A Survey of Parallel Machine Scheduling under Availability Constraints

Jihene Kaabi, Youssef Harrath
College of Information Technology, University of Bahrain
Kingdom of Bahrain, P. O. Box 32038
Email: yharrath {at} uob.edu.bh

Abstract – **This paper reviews results related to the parallel machine scheduling problem under availability constraints. The motivation of this paper comes from the fact that no survey focusing on this specific problem was published. The problems of single machine, identical, uniform, and unrelated parallel machines under different constraints and optimizing various objective functions were analyzed. For every surveyed problem, the solving algorithms and their complexities were presented.**

Keywords- Parallel machine scheduling; availability constraints; maintenance.

I. INTRODUCTION

Machine scheduling under availability constraints is a real industrial problem that attracted more and more attention during the last years. In fact, the machines may be subject to accidental breakdowns, periodic preventive maintenance, tool changes, workers availability, and availability of the resources used by the machines, and so on. Most of the research done in this area assumed that the non-availability of the machines is mainly due to preventive maintenance activities. There are two classes of problems considered depending on whether the scheduling of preventive maintenance activities is determined before the scheduling of jobs or jointly with the scheduling of jobs. The first problem is often referred in the literature as scheduling with machine availability where the maintenance periods are known and fixed in advance and denoted by *Pm* (Periodic maintenance). The second problem is called joint/simultaneous production and maintenance scheduling problem and denoted by *Jm* (Jointly production/maintenance). Despite the interdependent relationship between the production scheduling and the maintenance planning, the two activities are generally planned and executed separately in real manufacturing systems (class 1). This kind of schedules may result in an unsatisfied demand or machine breakdowns if the production and maintenance services do not respect the requirements of each others. Hence, it is crucial that the maintenance and production services collaborate in order to maximize the system productivity (class 2).

Maintenance strategies can be broadly classified into Corrective Maintenance (CM) and Preventive Maintenance (PM) strategies. Corrective maintenance is used to restore (repair or replace) some equipment to its required function after it has failed. This strategy leads to high levels of machine downtime (production loss) and maintenance (repair or replacement) costs due to sudden failure. The concept of PM involves the performance of maintenance activities prior to the failure of equipment. One of the main objectives of PM is to reduce the failure rate or failure frequency of the equipment. This strategy contributes to minimizing failure costs and machine downtime (production loss), and increasing product quality. The PM decision making is based on facts acquired through real data analysis. In the literature, several kinds of PM interventions are considered including time-based, condition-based, and experience-based. For more details about these PM strategies refer to [5].

Based on the degree of the information on the job data in advance, we can distinguish two kinds of scheduling problems. The first one, known as offline problem is when all job information is available at one time before scheduling. The second one referred to as online scheduling, builds the schedule of jobs on the machines as soon as they arrive, without any knowledge about the next jobs that follow [14].

In the last decade, the job scheduling under maintenance constraints was widely applied to a variety of machine models. These models can be classified into two categories: single machine and multi-machine models. The multi-machine model includes parallel machines, performing the same function, and dedicated machines being specialized of the execution of certain operations. A dedicated machine model can be flow shop, open shop, and job shop. In this paper, we deal only with scheduling parallel machines subject to availability constraints.

The remaining part of this paper is structured as follows. Section 2 introduces the parallel machine scheduling problem, its classes and the classical rules and heuristics used to solve it. The most common notations to describe the studied scheduling problems will be given in section 3. Section 4, representing the main part of this paper, includes a deep survey of the published results about the one machine as well as the parallel machine cases. The last section presents the conclusions.

II. PARALLEL MACHINE SCHEDULING

Parallel machines scheduling is at hand when machines of similar type and eventually slightly different in characteristics are available in multiple numbers. Jobs can be processed over these machines simultaneously. The Fig. 1 illustrates the Parallel machines scheduling environment.

Parallel machines can be classified into three categories; identical, uniform and unrelated parallel machines [9]. Let p_{ij} be the processing time of the job j on the machine M_i where $1 \le i \le m$ and $1 \le j \le n$.

- *Identical parallel machines*: All the parallel machines are identical in terms of their speed. Thus, each and every job will take the same amount of processing time on each machine: $p_{ij} = p_{1j}$ for all i and j.
- *Uniform parallel machines*: The parallel machines have different speeds. Consider $s_1, s_2, \ldots, s_m$ be the speeds of the machines 1, 2, …, m respectively with the relation $s_1 < s_2 < \ldots < s_m$. The processing times of the jobs are given by $p_{ij} = p_{1j}/s_i$ for all i and j.
- *Unrelated parallel machines*: The processing times p_{ij} for all i and j are arbitrary and have no special characteristics.

