

## **A new model and heuristic to a job and maintenance machine scheduling problem with sequence dependent deterioration**

### **Primeiro Autor**

Filiação  
Endereço da Instituição  
e-mail

### **Segundo Autor**

Filiação  
Endereço da Instituição  
e-mail

### **RESUMO**

Máquinas que possuem alto nível de ocupação comumente apresentam problemas de desgaste que impactam o seu desempenho e fazem com que manutenções corretivas sejam necessárias. Neste trabalho tratamos um problema de escalonamento de tarefas em máquinas paralelas não correlatas com tempo de processamento dependente da deterioração e manutenção. Neste problema buscamos minimizar o *makespan* através da definição de sequências de tarefas e momentos de manutenções de um conjunto de máquinas. O tempo processamento de cada tarefa aumenta por um fator de deterioração  $q$  (dependente da tarefa e máquina) a cada tarefa executada desde a última manutenção. Apresentamos neste artigo uma versão linearizada de um modelo matemático disponível na literatura que descreve o problema e uma heurística baseada na meta-heurística ILS para resolvê-lo. Então comparamos os resultados obtidos pela heurística com os resultados da modelo matemático e de algoritmos da literatura.

**PALAVRAS CHAVE.** Scheduling. Modelagem matemática. Meta-heurística.

**Tópicos** (Escalonamento de tarefas, Modelagem matemática, Linearização, Meta-heurística, ILS )

### **ABSTRACT**

Machines that have a high occupation level commonly presents deterioration problems which impacts on its performance, making needed corrective maintenances. In this paper we deal with a unrelated parallel machine job scheduling with time processing depending of deterioration and maintenance. In this problem, we aim to minimize the makespan through the definition of job sequences and maintenance events to a set of machines. The processing time of each job rises by a deterioration factor  $q$  (depending on machine and job) after each job executed after the last maintenance. We present here a linear version of a mathematical model available on the literature to describe the problem as well a ILS based heuristic to solve it. So we compare the results obtained by the proposed heuristic with the results got through the model and literature algorithms.

**KEYWORDS.** Scheduling. Mathematical modeling. Meta-heuristic.

**Paper topics** (Job scheduling, Mathematical model, Linearization, Meta-heuristic, ILS )

## 1. Introduction

In any continuous activity performed by a machine or person fatigue and deterioration factors affect the execution performance and increase the processing time. In some cases the loss of performance happens so frequently that it is necessary to include stop and maintenance events on the scheduling in order to recover the full performance of the executor and provide a overall best functioning of the system.

In this paper we deal with the machine job scheduling problem on environments where deterioration is a relevant factor in a short time horizon, like medical services, student examination, computer processing. In those scenarios, we have a significant increase of time processing after the execution of each job due deterioration and a short maintenance procedure that could fix the accumulated deterioration, where maintenance could be a repair, stop, cooling, warming, etc...

Looking for the optimization of some metric, in this case the makespan, we define which jobs will be processed in each machine, in which order and also when to execute a maintenance procedures.

Our main contribution in this paper is a linear and improved version of the model proposed by [Ruiz-Torres et al., 2017] to describe the model that makes possible the use of any of the main MILP solvers to solve it. We also provide a heuristic procedure that allows to solve the version of problem that considers unrelated machines (with different deterioration rates and different processing and maintenance times).

In the next session we present a short view of the previous works that deal with scheduling with deterioration problem. In the section 3 we describe the problem and present the model created by [Ruiz-Torres et al., 2017] and our linear version. In the section 4 we present the proposed heuristic and a adaptation of the heuristic proposed in .... to solve a identical machine version of the problem. In the section 5 we describe the computational experiments and finally on section 6 we present the results of the experiments performed.

## 2. Literature review

Scheduling with deterioration problems have been investigated since the late 80's, being first study ...

## 3. Problem description

The job and maintenance machine scheduling problem with sequence dependent deterioration is basically a problem where a set of jobs should be scheduled on a set of machines in order to optimize some scheduling metric. The processing time of each job on each machine depends of the machine deterioration, that increases after a job be executed and is reset after a maintenance, that takes a significant amount of time to be done. Then, we can describe the problem as follows.

Let  $J = 1, 2, 3, \dots, |J|$  be a set of independent and non preemptive jobs to be processed on a set  $M = 1, 2, \dots, |M|$  of unrelated parallel machines that suffer a significant deterioration after each job processed.

