https://doi.org/10.1016/j.ijpe.2014.09.038

Scheduling with Processing Set Restrictions: A Literature Update

Joseph Y.-T. Leung[a,b,c] and Chung-Lun Li[d]

[a] School of Management, Hefei University of Technology, Hefei 230009, PR China

[b] Key Laboratory of Process Optimization and Intelligent Decision-making, Ministry of Education, Hefei 230009, PR China

[c] Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102, USA

[d] Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

E-mail addresses: joseph.y.leung@njit.edu (J.Y.-T. Leung), chung-lun.li@polyu.edu.hk (C.-L. Li)

November 11, 2015
Revised January 15, 2016

Abstract

In 2008, we published a survey paper on machine scheduling with processing set restrictions [Leung, J.Y.-T., Li, C.-L., 2008. Scheduling with processing set restrictions: A survey. International Journal of Production Economics 116 (2), 251–262]. Since the appearance of that survey paper, there has been a significant increase in interest in this field. In this paper, we provide an expository update of this line of research. Our survey covers five types of processing set restrictions, namely inclusive processing sets, nested processing sets, interval processing sets, tree-hierarchical processing sets, and arbitrary processing sets, and it covers both offline and online problems. While our main focus is on scheduling models with a makespan objective, other performance criteria are also discussed.

Keywords: Scheduling; parallel machines; processing set restrictions; computational complexity

1 Introduction

In 2008, we published the survey paper "Scheduling with Processing Set Restrictions: A Survey" (Leung and Li 2008). Since then, there has been a significant increase in interest in machine scheduling problems with processing set restrictions. In this paper, we provide an expository update of this line of research. Note that some of the works covered in this survey have been included in the paper by Lim (2010), who has conducted a survey on scheduling problems with inclusive processing set restrictions, and in the papers by Lee *et al.* (2010, 2013), who have conducted surveys on online algorithms for scheduling problems with processing set restrictions.

1.1 Definitions

As recalled from Leung and Li (2008), the problem concerned can be stated as follows: We are given a set of n jobs $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$ and a set of m parallel machines $\mathcal{M} = \{M_1, M_2, \ldots, M_m\}$. Each job J_j has a processing time p_j and a set of machines $\mathcal{M}_j \subseteq \mathcal{M}$ to which it can be assigned. We aim to find a schedule such that each job J_j is assigned to one of the machines in $\mathcal{M}_j$ and that the makespan $C_{\max}$ is minimized. This problem is denoted as $P|\mathcal{M}_j|C_{\max}$ if the machines are identical, $Q|\mathcal{M}_j|C_{\max}$ if the machines are uniform, and $R|\mathcal{M}_j|C_{\max}$ if the machines are unrelated. These three problems are denoted as $Pm|\mathcal{M}_j|C_{\max}$, $Qm|\mathcal{M}_j|C_{\max}$, and $Rm|\mathcal{M}_j|C_{\max}$, respectively, if the number of machines, m, is fixed. Note that $R|\mathcal{M}_j|C_{\max}$ is equivalent to the classical unrelated parallel machine problem $R||C_{\max}$ (see Leung and Li 2008). Because of this equivalence, we will not review any work about $R|\mathcal{M}_j|C_{\max}$.

There are several important special forms of processing set restrictions, namely the *inclusive* processing set, *nested* processing set, *interval* processing set, and *tree-hierarchical* processing set restrictions. The inclusive processing set restriction has the property that for each pair $\mathcal{M}_j$ and $\mathcal{M}_k$, either $\mathcal{M}_j \subseteq \mathcal{M}_k$ or $\mathcal{M}_k \subseteq \mathcal{M}_j$. The nested processing set restriction has the property that for each pair $\mathcal{M}_j$ and $\mathcal{M}_k$, either $\mathcal{M}_j \cap \mathcal{M}_k = \emptyset$, $\mathcal{M}_j \subseteq \mathcal{M}_k$, or $\mathcal{M}_k \subseteq \mathcal{M}_j$. The interval processing set restriction has the property that for any job J_j, $\mathcal{M}_j = \{M_{a_j}, M_{a_j+1}, \ldots, M_{b_j}\}$ for some $1 \leq a_j \leq b_j \leq m$. The tree-hierarchical processing set restriction has the property that each machine M_i is represented by a node of a tree, and that the processing set of a job J_j is the set of machines consisting of its associated node, say M_{a_j}, and all the nodes on the unique path from M_{a_j} to the root of the tree. We denote a problem with such a special form of processing set restriction as $\alpha|\mathcal{M}_j(\delta),\beta|\gamma$, where α is the machine environment ($\alpha \in \{P, Q\}$), δ is the processing set restriction ($\delta \in \{inclusive, nested, interval, tree\}$), γ is the scheduling criterion, and β states the special processing characteristics (if any). While the main focus of this survey is on scheduling models with a makespan minimization objective (i.e., $\gamma = C_{\max}$), other performance criteria are also discussed. We let $\sum C_j$, $\sum w_jC_j$, $\sum w_jT_j$, $\sum U_j$, $L_{\max}$, and $C_{\min}$ denote the objectives of minimizing the sum of job completion times, minimizing the weighted sum of job completion times, minimizing the total weighted tardiness, minimizing the number of tardy jobs, minimizing the maximum lateness, and maximizing the minimum machine load (i.e., minimum completion time of all the machines), respectively. Special processing characteristics β include, but not limited to, job release dates (r_j), identical job processing times ($p_j = p$), unit processing times ($p_j = 1$), fractional assignment or job splitting (*split*), and job preemption (*pmtn*).

Note that $P|\mathcal{M}_j(inclusive)|C_{\max}$ and $Q|\mathcal{M}_j(inclusive)|C_{\max}$ are NP-hard in the strong sense (see, e.g., Leung and Li 2008). Note also that $P|\mathcal{M}_j(nested)|C_{\max}$, $P|\mathcal{M}_j(interval)|C_{\max}$, and $P|\mathcal{M}_j(tree)|C_{\max}$ are generalizations of $P|\mathcal{M}_j(inclusive)|C_{\max}$ but are special cases of $P|\mathcal{M}_j|C_{\max}$. Similarly, $Q|\mathcal{M}_j(nested)|C_{\max}$, $Q|\mathcal{M}_j(interval)|C_{\max}$, and $Q|\mathcal{M}_j(tree)|C_{\max}$ are generalizations of $Q|\mathcal{M}_j(inclusive)|C_{\max}$ but are special cases of $Q|\mathcal{M}_j|C_{\max}$. We refer to the most general form of processing set restrictions, where $\mathcal{M}_j$ can be an arbitrary subset of $\mathcal{M}$, as *arbitrary* processing set.

In this survey, our emphasis is on optimal and approximation solution methods for these scheduling models, and we consider both online and offline models. An underlying assumption of an offline scheduling model is that all the input data are known in advance. In an online scheduling model, the scheduler obtains the job data piece by piece and has to make decision with only a partial knowledge of the input. There are two versions of online algorithms: "online over list" and "online over time." In the former case, all the jobs arrive at time zero, but the jobs are given to the scheduler one at a time. The scheduler has to schedule the job as soon as it is released, and the scheduling decision cannot be revoked. In the latter case, the jobs arrive at different times and the scheduler has no information about the release times of the jobs. When a job J_j is released at r_j, the scheduler can schedule the job or delay scheduling the job.

