

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 levels commonly present deterioration problems which impacts on their 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 an 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

Fatigue and deterioration commonly affect the performance of persons or machine executing an activity, causing the increase of processing time, errors or resource waste. In some cases it happens frequently, making 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 and computer processing. In those scenarios, we have a significant increase of processing time after the execution of each job due deterioration and a short maintenance procedure (as a repair, stop, cooling, warming, etc...). In this way, we aim to optimize a scheduling metric (makespan) by choosing which jobs will be processed in each machine, in the process 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 problem for making possible it be solved in MILP solver. 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 next, on section 4, the model proposed by [Ruiz-Torres et al., 2017] and our linear version. In the section 5 we present the proposed heuristic and a adaptation of the heuristic proposed in [Ruiz-Torres et al., 2017] to solve a identical machine version of the problem. In the section 6 we describe the computational experiments and finally on section 7 we present the results of the experiments performed.

2. Literature review

Scheduling with deterioration problems have been investigated since the late 80's, with the first known study presented in [Gupta e Gupta, 1988], where the processing time depends on the job starting time. This article was followed in next years by [Browne e Yechiali, 1990], [Mosheiov, 1991] and [Mosheiov, 1994] that used similar ways to evaluate the deterioration.

The published studies can be divided in two groups according to how deterioration is estimated. The first and older group estimate the deterioration based on the job starting time, supposing that a machine is deteriorated with the same rate by any job, without considering the nature of job. The second group estimate the deterioration based on the set of jobs processed previously, considering the eventual different deterioration rates due to the hardness of some jobs.

Among the works in which the deterioration is dependent on starting time we can highlight yet [Mosheiov, 1998] that proved that the makespan minimization with parallel machines problem is strongly NP-hard even for a two machine case and [Ji e Cheng, 2008] that created a polynomial-time approximation scheme for the case with a fixed number of machines. In [Cheng e Ding, 2001], [Leung et al., 2008] and [Lalla-Ruiz e Voß, 2016] the concept of step-deterioration is used to describe situations where the processing time just change if the job is processed after a threshold time.

The first paper with a sequence-dependent deterioration was [Ruiz-Torres et al., 2013]. In this paper is presented a non-linear model to describe the problem. This same problem is the subject in [Santos e Arroyo, 2015] paper, where is proposed an ILS and a combined ILS with RVND algorithm to solve it.

Maintenance activities are introduced on [Yang, 2011] and [Yang et al., 2012], in a parallel machine scheduling with position-dependent deterioration, where the time is not important to define the deterioration but the jobs are yet considered equivalent on deterioration potential. We can cite yet [Huang e Wang, 2015] that still uses the position-dependent deterioration concept, but without maintenance activities.

The first work where there is a sequence-dependent deterioration and maintenance activities is [Ruiz-Torres et al., 2017]. In this paper, the processing time of a job depend on the sequence of activities performed before it. After this, in the best of our knowledge only [Ding et al., 2019] worked in the same subject, changing slightly the problem definition to make the job deterioration influences its own processing time.

This paper is based on the problem defined by [Ruiz-Torres et al., 2017] and in the next sections we will describe it, present a new mathematical formulation to replace the existent non-linear formulation and propose an ILS based heuristic to solve it.

3. Problem description

The job and maintenance machine scheduling problem with sequence dependent deterioration looks to define where a set of jobs should be scheduled on a set of machines and when to proceed manteinances in those machine 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. A maintenance takes a significant amount of time to be done. Then, we can describe the problem as a 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 [Ruiz-Torres et al., 2017] 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 makespan 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