Classical parallel machines scheduling problems are often solved using priority rules-based algorithms and heuristics. Theses priority rules are easy to implement and their computational complexity is low. The commonly used priority rules and algorithms are:

- Shortest Processing Time rule (SPT): Jobs are arranged in ascending order of their processing times. Jobs with small values of processing times are given high priority for scheduling.
- Preemptive Shortest Processing Time rule (PSPT): This rule is a variation of SPT. At any time, the job with the minimum remaining time is to be processed first.
- Weighted Shortest Processing Time rule (WSPT): For some scheduling problems, each job j, $1 \le j \le n$ is described by its processing times p_{ij} on the machine M_i, $1 \le i \le m$ and a weight w_j indicating certain priority. According to WSPT rule, jobs are arranged in increasing order of the ratios p_{ij}/w_j.
- Earliest Due Date rule (EDD): Jobs are processed according to the increasing order of their due dates.
- Longest Processing Time rule (LPT): Jobs are arranged in decreasing order of their processing times. Jobs with large values of processing times are given high priority for scheduling.
- Critical Path rule (CP): This rule is applied when precedence relationships represented by a tree are considered between jobs. Jobs of the highest level are scheduled first.
- Largest Number of Successors rule (LNS): In the precedence graph, jobs with largest number of succors are given highest priority.
- Longest Remaining Processing Time first rule (LRPT): This rule is to be applied when job preemptions are allowed. Jobs having longest remaining processing time are scheduled first.
- Longest Remaining Processing Time first-Fastest Machine rule (LRPT-FM): This rule is to be applied when job preemptions are allowed. Jobs having longest remaining processing time are assigned to the fastest machine.
- Shortest Remaining Processing Time rule (SRPT): This rule is to be applied when job preemptions are allowed. Jobs having shortest remaining processing time are scheduled first.
- List Scheduling (LS): Given a sequence of jobs, assign them one by one according to the list. Each job is assigned to the machine where the job can be finished as early as possible.
- Best Fit Decreasing (BFD): An efficient approximation algorithm for Bin-Packing problem [48].

Many methods can be used for evaluating the performance of the approximation algorithms. A classical method was to run these algorithms on selected sample problem instances. An alternative theoretical approach is to evaluate approximation algorithms using probabilistic techniques. These methods suffer from some major drawbacks. A commonly used evaluation technique is based on bounding the worst-case behavior of a particular approximation algorithm using the best worst-case ratio r—the closest rate to 1 obtained by dividing the optimal value by the best bounded value of the approximation algorithm [27].

III. NOTATIONS

The most common notations used in the literature are summarized in table I.

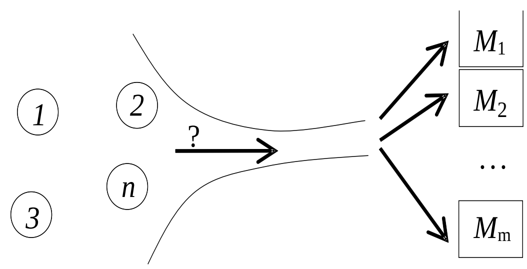

Figure 1: Parallel machines scheduling environment.

TABLE I: MOST OFTEN USED NOTATIONS IN SCHEDULING

j	Job; $1 \le j \le n$
i	Machine or resource index; $1 \le i \le m$
M_i	Machine or resource i.
M_j	Specific subset of machines that can process job j.
H	Planning horizon.
p_{ij}	Processing time of job j on machine M_i. The subscript i is dropped for single machine environment or if the job j has a constant processing time on all machines.
p_{jri}	Processing time of job j if it is scheduled in the r^{th} position after a maintenance activity on the machine M_i.
b_{ji}	Aging factor of job j scheduled on machine M_i.
r_j	Release date of the job j.
d_j	Due date of the job j.
w_j	Weight of the job j.
C_j	Completion time of the job j.
W_j	Waiting time of the job j, $W_j = C_j - p_j$
q_j	Latency duration or tail of the job j.
L_j	Lateness of the job j; $L_j = C_j - d_j$
T_j	Tardiness of the job j; $T_j = \max(C_j - d_j, 0)$
$T_{j,1}$ and $T_{j,2}$	Starting and ending dates respectively of the unavailability period on machine M_j .
U_j	$U_j = \begin{cases} 1 \text{ if } C_j > d_j \\ 0 \text{ otherwise} \end{cases}$
T_M	Time needed to perform one maintenance task.
$MS[T,T']$	Maintenance spacing: the time between any two consecutive maintenance is within the interval $[T,T'] = [T,T+\varepsilon]$
h_{jk}	k holes (unavailability periods) on machine M_j.
s_i^k	Starting time of the k^{th} unavailability period on M_i; in case there is only one hole, the superscript k is dropped.
e_i^k	Ending time of the k^{th} unavailability period on M_i; in case there is only one hole, the superscript k is dropped.
B_i^l	Beginning time of the l^{th} availability interval of M_i; the subscript i is dropped for single machine.
F_i^l	Finishing time of the l^{th} availability interval of M_i; the subscript i is dropped for single machine.
$[b_i,e_i)$	Interval in which machine M_i is planned to be shutdown
rm	Rate modifying maintenance activity
C_{max}	Makespan = $\max\{C_j, j = 1,2,\ldots,n\}$
$C^i_{\max}$	Makespan of machine M_i
$\sum_{i=1}^{m} C^i_{\max}$	Total machine load
L_{max}	Maximum lateness = $\max\{L_j, j = 1,2,\ldots,n\}$
T_{max}	Maximum tardiness = $\max\{U_j, j = 1,2,\ldots,n\}$
TC	Total completion times, $TC = \sum_{i=1}^{n} C_i$
TW	Total waiting times, $TW = \sum_{i=1}^{n} W_i$
$TADC$	Total Absolute Deviation of Completion times $TADC = \sum_{i=1}^{n}\sum_{j=i}^{n} \lvert C_i - C_j \rvert$
$TADW$	Total Absolute Deviation of Waiting times $TADW = \sum_{i=1}^{n}\sum_{j=i}^{n} \lvert W_i - W_j \rvert$
$MILP$	Mixed Integer Linear Programming
ubC	Upper bound on C_{max}