Worst-case error bounds are often used for measuring the performance of an offline approximation algorithm. For a given instance $\mathcal{I}$ of an offline problem and an approximation algorithm A, let $A(\mathcal{I})$ and $OPT(\mathcal{I})$ denote the objective value obtained by algorithm A and an optimal algorithm, respectively, when applied to $\mathcal{I}$. For a minimization (respectively maximization) problem, we say that algorithm A has an *absolute worst-case error bound* (or simply *worst-case bound*) q if $A(\mathcal{I})/OPT(\mathcal{I}) \leq q$ (respectively $OPT(\mathcal{I})/A(\mathcal{I}) \leq q$) for all $\mathcal{I}$, and we say that algorithm A has an *asymptotic worst-case error bound* q if there exists $N > 0$ such that $A(\mathcal{I})/OPT(\mathcal{I}) \leq q$ (respectively $OPT(\mathcal{I})/A(\mathcal{I}) \leq q$) for all $\mathcal{I}$ satisfying $OPT(\mathcal{I}) \geq N$. A *polynomial time approximation scheme* (PTAS) for a minimization (respectively maximization) problem with input size N is an approximation algorithm which takes an accuracy requirement $\epsilon > 0$ as input and produces a solution with a running time polynomial in N for any fixed ϵ, such that $A(\mathcal{I})/OPT(\mathcal{I}) \leq 1 + \epsilon$ (respectively $OPT(\mathcal{I})/A(\mathcal{I}) \leq 1+\epsilon$). An *efficient PTAS* (EPTAS) is a PTAS with an $O(f(1/\epsilon)N^c)$ running time, where $f(\cdot)$ is an arbitrary function and c is a constant independent of ϵ. If $f(\cdot)$ is a polynomial function, then the EPTAS is called a *fully polynomial time approximation scheme* (FPTAS).

Competitive ratios are often used for measuring the performance of an online algorithm. An online algorithm A for a minimization (respectively maximization) problem is said to be *c-competitive* if $A(\mathcal{I}) \leq c \cdot OPT(\mathcal{I})$ (respectively $OPT(\mathcal{I}) \leq c \cdot A(\mathcal{I})$) for all instances $\mathcal{I}$ of the problem, where $OPT(\mathcal{I})$ is the objective value obtained by an optimal offline algorithm. If algorithm A is c-competitive, then we say that it has a competitive ratio of c (Leung and Li 2008). Thus, an online algorithm A is 1-competitive, or, equivalently, has a competitive ratio of 1, if it produces a solution whose objective value is exactly the same as that of an optimal offline algorithm.

1.2 Examples of applications

Problem $P|\mathcal{M}_j(inclusive)|C_{\max}$ has many applications. For example, Ou *et al.* (2008) describe an application in scheduling vessel-loading cranes where multiple cranes with different weight capacity limits are working in parallel. Machines with inclusive processing set restrictions are also known as machines with a "linear hierarchy" (see Bar-Noy *et al.* 2001) as well as machines with "Grade of Service" (GoS) provision (see Hwang *et al.* 2004). Hwang *et al.* (2004) describe an application in the service industry in which a service provider has customers categorized as platinum, gold, silver, and regular members. Those "special members" are entitled to premium services. In order to provide premium services to "special members", servers (i.e., machines) and customers (i.e., jobs) are labeled with GoS levels, and a customer is allowed to be served by a server only when the GoS level of the customer is not lower than the GoS level of the server. In this paper, we will use GoS provision and inclusive processing set restrictions interchangeably.

An example of the nested processing set model is given as follows: Consider the Operations Management and Information Systems department of a university. Each undergraduate of this department is majoring in either Operations Management (OM) or Information Systems (IS). Every year the department has to assign students to different academic advisors, where an academic advisor is a faculty member in either the OM discipline or the IS discipline. Each of these two disciplines has a faculty member serving as a discipline coordinator. Each student has to be assigned to one faculty member, and each faculty member can serve as academic advisors of multiple students. Academic advisors are assigned to students according to the following rules: (i) A first-year student who is not on academic probation can be assigned to any faculty member in the department; (ii) a second-, third-, or fourth-year student must be assigned to a faculty member in the student's discipline; (iii) a student who is on academic probation must be assigned to the discipline coordinator of the student's discipline. Assigning a student to a faculty member will bring some work to the faculty member, and the objective is to minimize the maximum workload among all faculty members in the department. In this example, the processing sets (i.e., sets of eligible academic advisors for the students) has a nested structure.

The nested processing set restriction is a special case of the interval processing set restriction (see, e.g., Lee *et al.* 2013). Hence, the above example is also an example of the interval processing set model. Suppose that there are u faculty members $M_1, M_2, \ldots, M_u$ in the OM discipline and v faculty members $M_{u+1}, M_{u+2}, \ldots, M_{u+v}$ in the IS discipline, where M_1 is the OM discipline coordinator and M_{u+1} is the IS discipline coordinator. Then, the processing set of a (non-probationary) first-year student is $\{M_1, M_2, \ldots, M_{u+v}\}$; the processing set of a (non-probationary) second-, third-, or fourth-year OM major is $\{M_1, M_2, \ldots, M_u\}$; the processing set of a (non-probationary) second-, third-, or fourth-year IS major is $\{M_{u+1}, M_{u+2}, \ldots, M_{u+v}\}$; the processing set of an OM major who is on academic probation is $\{M_1\}$; and the processing set of an IS major who is on academic probation is $\{M_{u+1}\}$.

An example of the tree-hierarchical processing set model is given as follows: Consider the purchasing office of a university in which a team of administrators whose daily tasks are to process purchase requisitions submitted by various departments. There are two divisions in the purchasing office. One division supports the procurement activities of academic departments, and the other division supports the administrative departments. Each division has a supervisor and a group of clerical staff, and the head of the purchasing office oversees both divisions. Each clerical staff in a division is designated to a specific category of purchases such as equipment, office furniture, stationeries, etc. If a purchase requisition involves a monetary amount no more than a threshold $\$X$, then it can be handled by any clerical staff member in the division, the supervisor of the division, or the head of the purchasing office. If a purchase requisition involves an amount higher than $\$X$ but no more than $\$Y$, where $Y > X$, then it must be handled by the supervisor of the division or the head of the purchasing office. A purchase requisition involving an amount higher than $\$Y$ can only be handled by the head of the purchasing office, regardless of whether it comes from an academic department or an administrative department. Assigning a purchase requisition to a person will increase the workload of that person, and the objective is to minimize the maximum workload among all staff in the purchasing office. In this example, the "machines" have a tree-hierarchical structure. The head of the purchasing office is the root of the tree. The supervisors of the two divisions are directly underneath the root. The clerical staff members in each division are directly under the supervisor of the division.

1.3 Outline of the paper

In Section 2 we review the works on inclusive processing sets. In Section 3, we review the works on nested, interval, tree-hierarchical, and arbitrary processing sets. In these sections, unless stated otherwise, the scheduling objective is always makespan minimization. In Section 4, we review some related works, including models with batch processing, models with resource constraints, etc. We conclude the paper by providing some suggested future research directions in Section 5.

2 Inclusive Processing Sets

In this section we review the recent works on scheduling with inclusive processing set restrictions. We divide our discussions into offline and online models.

2.1 Offline models

We first consider the problem $P|\mathcal{M}_j(inclusive)|C_{\max}$. As stated in Leung and Li (2008), a polynomial-time algorithm with an absolute worst-case error bound of 4/3 has been developed by Ou *et al.* (2008). A PTAS has also been given by Ou *et al.* (2008), which has been extended to include release dates by Li and Wang (2010).