In this section we present the heuristic we propose to solve the job and maintenance machine scheduling problem with sequence-dependent deterioration. It is a ILS based algorithm with five local searches and two perturbations, that are used in an alternate way.

```

input : J, M, p, d, t
output: s*
1 nonImprovellers  $\leftarrow$  0 ;
2 startTimer (temp)
3 s*  $\leftarrow$  constructiveHeuristic (J, M, p, d, t);
4 s  $\leftarrow$  s*
5 while nonImprovellers  $\leq$  MAX_IT and temp < MAX_TIME do
6   nonImprovellers  $\leftarrow$  nonImprovellers + 1 ;
7   s  $\leftarrow$  internal2SwapLS (s, J, M, p, d, t);
8   s*  $\leftarrow$  updateBestGlobal (s, s*, nonImprovellers);
9   s  $\leftarrow$  internal3SwapLS (s, J, M, p, d, t);
10  s*  $\leftarrow$  updateBestGlobal (s, s*, nonImprovellers);
11  s  $\leftarrow$  external2SwapLS (s, J, M, p, d, t);
12  s*  $\leftarrow$  updateBestGlobal (s, s*, nonImprovellers);
13  s  $\leftarrow$  external3SwapLS (s, J, M, p, d, t);
14  s*  $\leftarrow$  updateBestGlobal (s, s*, nonImprovellers);
15  if nonImprovellers mod 5 = 0 then
16    | s  $\leftarrow$  doPerturbation (s)
17  end
18 end

```

Algorithm 1: ILS heuristic

In the Algorithm 1 we can see the ILS function. It receive as input a set of jobs J , a set of machines M , the maintenance time of each machine and the processing time and deterioration delay of each job on each machine.

At line 3 we the variable s^* representing the best solution gets the solution obtained in the constructive heuristic. In this paper we tried two constructive algorithms. The first, that we will reference as *longest processing time first* (LPTF), orders all jobs by non-increasing processing time order and after that, starting from the first machine, it is assigned to each machine the non used job with longest processing time on that machine. The second, referenced as *lowest delay first* (LDF) orders job by increasing deterioration delay order. After that, as in the previous method, we start from the first machine and insert one job by time on each machine, choosing the non used job with lowest delay.

Once the jobs are inserted in the machines, we put the maintenance operations. The strategy used is to evaluate the accumulated delay, starting from the last job to the first, and every time the deterioration delay becomes bigger than maintenance processing time, a maintenance is inserted in the position anterior to the job where the delay surpassed the maintenance processing time.

In the sequence of ILS method, we have from line 7 to 16 a sequence of local searches. After any local search the function *updateBestGlobal* is called to evaluate if the new solution is best than the global solution and set the *nonImproveIters* variable value to 0.

The first local search performed, *internal2SwapLS*, swaps the position, at each machine, of two jobs. Those jobs are chosen as the index of two nested *for* loops. The number of swaps tested

is limited by a big value, that eventually is not reached.

The second local search in the sequence(*internal3SwapLS*) changes the position of three jobs in two different ways. Starting from a tuple of three indexes (i_1, i_2, i_3) , taken from three nested for loops, it tests two movements (i_3, i_1, i_2) and (i_2, i_3, i_1) . The other three possible moves are equivalent to simple two-position swaps, and are not executed. The number of moves tested is limited, as in the previous function

The two following local searches are inter-machines movements. The *external2SwapLS* procedure swap two jobs between two machines (one job of first machine changes the position with one job of second machine). The machines are chosen by the index of two nested *for* loops and the jobs are chosen in the same way.

Finally the *external3SwapLS* procedure is analogous with the *external2SwapLS*, changing three jobs of three different machines but adopting the same move strategy of the *internal3SwapLS*.

After all these local searches, if the condition showed on line 17 is satisfied, we proceed a perturbation of the current solution.

In the local search described above there are no maintenance position swaps. The maintenance positions are evaluated only on constructive heuristics and perturbation procedures due to performance reasons.

The perturbation procedure used by us calls, alternately two functions. The first one reverses the order of all jobs in one machine and reevaluates the position of the maintenance activities. The second does a shift of one position on machine scheduling, assigning to the machine i the jobs of the machine $i+1$, and the jobs of the last machine to the first. The order of the jobs in this perturbation does not change and the position of maintenance activities are recalculated.