If  $p_{jm} \geq 0$  is the baseline processing time of the job  $j \in J$  on machine  $m \in M$ ,  $\delta_{jm} \geq 1$  is the accumulated delay factor caused by deterioration at machine  $m \in M$  just before the start of job  $j \in J$  and  $d_{jm} \geq 1$  the additional delay factor caused by deterioration on the machine  $m \in M$  after the job  $j \in J$ , we can state that:

- The actual processing time of machine  $j \in J$  on machine  $m \in M$  is equals to  $p_{jm} * \delta_{jm}$
- The accumulated delay factor caused by deterioration on machine  $m \in M$  after the processing in it of job  $j \in J$  is equals to  $\delta_{jm} * d_{jm}$

Considering yet that all jobs are available on time zero, the objective is create a job processing sequence on each machine in order to optimize a scheduling metric, in this article, the minimization of makespan.

In these version of the problem the number of maintenance is unbounded and can be done in any time between the end of a job processing and the beginning of other job processing. Empty intervals on machines are not forbidden but does not make sense. A maintenance in one machine does not impact a maintenance in any other machine and all the machines can be repaired at same time without loss of overall performance.

From this point, we will use the notation exposed on this section to all the rest of the paper.

#### 4. Mathematical model

In this section we present the mathematical model proposed by ... and the modifications done in it to create a linearized version. We also include some constraints to avoid some scenarios not previously predicted in the original model.

The proposed model works based on machine slots that could be filled with a job or a maintenance. Those slots are used to define the position of an activity on a machine schedule and have no fixed duration as it could have in a time indexed formulation.

In this way, we can define if a job  $j \in J$  will be executed in the slot  $h \in H$  of machine  $m \in M$  using the binary variable  $x_{jmh}$  and similarly if a maintenance will be executed at slot  $h \in H$  of the machine  $m \in M$  with binary variable  $s_{mh}$ .

According to the activity performed in a machine slot we can define the resulting performance in the next machine slot. This is stored at the variable  $q_{mh}$  that saves the current performance in the slot  $h \in H$  of the machine  $m \in M$ , being its value between 1 to represent a 100% performance (no deterioration) and 0 (total deterioration).

With these definitions we can state the model bellow.

$$\min C_{max} \quad (1)$$

Subject to

$$\sum_{j \in J} x_{jmh} + s_{mh} \leq 1 \quad \forall m \in M, h \in H \quad (2)$$

$$\sum_{m \in M} \sum_{h \in H} x_{jmh} = 1 \quad \forall j \in J \quad (3)$$

$$\sum_{j \in J} \sum_{h \in H} \frac{p_{j,m}}{q_{kh}} * x_{jmh} + \sum_{h \in H} t_h * s_{mh} - C_{max} \leq 0 \quad \forall m \in M \quad (4)$$

$$q_{mh} - \sum_{j \in J} 1 - d_{jm} * q_{m(h-1)} * x_{jm(h-1)} - s_{m(h-1)} = 0 \quad \forall h \in H \setminus \{1\}, m \in M \quad (5)$$

$$q_{m1} = 1 \quad \forall m \in M \quad (6)$$

$$x_{jmh} \in \{0, 1\} \quad \forall j \in J, m \in M, h \in H \quad (7)$$

$$s_{mh} \in \{0, 1\} \quad \forall m \in M, h \in H \quad (8)$$

$$q_{mh} \geq 0 \quad \forall m \in M, h \in H \quad (9)$$

In the Equation (1) we have the objective function, that is just the value of the variable  $C_{max}$  that represents the makespan. The Equation (2) states that just one activity can be done by slot. Equation (3) assigns an unique job to a machine slot. On Equation (4) computes the maskspan based on the finish time on each machine and (5) computes the deterioration on each machine slot. Equation (6) initializes the  $q$  value in each machine and finally Equations (7) to (9) defines the variables intervals.

As can be seen, the constraints represented in the Equations (4) and (5) are not linear. We could make them linear easily if there was multiplication instead of a division in the Equation (4). However, due the way used to define the machine performance (and consequently the deterioration) there is not a simple way to do this.

The new formulation proposed uses the variable  $q_{mh}$  to represent the delay factor (not the performance) of the machine  $m \in M$  on slot  $h \in H$  after a sequence of activities (jobs and maintenances). Using this approach, the variable lower bound is equals to 1 (representing a non delay on machine) and the upper bound is the product of all deteriorations in the machine.

With this variable meaning change the actual processing time of the job  $j \in J$  on machine  $m \in M$  in the period  $h \in H$  (that we will represent with the variable  $a_{jmh}$ ) can be calculated as  $p_{jm} * q_{mh}$ .

Another modification done is in the  $d_{jm}$  definition. Now it will represent the delay caused due the deterioration. It can be calculated using the inverse of the value used on the original model without any significant loss of information.

