

Scheduling a maintenance activity under skills constraints to minimize total weighted tardiness and late tasks

Djalal Hedjazi*

L@STIC Laboratory, Computer Science Department, University of Batna, Batna, Algeria

CHRONICLE

Article history:

Received November 12 2014

Received in Revised Format

January 10 2014

Accepted January 10 2015

Available online

January 11 2015

Keywords:

Scheduling

Maintenance activity

Resource constraints

Skills

Tardiness

ABSTRACT

Skill management is a key factor in improving effectiveness of industrial companies, notably their maintenance services. The problem considered in this paper concerns scheduling of maintenance tasks under resource (maintenance teams) constraints. This problem is generally known as unrelated parallel machine scheduling. We consider the problem with a both objectives of minimizing total weighted tardiness (TWT) and number of tardiness tasks. Our interest is focused particularly on solving this problem under skill constraints, which each resource has a skill level. So, we propose a new efficient heuristic to obtain an approximate solution for this NP-hard problem and demonstrate his effectiveness through computational experiments. This heuristic is designed for implementation in a static maintenance scheduling problem (with unequal release dates, processing times and resource skills), while minimizing objective functions aforementioned.

© 2015 Growing Science Ltd. All rights reserved

1. Introduction

The maintenance activity has become extremely important in the industry since of its advantages in terms of keeping system availability and safety, as well as improving both quality and productivity (Alyouf, 2007; Ait-Kadi et al., 2011). According to intervention type, the maintenance can be classified into two categories: corrective and preventive. Corrective maintenance consists of those tasks required to restore a system to a functioning state after an identified or suspicious failure has occurred. Preventive maintenance consists of scheduled tasks performed before a failure is likely to occur. It can reduce the number of failures but the costs will increase as the level of prevention increases. There is an optimum level of preventive maintenance that improves system performance and reduces total maintenance costs.

Naturally, the preventive maintenance tasks depend on the concerned system, nevertheless classical tasks include changing lubricants, replacing worn parts, re-calibrating adjustable subsystems, etc. It requires a fixed time interval during which the system is turned off and production is stopped. So, after the preventive maintenance time, the system becomes more efficient, but it will be indispensable to establish a strategy that spreads the tasks to be performed over a suitable time scale that is, available resources

* Corresponding author.

E-mail: djalal05@yahoo.fr djalal_hedjazi@univ-batna.dz (D. Hedjazi)

must be assigned to tasks, each one of which will include a temporal sequence of tasks. So, depending on the maintenance tasks' and resources' features, one of the primordial problems of the maintenance service manager will be to find for each task what resource must treat it? and at what time must accomplish it? This problem is generally known as Maintenance Scheduling (MS).

The MS problem has attracted large attention and many authors focalized their research on this area putting in evidence the increasing interest shown by the scientific and industrial community to face the related problematic. In fact, the exact solution methods such as branch and bound (Egan et al., 1976; Bagchi et al., 1987; Dorn & Kerr, 1994), dynamic programming (Zurn & Quintana, 1975; Abdul-Razaq & Potts, 1988), integer programming (Dopazo & Merrill, 1975; Edwin & Curtius, 1990) and Lagrangian relaxation (Li, 1997) generate optimal schedules. Practically, these methods are generally inefficient in computational time terms that they are not suggested for scheduling of flexible industrial systems under dynamically changing conditions. Moreover, these methods based on mathematical optimization techniques are used for the MS that belongs to combinatorial optimization problem. Indeed, they can generate an exacting optimal solution for small scale problems but are inefficient for the large scale problems because of considerable number of transitional solutions. So, it is very hard in MS context to find the global optimal solution of large scale problems within reasonable computing time. Thus, many approximation approaches, based on the characteristics of the problem, have been proposed. These approaches such as simulated annealing (Aarts et al. 1986; Satoh & Nara, 1991), tabu search (Glover 1986, Gopalakrishnan et al., 2001), and genetic algorithm (Goldberg, 1989; Besbes et al., 2010) are generally known as heuristics or meta-heuristics techniques, generate acceptable schedules with less computational time than the exact approaches. Practically, for greater acceptance, approximation approaches must take care of dominant uncertainty of changes such as system failures.

