the average CPU time, in seconds, to achieve the target solution, considering only the instances for which the target solution was found in the time limit of 1 hour ($TT_g(\text{sec})$), the number of instances for which the target solution was achieved in the time limit ($T_g(\#)$), and the number of instances for which we were able to improve the best known solution, obtaining a better solution than the target in the time limit of 1 hour ($BK(\#)$). It's interesting to see that in these experiments, formulation (P_3^+) is the winner on all the three measures. Clearly, the valid inequalities added in this problem were effective to increase the lower bounds of the subproblems computed in the branch-and-bound algorithm, without jeopardizing the achievement of good upper bounds on the time limit. For (P_3^{++}), we see the worst results on these experiments, showing that the delay on solving subproblems due to the addition of a large number of inequalities results also on a delay on the computation of good feasible solutions. One future experiment we could try, among other things, a DepthFirst Search (DFS) strategy on the branch-and-bound to avoid this delay.

4. Concluding remarks

The job scheduling problem with deterioration effects is quite relevant due to real world applications. It is NP-hard and has been approached mainly by heuristic procedures in the literature. We show in this paper that tight mixed integer linear programming formulations for this problem can be constructed with the knowledge that we have on certain characteristics of optimal schedulings and also with the development of valid inequalities. With the formulations proposed in this paper, we solved to optimality benchmark instances from the literature for the first time. We also improved the best known solution for about 40% of the 900 benchmark instances in the set of more difficult problems.

References

- Abramowitz, M. and Stegun, I. A. (1972). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. 9th printing, New York: Dover.
- Biskup, D. (1999). Single-machine scheduling with learning considerations. *European Journal of Operational Research*, 115(1):173 – 178. ISSN 0377-2217.
- Browne, S. and Yechiali, U. (1990). Scheduling deteriorating jobs on a single processor. *Oper. Res.*, 38(3):495–498. ISSN 0030-364X.