Theorem 1 (Li and Wang 2010). *There exists a PTAS for* $P|\mathcal{M}_j(inclusive), r_j|C_{\max}$.

Li *et al.* (2012) give an EPTAS for the special case where the GoS level of the machines is either 1 or 2. Several FPTASs are known for the case where the number of machines m is fixed; see Woeginger (2009) and Li *et al.* (2012). Epstein and Levin (2011) study the "speed hierarchical model," where the machines are uniformly related machines and each job has a minimum speed level necessary for a machine to process the job. They develop a PTAS for this problem.

Theorem 2 (Epstein and Levin 2011). *There exists a PTAS for* $Q|\mathcal{M}_j(inclusive)|C_{\max}$ *when each job has a minimum speed level necessary for a machine to process the job.*

For a set of equal processing time jobs, the problem can be solved in polynomial time even when there are release dates. Li and Li (2015) present an $O(n^2 + mn \log n)$ time algorithm for $P|\mathcal{M}_j(inclusive), r_j, p_j = p|C_{\max}$ (i.e., problem $P|\mathcal{M}_j(inclusive)|C_{\max}$ with job release dates and equal processing times). For uniform machines, they show that $Q|\mathcal{M}_j(inclusive), r_j, p_j = p|C_{\max}$ can be solved by an $O(mn^2 \log m)$ time algorithm. Recently, Li and Lee (2016) develop a modified algorithm for $P|\mathcal{M}_j(inclusive), r_j, p_j = p|C_{\max}$ with an improved running time of $O(\min\{m, \log n\} n \log n)$.

Theorem 3 (Li and Li 2015; Li and Lee 2016). *$P|\mathcal{M}_j(inclusive), r_j, p_j = p|C_{\max}$ can be solved in $O(\min\{m, \log n\} n \log n)$ time, whereas $Q|\mathcal{M}_j(inclusive), r_j, p_j = p|C_{\max}$ can be solved in $O(mn^2 \log m)$ time.*

The preemptive case can also be solved in polynomial time. Huo *et al.* (2009a) provide an $O(nk \log P + mnk^2 + m^3k)$ time algorithm for $P|\mathcal{M}_j(inclusive), r_j, pmtn|C_{\max}$, where k is the number of distinct release dates and P is the total processing time of all the jobs. They show that there is no 1-competitive online algorithm for this problem.

Theorem 4 (Huo *et al.* 2009a). *$P|\mathcal{M}_j(inclusive), r_j, pmtn|C_{\max}$ can be solved in $O(nk \log P + mnk^2 + m^3k)$ time, where k is the number of distinct release dates and $P = \sum_{j=1}^{n} p_j$.*

In some scheduling models, the objective is to maximize the minimum completion time of all the machines. Li *et al.* (2009) present a PTAS with running time of $O(mn^{O(1/\epsilon^2)})$ for problem $P|\mathcal{M}_j(inclusive)|C_{\min}$. When the number of GoS level is bounded above by a fixed constant k, their PTAS has a running time $O(n)$. When the number of machines is fixed, they give an FPTAS with running time $O(n)$. Table 1 summarizes the major results for the offline models with inclusive processing set restrictions.

2.2 Online models

As mentioned in Section 1.1, there are two versions of online algorithms, namely online over list and online over time. In the following, we will review online scheduling over list, unless stated otherwise.

As mentioned in Leung and Li (2008), Bar-Noy *et al.* (2001) have given an online algorithm with a competitive ratio of $e + 1 \approx 3.718$ for problem $P|\mathcal{M}_j(inclusive)|C_{\max}$. Tan and Zhang (2011) provide improved competitive ratios of 2.333 and 2.610 for the 4-machine and 5-machine cases, respectively. Lim *et al.* (2011) further improve the competitive ratios to 2.294 and 2.501 for these two cases.

Theorem 5 (Bar-Noy *et al.* 2001; Lim *et al.* 2011). *There exist online (over list) algorithms for* $P|\mathcal{M}_j(inclusive)|C_{\max}$*,* $P4|\mathcal{M}_j(inclusive)|C_{\max}$*, and* $P5|\mathcal{M}_j(inclusive)|C_{\max}$ *with competitive ratios* 3.718*,* 2.294*, and* 2.501*, respectively.*

Recently, Zhang (2015) studies the fractional assignment model $P|\mathcal{M}_j(inclusive), split|C_{\max}$, where each job can be arbitrarily split between the machines, and the split jobs can be processed, possibly in parallel, by different machines. He provides an optimal online algorithm for this problem based on the solution of linear programming.

Zhang *et al.* (2009) examine online scheduling of problem $P|\mathcal{M}_j(inclusive)|C_{\max}$ with two GoS levels. They present an online algorithm with a competitive ratio of $1 + \frac{m^2-m}{m^2-km+k^2} < \frac{7}{3}$, where k is the number of machines that can process all the jobs and $m-k$ is the number of machines that can only process a subset of jobs.

Lee *et al.* (2011b) study online scheduling over time for the two-machine problem $P2|\mathcal{M}_j(inclusive), r_j, p_j = p|C_{\max}$ in which the jobs have release dates and the job processing times are identical. They present an optimal online algorithm with a competitive ratio of $\sqrt{2}$. (An online algorithm is optimal if its competitive ratio matches the lower bound of the problem.) Xu and Liu (2015) give an optimal online algorithm with a competitive ratio of $\sqrt{2}$ for a more general problem $P|\mathcal{M}_j(inclusive), r_j, p_j = p|C_{\max}$ which has an arbitrary number of machines.

Theorem 6 (Xu and Liu 2015). *There exists an optimal online (over time) algorithm for* $P|\mathcal{M}_j(inclusive), r_j, p_j = p|C_{\max}$ *with a competitive ratio of* $\sqrt{2}$*.*

Shabtay and Karhi (2012a) consider the two-machine problem $P2|\mathcal{M}_j(inclusive), p_j = 1|\sum C_j$, where all jobs have unit processing time and the objective is to minimize the total job completion time. They present an online algorithm with a competitive ratio of $\rho_{LB} + O(\frac{1}{n})$, where ρ_{LB} is a lower bound on the competitive ratio of any online algorithm. The lower bound ρ_{LB} is given by $1 + \left(\frac{-\alpha+\sqrt{4\alpha^3-\alpha^2+2\alpha-1}}{2\alpha^2+1}\right)^2$, where $\alpha = \frac{1}{3} + \frac{1}{6}\left(116 - 6\sqrt{78}\right)^{1/3} + \frac{(58+3\sqrt{78})^{1/3}}{3(2)^{2/3}} \approx 1.918$. Since the competitive ratio differs from the lower bound by $O(\frac{1}{n})$, it approaches the lower bound arbitrarily closely when n gets arbitrarily large. Thus, their algorithm can be regarded as asymptotically optimal.