Nowadays, MS problem become a popular topic among researches and extensive efforts have been investigated in solving various problems (Kubzin and Strusevich, 2006; Levin et al., 2009; Mosheiov & Sarig, 2009a, 2009b); Yang, 2010; Yang & Yang, 2010, Sun & Li, 2010; Batun & Azizoglu, 2009; Hartmann & Briskorn, 2010; Mor & Mosheiov, 2012). We are interested in this paper to unrelated parallel-machine scheduling problems. Most of these problems are NP-hard and thus are computationally challenging (Lin et al., 2011). Therefore, the development of efficient heuristics is often important to solve large instances of these problems. So, we mentioned Pfund which presented in the first part of his study (Pfund et al., 2004) a state of the art of algorithms for single- and multi-objective unrelated parallel machine deterministic scheduling problems, than identified that unrelated parallel machine problems rest relatively unstudied. Notably, they indicated that there were little solution approaches to minimize due-date-related criteria. Many authors have considered these problems for individual minimization of three regular and important performance measures: total weighted completion time, makespan and total weighted tardiness. According to Pfund et al. (2004), the problem of minimizing the makespan is the most studied of all unrelated parallel machine scheduling problems, including (Lin et al., 2011; Gairinga et al., 2007; Azar & Epstein, 2005; Hariri & Potts, 1991; Davis & Jaffe, 1981; De & Morton, 1980) based on their solutions on two-phase heuristics that used LP relaxation followed by scheduling non-integer variables. In addition, the total weighted completion time which the weight of task represents a priority features, denoting the importance of concerned task relative to the others in the system. This problem is formulated by Azizoglu and Kirca (1999) as a mathematical program and then the solution given to solve it use a branch-and-bound algorithm. Cruz-Chávez et al. (2009) also presented a solution to this problem using a simulated annealing algorithm that solves it as a weighted bipartite matching problem. The total weighted tardiness is a measure of customer satisfaction as well as it has more or less been addressed in literature (Marmier et al., 2009a; Lee et al., 1996). However, it does not preclude reference to some research works proposed in this context; including Panwalkar et al. (1993) who proposed a heuristic to minimize the mean tardiness for the single-machine sequencing problem. This heuristic is based on the shortest processing time (SPT) and the earliest due date (EDD) principles. An extension of this solution from the single-machine tardiness problem to a parallel-machine setting was proposed by Koulamas (1997). Further, a heuristic was developed by Liaw et al. (2000) to give an upper bound for the problem. Three years after Liaw et al. (2003) solved the problem using a branch-and-bound algorithm. In addition,

Cao et al. (2005) proposed tabu search-based heuristic to solve the problem of simultaneously selecting and scheduling parallel machines to minimize the sum of job tardiness cost and machine holding cost.

Moreover, the industrial companies consider the management of human resources according to their skills is extremely important (Grabot et al., 2000). However, few authors have dealt the unrelated parallel machine scheduling problems under resources' and skills' constraints (Ait et al., 2011). Among researchers who have dealt with this problem, we reference the work of Marmier et al. (2009a) who proposed an approach to assign maintenance tasks to resources under skills constraint. In order to minimize the total weighted tardiness, the static scheduling algorithm has been introduced in the first part. In addition, to confer a maximum robustness to the obtained schedule a proactive methodology which takes into account possible variations has been also proposed. In this context, the same authors (Marmier et al., 2009b) proposed a multi-criteria approach to dynamically insert new tasks into an existing schedule. This approach gives to the maintenance manager a set of solutions with their evaluations following different criteria. Indeed, Fuzzy logic has been used to deal with uncertainties and to evaluate potential penalties.

The problem considered herein is MS under skills constraints. It consists of assigning N independent non-pre-emptive maintenance tasks: $j, j = \{1 \dots n\}$ to M resources, which represent the maintenance teams. The objective is to minimize total weighted tardiness and number of late tasks. According to Lawler (1993), this problem is an NP-hard in the strong sense and the exact solution appears very hard even on very small inputs (Congram et al., 2002). This paper presents a new heuristic as well as the numerical tests executed. These tests indicate that the proposed heuristic performs consistently well and the average optimality gaps are fairly small in the majority of them.

The remainder of the paper is organized as follows: The problem formulation is given in Section 2. Section 3 describes our scheduling heuristic for generating robust schedules. Section 4 is devoted to the presentation of computational experiments and conclusions plus perspectives are given in Section 5.

2. Formulation

This research work deals firstly with scheduling problem describing and secondly with resources assignment. We consider a manufacture system composed of several independent equipment, and sharing M resources. These resources are responsible for performing preventive and corrective maintenance tasks on different machines of the system. Often, the number of resources is very less than the number of tasks ($M \ll N$). We also assume that all tasks are performed without pre-emption. Thus, once begun, each task is performed to its accomplishment without interruption. During the maintenance process, no production is performed, as well as the maintenance tasks do not affect the successor tasks. In addition, it is assumed that the logistic times are small enough to be integrated in the maintenance tasks' processing times.