To be able to refer to the studied scheduling problems in a concise way, in the following we will use the standard three fields notation α|β|γ introduced in [26]. Each field may be a comma separated list of words.

The parameter $\alpha = \alpha_1, \alpha_2, \alpha_3, \alpha_4$ field describes the machine environment, β the job's characteristics, and γ the objective function. Since we focus our study only on parallel machines problems, we omit entries which are not relevant. The parameter $\alpha_1 \in \{1, P, Q, R\}$ where:

- 1: For parallel machine system with flexible single machine.
- P: For parallel machine system with identical machines.
- Q: For parallel machine system with uniform machines.
- R: For parallel machine system with unrelated machines.

The parameter α_2 denotes the number of machines. Different patterns of machine availability are often discussed. These are constant, zigzag, decreasing, increasing, staircase, and arbitrary. According to these cases, $\alpha_3 \in \{\phi, NC, NC_{zz}, NC_{dec}, NC_{inc}, NC_{sc}, NC_{win}\}$ denotes the machine availability [18]. In some studies, the machine availability constraint is denoted by *a* or *ma*. The parameter α_4 appears in only few studies to mention whether interrupted jobs are resumable (*rs*), semiresumable (*sr*), non resumable (*nr*), or a mixture between of them (*rs*/*nr*). These characteristics are added in some studies to the field β.

The parameter $\beta = \beta_1, \beta_2, \ldots, \beta_5$ describes the job characteristics. The parameter $\beta_1 = \phi$, β_1 = *t-pmtn, or* β_1 = *pmtn* indicates respectively nonpreemption, preemption and arbitrary preemption. An alternative notation β_1 = *r-a*, β_1 = *nr-a, or* β_1 = *sr-a* is sometimes used to denote respectively, resumable, nonresumable and semiresumable availability constraints. The parameter $\beta_2 \in \{\phi, r_j\}$ where ϕ indicates that all jobs are ready at time zero and r_j denotes jobs with different release dates. The parameter $\beta_3 \in \{\phi, d_j\}$ indicates whether the jobs come with due dates or not. The parameter $\beta_4 \in \{\phi, q_j\}$ indicates whether the jobs have tails. The parameter $\beta_5 \in \{offline, online\}$ denotes that the scheduling problem is either offline or online.

The third field denotes the optimization criterion $\gamma \in \{C_{\max}, \sum C_j, \sum w_j C_j, L_{\max}, T_{\max}, \ldots\}$

IV. SURVEY ANALYSIS

To the best of our knowledge, all the previous surveys studying the scheduling problems under machine availability didn't focus on parallel machines [14], [18], and [23]. In [23], the author classified his study into one machine, parallel machines, and flow shop problems. In each class of problems, many criteria such as the makespan and the sum of completion times were considered. The problem's classification in [14] is similar to [23] with addition of job shop and open shop cases. However, in [18] the authors structured their paper into deterministic and stochastic problems. For every problem type, the cases under resumable and nonresumable availability constraints for one machine, parallel machines, flow shop and job shop were surveyed.

Our survey analysis will be structured as follows. The first part focuses on one machine scheduling problems under availability constraints. The second part will concern the parallel machine problems. These latter will be grouped into two classes, the identical parallel machine and the uniform/unrelated parallel machines. Tables II and III summarize the criteria, the solving algorithms and methods, and their complexity/worst case ratios for the one machine and identical parallel machine scheduling problems respectively.