Hou and Kang (2012) consider online scheduling on uniform machines. There are k machines with speed s that can schedule all the jobs, and $m-k$ machines with speed 1 that can schedule only a subset of jobs. They present an online algorithm with the following competitive ratio. Let $s_1 \in (0,1)$ be the real root of the equation $k^2s^3+k(2m-2k-1)s^2+(m-k)(m-2k)s-(m-k)^2=0$ and s_2 be the positive root of the equation $ks^2-(2k-1)s-(m-k)=0$. When $0<s<1$, the competitive ratio is

$$\begin{cases} \max\left\{1+\frac{k-1}{k}, 1+\frac{ks+m-k-1}{m-k}\right\}, & \text{if } 0<s\le s_1; \\ 1+\frac{(m-1)(ks+m-k)}{k^2s^2+k(m-k)s+(m-k)^2}, & \text{if } s_1<s<1. \end{cases}$$

When $s \geq 1$, the competitive ratio is

$$\begin{cases} 1+\frac{k(k-1)s^2+(m-k)(m+k-1)s}{k^2s^2+k(m-k)s+(m-k)^2}, & \text{if } 1\le s<s_2; \\ 1+\frac{(k-1)s+m-k}{ks}, & \text{if } s\ge s_2. \end{cases}$$

Liu *et al.* (2009) study online scheduling on two uniform machines; that is, the problem $Q2|\mathcal{M}_j(inclusive)|C_{\max}$. They assume that M_1 has speed 1 and can process any job, while M_2 has speed $s>0$ and can only process a subset of jobs. They derive a lower bound on the competitive ratio, and propose and analyze two online algorithms. Lee *et al.* (2009c) point out an error in Liu *et al.* (2009) and propose several online algorithms for the problem. They derive the following competitive ratios for these algorithms: Let $s_1=\frac{\sqrt{5}-1}{2}$ and s_2 be the solution of the equation $s^3-s-1=0$ for $1.3<s<1.4$. When $s\in(0,s_1]$, the High Speed Machine First (HSF) algorithm has a competitive ratio of $1+s$, which matches its lower bound (i.e., the algorithm is optimal). When $s\in[s_2,\infty)$, HSF is also optimal and has a competitive ratio of $1+\frac{1}{s}$. When $s\in(s_1,1)$, they propose a "Modified ONLINE1" algorithm and show that it has a competitive ratio of $1+\frac{2s}{s^2+s+1}$, which does not match its lower bound. When $s\in(1,s_2)$, they propose a "Modified ONLINE2" algorithm and show that it has a competitive ratio of $1+\frac{s^2+s}{s^2+s+1}$, which also does not match its lower bound.

Tan and Zhang (2010) consider the same problem as in Lee *et al.* (2009c). They assume that M_1 has speed $s>0$ and can process any job, while M_2 has speed 1 and can only process a subset of jobs. For $s<1$, they propose an optimal algorithm with a competitive ratio of $\min\{1+s, 1+\frac{1+s}{1+s+s^2}\}$. For

$s > 1$, they propose another optimal algorithm with a competitive ratio of $\min\{\frac{1+s}{s}, 1 + \frac{2s}{1+s+s^2}\}$.

Chassid and Epstein (2008) study the online problem $Q2|\mathcal{M}_j(inclusive)|C_{\min}$. They assume that the first machine has speed $s > 0$ that can process all the jobs, while the second machine has speed 1 that can only process a subset of jobs. They show that no online algorithm with a constant competitive ratio exists for this problem. They then design an optimal online algorithm with competitive ratio $\frac{2s+1}{s+1}$ for the fractional assignment model $Q2|\mathcal{M}_j(inclusive), split|C_{\min}$.

Hou and Kang (2011) consider online scheduling on uniform machines with two GoS levels. There are k machines with speed s that can process all the jobs, and there are $m - k$ machines with speed 1 that can only process a subset of jobs. Both the $C_{\max}$ and $C_{\min}$ objectives are considered. For the $C_{\min}$ objective, they show that no online algorithm can achieve bounded competitive ratio. They also consider the fractional assignment model, for which they propose an optimal online algorithm with a competitive ratio of $\frac{2ks+m-k}{ks+m-k}$ for the $C_{\min}$ objective, and an optimal online algorithm with a competitive ratio of $\frac{(ks+m-k)^2}{k^2s^2+ks(m-k)+(m-k)^2}$ for the $C_{\max}$ objective. Table 2 summarizes the major results for the online models with inclusive processing set restrictions.

Studies have also been conducted on semi-online scheduling problems with inclusive processing set restrictions, where some partial information about the jobs such as sum of processing requirements of all jobs, bounds on job processing times, etc., are known to the scheduler. These works include Chassid and Epstein (2008), Hou and Kang (2011), Liu *et al.* (2011), Wu *et al.* (2012a,b), Chen *et al.* (2013), Lu and Liu (2013), Lee *et al.* (2014), Luo and Xu (2014a,b, 2016), Luo *et al.* (2014), Wu *et al.* (2014), Chen *et al.* (2015), Lu and Liu (2015), and Zhang *et al.* (2015), who consider various models with different partial information given to the schedulers.

Wang and Xing (2010) study an "online over list" scheduling model with inclusive processing set restrictions where the definition of online scheduling is different from what is defined above. They consider a problem with two GoS levels and derive competitive ratios for several service policies. There are ordinary and special jobs, and there are dedicated and flexible machines. Special jobs can only be processed by flexible machines, while ordinary jobs can be processed by any machine. Unlike the definition of "online over list" defined above, their service policies assume that the scheduler may select a special job waiting in the list and schedule it before some ordinary jobs in front of it. Wang *et al.* (2009) study the "online over time" version of Wang and Xing's (2010) problem where each job has a given release date.

3 Nested, Interval, Tree-Hierarchical, and Arbitrary Processing Sets

In this section, we review the works on nested, interval, tree-hierarchical, and arbitrary processing sets. We divide our discussions of these four types of processing set restrictions into four subsections. Major results for the offline and online models are summarized in Table 3 and Table 4, respectively.

3.1 Nested processing sets

As mentioned in Leung and Li (2008), Glass and Kellerer (2007) have provided a simple polynomial-time algorithm for $P|\mathcal{M}_j(nested)|C_{\max}$ with an absolute worst-case error bound of $2 - 1/m$. Huo and Leung (2010b) present a polynomial time algorithm with an improved absolute worst-case error bound of 7/4. For two- and three-machine cases, their algorithm offers a better worst-case error bound of 5/4 and 3/2, respectively. Huo and Leung (2010a) present a better algorithm with a worst-case error bound of 5/3. Muratore *et al.* (2010) and Epstein and Levin (2011) develop PTASs independently for this problem.

Theorem 7 (Muratore *et al.* 2010; Epstein and Levin 2011). *There exists a PTAS for* $P|\mathcal{M}_j(nested)|C_{\max}$.

Biró and McDermid (2014) study the problem $P|\mathcal{M}_j(nested), p_j \in \{1, 2, 4, \ldots, 2^k\}|C_{\max}$. They show that this problem can be solved optimally in $O(ke \log n)$ time, where $e = \sum_{j=1}^{n} |\mathcal{M}_j|$.

Huo *et al.* (2009b) consider the problem $P|\mathcal{M}_j(nested), pmtn|C_{\max}$, where job preemption is permitted. They propose an $O(n \log n)$ time algorithm to find an optimal schedule. They also present an $O(mn + n \log n)$ time algorithm to find a *maximal* schedule, where a schedule is said to be maximal if it processes as much work as any other schedule in any time interval $[0, t]$, $t > 0$.