Formally, our problem can be described similarly as the one developed in (Marmier et al., 2009a): Given a set of N tasks. The basic processing time of task j is denoted by $P_j, j = 1 \dots n$. Practically, the processing time is subject to variations depending on the skill of resource it is assigned to. So, we consider M resources characterized by a skill profiles. The processing time of task j if assigned to resource i is denoted by $P_{ij}, i = 1, \dots, m, j = 1, \dots, n$, which is given by :

$$p_{ij} = f(p_j, \text{Comp}_{i,j}) = p_j \text{Comp}_{ij}, \quad (1)$$

where each resource has a corresponding skill level for each task and the relation (task-skill-resource) can be given by a following matrix:

$$\text{Comp} = \begin{pmatrix} \text{comp}_{1,1} & \dots & \text{comp}_{1,n} \\ \dots & \dots & \dots \\ \text{comp}_{m,1} & \dots & \text{comp}_{m,n} \end{pmatrix} \quad (2)$$

Furthermore, we assume that all tasks cannot be scheduled before the machine availability. So, using the notation $r_j, j = 1 \dots n$ to denote release-date of task j , The starting of j must be after r_j denoted by $start_j > r_j$. In addition, the different machines in the system are weighted according to their sensitivity and their importance in the manufacture system. Thus, the weight (priority) of task j is denoted by w_j , which represent the weighted of the machine. Accordingly, the maintenance service is looking to maximize the equipment availability in manufacture system with the tasks having different levels; hence our performance measure of interest is total weighted tardiness. This last is described as a measure that incurs a penalty for each task that finishes processing after its promised date. This penalty increases with the magnitude of the tardiness, and therefore schedules that minimize the weighted sum of penalties provide good performance, whereas higher than required values of tardiness indicate that many important tasks are not being treated on time. Total weighted tardiness is the summation of the weighted tardiness over all tasks $j=1, \dots, n$. It is denoted as $w_j T_j$, where $T_j = \max(0, C_j - d_j)$. Further, C_j and d_j refer to the completion time and the due date of task j , respectively.

On the other words, our objective is to schedule N tasks on M resources, such that the total weighted tardiness and the number of late tasks are simultaneously minimized:

$$\min \sum_{j=1}^n w_j T_j \quad (3)$$

$$\min \sum_{j=1}^n U_j \quad (4)$$

3. Heuristic algorithm

In the industry world, the maintenance service is responsible to schedule a preventive maintenance tasks to available resources in the horizon time. This could be done with exploiting static scheduling algorithms. However, the industrial machines are often subject to random failures which must be planned in the current schedule. This could be done with exploiting dynamic scheduling algorithms. So, in this section we introduce our heuristic algorithm for the static scheduling after purposing some definitions that will be used in (Ouelhadj & Petrovic, 2009).

3.1 Definitions

- *Optimality Window (OW)*: As described in the figure bellow, for each task j the interval $[r_j, d_j]$ is called optimality window for which the cost of task processing is optimal. Therefore, within preventive strategy, the curve can be explained as: "When we intervene before r_j , the maintenance cost increases which mean that we do too often preventive maintenance. On the other hand, if we exceed the d_j date, the risk that the machine fails increases, which requires corrective maintenance. This last generates often additional costs notably these related to the cessation of production".

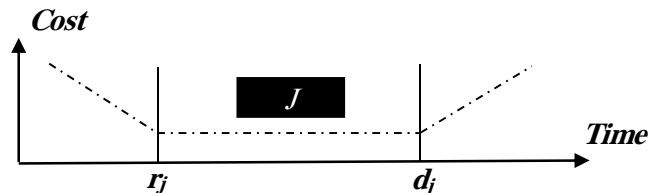


Fig. 1. Optimality window of task

- *Basic Window (BW)*: The BW represents the interval between two tasks j and $(j+1)$ assigned to the same resource (Fig. 2). It calculate by the following formula:

$$Bw[j][j+1] = [r_j + p_j, d_{j+1} - p_{j+1}] \quad (5)$$

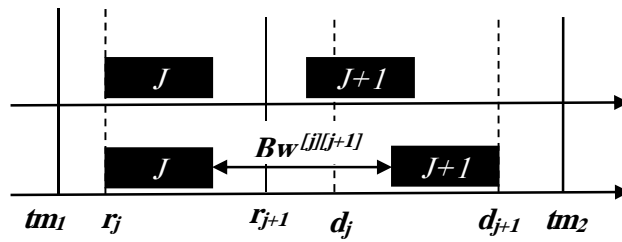


Fig. 2. Graphic representation of Basic window

- *BEst Window (BEW)*: The BEW_j for a task j is the BW (with $BW \leq P_{ij}$) and the total weighted tardiness is minimal. Once two BW give a same cost, the BEW_j is the one of smallest window.
- *Liberty of task*: The liberty of a task in a given scheduling is the possible dates of task processing in the optimality window. It indicates the ability of the task to move in the scheduling. The mathematical expression of liberty of task j if it processed by the resource i is as:

$$lib_j = d_j - r_j - p_{ij} \tag{6}$$

3.2 Static scheduling

The static scheduling heuristic consists of two main steps:

Step 1: pre-treatment of maintenance tasks list (L) which can be:

- The ascending sort: represents the sort in ascending order from r_j . When two tasks have the same r_j , the result is sorted in ascending order from due-date of task (d_j)
- The descending sort: represents the sort in ascending order from r_j . When two tasks have the same r_j , the result is sorted in descending order from due-date of task (d_j)

A comparative study is done to compare the both versions of heuristic (with ascending sort and descending sort). The results prove that the heuristic with ascending perform well in term of the both objective functions (TWT and number of late tasks). The Fig. 3 presents the comparative study results.