The resulting model is the following:

$$\min C_{max} \quad (10)$$

Subject to

$$\sum_{j \in J} x_{jmh} + s_{mh} \leq 1 \quad \forall m \in M, h \in H \quad (11)$$

$$\sum_{m \in M} \sum_{h \in H} x_{jmh} = 1 \quad \forall j \in J \quad (12)$$

$$a_{jmh} \leq M_1 * x_{jmh} \quad \forall j \in J, m \in M, h \in H \quad (13)$$

$$a_{jmh} \geq p_{jm} * q_{mh} - M_1 * (1 - x_{jmh}) \quad \forall j \in J, m \in M, h \in H \quad (14)$$

$$\sum_{j \in J} \sum_{h \in H} a_{jmh} + \sum_{h \in H} t_h * s_{mh} - C_{max} \leq 0 \quad \forall m \in M \quad (15)$$

$$q_{mh} \geq d_{jm} * q_{m(h-1)} - M_2 * [s_{m(h-1)} + (1 - x_{jm(h-1)})] \quad j \in J, m \in M, h \in H \setminus \{1\} \quad (16)$$

$$q_{m1} = 1 \quad \forall m \in M \quad (17)$$

$$x_{jmh} \in \{0, 1\} \quad \forall j \in J, m \in M, h \in H \quad (18)$$

$$s_{mh} \in \{0, 1\} \quad \forall m \in M, h \in H \quad (19)$$

$$a_{jmh} \geq 0 \quad \forall j \in J, m \in M, h \in H \quad (20)$$

$$q_{mh} \geq 1 \quad \forall m \in M, h \in H \quad (21)$$

In this model the Equations (10) to (12) are the same of original model. Equations (13) and (14) are the constraints used to define the actual processing time of the job  $j \in J$  on slot  $h \in H$  of machine  $m \in M$  and uses the constant  $M_1 \geq \sum_{j \in J} p_{jm}$  to free the value of variable  $a_{jmh}$  according to the value of  $x_{jmh}$ . Equation (15) is the linear version of Equation (4) used to calculate the makespan and the Equation (16), with the constant  $M_2 \geq \prod_{j \in J} d_{jm}$ , is the linear version of Equation (5) constraint used to evaluate the delay in the processing time. Equation (21) is correspondent to the Equation (9) and Equations (18) to (21) are constraints to define the bound of the variables.

Beside those constraints we define some others to better describe some situations in the problem, mainly yielding a consistent schedule of the activities. They are presented bellow:

$$s_{k(h-1)} + s_{kh} \leq 1 \quad \forall h \in H \setminus \{1\} \quad (22)$$

$$s_{k(h-1)} + \sum_{j \in J} x_{jm(h-1)} \geq s_{kh} + \sum_{j \in J} x_{jmh} \quad \forall h \in H \setminus \{1\}, m \in M \quad (23)$$

$$x_{jkh} - \sum_{l \in N} x_{lm}(h-1) - s_{m(h-1)} \leq 0 \quad \forall j \in J, m \in M, h \in H \setminus \{0\} \quad (24)$$

The Equation (22) states that is not permitted to proceed two maintenances in two consecutive slots. Actually nothing avoids it, but in a optimal solution it does not make senses, because the second maintenance has no effect on deterioration and spends more time (unless the maintenance time is equals to zero, a non-realistic scenario).

The Equations (23) and (24) are complementary and states that empty slots in the middle of a scheduling are not allowed. In other words, if the previous slot has not activity, it is because the machine use stopped there and its not possible to restart it. As the machine slots have no minimum time duration, it avoids strange solutions with several non used slot between two used slots.

Then, we have in the Equations (10) to (24) the complete formulation of the model proposed by us and used in the computational experiments of the paper.

#### 4.1. Considerations about the slot based formulation

One of the main concerns about the slot formulation to describe the job scheduling with sequence dependent deterioration is to define the number of slots available. Once any job (except the last) in a machine can be followed by a maintenance, a pessimistic value to the number of slots could be  $2 * |J|$ . However, it is obviously a excessive value, that could be lowered.

The option used by us in this formulation is to check if, to all machines, the sum of the shortest jobs processing times one machine (without considering deterioration) is bigger than the processing time of a single job in any of the other machines. If it is true, we can be sure that the optimal solution has at most  $(|J| - |M|)$  on a machine and at most  $2(|J| - |M|)$  slots used. As the number of jobs is often much greater than the number of machines, this reduction is not so big, but allows us to get a significant reduction on number of variables and a observable improvement on solving performance.

In other hand, this formulation allow us to decide if the next activity will be a maintenance or not just by looking the current slot state, as in a Markov chain model. It make easy to model some problem characteristics, specially the accumulated deterioration.

## 5. Heuristic algorithms

### Referências

Ruiz-Torres, A. J., Paletta, G., e M'Hallah, R. (2017). Makespan minimisation with sequence-dependent machine deterioration and maintenance events. *International Journal of Production Research*, 55(2):462–479.