Table 1: Numerical experiments aiming at optimality

| p_{range} | d_{range} (%) | m | n | (P_3) | | | (P_3^+) | | | (P_3^{++}) | | |
|-------------|--------------------|-----|-----|------------|------------|------------|------------|------------|------------|--------------|------------|------------|
| | | | | T (sec) | Gap (%) | Opt (#) | T (sec) | Gap (%) | Opt (#) | T (sec) | Gap (%) | Opt (#) |
| [1,100] | [1,5] | 4 | 20 | 0,09 | 0,000 | 25 | 0,08 | 0,000 | 25 | 0,21 | 0,000 | 25 |
| | | | 35 | 0,53 | 0,000 | 25 | 0,43 | 0,000 | 25 | 1,94 | 0,000 | 25 |
| | | | 50 | 20.94 | 0.000 | 25 | 5.10 | 0.000 | 25 | 65.52 | 0.000 | 25 |
| | | 7 | 20 | 0.10 | 0.000 | 25 | 0.10 | 0.000 | 25 | 0.26 | 0.000 | 25 |
| | | | 35 | 0.75 | 0.000 | 25 | 0.91 | 0.000 | 25 | 3.88 | 0.000 | 25 |
| | | | 50 | 20.17 | 0.000 | 25 | 9.05 | 0.000 | 25 | 31.94 | 0.000 | 25 |
| | | 10 | 20 | 0.09 | 0.000 | 25 | 0.10 | 0.000 | 25 | 0.21 | 0.000 | 25 |
| | | | 35 | 1.00 | 0.000 | 25 | 1.58 | 0.000 | 25 | 4.04 | 0.000 | 25 |
| | | | 50 | 7.90 | 0.000 | 25 | 8.81 | 0.000 | 25 | 19.44 | 0.000 | 25 |
| | [5,10] | 4 | 20 | 0.13 | 0.000 | 25 | 0.12 | 0.000 | 25 | 0.30 | 0.000 | 25 |
| | | | 35 | 25.24 | 0.000 | 25 | 6.36 | 0.769 | 23 | 19.64 | 0.000 | 25 |
| | | | 50 | 3582.27 | 6.907 | 1 | 882.74 | 0.464 | 23 | 2588.04 | 1.070 | 13 |
| | | 7 | 20 | 0.12 | 0.000 | 25 | 0.15 | 0.000 | 25 | 0.33 | 0.000 | 25 |
| | | | 35 | 5.63 | 0.000 | 25 | 5.70 | 0.353 | 24 | 12.11 | 0.000 | 25 |
| | | | 50 | 1717.33 | 1.248 | 17 | 238.59 | 0.000 | 25 | 640.11 | 0.000 | 25 |
| | | 10 | 20 | 0.10 | 0.000 | 25 | 0.12 | 0.000 | 25 | 0.18 | 0.000 | 25 |
| | | | 35 | 1.37 | 0.000 | 25 | 3.47 | 0.000 | 25 | 7.34 | 0.000 | 25 |
| | | | 50 | 311.87 | 0.000 | 24 | 158.91 | 0.000 | 25 | 160.92 | 0.000 | 25 |
| [100,200] | [1,5] | 4 | 20 | 11.68 | 0.000 | 25 | 5.67 | 0.000 | 25 | 2.39 | 0.000 | 25 |
| | | | 35 | 1792.02 | 0.309 | 18 | 879.48 | 0.498 | 21 | 2139.66 | 0.287 | 15 |
| | | | 50 | 3600.00 | 4.744 | 0 | 3600.00 | 1.763 | 0 | 3600.00 | 1.540 | 0 |
| | | 7 | 20 | 15.73 | 0.000 | 25 | 6.62 | 0.000 | 24 | 1.75 | 0.000 | 25 |
| | | | 35 | 3479.91 | 2.168 | 3 | 1878.28 | 1.015 | 17 | 235.71 | 0.264 | 23 |
| | | | 50 | 3600.00 | 6.913 | 0 | 3600.00 | 4.763 | 0 | 3600.00 | 4.449 | 0 |
| | | 10 | 20 | 27.27 | 0.000 | 25 | 1.97 | 0.000 | 25 | 1.04 | 0.000 | 25 |
| | | | 35 | 3600.00 | 9.549 | 0 | 3600.00 | 7.981 | 0 | 2155.21 | 2.550 | 14 |
| | | | 50 | 3600.00 | 4.291 | 0 | 3600.00 | 2.453 | 0 | 2605.62 | 0.358 | 11 |
| | [5,10] | 4 | 20 | 221.35 | 0.130 | 24 | 193.31 | 0.083 | 24 | 195.52 | 0.097 | 24 |
| | | | 35 | 3600.00 | 13.646 | 0 | 3600.00 | 9.355 | 0 | 3600.00 | 5.694 | 0 |
| | | | 50 | 3600.00 | 28.222 | 0 | 3600.00 | 21.375 | 0 | 3600.00 | 17.745 | 0 |
| | | 7 | 20 | 426.29 | 0.000 | 25 | 319.74 | 0.000 | 24 | 32.42 | 0.000 | 25 |
| | | | 35 | 3600.00 | 8.629 | 0 | 3600.00 | 5.850 | 0 | 3600.00 | 3.053 | 0 |
| | | | 50 | 3600.00 | 20.224 | 0 | 3600.00 | 15.872 | 0 | 3600.00 | 11.994 | 0 |
| | | 10 | 20 | 180.76 | 0.000 | 25 | 12.61 | 0.000 | 25 | 1.87 | 0.000 | 25 |
| | | | 35 | 3600.00 | 14.668 | 0 | 3600.00 | 12.899 | 0 | 3600.00 | 10.912 | 0 |
| | | | 50 | 3600.00 | 12.455 | 0 | 3600.00 | 9.445 | 0 | 3600.00 | 6.648 | 0 |
| Total | | | | | | 587 | | | 630 | | 650 | |
| Average | | | | 1329.19 | 3.725 | | 1128.33 | 2.637 | | 1103.54 | 1.852 | |