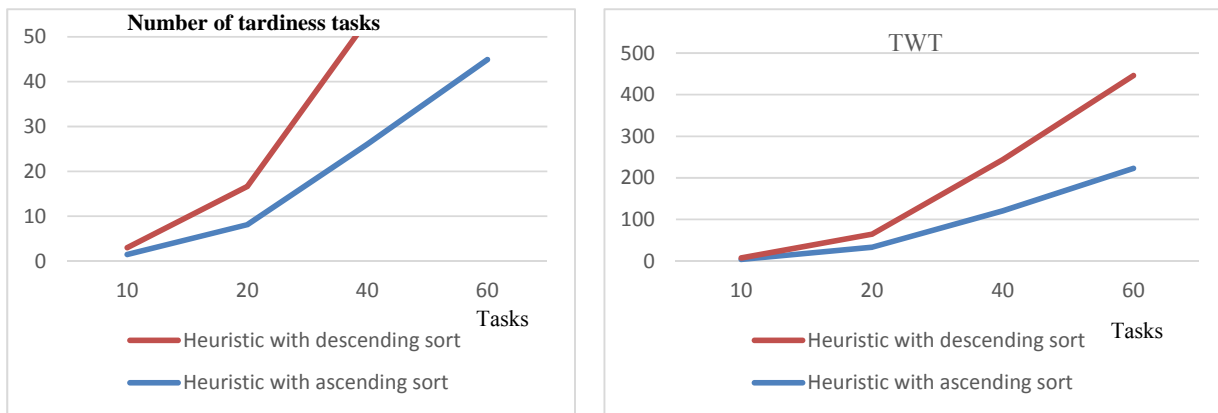
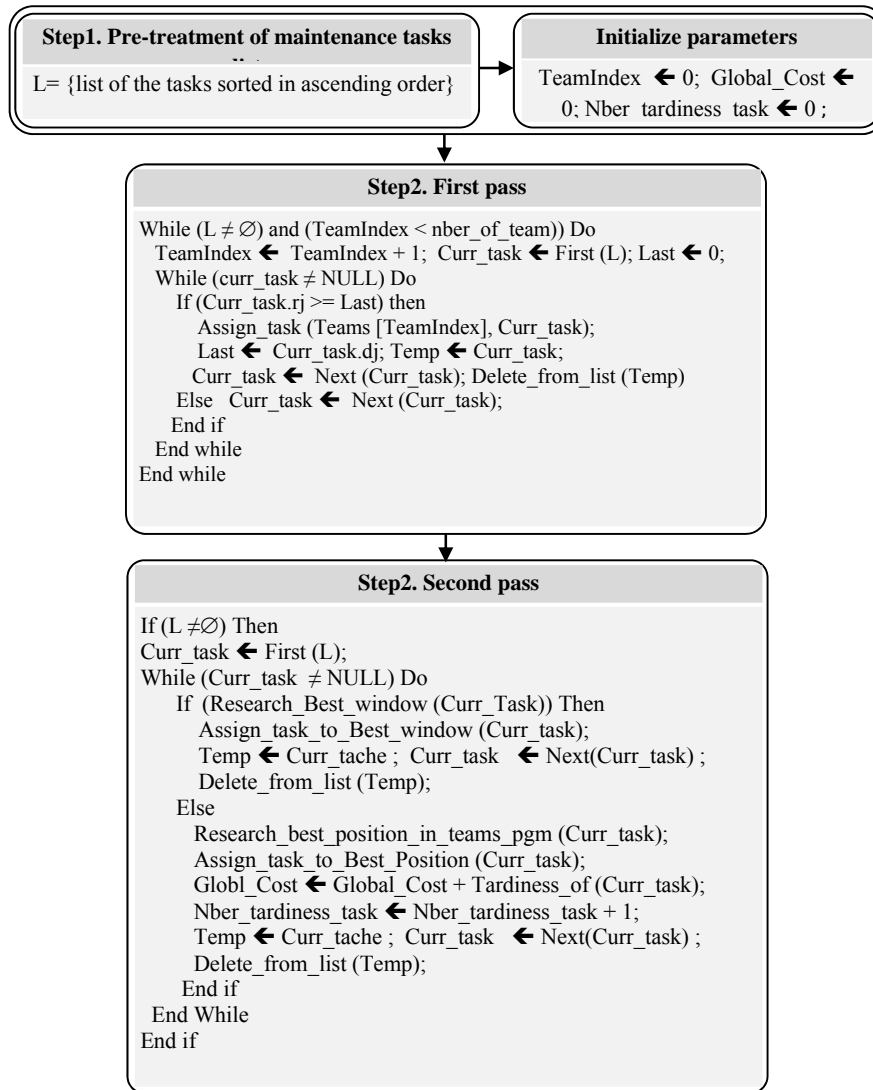


Fig. 3. Comparative study between both versions of heuristic (with ascending and descending sort)

Step 2: In this step, two passes on the list (L) will be done to assign tasks to resources. In the first one, we try to schedule and assign to each resource a maximum of tasks which their OW are not overlapped. In the second one, we aim to schedule and assign unassigned tasks while seeking the best position in resource planning's. The best position is the one of the BEW (see §3.1.).