Theorem 8 (Huo *et al.* 2009b). $P|\mathcal{M}_j(nested), pmtn|C_{\max}$ *can be solved in* $O(n \log n)$ *time.*

Lim *et al.* (2011) consider the online version of problem $P|\mathcal{M}_j(nested)|C_{\max}$. They present several lower bounds on the competitive ratio for problems with different number of machines. Lee *et al.* (2011b) study online scheduling over time and point out that the Least Flexible Job first (LFJ) algorithm, which processes jobs in nondecreasing order of the cardinality of their processing sets, is 1-competitive for the problem $P|\mathcal{M}_j(nested), r_j, p_j = 1|C_{\max}$. Xu and Liu (2015) study online scheduling over time for the more general problem $P|\mathcal{M}_j(nested), r_j, p_j = p|C_{\max}$ and provide an optimal online algorithm with a competitive ratio of $\frac{\sqrt{5}+1}{2}$.

Theorem 9 (Lee *et al.* 2011b; Xu and Liu 2015). *There exist online (over time) algorithms for* $P|\mathcal{M}_j(nested), r_j, p_j = 1|C_{\max}$ *and* $P|\mathcal{M}_j(nested), r_j, p_j = p|C_{\max}$ *with competitive ratios 1 and* $\frac{\sqrt{5}+1}{2}$*, respectively.*

3.2 Interval processing sets

Shabtay and Karhi (2012b) study a special case of $P|\mathcal{M}_j(interval), p_j = 1|C_{\max}$ with two job types. The processing set of the first job type is $\{M_1, M_2, \ldots, M_k\}$, while the processing set of the second job type is $\{M_{s+1}, M_{s+2}, \ldots, M_m\}$, where $1 \leq s \leq k \leq m$. They provide a linear-time offline algorithm for constructing an optimal schedule. They also give an optimal online algorithm with a competitive ratio of $\frac{mk}{(m-s)k+s^2}$. This online algorithm becomes 1-competitive when $k = s$ and becomes 4/3-competitive when $k = m = 2s$. Karhi and Shabtay (2013) show that the online algorithm of Shabtay and Karhi is also optimal for two other special cases, namely the case where the processing times are job type dependent and the case where the processing times are machine set dependent.

Karhi and Shabtay (2014) study the same problem as in Shabtay and Karhi (2012b), except that the processing times of the jobs are arbitrary. They give an online algorithm with competitive ratio of $1 + \frac{k(m-1)}{k(m-s)+s^2}$ as well as lower bounds for the competitive ratio. Although their online algorithm has not been shown to be optimal, the gap between its competitive ratio and the lower bound is quite small.

Lim *et al.* (2011) study the online version of problem $P|\mathcal{M}_j(interval)|C_{\max}$ and present several lower bounds on the competitive ratio for problems with different number of machines.

3.3 Tree-hierarchical processing sets

Huo and Leung (2010a) present a fast approximation algorithm for problem $P|\mathcal{M}_j(tree)|C_{\max}$ with an absolute worst-case error bound of 4/3. Epstein and Levin (2011) develop a PTAS for this problem.

Theorem 10 (Epstein and Levin 2011). *There exists a PTAS for* $P|\mathcal{M}_j(tree)|C_{\max}$.

Li and Li (2015) consider problem $P|\mathcal{M}_j(tree), r_j, p_j = p|C_{\max}$ in which the jobs have release dates and the job processing times are identical. They develop an $O(n^2 + mn \log n)$ time algorithm for this problem. They also present an $O(mn^2 \log m)$ time algorithm for the uniform machine case. Recently, Li and Lee (2016) provide an improved algorithm for $P|\mathcal{M}_j(tree), r_j, p_j = p|C_{\max}$ with a running time of $O(mn \log n)$.

Theorem 11 (Li and Li 2015; Li and Lee 2016). $P|\mathcal{M}_j(tree), r_j, p_j = p|C_{\max}$ *can be solved in* $O(mn \log n)$ *time, whereas* $Q|\mathcal{M}_j(tree), r_j, p_j = p|C_{\max}$ *can be solved in* $O(mn^2 \log m)$ *time.*

Xu and Liu (2015) study online scheduling over time for the three-machine problem $P3|\mathcal{M}_j(tree), r_j, p_j = p|C_{\max}$. They design an optimal algorithm with a competitive ratio of 3/2.

3.4 Arbitrary processing sets

Low (2006) propose an approximation algorithm for problem $P|\mathcal{M}_j|C_{\max}$ and present an absolute worst-case error bound. Lee *et al.* (2009a) point out an error in Low's analysis and show that his algorithm does not have a constant approximation ratio. Huang and Yu (2010) present a simple heuristic for problem $P|\mathcal{M}_j|C_{\max}$ and test its effectiveness computationally.

Recalde *et al.* (2010) propose four different neighborhood search methods for the problem $Q|\mathcal{M}_j|C_{\max}$: the jump, swap, push, and lexicographical jump (lexjump) neighborhood. The jump/swap/push/lexjump-optimal assignment is one where the jump/swap/push/lexjump neighborhood is applied until no further improvement can be made. Recalde *et al.* show that the jump/swap/push-optimal assignment for $Q|\mathcal{M}_j|C_{\max}$ has an absolute worst-case error bound of $1/2 + \sqrt{1/4 + (m-1)\tilde{s}}$, where $\tilde{s} = \frac{\max\{s_i\}}{\min\{s_i\}}$ is the ratio of the maximum machine speed versus the minimum machine speed. They also show that the lexjump-optimal assignment for $Q|\mathcal{M}_j|C_{\max}$ has an absolute worst-case error bound of $O(\frac{\log \sum \tilde{s_i}}{\log \log \sum \tilde{s_i}})$, where $\tilde{s_i} = \frac{s_i}{\min\{s_i\}}$ is the relative speed of machine M_i. For equal-processing-time jobs, they show that jump/swap/push-optimal assignment has an absolute worst-case error bound of $\sqrt{(1+\frac{m-1}{n})\sum \tilde{s_i}}$, while lexjump-optimal assignment has an absolute worst-case error bound of $O(\frac{\log n}{\log \log n})$. Rutten *et al.* (2012) show that the above bounds are tight up to a constant factor.

For the problem $P|\mathcal{M}_j|C_{\max}$, Recalde *et al.* (2010) show that the "jump/swap/push-optimal assignment" has an absolute worst-case error bound of $1/2+\sqrt{m-3/4}$ while the lexjump-optimal assignment has an absolute worst-case error bound of $O(\frac{\log m}{\log \log m})$. Both results are obtained by specializing the bounds in $Q|\mathcal{M}_j|C_{\max}$ to identical speeds of the machines.

Lee *et al.* (2009b) consider the problem $P|\mathcal{M}_j, |\mathcal{M}_j| \leq 2|C_{\max}$ (i.e., the problem in which each eligible set contains at most two machines). They formulate the problem as a graph balancing problem which is defined as follows: Given an undirected multigraph with weights on the edges, we would like to orient the edges so that the maximum of the loads of the vertices is minimized, where the load of a vertex is the sum of the weights of the incoming edges. Lee *et al.* develop an FPTAS when the simplified graph of the multigraph is a tree (the simplified graph of a multigraph is obtained by deleting all self loops and replacing the multiple edges by a single edge that connects the same vertices). Recently, Ebenlendr *et al.* (2014) design a polynomial-time 1.75-approximation algorithm for the problem $P|\mathcal{M}_j, |\mathcal{M}_j| \leq 2|C_{\max}$.