5.1. Computational experiments

To test the performance of the heuristics in solving the problem and the fitness of the model to describe the problem we created a instance set based on the parameters described by Ruiz-Torres et al. [2017]. To test the model were created three instances to each parameter set combination (number of machines, ration machines by job, deterioration interval and maintenance time interval) summing 144 instances. The heuristics were tested in a large set with 10 instances to each parameter set combination.

The instances created have machines identical processing times to all jobs but different deterioration rates and maintenance times, in a semi-unrelated configuration. This approach was used to make easier to compare the results obtained by our heuristics and the —nome heuristica— heuristic proposed by Ruiz-Torres et al. [2017] created to solve a identical machines scenario. To make the result more directly comparable the — nome heuristica— heuristic was adapted to consider unrelated machines, as described in the previous version. All the algorithms, however, works with instances having machines different processing times.

All the experiments were done in a Linux machine with 32 GB of ram and processor Core i7. The algorithms were implemented using Julia language (version 1.07) and the model was solved by Cplex 12.6 called through the Julia optimization library JuMP. As none of the algorithms uses random walks we run each instance just one time to each algorithm.

6. Results

In this section we make a brief description of the results obtained by each one of methods used to solve the problem.

7. Conclusion

Referências

- Browne, S. e Yechiali, U. (1990). Scheduling deteriorating jobs on a single processor. *Operations Research*, 38(3):495–498.
- Cheng, T. e Ding, Q. (2001). Single machine scheduling with step-deteriorating processing times. *European Journal of Operational Research*, 134(3):623–630.
- Ding, J., Shen, L., Lü, Z., e Peng, B. (2019). Parallel machine scheduling with completion-time-based criteria and sequence-dependent deterioration. *Computers & Operations Research*, 103:35 – 45. ISSN 0305-0548.
- Gupta, J. N. e Gupta, S. K. (1988). Single facility scheduling with nonlinear processing times. *Computers & Industrial Engineering*, 14(4):387–393.
- Huang, X. e Wang, J.-J. (2015). Machine scheduling problems with a position-dependent deterioration. *Applied Mathematical Modelling*, 39(10):2897 – 2908. ISSN 0307-904X.
- Ji, M. e Cheng, T. (2008). Parallel-machine scheduling with simple linear deterioration to minimize total completion time. *European Journal of Operational Research*, 188(2):342 – 347. ISSN 0377-2217.
- Lalla-Ruiz, E. e Voß, S. (2016). Modeling the parallel machine scheduling problem with step deteriorating jobs. *European Journal of Operational Research*, 255(1):21–33.
- Leung, J. Y.-T., Ng, C., e Cheng, T. E. (2008). Minimizing sum of completion times for batch scheduling of jobs with deteriorating processing times. *European Journal of Operational Research*, 187(3):1090–1099.
- Mosheiov, G. (1991). V-shaped policies for scheduling deteriorating jobs. *Operations Research*, 39 (6):979–991.
- Mosheiov, G. (1994). Scheduling jobs under simple linear deterioration. *Computers & Operations Research*, 21(6):653–659.
- Mosheiov, G. (1998). Multi-machine scheduling with linear deterioration. *INFOR: Information Systems and Operational Research*, 36(4):205–214.
- 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.
- Ruiz-Torres, A. J., Paletta, G., e Pérez, E. (2013). Parallel machine scheduling to minimize the makespan with sequence dependent deteriorating effects. *Computers & Operations Research*, 40 (8):2051 – 2061. ISSN 0305-0548.
- Santos, V. L. A. e Arroyo, J. E. C. (2015). Sequenciamento de tarefas em máquinas paralelas considerando desgastes dependentes da sequência. In *XLVII SBPO, SÃmposio Brasileiro de Pesquisa Operacional*, p. 2572 – 2583.
- Yang, D.-L., Cheng, T., Yang, S.-J., e 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.