A. One machine

The one machine scheduling under availability constraints was of big interest in the last decade. The majority of related works considered fixed maintenance periods (frequently one) known as scheduling under machine availability problems. Few studies treated the case of jointly scheduling jobs and maintenance [38], [47], [46], and [35]. Various job/maintenance constraints were considered; resumable/non-resumable, job deterioration [28], repair rate-modifying maintenance activities [35], [47], sequence dependent setup costs [29], [31], etc. To the best of our knowledge, no study considered the online scheduling of maintenance and jobs on one machine. Table II summarizes the main results of one machine scheduling under availability constraints.

B. Identical Parallel machines

The identical parallel machine scheduling problem is the most studied among the three types of parallel machines. The special case of two machines was of bigger interest and was often solved polynomially. In [3], [7], [13], [15], and [21] the jobs are non-resumable after they have been interrupted by an availability period. Rules like LPT, LS, CLS, and approximation scheme found near optimal solutions with small worst case ratios for the problems in [7], [15], and [21]. Due to the complexity of the problem, the online scheduling or the resumable jobs constraints were rarely considered ([15], [21], and [10]).

The general problem of *m* machines ($m > 2$) is widely studied under different constraints such as resumable/non-resumable jobs, and offline/online scheduling and various objective functions including makespan, total completion times, etc ([6], [8], [11], and [12]). Results of identical machine scheduling under availability are summarized in the table III.

TABLE II: MAIN RESULTS OF ONE MACHINE SCHEDULING UNDER AVAILABILITY CONSTRAINTS

Problem	**Solving approaches**	**Worst case/competitive ratio or Complexity**	**Reference**
$1 \mid nr-PM \mid C_{\max}$	Two stage binary integer programming model	Optimal	[49]
	Two heuristics (for large-sized problems)	< 0.01	
$1 \mid p_j g_j^{[x]}, MP^{[k-1]} \mid C_{\max}$	4 polynomial algorithms	-	[47]
$1 \mid nr-a, p_j = \alpha S_j, online \mid C_{\max}$	LS-based Optimal approximation algorithm	b_1/t_0	[33]
$1 \mid nr-a, p_j = \alpha S_j \mid C_{\max}$	Modified LS-based Optimal approximation algorithm – Largest Growth Rate	-	
$1 \mid nr-a, p_j = \alpha S_j \mid \sum C_j$	Smallest Growth Rate-based heuristic algorithm	0.63	
$1 \mid h(1), N-res \mid \sum w_j C_j$	Approximation algorithm based on the knapsack problem	$2+\varepsilon$	[41]
$1 \mid nr, (s,t), p_{jr} \mid C_{\max}$	An $O(n^3)$ – heuristic algorithm	2	[39]
$1 \mid nr, (s,t), p_{jr} \mid \sum C_j$	An $O(n^3)$ – heuristic algorithm	Unbounded	
$1 \mid nr, (s,t), p_{jr}, d_j = d \mid \sum U_j$	An $O(n^3 \log n)$ – heuristic algorithm	–	
$1 \mid h(1) \mid \sum w_j C_j$	$O(n \log n), O(n^3), O(n^2)$ heuristics and Branch & Bound algorithm (B&B)	–	[42]
$1 \mid nr-pm \mid C_{\max}$	LPT	2	[37]
$1 \mid rm, r-a \mid E[C_{\max}]$	Lemma	–	[35]
$1 \mid rm, r-a \mid \sum E[C_j]$	SPT	Optimal	
$1 \mid rm, r-a \mid MaxE[L_j]$	EDD	Optimal	
$1 \mid rm, r-a \mid E[\max L_j]$	EDD	Optimal	
$1 \mid rm, nr-a \mid E[C_{\max}]$	Theorem	–	
$1 \mid rm, nr-a \mid \sum E[C_j]$	Theorem	–	
$1 \mid rm, nr-a \mid MaxE[L_j]$	Theorem	–	
$1 \mid rm, nr-a \mid E[MaxL_j]$	Theorem	–	
$1 \mid nr-a \mid T_{Max}$	Heuristic + B&B	$O(2^n \sum_{i=1}^{n} p_i)$	[36]
$1 \mid pfm \mid C_{\max}$	6 Bin packing-based heuristics	2, 3, 4	[32]
$1 \mid nr-h(1) \mid \sum w_i C_i$	Fully polynomial-time approximation algorithm	$O(2^n / \varepsilon^2)$	[40]
$1 \mid nr-fpa \mid C_{\max}$	Heuristic	2	[30]
$1 \mid r-a, p_j = \alpha_j w_j \mid C_{\max}$	0-1 integer programming algorithm	$1+\varepsilon$	[34]
$1 \mid p_{ir} = (p_i - at)r^b, ma \mid C_{\max}$	Sorting algorithm + theorem	$O(n^2 \log n)$	[38]
$1 \mid p_{ir} = (p_i - at)r^b, ma \mid \sum C_i$	Sorting algorithm + theorem	$O(n \log n)$	
$1 \mid p_{ir} = (p_i - at)r^b, ma \mid TADC$	Theorem	$O(n^2 \log n)$	
$1 \mid p_{ir} = (p_i - at)r^b, ma \mid \sum \alpha E_i + \beta T_i + \gamma d + \delta D$	Theorem	$O(n^2 \log n)$	
$1 \mid a \mid \sum U_i$	2-phase heuristic algorithm	–	[45]
$1 \mid h(1) \mid \sum C_i$	Three exact methods: (B&B), Mixed Integer Programming (MIP), and Dynamic Programming (DP)	Optimal	[44]
$1 \mid r-a \mid \sum C_i$	SPT	Optimal	[42]
$1 \mid nr-a \mid \sum C_i$	SPT	$2/7$	
	B&B and DP	Optimal	