Lee *et al.* (2011b) analyze the problem $P|\mathcal{M}_j, r_j, p_j = p|C_{\max}$, where jobs have release dates and identical processing times. They design an $O(m^{3/2}n^{5/2}\log n)$ algorithm for the problem. They also extend the result to solve problem $Q|\mathcal{M}_j, r_j, p_j = p|C_{\max}$ in $O(m^{3/2}n^{5/2}\log nm)$ time.

Theorem 12 (Lee *et al.* 2011b). *$P|\mathcal{M}_j, r_j, p_j = p|C_{\max}$ can be solved in $O(m^{3/2}n^{5/2}\log n)$ time, whereas $Q|\mathcal{M}_j, r_j, p_j = p|C_{\max}$ can be solved in $O(m^{3/2}n^{5/2}\log nm)$ time.*

Biró and McDermid (2014) study the problem $P|\mathcal{M}_j, p_j \in \{1, 2, 4, \ldots, 2^k\}|C_{\max}$. They give a polynomial-time approximation algorithm with a worst-case error bound of $2 - \frac{1}{2^k}$. The running time of their algorithm is $O(2^k ne)$, where $e = \sum_{j=1}^{n} |\mathcal{M}_j|$.

Some studies consider online scheduling of problems with arbitrary processing sets. Lim *et al.* (2011) study the online version of problem $P|\mathcal{M}_j|C_{\max}$ and show that Algorithm AW developed by Azar *et al.* (1995) achieves a competitive ratio of $\lfloor \log_2 m \rfloor + \frac{m}{2^{\lfloor \log_2 m \rfloor}}$.

Theorem 13 (Lim *et al.* 2011). *There exists an online (over list) algorithm for $P|\mathcal{M}_j|C_{\max}$ with a competitive ratio of $\lfloor \log_2 m \rfloor + \frac{m}{2^{\lfloor \log_2 m \rfloor}}$.*

Lee *et al.* (2009c) study the online version of problem $Q2|\mathcal{M}_j|C_{\max}$. Assuming that M_1 has speed 1 and M_2 has speed $s > 0$, they show that the HSF algorithm is an optimal online algorithm with competitive ratio $1+\min\{s, \frac{1}{s}\}$. Mandelbaum and Shabtay (2011) study a semi-online version of $P|\mathcal{M}_j, p_j = 1|C_{\max}$ in which the job types of the next h jobs beyond the current one in the job list, as well as the total number of jobs, are known to the scheduler. Lee *et al.* (2011b) study online over time for the problem $P2|\mathcal{M}_j, r_j, p_j = p|C_{\max}$ and present an optimal algorithm with a competitive ratio of $\frac{\sqrt{5}+1}{2}$.

Theorem 14 (Lee *et al.* 2011b). *There exists an optimal online (over time) algorithm for $P2|\mathcal{M}_j, r_j, p_j = p|C_{\max}$ with a competitive ratio of $\frac{\sqrt{5}+1}{2}$.*

Some studies consider arbitrary processing set models with non-makespan objectives. Hao *et al.* (2009) develop a chaotic particle swarm optimization based hybrid algorithm for the problem $P|\mathcal{M}_j|\sum U_j$. Hao *et al.* (2010) develop a particle swarm based algorithm for the problem $P|\mathcal{M}_j|\sum w_j T_j$.

4 Other Related Works

In this section, we consider other scheduling problems with processing set restrictions.

4.1 Models with batch processing

Inclusive processing set restrictions have been extended to parallel-batching (p-batch) machines. Each p-batch machine M_i has a size capacity S_i. Each job J_j has a size s_j, where $0 < s_j \leq$

$\max_{i=1,\ldots,m}\{S_i\}$ for all j. Several jobs can be batched together and assigned to a machine M_i, provided that the total size of the jobs in the batch does not exceed S_i. The processing time of a batch is the longest processing time of all the jobs in the batch. We use $P|p\text{-}batch, \mathcal{M}_j(inclusive)|C_{\max}$ to denote problem $P|\mathcal{M}_j(inclusive)|C_{\max}$ with p-batch machines. For the problem $P|p\text{-}batch, \mathcal{M}_j(inclusive), p_j = 1|C_{\max}$, Wang and Leung (2014) present an $O(n \log n)$ time algorithm with an absolute worst-case error bound of 2. They show that, unless $P = NP$, there is no polynomial-time approximation algorithm with an absolute worst-case error bound less than 2. They also provide an $O(mn^2)$ time algorithm with an asymptotic worst-case error bound of 3/2. For the problem $P|p\text{-}batch, \mathcal{M}_j(inclusive)|C_{\max}$, Jia *et al.* (2015) give a deterministic algorithm based on the first-fit-decreasing rule and a meta-heuristic based on the max-min ant system to solve the problem, and they show that both heuristics outperform previously studied heuristics.

Motivated by the operation of a restaurant, Tadayon and Salmasi (2012, 2013) study a hybrid flowshop with jobs arriving the system in groups at different times. There are parallel machines with arbitrary processing set restrictions in each stage, and the objective is to minimize a combination of (i) the sum of completion times of the groups and (ii) the sum of the differences between the completion time of each job and the completion time of the group that the job belongs to. A particle swarm optimization algorithm is presented.

Wang *et al.* (2012) consider a two-stage hybrid flowshop problem encountered in semiconductor manufacturing. Each job has a given release date and a given due date. In the first stage, each machine processes at most one job at a time with an arbitrary processing set restriction. In the second stage, each machine can process a batch of no more than two jobs of the same "recipe" simultaneously, again with an arbitrary processing set restriction. The objective is to minimize the makespan of the schedule. Dispatching rules and reoptimization techniques are proposed for tackling the problem.

4.2 Models with resource constraints

Motivated by the operation of an injection molding department of an electrical appliance company, Edis *et al.* (2008) consider the problem $P|\mathcal{M}_j|\sum C_j$ with additional resource constraints due to the limited number of available operators. Each job J_j requires res_j operators during the time period the job is processed, and b operators are available. They present a Lagrangian-based solution method and a problem-specific heuristic, and evaluate their performance via computational experiments. Edis and Ozkarahan (2011) consider problem $P|\mathcal{M}_j|C_{\max}$ with the same resource constraints as in Edis *et al.* (2008). They study three optimization models: (1) an integer programming (IP) model, (2) a constraint programming (CP) model, and (3) a combined IP/CP model. Computational results show that the combined IP/CP model outperforms the IP model and the CP model. Wang *et al.* (2015) study a surgical operations scheduling problem and model it as problem $P|\mathcal{M}_j|C_{\max}$ with resource constraints. In their model, there are λ types of resource with one unit of each type, and each job requires only one unit of resource to process. They propose heuristic algorithms and evaluate them via computational experiments.

Su *et al.* (2011) consider the maximum lateness minimization problem $P|\mathcal{M}_j|L_{\max}$ with resource constraints, but the resources are allocated to machines and not to jobs. There are R units of a resource available. The speed of a machine M_i depends on the amount of resource r_i allocated to it. All machines have a common minimum and maximum number of units of the resource, $r_{\min}$ and $r_{\max}$, respectively. Thus, r_i needs to satisfy $r_{\min} \leq r_i \leq r_{\max}$ and $\sum_{i=1}^{m} r_i \leq R$. A network flow algorithm and a heuristic are developed for this problem. The effectiveness of the heuristic is tested computationally.

