

**INTEGRATIVE PROGRAMMING OF PRODUCTION
AND PREVENTIVE MAINTENANCE TASKS IN AN ENVIRONMENT
OF IDENTICAL PARALLEL MACHINES: PROPOSAL
OF A CONSTRUCTION HEURISTIC**

*²Daynier Rolando DELGADO SOBRINO, ¹Ronald DÍAZ CAZAÑAS,
²Roman RUŽAROVSKÝ, ²Radovan HOLUBEK

¹CENTRAL UNIVERSITY “MARTA ABREU” DE LAS VILLAS, FACULTY
OF MECHANICAL AND INDUSTRIAL ENGINEERING,
5½ KM. CAMAJUANÍ ROAD, 54 830, SANTA CLARA, VILLA CLARA, CUBA
e-mail: ^bronalddc@uclv.edu.cu

²SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA,
FACULTY OF MATERIALS SCIENCE AND TECHNOLOGY IN TRNAVA,
INSTITUTE OF PRODUCTION TECHNOLOGIES,
ULICA JÁNA BOTTU 2781/25, 917 24 TRNAVA, SLOVAK REPUBLIC
e-mail: daynier_sobrino@stuba.sk, roman.ruzarovsky@stuba.sk, radovan.holubek@stuba.sk
(*Corresponding author)

Received: 31.05.2018, Accepted: 26.06.2018, Published: 19.09.2018

Abstract

*The article presents a constructive heuristic for the integrated programming of production orders and preventive maintenance (PM) interventions in an environment of identical parallel machines, aimed at minimizing the completion time of the last scheduled job (makespan). Unlike other proposals found in the literature, the one here considers that the time between preventive interventions is not previously known, but instead, this is assumed as one of the decision variables of the system in the beginning of each of these interventions; in addition, the proposal also considers the possibility that not all jobs have the same release time. The design of the heuristic algorithm is based on the general approach of the Longest Processing Time dispatch rule, with a slight modification in its conception to consider the random character of the equipment failures and their impact on the execution time of the tasks. Consequently, a simple dispatch rule called *rj - LPT* was also proposed to comply with one of the steps of the heuristic, as well as an algorithm to determine the minimum dimension for the makespan, which constitutes the basis of the stop criterion linked to the initial solution.*

Key words

Production programming, maintenance programming, construction heuristic, algorithm, Longest Processing Time rule (LPT), pseudo-code

INTRODUCTION

Over the years, it has been proved that the process of making decisions in Maintenance should not only be quick and precise, it should be also closely coordinated with the production needs and requirements (1). Similarly, the operating and maintenance costs, as well as the reliability concerning production performance requirements are also a decisive factor for success, which can be achieved through the efficient collaboration among the maintenance decision levels at the shopfloor, production scheduling and quality control (2, 3).

Despite the quite well-known need for a production-maintenance integration followed by a solid system of quality control, reality often shows the opposite, evidencing over the years a scenario characterized by the presence of conflicting interests between both areas. In general terms, authors such as Ruiz et al. (4), Hadidi et al. (5), Kumar et al. (6), Da et al. (7), Zahedi and Salim (8), Jing et al. (9) and Boudjelida (10) coincide in the fact that production and maintenance programming are very often treated independently and designed by different work teams, causing the development of separate models for each function that leads to obtain suboptimal solutions. This situation at the end exacerbates the conflict that is generated by the fact that, on the one hand, maintenance activities consume production time and, on the other, maintenance tasks are often postponed until the production sequence allows it what can significantly increase the probability of failure of the machines.

In a productive environment characterized by parallel machines two fundamental decisions must be made: firstly, the resource (machines) allocation, i.e.: these are to be assigned to the different jobs, and, subsequently, the sequencing of the jobs on each machine should be established, looking for ways to optimize some function related to the completion time of these and/or to the total cost. Pinedo (11) highlights the fact that when it comes to this type of productive configuration, the makespan becomes an objective of special interest, since in practice production programmers frequently try to achieve an adequate balance in the use of the resources. In these cases, the assignment of the machines to the corresponding jobs is considered as the most important task, besides it must also consider that the realization of the jobs is or may be influenced by possible interruptions due to failures in the production equipment. In dealing with this kind of problems and shopfloor configurations, solution strategies have often comprised constructing algorithms that have been mainly designed based on the well-known and long-used Largest Processing Time dispatch rule (LPT) and on the MULTIFIT algorithm proposed by Coffman et al. (12).