Table 2: Numerical experiments aiming at a target solution

| P_{range} | d_{range} (%) | m | n | (P_3) | | | (P_3^+) | | | (P_3^{++}) | | | | |
|-------------|--------------------|--------|-----|-----------------|--------------|-----------|-----------------|--------------|-----------|-----------------|--------------|-----------|----|----|
| | | | | TT_g (sec) | T_g (#) | BK (#) | TT_g (sec) | T_g (#) | BK (#) | TT_g (sec) | T_g (#) | BK (#) | | |
| [1,100] | 4 | 20 | 20 | 0.067 | 25 | 3 | 0.040 | 25 | 3 | 0.122 | 25 | 3 | | |
| | | | 35 | 0.095 | 25 | 1 | 0.244 | 25 | 1 | 1.182 | 25 | 2 | | |
| | | | 50 | 0.913 | 25 | 10 | 0.840 | 25 | 10 | 12.126 | 25 | 10 | | |
| | [1,5] | 7 | 20 | 20 | 8.343 | 25 | 6 | 0.072 | 25 | 6 | 0.216 | 25 | 6 | |
| | | | | 35 | 0.065 | 25 | 14 | 0.421 | 25 | 14 | 1.785 | 25 | 14 | |
| | | | | 50 | 0.076 | 25 | 18 | 1.464 | 25 | 18 | 7.854 | 25 | 18 | |
| | [1,100] | 10 | 20 | 20 | 3.572 | 25 | 4 | 0.075 | 25 | 4 | 0.156 | 25 | 4 | |
| | | | | 35 | 27.210 | 25 | 16 | 0.695 | 25 | 16 | 1.907 | 25 | 17 | |
| | | | | 50 | 0.048 | 25 | 21 | 2.085 | 25 | 21 | 5.902 | 25 | 21 | |
| | [5,10] | 4 | 20 | 20 | 0.046 | 25 | 1 | 0.067 | 25 | 1 | 0.152 | 25 | 1 | |
| | | | | 35 | 0.355 | 25 | 4 | 89.774 | 25 | 4 | 2.674 | 25 | 4 | |
| | | | | 50 | 3.508 | 24 | 11 | 24.116 | 25 | 12 | 183.372 | 25 | 12 | |
| | | [5,10] | 7 | 20 | 20 | 0.197 | 25 | 5 | 0.091 | 25 | 5 | 0.233 | 25 | 5 |
| | | | | | 35 | 0.918 | 25 | 12 | 95.978 | 25 | 12 | 2.410 | 25 | 12 |
| | | | | | 50 | 496.951 | 25 | 19 | 4.500 | 25 | 19 | 16.783 | 25 | 19 |
| | | [5,10] | 10 | 20 | 20 | 1219.043 | 25 | 7 | 0.076 | 25 | 7 | 0.109 | 25 | 7 |
| | | | | | 35 | 0.388 | 25 | 17 | 0.731 | 25 | 17 | 1.766 | 25 | 17 |
| | | | | | 50 | 0.752 | 25 | 19 | 4.461 | 25 | 19 | 9.891 | 25 | 19 |
| (100,200) | | 4 | 20 | 20 | 348.696 | 25 | 1 | 2.356 | 25 | 1 | 1.346 | 25 | 1 | |
| | | | | 35 | 1270.455 | 24 | 9 | 171.997 | 25 | 9 | 635.609 | 25 | 7 | |
| | | | | 50 | 0.365 | 16 | 6 | 1201.966 | 18 | 6 | 1914.423 | 12 | 6 | |
| | [1,5] | 7 | 20 | 20 | 0.709 | 25 | 7 | 2.603 | 25 | 7 | 0.903 | 25 | 7 | |
| | | | | 35 | 536.248 | 25 | 12 | 38.552 | 25 | 13 | 9.855 | 25 | 16 | |
| | | | | 50 | 680.375 | 23 | 13 | 160.127 | 23 | 13 | 792.292 | 20 | 10 | |
| | [1,5] | 10 | 20 | 20 | 1.750 | 25 | 6 | 0.252 | 25 | 6 | 0.407 | 25 | 8 | |
| | | | | 35 | 378.322 | 24 | 15 | 204.026 | 25 | 16 | 162.474 | 25 | 17 | |
| | | | | 50 | 1394.737 | 24 | 17 | 414.237 | 25 | 19 | 484.674 | 25 | 19 | |
| | [5,10] | 4 | 20 | 20 | 1062.057 | 25 | 2 | 3.498 | 25 | 2 | 6.185 | 25 | 2 | |
| | | | | 35 | 1.567 | 16 | 3 | 2185.222 | 21 | 6 | 2948.227 | 12 | 4 | |
| | | | | 50 | 146.666 | 10 | 7 | 1214.607 | 9 | 5 | — | 0 | 0 | |
| | | [5,10] | 7 | 20 | 20 | 260.337 | 25 | 6 | 16.966 | 25 | 6 | 9.237 | 25 | 6 |
| | | | | | 35 | 924.359 | 25 | 8 | 550.985 | 23 | 8 | 523.147 | 22 | 8 |
| | | | | | 50 | 1.427 | 21 | 7 | 1200.073 | 23 | 8 | 1284.066 | 2 | 1 |
| | | [5,10] | 10 | 20 | 20 | 7.717 | 25 | 7 | 0.293 | 25 | 7 | 0.628 | 25 | 7 |
| | | | | | 35 | 897.216 | 23 | 16 | 607.948 | 22 | 16 | 1031.843 | 19 | 11 |
| | | | | | 50 | 1742.980 | 20 | 11 | 1423.454 | 23 | 12 | 2664.350 | 2 | 1 |
| Total | | | | | 850 | 341 | | 862 | 349 | | 789 | 322 | | |
| Average | | | | | 317.181 | | | 267.358 | | | 353.543 | | | |