4.3 Models with coordination mechanisms

Lee *et al.* (2011a) study coordination mechanisms for scheduling jobs on m parallel machines with processing set restrictions, where agents who own the jobs acts selfishly to minimize his/her own completion time. To resolve the conflict, each machine will announce in advance the local policy (i.e., sequencing rule) adopted by the machine to sequence the jobs assigned to that machine. Each agent is aware of the local policy of each machine as well as all the information about the other jobs. The social objective is to minimize the makespan. The price of anarchy (POA), which is the ratio between the worst objective function value of an equilibrium of the game and the objective function value of an optimal outcome, is used to quantify the inefficiency of the equilibrium.

Lee *et al.* (2011a) consider the Lowest Grade and Longest Processing Time first (LG-LPT) local policy which processes jobs in nondecreasing order of the flexibility of the job, and in case of a tie, in nonincreasing order of their processing times. For the case with inclusive processing sets, they show that the POA of the LG-LPT policy is $\frac{5}{4}$ when $m = 2$, and is $2 - \frac{1}{m-1}$ when $m \geq 3$. For the case with interval processing sets, they show that the POA of the LF-LPT policy is $\frac{5}{4}$ when $m = 2$, at most $4 - \frac{3}{m}$ when $m \geq 3$, and at least $4 - \frac{28}{m+12}$ when m is a multiple of 8. For the case with arbitrary processing sets, they consider the LFJ local policy which processes jobs in nondecreasing order of the cardinality of their processing sets. They show that the POA of the LFJ policy is $\frac{3}{2}$ when $m = 2$, and is within the interval $\left[\lceil \log_2(m+1) \rceil - 1, \log_2 m + \frac{1}{2}\right]$ when $m \geq 3$.

Guan and Li (2013) also consider the LG-LPT local policy for the case with inclusive processing sets. Same as Lee *et al.* (2011a), they show that the POA of this policy is $\frac{5}{4}$ when $m = 2$, and is $2 - \frac{1}{m-1}$ when $m \geq 3$. They also consider a makespan policy where jobs are ordered arbitrarily and show that its POA is $\frac{3}{2}$ when $m = 2$, and is $\Theta\left(\frac{\log m}{\log \log m}\right)$ when $m \geq 3$.

4.4 Models with uncertainty

As mentioned in Section 3.4, Mandelbaum and Shabtay (2011) study a semi-online version of $P|\mathcal{M}_j, p_j = 1|C_{\max}$, where the scheduler has lookahead abilities. In this study, they also consider a stochastic version of this problem in which the scheduler only knows the probability distribution of the jobs types, and the objective is to minimize the expected makespan of the schedule. They develop optimal dynamic programming algorithms for the problem.

Pinedo and Reed (2013) consider a stochastic uniform machine scheduling problem with nested processing sets. Each job has an exponentially distributed arrival time and an exponentially distributed processing time. Job preemption is permitted. They show that the "least flexible job to the fastest machine" rule minimizes the expected makespan as well as the total expected completion time.

Some studies investigate the robustness of scheduling models with processing set restrictions. Rossi *et al.* (2011) consider problem $Q|\mathcal{M}_j, split|C_{\max}$, where "*split*" indicates that job splitting and preemption are allowed. They conduct sensitivity analysis for this problem by analyzing how a change in demand of the job types affect the makespan of the optimal schedule. Rossi (2010) studies the robustness measures of the configuration of the machines in problem $Q|\mathcal{M}_j, split|C_{\max}$, where the configuration is a Boolean matrix $\{Q_{ji}\}$ such that $Q_{ji} = 1$ if and only if machine $M_i \in \mathcal{M}_j$. Aubry *et al.* (2012) consider a model which maximizes the robustness of the configuration in problem $Q|\mathcal{M}_j, split|C_{\max}$, subject to a setup cost constraint, where a setup cost is incurred when a machine is made eligible for a job.

4.5 Models with other structures or requirements

Shabtay *et al.* (2015) study uniform machine scheduling of equal-processing-time jobs with rejection and arbitrary processing set restrictions. A job J_j can either be rejected and incur a cost of e_j, or be accepted and scheduled on a machine that belongs to processing set $\mathcal{M}_j$. Their model consists of two criteria Criterion F_1 is either $f_{\max}(A) = \max_{J_j \in A}\{f_j(C_j)\}$ or $\sum_{J_j \in A} f_j(C_j)$, where A is the set of accepted jobs, C_j is the completion time of J_j, and f_j is a nondecreasing function of C_j. Criterion F_2 is the total rejection cost; that is, $F_2 = \sum_{J_j \in \bar{A}} e_j$. They study four bi-criteria problems: (1) minimize $F1 + F2$; (2) minimize $F1$ subject to the constraint that F_2 is not greater than a given threshold; (3) minimize F_2 subject to the constraint that F_1 is not greater than a given threshold; and (4) identify a Pareto-optimal solution for each Pareto-optimal point. They show that if F_1 is $f_{\max}(A)$, then all four problems can be solved in polynomial time. On the other hand, if F_1 is $\sum_{J_j \in A} f_j(C_j)$, then only the first problem is solvable in polynomial time, while the other three problems are NP-hard.

Eliiyi *et al.* (2009) also study a problem with job rejection. They consider the setting where each job has a time window and a processing set; that is, each job has to be scheduled completely inside its given time window on a machine that belongs to its processing set. The objective is to maximize the total profit of the scheduled jobs, and no penalty is incurred on the rejected jobs. A constraint-graph based construction algorithm is developed for solving the problem.

Liao (2009) consider a parallel-machine makespan minimization scheduling problem in which each machine M_i has a set of time intervals $\{[b_i^r, f_i^r) \mid 1 \le i \le m; 1 \le r \le q_i\}$ that the jobs can be processed. Each availability interval $[b_i^r, f_i^r)$ has a service level k, where $1 \le k \le h$, and h is the total number of service levels. Each job J_j has a unit processing time and a required service level rs_j ($1 \leq rs_j \leq h$). Each availability interval with service level k can serve jobs with required service level less than or equal to k. Liao shows that a modified least flexible job (LFJ) rule is optimal for this problem.

Scheduling problems with processing set restrictions appear in various industrial applications. Hu *et al.* (2010) describe an application in block erection in shipbuilding and solve the problem as a parallel-machine scheduling problem with arbitrary processing set restrictions and precedence constraints. Gokhale and Mathirajan (2012) describe an application automobile gear manufacturing and solve the problem as a parallel-machine scheduling problem with arbitrary processing set restrictions and sequence-dependent setup time. Xu *et al.* (2012) apply a scheduling model with inclusive processing sets to berth allocation in container terminal operation. Ghandour *et al.* (2014) apply a scheduling model with arbitrary processing sets to safety message dissemination in vehicular networks. Jin *et al.* (2015) consider energy consumption control of parallel processors and formulate the problem as a parallel-machine scheduling model with arbitrary processing sets.

5 Future Research and Recommendations

We have presented an update of recent results related to scheduling problems with processing set restrictions. There are many interesting problems in this area remain to be resolved. A partial list of these problems is given below.

1. As mentioned in Section 4.1, Wang and Leung (2014) have given a fast heuristic for problem $P|p\text{-}batch, \mathcal{M}_j(inclusive), p_j = 1|C_{\max}$ with an asymptotic worst-case error bound of 3/2. Could there be a fast heuristic for the more general problem $P|p\text{-}batch, \mathcal{M}_j(inclusive)|C_{\max}$ with an asymptotic worst-case error bound less than 2?