As an illustration of the heuristic, the Fig. 4 explains the principle of static scheduling heuristic progression on an example with seven preventive maintenance tasks.

4. Experimental results and discussion

In this section, we demonstrate the efficiency of the proposed heuristic. This will be confirmed in three points: (1) to ascertain that the heuristic generates the good schedules in terms of total weighted tardiness and number of late tasks, (2) to compare its quality to some other heuristics and (3) to point out other advantages and disadvantages of the heuristic. All the programs, implementing our heuristic algorithm and some algorithms from literature, coded in MATLAB language.

For reasons of comparing with the performances of the heuristic, some computational experiments are conducted. So, we built a generator that randomly generates test cases with different problem dimensions, where the problem dimension is the number of tasks and resources in each case. Thus, we conducted experiments with number of tasks increasing up to 160 as well as the number of resources increasing up to 8. Additionally, the data used in the generator was generated from a uniform distribution.

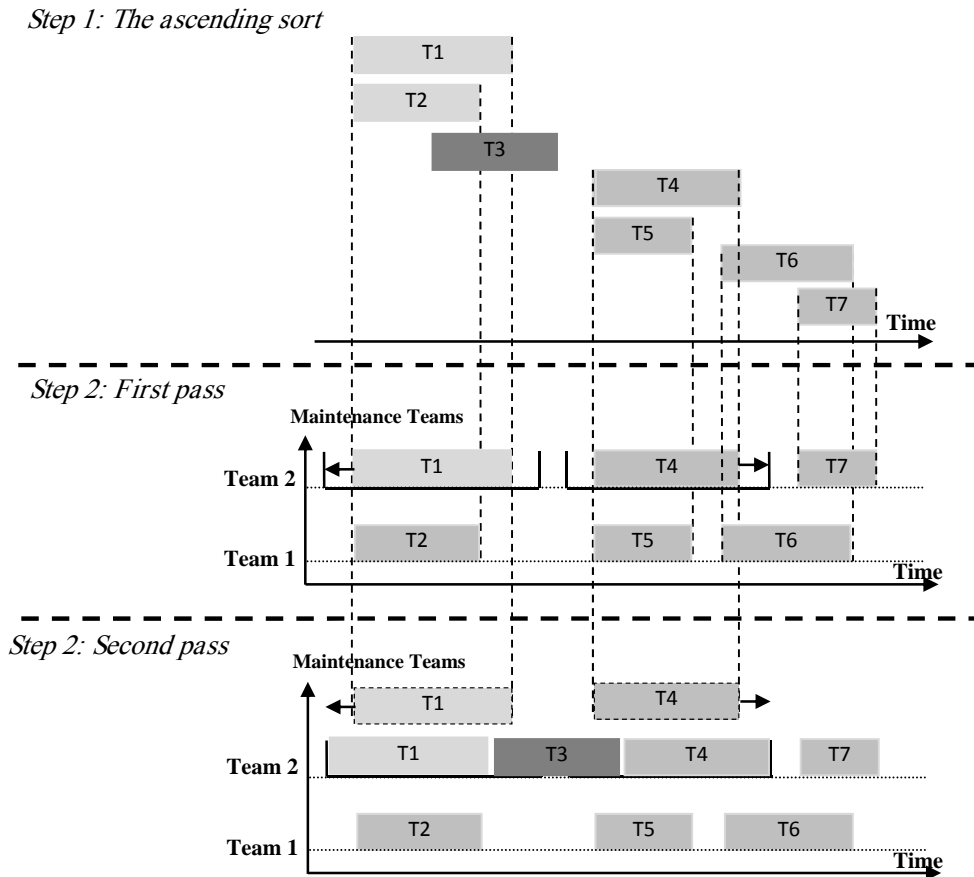


Fig. 4. Static scheduling heuristic: an example

Similarly to the random generator adopted by (Marmier et al, 2009a), our generator is characterized as follow: For each task, the values of p_j are generated as an integer obtained in the interval $[1, 7200]$ (varying between one second and two hours), as well as the skill required for the tasks is randomly determined by taking an integer value which can be 1, 2 or 3 (implying that there are three skills). So, for each resource the level of skill is a real value generated in $[1.01, 2.00]$. Moreover, the availability dates (r_j) and the weights of penalty (w_j) are respectively generated in the intervals: $[0, 86400]$ (varying between 0 and 24 hours), $[1, 100]$. The due-dates (d_j) are generated as real values in the interval: $[r_j+2 \times p_j, r_j+2 \times p_j+86400]$ (To ensuring that tasks are achievable in the time, due dates cannot be fixed before $r_j+2 \times p_j$ and after two hours).