Other important contributions in parallel machines minimizing the makespan correspond to Xu et al. (13) and Lee and Wu (14). The first studied the problem that is generated when considering a relatively long programming horizon that includes several preventive interventions, and the time between these interventions is not constant, but deterministic. On the other hand, Lee and Wu (2008) considered a single maintenance period within the planning horizon having known starting and ending instants for each machine; however, in this case the processing time of the jobs is a function of the moment these start at, thus the later it begins, the longer it will last (deteriorating jobs).

In recent years, most of the solution strategies for this type of problem have been based on the local search algorithms, artificial intelligence and the use of simulation. In this regard, authors such as Mirabedini and Iranmanesh (15) applied a multi-objective dynamic genetic algorithm and an optimization scheme based on a swarm of particles to address these issues. Similarly, Da et al. (7) developed a genetic algorithm of the type NSGA-II as a solution method for an integrated model that includes uniform parallel machines, defining as objectives to minimize the cost of maintenance and makespan, considering besides, that both the equipment and the processing time of the jobs deteriorate over time. In the literature it is also possible to find other contributions developed for this type of productive configuration but aimed at the

optimization of different objectives; such is the case of those presented by Lee et al (16) and Kumar and Kumar - Lad (17).

However, despite the increasing interest in the use of these local search methods, included in the field of Artificial Intelligence, constructive heuristics continue to be a tempting alternative when it comes to implementing relatively simple methods capable of generating good solutions quickly. Besides these have also proven useful as part of the development of hybrid methods in which quite often a simple heuristic generates an initial solution for a given local search algorithm that then further explores it and improves it in the search for a “more global” optimum.

In order to contribute to the solution of the above-mentioned gaps and problematic issues of the state of the art and practice, the article presents a constructive heuristic that generates an integrated program of production orders and preventive maintenance (PM) interventions in an environment of parallel machines. The proposal has the objective of minimizing the makespan (C_{max}), considering that different jobs may have different release times, and the time between PM interventions is not fixed, but instead a decision variable within the algorithm. On the other hand, the influence of failure-related interruptions in the jobs' completion time is also taken into account with the proposal.

DEFINITION OF THE PROBLEM

The productive system consists of a set of m machines, arranged in parallel, capable of processing indifferently n types of products. Each product is processed in one and only one of the machines available, which are assumed to be identical in the sense that the processing time of each job (product) is the same in all machines. The processing time of the jobs is assumed constant and known, besides, once a job has started to be processed it will not be interrupted except for rare situations of *force majeure*, and, in these cases, the processing of the product will continue once the operating conditions return to normal (resumeable case). There is also an assumption that the time between failures of the equipment can be described by the Weibull probability density distribution. Based on these assumptions, the proposed algorithm is aimed at establishing an assignment of jobs to the different machines, also indicating the most convenient time to carry out preventive maintenance interventions on these, in order to minimize the total manufacturing time (makespan).

MATERIALS AND METHODS

Definition of variables and parameters of the heuristic algorithm

n : Number of jobs (products)

m : Number of machines

J : Set of jobs (products) to be programmed

M : Set of machines (productive resources)

r_j : Time instant where the job j is ready to start being processed

p_j : Processing time of the job j

t_p : Average duration time of the preventive maintenance interruption

t_r : Average duration time of the corrective maintenance interruption

β_k : Weibull shape parameter for the variable “time between failures” associated to the machine k

η_k : Weibull scale parameter for the variable “time between failures” associated to the machine k

j_l, j_t : Subscripts used to denote the jobs

I_k : Number of jobs assigned to the machine k

Tt_k : Ending time for the machine k.