TABLE III: MAIN RESULTS ON SCHEDULING IDENTICAL PARALLEL MACHINES UNDER AVAILABILITY CONSTRAINTS

Problem	Solving approaches	Worst case/competitive ratio (*cr*) or Complexity	Reference
$P2\|nr-a\|\sum w_jC_j$	Fully polynomial-time approximation scheme	$1+\varepsilon$	[7]
$Pm\|nr-a\|\sum w_jC_j$			
$P_{1,m},h_{k_1},h_{k_2},...,h_{k_m}\|nr-a,w_i=p_i\|\sum w_iC_i$	Polynomial-time approximation scheme	$1+\varepsilon$	[6]
$P_{0,m},h_{k_1},h_{k_2},...,h_{k_m}\|nr-a,fixed,w_i=p_i\|\sum w_iC_i$			
$P_{1,m}\|online\|C_{\max}$	Optimal algorithm	2	[8]
$Pm,k\|nr-a\|C_{\max}$	LS	$1+\frac{m-1}{m-k}$, when $k<m$	[11]
$Pm,k\|nr-a\|\sum C_i$	SPT	At most $1+\frac{m-1}{m-k}$, when $k<m$	
$P2,k\|nr-pm\|C_{\max}$	Bin-packing based heuristic	At most $\max(1.6+1.2\sigma,2)/O(n^2)$	[13]
$P2,k\|nr-jm\|\sum_{j=1}^{n}C_j$	SPT	$1+2\sigma$	
$Pm\|nr-a\|C_{\max},\min\{\max_{t\in D}\{\bar{A}_S(t)\}\}$	Ant Colony based Algorithm	$O(mn^2)$	[12]
$P2,M_1PU\|\|C_{\max}$ (PU: Periodically Unavailable)	LPT	3/2	[15]
$P2,M_1PU\|online\|C_{\max}$	LS	2	
$Pm,h_{j1}\|\|\sum C_i$	MILP, DP, and B&B	$O(nm\prod_{j=1}^{m}T_{j,1}\prod_{j=1}^{m-1}(ubC-T_{j,2}))$	[16]
$Pm,MS[T,T']\|\|C_{\max}$	$T'\neq T$: BFD-LPT	$\frac{2T'}{T}$	[17]
	$T'=T$: BFD-LPT	At most 3/2	
$Pm,NC_{win}\|pmtn,r_j,M_j\|C_{\max}$	Binary search algorithm	$O\left(((n+m)^3(n+k)^3)\log H\right)$	[19]
$Pm,r_j\|ordinal\ \ online\|C_{\max}$	Ordinal algorithms	$m=2,3,m\geq 4,cr\geq\frac{3}{2},\frac{13}{8},2-\frac{1}{m-1}$ respectively	[22]
$Pm\|nr\text{ - }a\|C_{\max}$	LPT-based Matching technique	$\frac{2m-1}{3m-2}$	[24]
$Pm,[b_i,e_i)\|\|C_{\max}$	LPT	2 (if no more than $\frac{m}{2}$ machines are simultaneously shutdown)	[25]
$Pm,NC_{inc}\|r_j,q_j\|C_{\max}$	B&B	Optimal	[20]
$P2\|nr-a,online\|C_{\max}$	Modified LS algorithm: CLS	5/2	[21]
$P2,r_j\|nr-a\|\delta_1TC+\delta_2TADC$	Assignment problem-based algorithm	$O(n^4)$	[3]
$P2,r_j\|nr-a\|\delta_1TW+\delta_2TADW$	Assignment problem-based algorithm	$O(n^4)$	
$Pm,r_j\|nr-a\|\delta_1TC+\delta_2TADC$	Theorem	$O(n^{m+3})$ if $m>2$	
$Pm,r_j\|nr-a\|\delta_1TW+\delta_2TADW$	Theorem	$O(n^{m+3})$ if $m>2$	
$P_{1,1}\|r-a,prmt\|\sum C_j/C_{\max}$	PSPT-based algorithm	$O(n\log n)$	[10]
$P_{1,1}\|r-a,prmt\|C_{\max}/\sum C_j$	PSPT-based algorithm	$O(n\log n)$	
$P2\|r-a,prmt\|C_{\max}/\sum C_j$	PSPT-based algorithm	$O(n\log n)$	