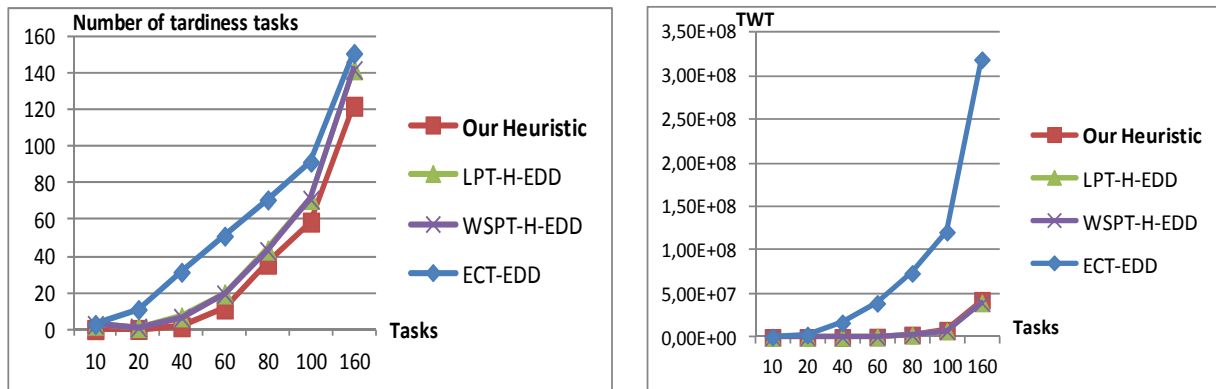
Firstly, we examine the performance of the presented static scheduling heuristic using test case with different problem dimensions, and we generate randomly a set of 100 examples for each problem dimension. Table 1 illustrates experimental results for test cases that include 10, 20, 40, 60, 80, 100, 160 tasks, as well as 4, 6, 8 resources. Columns give the values of both objective functions compared to the LPT-H-EDD, ECT-EDD and WSPT-H-EDD algorithms published in (Marmier et al., 2009a). Our results shown in the Table 1 reveal clearly that our static scheduling generates better schedules.

Additionally, the curves plotted on the Fig. 5 presents the evolution of both objective functions of the proposed heuristic with 4 resources compared to LPT-H-EDD, WSPT-H-EDD and ECT-EDD respectively.

Table 1

Results of all comparison experiments

M	N	Our Heuristic		LPT-H-EDD		WSPT-H-EDD		ECT-EDD	
		$\sum_{j=1}^n w_j T_j$	$\sum_{j=1}^n U_j$	$\sum_{j=1}^n w_j T_j$	$\sum_{j=1}^n U_j$	$\sum_{j=1}^n w_j T_j$	$\sum_{j=1}^n U_j$	$\sum_{j=1}^n w_j T_j$	$\sum_{j=1}^n U_j$
4	10	0	0	3173,1	2,4	3373,1	2,6	589430	2,8
	20	932,39	0,05	21352	1,11	17612	1,08	2812200	10,92
	40	29459	1,54	178600	6,95	153880	6,47	16676000	31,34
	60	438040	11,28	693300	19,53	666100	19,32	39164000	51,12
	80	2882200	35,82	2707900	43,56	2537100	43,07	73499000	70,94
6	10	0	0	1208,6	0,1	1784,3	0,11	283330	1,73
	20	0	0	10724	0,65	7539,2	0,55	2021200	8,3
	40	555,62	0,02	62367	3,5	43243	2,94	10142000	26,21
	60	15126	0,88	202840	9,04	138400	8,06	28678000	46,76
	80	179220	7,21	607740	19,83	509370	19,09	59909000	66,62
	100	975830	23,06	1500100	36,25	1309900	35,74	101160000	87,28
8	10	0	0	599,8	0,09	1247,3	0,07	124390	0,66
	20	0	0	8109,3	0,44	5356,6	0,41	1318000	6,22
	40	0	0	35704	2,21	15561	1,69	7360000	23,11
	60	0	0	112930	6,03	83928	5,34	22753000	42,73
	80	9991,4	0,76	274840	12,16	189620	10,32	45061000	62,07
	100	66053	4,09	602440	21,97	4689700	20,85	83482000	82,77
	160	4606500	65,97	4755000	83,35	4155900	81,17	258180000	142,75

**Fig. 5.** Evolution of the both objective functions (TWT and number of late tasks) on 4 resources

5. Conclusion

We addressed the maintenance scheduling problem under constraints of resources and skills developing a new heuristic for its solution. The objective was to find for each resource a maintenance tasks sequence and a starting time of each task to minimize the total weighted tardiness and the number of late tasks. Our method for this NP-hard problem has been shown to produce good solution and it focused on the off-line scheduling problem. Finally, experiments using randomly generated problems were conducted to demonstrate the effectiveness of the proposed heuristic. The good results produced in terms of minimizing number of late tasks proved that the heuristic seem more preferred for the maintenance companies in order to satisfy a maximum of customers.