$a_{k(lk^-)}$: Effective age of the machine k before starting the processing of the job $l_{(k-th)}$

$a_{k(lk)}$: Effective age of the machine k before ending the processing of the job $l_{(k-th)}$

TCE_{lk} : Expected completion time of the job l in the machine k

TE_{lkPM1} : Processing time of the job l in the machine k in the case this is subjected to a PM intervention before processing the job $l_{(k-th)}$

TE_{lkPM0} : Processing time of the job l in the machine k in the case this is NOT subjected to a PM intervention before processing the job $l_{(k-th)}$

\overline{TCE}_{lkPM1} : Mean expected completion time of the first I_k Jobs processed in the machine k in the case this is subjected to a PM intervention before initiating the processing the job $l_{(k-th)}$

\overline{TCE}_{lkPM0} : Mean expected completion time of the first I_k Jobs processed in the machine k in the case this is NOT subjected to a PM intervention before initiating the processing the job $l_{(k-th)}$

PM_{lk} : Binary variable indicating if the PM intervention is to take place on the machine k before initiating the job l

$A(k), B(k)$: Arrays representing the job subsets assigned to the machine k, and the PM interventions that should take place before initiating each job

$\#(\alpha)$: Cardinality of the subset α

C_{\max} : Makespan value associated to the assignment of jobs and maintenance interventions assumed as the solution

\hat{C}_{\min} : Minimum level for the value of the makespan obtained through the application of some assignment method

ρ : Established limit for the solution error obtained by the first phase of the algorithm (steps 1 to 6)

The steps of the algorithm

1. Initialize: $Tt_k = I_k = 0 \forall k \in M; l = 0; A(1) = A(2) = \dots = A(m) = \emptyset; r_j \forall j = \{1, 2, \dots, n\}$

2. Generate the set of jobs J ordered in a non-increasing way according to p_j

3. Make: $t = 1; S = \emptyset; P = \emptyset; l = l + 1$

If $r_l = \min(r_j) \forall j = \{l, l + 1, \dots, n\}$ assign the job l to the machine k | $TCE_{lk} = \min\{TCE_{lk}\} \forall k \in M$. In case of a match, select the machine k arbitrarily

$I_k = I_k + 1$

$$TCE_{lk} = \begin{cases} \max(Tt_k; r_l) + TE_{lkPM1} & \text{if } \overline{TCE}_{lkPM1} < \overline{TCE}_{lkPM0} \\ \max(Tt_k; r_l) + TE_{lkPM0} & \text{otherwise} \end{cases} \quad [1]$$

$$PM_{lk} = \begin{cases} 1 & \text{si } \overline{TCE}_{lkPM1} < \overline{TCE}_{lkPM0} \\ 0 & \text{otherwise} \end{cases}$$

$$TE_{lkPM1} = p_l + t_p + t_r \left[\left(\frac{p_l}{\eta_k} \right)^{\beta_k} \right] \quad [2]$$

$$TE_{lkPM0} = p_l + t_r \left[\left(\frac{a_{k(lk^-)} + p_l}{\eta_k} \right)^{\beta_k} - \left(\frac{a_{k(lk^-)}}{\eta_k} \right)^{\beta_k} \right] \quad [3]$$

$$a_{k(lk^-)} = \begin{cases} 0 & \text{if } PM_{lk} = 1 \\ a_{k(lk^- - 1)} + p_{lk^- - 1} & \text{if } PM_{lk} = 0 \end{cases} \quad [4]$$

$$\overline{TCE}_{lkPM1} = \frac{\sum_{j=1}^{I_k} p_j + \sum_{j=1}^{I_k - 1} PM_{jk} * t_p + \sum_{j=1}^{I_k - 1} t_r \left[\left(\frac{a_{k(j)}}{\eta_k} \right)^{\beta_k} - \left(\frac{a_{k(j^-)}}{\eta_k} \right)^{\beta_k} \right] + t_p + t_r \left[\left(\frac{p_l}{\eta_k} \right)^{\beta_k} \right]}{I_k} \quad [5]$$

$$\overline{TCE}_{IkPM0} = \frac{\sum_{j=1}^{I_k} p_j + \sum_{j=1}^{I_k-1} PM_{jk} * tp + \sum_{j=1}^{I_k} t_r \left[\left(\frac{a_{k(j)}}{\eta_k} \right)^{\beta_k} - \left(\frac{a_{k(j-1)}}{\eta_k} \right)^{\beta_k} \right]}{I_k} \quad [6]$$

$Tt_k = TCE_{Ik}$; $A(k) = A(k) \cup \{j_l\}$.