C. Uniform and Unrelated parallel machines

According to the exhaustive search we've made concerning the uniform and unrelated parallel machine scheduling under availability constraints, few recent papers were found. In [8] the authors investigated an online scheduling on two uniform parallel machines where one machine is periodically unavailable. This problem is denoted by $Q_{1,2}|online|C_{max}$ and solved by an optimal algorithm with competitive ratio $1+1/s$ ($s \geq 1$: speed of the 2nd machine). In case when $0 < s < 1$, the authors proposed some lower bounds on competitive ratio. They also studied special case where s=1 and jobs arrive in decreasing sequence and proved that LPT proposed in [15] is optimal with competitive ratio 3/2.

The unrelated parallel machine scheduling including availability constraints was studied in [1], [2], [4], and [9]. All these papers published by the same group of researchers, treat the availability constraints in different aspects. In [4] the authors investigated the parallel machine scheduling problem with aging effect and multi-maintenance activities simultaneously. They studied the following two problems $Rm \mid p_{jri}, ma \leq h_{jk} \mid \sum_{i=1}^{m} C_{\max}^{i}$ and $Rm \mid p_{jri} = p_{ij} + \alpha b_{ji}, ma \leq h_{jk} \mid \sum_{i=1}^{m} C_{\max}^{i}$ and proposed two efficient algorithms with complexity $O(n^{m+3})$. In [9] deteriorating maintenance activities were considered. The objective is to minimize the total completion time or the total machine load. They showed that both versions of the problem can be optimally solved in $O(n^{m+3})$. Advanced complexity studies related to the paper [9] were published in [1]. It was proved that the algorithm complexity remains the same no matter the processing time of a job after a maintenance activity is greater or less than its processing time before the maintenance activity. Furthermore, in [2] the authors considered simultaneously deterioration effects and deteriorating multi-maintenance activities. The objective is to find jointly the optimal maintenance frequencies, the optimal maintenance positions, and the optimal job sequences such that the total completion time is minimized. All the versions of the problem under study were solved in a polynomial time.

V. CONCLUSION

In this paper we presented a large survey on parallel machine scheduling with availability constraints. We decomposed the problems according to their machine environment; one machine, identical parallel machines, and uniform and unrelated parallel machines. The results related to the problem description, solving algorithms and their complexities for one machine and identical parallel machines were summarized in Table II and Table III. The uniform and the unrelated parallel machine scheduling problems were rarely studied and thus few results were published.

The survey analysis showed that almost all results are concentrated on the offline version of problems. Few results considered the online case. This is not always realistic in the real industry. In addition to that, the jobs are generally considered non resumable. However, this assumption needs to be taken into account for many real life problems.

Concerning the machine availability constraints, all the studied problems assume that the periods during which the machines are not available for processing jobs are fixed in time and number. But, the maintenance activities can be planned in a flexible or in a non-deterministic ways. In fact, the machines are subject to random breakdowns.

We think that considering more realistic constraints such as online scheduling, resumable jobs, and nondeterministic availability constitute interesting research directions.

REFERENCES

[1] Suh-Jeng, Chou-Jung Hsu, Dar-Li Yang, Unrelated parallel-machine scheduling with deteriorating maintenance activities, Computers & Industrial Engineering, vol. 62, pp. 1141-1143, 2012.

[2] Suh-Jeng Yang, Unrelated parallel-machine scheduling with deterioration effects and deteriorating multi-maintenance activities for minimizing the total completion time, Applied Mathematical Modelling, vol. 37-5, pp. 2995-3005, 2013.

[3] Lixin Shen, Dan Wang, Xiao-Yuan Wang, Parallel-machine scheduling with non-simultaneous machine available time, Applied Mathematical Modelling, vol. 37, pp. 5227-5232, 2013.

[4] Dar-Li Yang, T.C.E Cheng, Suh-Jenq Yang, Chou-Jung Hsu, Unrelated parallel-machine scheduling with aging effects and multi-maintenance activities, Computers & Operations Research, Vo. 39, p. 1458-1464, 2012.

[5] Rosmaini Ahmad, Shahrul Kamaruddin, An overview of time-based and condition-based maintenance in industrial application, Computers & Industrial Engineering Vol. 63, p. 135-149, 2012.

[6] Bin Fu, Yumei Huo, Hairong Zhao, Approximation schemes for parallel machine scheduling with availability constraints, Discrete Applied Mathematics, Vol. 159, p. 1555-1565, 2011.

[7] Chuanli Zhao, Min Ji, Hengyong Tang, Parallel-machine scheduling with an availability constraint, Computers & Industrial Engineering, Vol. 61-3, p. 778-781, 2011.