Future works will follow principally one line of research. So, we are focusing on integration in our model the criteria which allow sharing material means between resources, as well as adapting the proposed heuristic to this new problem.

Acknowledgement

The work had been carried out in the context of a research project financed by the Algerian Ministry of Higher Education and Scientific Research. Also, we thank several people for their contribution in overhauling this paper. Finally, we thank and acknowledge the reviewers, who have donated their time and expertise to assist in revising this research work.

References

- Aarts, E. H., de Bont, F. M., Habers, E. H., & van Laarhoven, P. J. (1986). Parallel implementations of the statistical cooling algorithm. *INTEGRATION, the VLSI journal*, 4(3), 209-238.
- Abdul-Razaq, T. S., & Potts, C. N. (1988). Dynamic programming state-space relaxation for single-machine scheduling. *Journal of the Operational Research Society*, 39(2), 141-152.
- Ait-Kadi, D., Menye, J. B., & Kane, H. (2011). Resources assignment model in maintenance activities scheduling. *International Journal of Production Research*, 49(22), 6677-6689.
- Alsyouf, I. (2007). The role of maintenance in improving companies' productivity and profitability. *International Journal of Production Economics*, 105(1), 70-78.
- Azar, Y., & Epstein, A. (2005, May). Convex programming for scheduling unrelated parallel machines. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing* (pp. 331-337). ACM.
- Azizoglu, M. E. R. A. L., & Kirca, O. M. E. R. (1999). Scheduling jobs on unrelated parallel machines to minimize regular total cost functions. *IIE transactions*, 31(2), 153-159.
- Bagchi, U., Sullivan, R. S., & Chang, Y. L. (1987). Minimizing mean squared deviation of completion times about a common due date. *Management Science*, 33(7), 894-906.
- Batun, S., & Azizoglu, M. (2009). Single machine scheduling with preventive maintenances. *International Journal of Production Research*, 47(7), 1753-1771.
- Besbes, W., Teghem, J., & Loukil, T. (2010). Scheduling hybrid flow shop problem with non-fixed availability constraints. *European Journal of Industrial Engineering*, 4(4), 413-433.
- Cao, D., Chen, M., & Wan, G. (2005). Parallel machine selection and job scheduling to minimize machine cost and job tardiness. *Computers & operations research*, 32(8), 1995-2012.
- Congram, R. K., Potts, C. N., & van de Velde, S. L. (2002). An iterated dynasearch algorithm for the single-machine total weighted tardiness scheduling problem. *INFORMS Journal on Computing*, 14(1), 52-67.
- Cruz-Chávez, M. A., Juárez-Pérez, F., Ávila-Melgar, E. Y., & Martínez-Oropeza, A. (2009, September). Simulated annealing algorithm for the weighted unrelated parallel machines problem. In *Electronics, Robotics and Automotive Mechanics Conference, 2009. CERMA'09.* (pp. 94-99). IEEE.
- Davis, E., & Jaffe, J. M. (1981). Algorithms for scheduling tasks on unrelated processors. *Journal of the ACM (JACM)*, 28(4), 721-736.
- De, P., & Morton, T. E. (1980). Scheduling to minimize makespan on unequal parallel processors. *Decision Sciences*, 11(4), 586-602.
- Dopazo, J. F., & Merrill, H. M. (1975). Optimal generator maintenance scheduling using integer programming. *Power Apparatus and Systems, IEEE Transactions on*, 94(5), 1537-1545.
- Dorn, J., & Kerr, R. M. (1994, June). Co-operating scheduling systems communicating through fuzzy sets. In *Preprints of the 2nd IFAC/IFIP/IFORS-Workshop on Intelligent Manufacturing Systems* (pp. 367-373).
- Edwin, K. W., & Curtius, F. (1990). New maintenance-scheduling method with production cost minimization via integer linear programming. *International Journal of Electrical Power & Energy Systems*, 12(3), 165-170.
- Egan, G. T., Dillon, T. S., & Morsztyn, K. (1976). An experimental method of determination of optimal maintenance schedules in power systems using the branch-and-bound technique. *Systems, Man and Cybernetics, IEEE Transactions on*, (8), 538-547.
- Gairing, M., Monien, B., & Woelaw, A. (2007). A faster combinatorial approximation algorithm for scheduling unrelated parallel machines. *Theoretical Computer Science*, 380(1), 87-99.
- Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5), 533-549.
- Goldberg, D.E., (1989). *Genetic algorithms in search, optimization and machine learning*. First Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.
- Gopalakrishnan, M., Mohan, S., & He, Z. (2001). A tabu search heuristic for preventive maintenance scheduling. *Computers & industrial engineering*, 40(1), 149-160.