If $\sum_{k=1}^m I_k < n$ repeat step 3 otherwise then obtain $C_{max} = \max \{Tt_k\} \forall k \in M$

If $r_l \neq \min(r_j) \forall j = \{1, 2, \dots, n\}$ move to step 4

4. Generate the subset of jobs P ordered in a non-decreasing according to r_j ; , $P \subset J \setminus \{j_l; A(k) \forall k \in M\} | r_l > r_p \forall p \in P$; Make $I_k = I_k + 1 \forall k \in M$ and calculate the expected completion time of the job t for each machine k

$$TCE_{tk} = \begin{cases} \max(Tt_k; r_t) + TE_{tkPM1} & \text{if } \overline{TCE}_{IkPM1} < \overline{TCE}_{IkPM0} \\ \max(Tt_{ki}; r_t) + TE_{tkPM0} & \text{otherwise} \end{cases}$$

Identify the machine $k | TCE_{tk} = \min \{TCE_{tk}\}$. In case of a match, select arbitrarily

If $\min \{TCE_{tk}\} \leq r_l$ $Tt_k = TCE_{tk}$; $S = S \cup \{j_t\}$; $I_k = I_k - 1 \forall k \setminus k: Tt_k = TCE_{tk}$; otherwise $I_k = I_k - 1 \forall k = \{1, 2, \dots, m\}$

Make $t = t + 1$. If $t \leq \#(P)$ repeat the step 4, otherwise move to step 5

5. If $S \neq \emptyset$ assign the jobs from the subset S using the combined rule $r_j - LPT$ and then update the arrays $A(k)$. Update the values of Tt_k for each machine $k | \exists j_t \in A(k)$

$A(k) = A(k) \cup \{j\} | j \in A(k)$ and update the values of $I_k \forall k \in M$

$Tt_k = TCE_{tk} \forall k | j_t \in A(k)$

$$TCE_{tk} = \begin{cases} Tt_k + TCE_{tkPM1} & \text{if } \overline{TCE}_{IkPM1} < \overline{TCE}_{IkPM0} \\ Tt_k + TCE_{tkPM0} & \text{otherwise} \end{cases}$$

$$PM_{tk} = \begin{cases} 1 & \text{if } \overline{TCE}_{IkPM1} < \overline{TCE}_{IkPM0} \\ 0 & \text{otherwise} \end{cases}$$

If $S = \emptyset$ move to step 6

6. Assign the job l to the machine $k | TCE_{lk} = \min \{TCE_{lk}\} \forall k = \{1, 2, \dots, m\}$. In case of a match select the machine k arbitrarily; $A(k) = A(k) \cup \{j_l\}$. Update Tt_k and the values $I_k \forall k \in M$

$Tt_k = TCE_{lk}$

If $\sum_{k=1}^m I_k < n$ move to step 3, otherwise then obtain $C_{max} = \max \{Tt_k\} \forall k \in M$

If $C_{max} \leq \rho * \hat{C}_{min}$ the algorithm ends establishing for each machine k the assignment resulting from the arrays $A(k)$ y PM_{lk} , and the makespan is equal to C_{max}

Otherwise, then move to step 7

If $C_{max} > \rho * \hat{C}_{min}$ re-establish the set of jobs J ordered in not-increasing way according to p_j and readjust: $Tt_k = I_k = 0 \forall k = \{1, 2, \dots, m\}$; $B(1) = B(2) = \dots = B(m) = \emptyset$; $l = 1$; $k = 1$; $LS = C_{max}$ then move to step 8.

7. Calculate the expected completion time of the job l for each machine k (TCE_{lk}) using equation 1.

If $TCE_{lk} < LS$ $Tt_k = TCE_{lk}$; $C = C \cup \{j_l\}$; $l = l + 1$ and repeat the step 8

If $TCE_{lk} > LS$ Make $l = l + 1$

If $l \leq n$ repeat the step 8; otherwise move to step 9

8. If $C \neq \emptyset$ Assign the jobs of C to k using the rule $r_j - LPT$ and obtain $B(k)$; update Tt_k and readjust $J = J \setminus C$

9. Increase $k = k + 1$ and adjust $l = 1$. If $k \leq m$ repeat the step 8, otherwise move to step 11

10. If $J = \emptyset$ set $C_{max} = \max \{Tt_{ki}\} \forall k \in M$; replace $A(k)$ by $B(k) \forall k = \{1, 2, \dots, m\}$ and move to the step 7; otherwise the algorithm ends establishing for each machine k the assignment resulting from the arrays $A(k)$ y PM_{lk} , being the makespan equal to LS