[8] Ming Liu, Feifeng Zheng, Chengbin Chu, Yinfeng Xu, Optimal algorithms for online scheduling on parallel machines to minimize the makespan with a periodic availability constraint, Theoretical Computer Science, Vol. 412, p. 5225-5231, 2011.

[9] T. C. E. Chang, Chou-Jung Hsu, Dar-Li Yang, Unrelated parallel-machine scheduling with deterioring maintenance activities, Computers & Industrial Engineering Vol. 60, p. 602-605, 2011.

[10] Yumei Huo, Hairong Zhao, Bicriteria scheduling concerned with makespan and total completion time subject to machine availability constraints, Theoretical Computer Science, Vol. 412, p. 1081-1091, 2011.

[11] Zhiyi Tan, Yong Chen, An Zhang, Parallel machines scheduling with machine maintenance for minsum criteria, European Journal of Operational Research, Vol. 212, p. 287-292, 2011.

[12] A. Berrichi, F. Yalaoui, L. Amodeo, M. Mezghiche, Bi-Objective Ant Colony Optimization approach to optimize production and maintenance scheduling, Computers & Operations Research, Vol. 37, p. 1584-1596, 2010.

[13] Kaibiao Sun, Hongxing Li, Scheduling problems with multiple maintenance activities and non-preemptive jobs on two identical parallel machines, International Journal of Production Economics, Vol. 124, P. 151-158, 2010.

[14] Ying Ma, Chengbin Chu, Chunrong Zuo, A survey of scheduling with deterministic machine availability constraints, Computers & Industrial Engineering, Vol 58, p. 199-211, 2010.

[15] Dehua Xu, Zhenmin Cheng, Yunqiang Yin, Hongxing Li, Makespan minimization for two parallel machines scheduling with a periodic availability constraint, Computers & Operations Research, Vol. 36, p. 1809-1812, 2009.

[16] Racem Mellouli, Cherif Sadfi, Chengbin Chu, Imed Kacem, Identical parallel-machine scheduling under availability constraints to minimize the sum of completion times, European Journal of Operational Research, Vol. 197, p. 1150-1165, 2009.

[17] Dehua Xu, Kaibiao Sun, Hongxing Li, Parallel machine scheduling with almost periodic maintenance and non-preemptive jobs to minimize makespan, Computers & operations research, Vol. 35, p. 1344-1349, 2008.

[18] Hamid Reza Dehnar Saidy, Mohammad Taghi Taghavi-Fard, Study of Scheduling Problems with machine Availability Constraint, Journal of Industrial and Systems Engineering, Vol. 1, No. 4, p. 360-383, 2008.

[19] Lu-Wen Liao, Gwo-Ji Sheen, Parallel machine scheduling with machine availability and eligibility constraints, European Journal of Operational Research, Vol. 184, p. 458-467, 2008.

[20] Anis Gharbi, Mohamed Haouari, Optimal parallel machines scheduling with availability constraints, Discrete Applied Mathematics, Vol. 148, p. 63-87, 2005.

[21] Zhiyi Tan, Yong He, Optimal online algorithm for scheduling on two identical machines with machine availability constraints, Information Processing Letters, Vol. 83, p. 323-329, 2002.

[22] Zhiyi Tan, Yong He, Ordinal Algorithm for Parallel Machine Scheduling with nonsimultaneous Machine Available Times, Computers & Mathematics with applications, Vol. 43, p. 1521-1528, 2002.

[23] Gunter Shmidt, Scheduling with limited machine availability, European Journal of Operational Research, Vol. 121, p. 1-15, 2000.

[24] Guo-Hui Lin, En-Yu Yao, Yong He, Parallel machine scheduling to maximize the minimum load with nonsimultaneous machine available times, Operations research letters, Vol. 22, p. 75-81, 1998.

[25] H.-C. Hwang, S. Y. Chang, Parallel Machines Scheduling with Machine Shutdowns, Computers & Mathematics with Applications, Vol. 36, No. 3, p. 21-31, 1998.

[26] R. L. Graham, E. L. Lawler, J. K. Lenstra, A. H. G. Ken Rinnooy, Optimization and approximation in deterministic sequencing and scheduling: A survey, Annals of Discrete Mathematics, Vol. 5, p. 287-326, 1979.

[27] M. R. Garey, R. L. Graham, D. S. Johnson, Performance Guarantees for Scheduling Algorithms, Operations research, Vol. 26, No. 1, 1978.

[28] A. Ghodratnama, M. Rabbani, R. Tavakkoli-Moghaddam, A. Baboli, Solving a single-machine scheduling problem with maintenance, job deterioration and learning effect by simulated annealing, Journal of Manufacturing Systems, Vol. 29, p. 1-9, 2010.