- Grabot, B., & Letouzey, A. (2000). Short-term manpower management in manufacturing systems: new requirements and DSS prototyping. *Computers in industry*, 43(1), 11-29.
- Hariri, A. M. A., & Potts, C. N. (1991). Heuristics for scheduling unrelated parallel machines. *Computers & operations research*, 18(3), 323-331.
- Hartmann, S., & Briskorn, D. (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, 207(1), 1-14.
- Koulamas, C. (1997). Decomposition and hybrid simulated annealing heuristics for the parallel-machine total tardiness problem. *Naval Research Logistics (NRL)*, 44(1), 109-125.
- Kubzin, M. A., & Strusevich, V. A. (2006). Planning machine maintenance in two-machine shop scheduling. *Operations Research*, 54(4), 789-800.
- Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H., & Shmoys, D. B. (1993). Sequencing and scheduling: Algorithms and complexity. *Handbooks in operations research and management science*, 4, 445-522.
- Lee, H. S., Murthy, S. S., Haider, S. W., & Morse, D. V. (1996). Primary production scheduling at steelmaking industries. *IBM Journal of Research and Development*, 40(2), 231-252.
- Levin, A., Mosheiov, G., & Sarig, A. (2009). Scheduling a maintenance activity on parallel identical machines. *Naval Research Logistics (NRL)*, 56(1), 33-41.
- Li, G. (1997). Single machine earliness and tardiness scheduling. *European Journal of Operational Research*, 96(3), 546-558.
- Liaw, C.F, Lin, Y.K., Chen, M.C., (2000). Scheduling unrelated parallel machines to minimize total weighted tardiness. In Proceedings of the fifth annual international conference on industrial engineering—theory, applications, and practice, pp. 1–11.
- Liaw, C. F., Lin, Y. K., Cheng, C. Y., & Chen, M. (2003). Scheduling unrelated parallel machines to minimize total weighted tardiness. *Computers & Operations Research*, 30(12), 1777-1789.
- Lin, Y. K., Pfund, M. E., & Fowler, J. W. (2011). Heuristics for minimizing regular performance measures in unrelated parallel machine scheduling problems. *Computers & Operations Research*, 38(6), 901-916.
- Marmier, F., Varnier, C., & Zerhouni, N. (2009). Static et dynamic scheduling of maintenance activities under the constraints of skills. *Journal of Operations and Logistics*, 2(3), 1-16.
- Marmier, F., Varnier, C., Zerhouni, N., (2009b), 'Proactive, dynamic and multi-criteria scheduling of maintenance activities. *International Journal of Production Research*, 47(8), 2185-2201.
- Mor, B., & Mosheiov, G. (2012). Scheduling a maintenance activity and due-window assignment based on common flow allowance. *International Journal of Production Economics*, 135(1), 222-230.
- Mosheiov, G., & Sarig, A. (2009). Scheduling a maintenance activity to minimize total weighted completion-time. *Computers & Mathematics with Applications*, 57(4), 619-623.
- Mosheiov, G., & Sarig, A. (2009). A note: Simple heuristics for scheduling a maintenance activity on unrelated machines. *Computers & Operations Research*, 36(10), 2759-2762.
- Ouelhadj, D., & Petrovic, S. (2009). A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling*, 12(4), 417-431.
- Panwalkar, S. S., Smith, M. L., & Koulamas, C. P. (1993). A heuristic for the single machine tardiness problem. *European Journal of Operational Research*, 70(3), 304-310.
- Pfund, M., Fowler, J. W., & Gupta, J. N. (2004). A survey of algorithms for single and multi-objective unrelated parallel-machine deterministic scheduling problems. *Journal of the Chinese Institute of Industrial Engineers*, 21(3), 230-241.
- Satoh, T., & Nara, K. (1991). Maintenance scheduling by using simulated annealing method [for power plants]. *Power Systems, IEEE Transactions on*, 6(2), 850-857.
- Sun, K., & Li, H. (2010). Scheduling problems with multiple maintenance activities and non-preemptive jobs on two identical parallel machines. *International Journal of Production Economics*, 124(1), 151-158.
- Yang, S. J. (2010). Single-machine scheduling problems with both start-time dependent learning and position dependent aging effects under deteriorating maintenance consideration. *Applied Mathematics and Computation*, 217(7), 3321-3329.
- Yang, S. J., & Yang, D. L. (2010). Minimizing the total completion time in single-machine scheduling with aging/deteriorating effects and deteriorating maintenance activities. *Computers & Mathematics with Applications*, 60(7), 2161-2169.
- Zurn, H. H., & Quintana, V. H. (1975). Generator maintenance scheduling via successive approximations dynamic programming. *Power Apparatus and Systems, IEEE Transactions on*, 94(2), 665-671.