Definition of the pseudo-code of the proposed combined rule $r_j - LPT$

1. Set $t_0 = \max(\min(r_j), \min(Tt_k)) \forall j \in S, k \in M, JA = \emptyset, MA = \emptyset$
2. Set $JA \subset S \mid r_j \leq t_0 \forall j \in S$.
3. Set $MA \subset M \mid Tt_k \leq t_0 \forall k \in M$
4. If $\#(JA) \leq \#(MA)$ assign each job of JA to some machine k of MA on the time instant t_0 , otherwise assign these jobs according to the rule LPT.
Update $A(k) \forall k \in MA$ y $Tt_k = TCE_{jk} \mid j \in A(k) \forall k \in MA$
5. Set $S = S \setminus JA$. If $S \neq \emptyset$ repeat the step 1 otherwise stop

Steps for the determination of \hat{C}_{min}

1. Obtain the incremented processing time p'_j for each job j

$$p'_j = p_j + \frac{tp * \overline{Np} + tr * Nf_{min}}{\sum_{j=1}^n (\frac{p_j}{\eta_{min}})^{\beta_{max}}}$$

2. Obtain the number of jobs \overline{Np} processed up to the beginning of the PM intervention
Generate the equation of the total time dedicated to maintenance (TTM) according to N_p

$$TTM = t_p \frac{n}{N_p} + t_r \left[\frac{n}{N_p} * \left(\frac{\sum_{j=1}^{N_p} p_j}{\eta_{min}} \right)^{\beta_{max}} \right]$$

- 2.1 Obtain the value N_p^* in a way that $TTM_{(N_p^*-1)} > TTM_{(N_p^*)} < TTM_{(N_p^*+1)}$ and then make $\overline{Np} = N_p^*$
3. Apply some known heuristic for minimizing the makespan in the context of parallel machines, e.g.: LPT, and this without considering the influence of maintenance and using the new processing times $p'_j, \forall j = \{1, 2, \dots, n\}$
 $\hat{C}_{min} = \max(Tt_k) \forall k \in M$

The first six steps of the proposed algorithm are aimed at searching for an initial solution under the principles of the LPT rule, but with a slight modification that is implemented in step 3. The modification considers that the job with the longer processing time available at a given time will be assigned to the machine that achieves for it a shorter expected completion time, instead of assigning it directly to the first available machine, as indicated by the original LPT. This way, the impact of interruptions due to failures on the processing time of the jobs is also considered, what enhances the chances for the initial solution of not necessarily being trapped in a local optimum as it often happens with the original LPT rule.

The step 6 analyses and decides if the quality of the solution obtained up this point is adequate so as to stop the algorithm. To do this, the obtained makespan (C_{max}) is compared with a minimum level calculated for it (\hat{C}_{min}) increased by an amount ρ . To obtain the value of \hat{C}_{min} , an algorithm that takes advantage of the convex character of the TTM function is proposed.

Subsequently, the steps from 7 to 11 seek to refine the solution in case the stop criterion established in step 6 is not met, for which an exhaustive search approach is implemented.

CONCLUSIONS AND FURTHER RESEARCH ISSUES

This article addressed one of the most significant problems that occur at the shopfloor level, related to the need of establishing integrated production and preventive maintenance programs. To contribute to the solution of this problem, a constructive heuristic was developed, which makes the use of the LPT dispatch rule principles, and introduces a modification that allows to consider the impact of interruptions due to failure on the production time of the product, and all this in a context and productive configuration of identical parallel machines.

The proposed algorithm is implicitly divided into two phases. In the first one (steps 1 to 6), an initial solution is obtained as a result of the slightly modified LPT rule and a new rule defined during its development, called r_j - LPT. If this previous solution does not meet the required quality level, the second phase determined by the last four steps is implemented, which implements an exhaustive search strategy.

Future research is aimed at analyzing the feasibility of developing a hybrid method in which the proposed constructive heuristic is combined with some of the local search algorithms that have shown their ability to be applied in this type of combinatorial optimization problems.