[29] Francisco Angel-Bello, Ada Alvarez, Joaquin Pacheco, Iris Martinez, A heuristic approach for a scheduling problem with periodic maintenance and sequence-dependant setup times, Computers and Mathematics with Applications, Vol. 61, p. 797-808, 2011.

[30] Dehua Xu, Yunqiang Yin, Hongxing Li, A note on "scheduling of nonresumable jobs and flexible maintenance activities on a single machine to minimize makespan, European Journal of Operational Research, Vol. 197, p. 825-827, 2009.

[31] Francisco Angel-Bello, Ada Alvarez, Joaquin Pacheco, Iris Martinez, A single machine scheduling problem with availability constraints and sequence-dependant setup costs, Applied Mathematical Modelling, Vol. 35, p. 2041-2050, 2011.

[32] Chinyao Low, Min Ji, Chou-Jung Hsu, Chwen-Tzeng Su, Minimizing the makespan in a single machine scheduling problems with flexible and periodic maintenance, Applied Mathematical Modelling, Vol. 34, p. 334-342, 2010.

[33] Min Ji, Yong He, T.C.E. Cheng, Scheduling linear deterioring jobs with an availability constraint on a single machine, Theoretical Computer Science, Vol. 362, p. 115-126, 2006.

[34] Min Ji, T.C.E Cheng, Scheduling resumable simple linear deteriorating jobs on a single machine with an availability constraint to minimize makespan, Computers & Industrial Engineering, Vol. 59, p. 794-798, 2010.

[35] Chung-Yee Lee, Chen-Sin Lin, Single-machine scheduling with maintenance and repair rate-modifying activities, European Journal of Operational Research, Vol. 135, p. 493-513, 2001.

[36] C.J. Liao, W.J. Chen, Single-machine scheduling with periodic maintenance and nonresumable jobs, Computers & Operations Research, Vol. 30, p. 1335-1347, 2003.

[37] Min Ji, Yong He, T.C.E. Cheng, Single-machine scheduling with periodic maintenance to minimize makespan, Computers & Operations Research, Vol. 34, p. 1764-1770, 2007.

[38] Suh-Jenq Yang, Single-machine scheduling problems with both start-time dependent learning and position dependent aging effects under deterioring maintenance consideration, Applied Mathematics and Computation, Vol. 217, p. 3321-3329, 2010.

[39] Baruch Mor, Gur Mosheiov, Heuristics for scheduling problems with an unavailability constraint and position-dependent processing times, Computers & Industrial Engineering, Vol. 62, p. 908-916, 2012.

[40] Imed Kacem, A. Ridha Mahjoub, Fully polynomial time approximation scheme for the weighted flow-time minimization on a single machine with a fixed non-availability interval, Computers & Industrial Engineering, Vol. 56, p. 1708-1712, 2009.

[41] Hans Kellerer, Mikhail A. Kubzin, Vitaly A. Strusevich, Two simple constant ratio approximation algorithms for minimizing the total weighted completion time on a single machine with a fixed non-availability interval, European Journal of Operational Research, Vol. 199, p. 111-116, 2009.

[42] Shan-lin, Ying Ma, Dong-ling Xu, Jian-bo Yang, Minimizing total completion time on a single machine with flexible maintenance activity, Computers & Operations Research, Vol. 38, p. 755-770, 2011.

[43] Imed Kacem, Chengbin Chu, Efficient branch-and-bound algorithm for minimizing the weighted sum of completion times on a single machine with one availability constraint, International Journal of Production Economics, Vol. 112, p. 138-150, 2008.

[44] Imed Kacem, Chengbin Chu, Ahmed Souissi, Single-machine scheduling with an availability constraint to minimize the weighted sum of the completion times, Computers & Operations Research, Vol. 35, p. 827-844, 2008.

[45] Ju-Yong Lee, Yeong-Dae Kim, Minimizing the number of tardy jobs in a single-machine scheduling problem with periodic maintenance, Computers & Operations Research, Vol. 39, p. 2196-2205, 2012.

[46] Mohamed-Chahir Fitouhi, Mustapha Nourelfath, Integrating noncyclical preventive maintenance scheduling and production planning for a single machine, International Journal of Production Economics, Vol. 136, p. 344–351, 2012.

[47] Kabir Rustogi, Vitaly A. Strusevich, Single machine scheduling with general positional deterioration and rate-modifying maintenance, Omega, Vol. 40, p. 791–804, 2012.

[48] Coffman EG, Garey MR, Johnson DS, Approximation algorithms for bin packing: a survey. In: Hochbaum DS, editor. Approximation algorithms for NP-hard problems. Boston: PWS Publishing Company, p. 46-93, 1997.

[49] Chou-Jung Hsu, Chinyao Low, Chwen-Tzeng Su, A single-machine scheduling problem with maintenance activities to minimize makespan, Applied Mathematics and Computation, Vol. 215, p. 3929-3935, 2010.