- Gupta, S. K., Kunnathur, A. S., and Dandapani, K. (1987). Optimal repayment policies for multiple loans. *Omega*, 15(4):323 – 330. ISSN 0305-0483.
- Mosheiov, G. (2012). A note: Multi-machine scheduling with general position-based deterioration to minimize total load. *International Journal of Production Economics*, 135(1):523 – 525. ISSN 0925-5273. *Advances in Optimization and Design of Supply Chains*.
- Niu, Y.-P., Wang, J., and Yin, N. (2015). Scheduling problems with effects of deterioration and truncated job-dependent learning. *Journal of Applied Mathematics and Computing*, 47(1):315–325.
- Rostami, M., Pilerood, A. E., and Mazdeh, M. M. (2015). Multi-objective parallel machine scheduling problem with job deterioration and learning effect under fuzzy environment. *Computers & Industrial Engineering*, 85:206 – 215. ISSN 0360-8352.
- Ruiz-Torres, A. J., Paletta, G., and Pérez, E. (2013). Parallel machine scheduling to minimize the makespan with sequence dependent deteriorating effects. *Computers & Operations Research*, 40:2051–2061.
- Ruiz-Torres, A. J., Ablanedo-Rosas, J. H., Alomoto, N., and Avils, D. J. (2014). Minimización de la tardanza en problemas de programación de tareas en máquinas paralelas con deterioro de los recursos. *Revista Ingeniería Industrial*, 13(1):51–75. ISSN 0717-9103.
- Ruiz-Torres, A. J., Paletta, G., and M'Hallah, R. (2017). Makespan minimisation with sequence-dependent machine deterioration and maintenance events. *International Journal of Production Research*, 55(2):462–479.
- Ruiz-Torres, A. J., Paletta, G., and Perez-Roman, E. (2015). Maximizing the percentage of on-time jobs with sequence dependent deteriorating process times. *Int. J. Operat. Res. Inf. Syst.*, 6(3): 1–18. ISSN 1947-9328.
- Santos, V. L. A. and Arroyo, J. E. C. (2015). Sequenciamento de tarefas em máquinas paralelas considerando desgastes dependentes da sequência. In *Anais do XLVII Simpósio Brasileiro de Pesquisa Operacional*, p. 2572–2583.
- Wang, X.-Y. and Wang, J.-J. (2014). Scheduling deteriorating jobs with a learning effect on unrelated parallel machines. *Applied Mathematical Modelling*, 38(21 - 22):5231 – 5238. ISSN 0307-904X.
- Yang, D.-L., Cheng, T., Yang, S.-J., and Hsu, C.-J. (2012). Unrelated parallel-machine scheduling with aging effects and multi-maintenance activities. *Computers & Operations Research*, 39(7): 1458 – 1464. ISSN 0305-0548.
- Yang, S.-J. (2011). Parallel machines scheduling with simultaneous considerations of position-dependent deterioration effects and maintenance activities. *Journal of the Chinese Institute of Industrial Engineers*, 28(4):270–280.