Acknowledgement

This paper was supported by the KEGA-021STU-4/2018 Project of Development of a laboratory for the design and maintenance of production systems supported by the use of Virtual Reality. This support is gratefully acknowledged.

References:

1. SWANSON, L. 1997. An empirical study of the relationship between production technology and maintenance management. *Int. Journal of Production Economics*, 53, pp.199 – 207. ISSN 0925-5273.
2. GAO, X., BARABADY, J., MARKESET, T. 2010. Criticality analysis of a production facility using cost importance measures. *Int. Journal of Systems Assurance Engineering and Management*. 1, pp. 17–23.
3. TAMBE, P. P., KULKARNI, M. S. 2013. A novel approach for production scheduling of a high pressure die casting machine subjected to selective maintenance and a sampling procedure for quality control. *Int. Journal of Systems Assurance Engineering and Management*.
4. RUIZ, R., GARCÍA-DÍAZ, J., MAROTO, C. 2007. Considering scheduling and preventive maintenance in the flowshop sequencing problem. In *Computers & Operations Research*, **34**(11), pp. 3314-3330. ISSN: 0305-0548.
5. HADIDI, L. A., AL-TURKI, U. M. 2012. Integrated models in production planning and scheduling, maintenance and quality: a review. *Int. Journal of Industrial and Systems Engineering*, **10**(1).
6. KUMAR, S., PUROHIT, B. S., KUMAR, B, L. 2014. *Integrated Approach for Job Scheduling and Multi-Component Maintenance Planning in a Production System*. 5th Int. & 26th All India Manufacturing Technology, Design and Research Conference (AIMTDR 2014).
7. DA, W., FENG, H., PAN, E. 2016. Integrated preventive maintenance and production scheduling optimization on uniform parallel machines with deterioration effect. In *Industrial Engineering and Engineering Management (IEEM)*. Int. Conference on. IEEE, pp. 951-955.
8. ZAHEDI, Z., SALIM, A. 2017. Integrating Preventive Maintenance Scheduling as Probability Machine Failure and Batch Production Scheduling. *ComTech: Computer, Mathematics and Engineering Applications*, **7**(2), pp. 105-112.
9. JING, Z., HUA, J., ZHU, Y. 2017. Multi-objective Integrated Optimization Problem of Preventive Maintenance Planning and Flexible Job-Shop Scheduling. In: *Proceedings of the 23rd Int. Conference on Industrial Engineering and Engineering Management*. DOI 10.2991/978-94-6239-255-7_25.
10. BOUDJELIDA, A. 2017. On the robustness of joint production and maintenance scheduling in presence of uncertainties. *Journal of Intelligent Manufacturing*, pp. 1-16.
11. PINEDO, M. L. 2012. *Scheduling. Theory, Algorithms, and Systems*. Fourth edition. Springer. New York University. New York.
12. COFFMAN, E. G., GAREY, M. R., JOHNSON, D. S. 1978. An application of Bin-Packing to multiprocessor scheduling. *SIAM J. COMPUT.*, **7**(1), pp. 1-17.
13. XU, D.; SUN, K.; LI, H. 2008. Parallel machine scheduling with almost periodic maintenance and non-preemptive jobs to minimize makespan. In *Computers & Operations Research* **35**, pp. 1344 – 1349.

14. LEE, W., CH., WU, CH, CH. 2008. Multi machine scheduling with deteriorating jobs and scheduled maintenance. In *Applied Mathematical Modelling*, 32, pp. 362–373.
15. MIRABEDINI, S, N., IRANMANESH, H. 2014. A scheduling model for serial jobs on parallel machines with different preventive maintenance (PM). *Int. Journal of Advanced Manufacturing Technologies*, 70, 1579–1589.
16. LEE, W. CH., WANG, J. Y., LEE, L. Y. 2015. A hybrid genetic algorithm for an identical parallel machine problem with maintenance activity. *Journal of the Operational Research Society*, pp. 1–13.
17. KUMAR, S., KUMAR – LAD, B. 2016. Integrated production and maintenance planning for parallel machine system considering cost of rejection. *Journal of the Operational Research Society*, pp. 1-13.

ORCID:

Daynier Rolando Delgado Sobrino	0000-0001-9253-6141
Roman Ružarovský	0000-0002-9465-4544
Radovan Holubek	0000-0003-0844-8603