2. As mentioned in Section 3.1, Huo and Leung (2010a) have given a 5/3-approximation algorithm for problem $P|\mathcal{M}_j(nested)|C_{\max}$. Could there be a fast heuristic for this problem with a smaller absolute worst-case error bound?

3. Is there any polynomial-time approximation algorithm for $P|\mathcal{M}_j|C_{\max}$ with a constant worst-case error bound? Or can we prove that no such algorithm exists?

4. Besides the work of Xu and Lu (2015), there is a pronounced absence of online (over time) algorithms for $P|\mathcal{M}_j(tree), r_j|C_{\max}$. It will be useful to develop online algorithms and analyze their competitive ratios.

There are also some important related research topics that are worth investigating, including the following:

1. The current research on scheduling problems with processing set restrictions is dominated by models with a makespan objective. However, scheduling objectives such as $\sum C_j$ and $\sum w_j C_j$ are useful when the waiting time of the jobs is the main concern of the system. Developing efficient approximation algorithms for scheduling problems with processing set restrictions involving these objectives is an important direction.

2. Processing set restrictions do not only appear in scheduling models. Kellerer *et al.* (2011) have studied a problem of packing a given set of items into a given set of bins so that the total weight of the items packed is maximized (i.e., a multiple subset sum problem), where the bins have inclusive processing set restrictions. Interesting extensions of Kellerer *et al.*'s model that are worth investigating include (i) problem with heterogeneous bin capacities, (ii) problem with a profit maximization objective where each item is associated with a given profit, and (iii) problem with other processing set restrictions (see Kellerer *et al.* 2011, Sec. 5).

Acknowledgment

This research was supported in part by The Hong Kong Polytechnic University under grant 1-BBZL.

References

Aubry, A., Jacomino, M., Rossi, A., Espinouse, M.-L., 2012. Maximizing the configuration robustness for parallel multi-purpose machines under setup cost constraints. Journal of Scheduling

15 (4), 457–471.

Azar, Y., Naor, J., Rom, R., 1995. The competitiveness of on-line assignments. Journal of Algorithms 18 (2), 221–237.

Bar-Noy, A., Freund, A., Naor, J., 2001. On-line load balancing in a hierarchical server topology. SIAM Journal on Computing 31 (2), 527–549.

Biró, P., McDermid, E., 2014. Matching with sizes (or scheduling with processing set restrictions). Discrete Applied Mathematics 164 (1), 61–67.

Chassid, O., Epstein, L., 2008. The hierarchical model for load balancing on two machines. Journal of Combinatorial Optimization 15 (4), 305–314.

Chen, X., Ding, N., Dósa, G., Han, X., Jiang, H., 2015. Online hierarchical scheduling on two machines with known total size of low-hierarchy jobs. International Journal of Computer Mathematics 92 (5), 873–881.

Chen, X., Xu, Z., Dósa, G., Han, X., Jiang, H., 2013. Semi-online hierarchical scheduling problems with buffer or rearrangements. Information Processing Letters 113 (4), 127–131.

Ebenlendr, T., Krčál, M., Sgall, J., 2014. Graph balancing: A special case of scheduling unrelated parallel machines. Algorithmica 68 (1), 62–80.

Edis, E.B., Araz, C., Ozkarahan, I., 2008. Lagrangian-based solution approaches for a resource-constrained parallel machine scheduling problem with machine eligibility restrictions. Lecture Notes in Computer Science 5027, 337–346.

Edis, E.B., Ozkarahan, I., 2011. A combined integer/constraint programming approach to a resource-constrained parallel machine scheduling problem with machine eligibility restrictions. Engineering Optimization 43 (2), 135–157.

Eliiyi, D.T., Korkmaz, A.G., Çiçek, A.E., 2009. Operational variable job scheduling with eligibility constraints: A randomized constraint-graph-based approach. Technological and Economic Development of Economy 15 (2), 245–266.

Epstein, L., Levin, A., 2011. Scheduling with processing set restrictions: PTAS results for several variants. International Journal of Production Economics 133 (2), 586–595.

Ghandour, A.J., Di Felice, M., Artail, H., Bononi, L., 2014. Dissemination of safety messages in IEEE 802.11p/WAVE vehicular network: Analytical study and protocol enhancements. Pervasive and Mobile Computing 11, 3–18.

Glass, C.A., Kellerer, H., 2007. Parallel machine scheduling with job assignment restrictions. Naval Research Logistics 54 (3), 250–257.

Gokhale, R., Mathirajan, M., 2012. Scheduling identical parallel machines with machine eligibility restrictions to minimize total weighted flowtime in automobile gear manufacturing. The International Journal of Advanced Manufacturing Technology 60 (9–12), 1099–1110.

Guan, L., Li, J., 2013. Coordination mechanism for selfish scheduling under a grade of service provision. Information Processing Letters 113 (8), 251–254.

Hao, J., Liu, M., Wu, C., 2009. CPSO-based hybrid approach for scheduling parallel machines with processing set restrictions. In: Proceedings of WRI Global Congress on Intelligent Systems, pp. 8–13.

Hao, J., Liu, M., Wu, C., 2010. Particle swarm optimization for parallel machine scheduling problem with machine eligibility constraints. Chinese Journal of Electronics 19 (1), 103–106.

Hou, L.-Y., Kang, L., 2011. Online and semi-online hierarchical scheduling for load balancing on uniform machines. Theoretical Computer Science 412 (12–14) 1092–1098.

Hou, L.-Y., Kang, L., 2012. Online scheduling on uniform machines with two hierarchies. Journal of Combinatorial Optimization 24 (4), 593–612.

Hu, X., Bao, J.-S., Jin, Y., 2010. Minimising makespan on parallel machines with precedence constraints and machine eligibility restrictions. International Journal of Production Research 48 (6), 1639–1651.

Huang, R.H., Yu, T.H., 2010. An effective heuristic considering machine flexibility for parallel machine with eligibility problem In: Proceedings of 2010 IEEE International Conference on Industrial Engineering and Engineering Management, pp. 1363–1366.

Huo, Y., Leung, J.Y.-T., 2010a. Fast approximation algorithms for job scheduling with processing set restrictions. Theoretical Computer Science 411 (44–46), 3947–3955.

Huo, Y., Leung, J.Y.-T., 2010b. Parallel machine scheduling with nested processing set restrictions. European Journal of Operational Research 204 (2), 229–236.

Huo, Y., Leung, J.Y.-T., Wang, X., 2009a. A fast preemptive scheduling algorithm with release times and inclusive processing set restrictions. Discrete Optimization 6 (3), 292–298.

Huo, Y., Leung, J.Y.-T., Wang, X., 2009b. Preemptive scheduling algorithms with nested processing set restrictions. International Journal of Foundations of Computer Science 20 (6), 1147–1160.

Hwang, H.-C., Chang, S.Y., Lee, K., 2004. Parallel machine scheduling under a grade of service provision. Computers & Operations Research 31 (12), 2055–2061.

Jia, Z.-H., Li, K., Leung, J.Y.-T., 2015. Effective heuristic for makespan minimization in parallel batch machines with non-identical capacities. International Journal of Production Economics 169, 1–10.

Jin, X., Zhang, F., Fan, L., Song, Y., Liu, Z., 2015. Scheduling for energy minimization on restricted parallel processors. Journal of Parallel and Distributed Computing 81–82, 36–46.

Karhi, S., Shabtay, D., 2013. On the optimality of the TLS algorithm for solving the online-list scheduling problem with two job types on a set of multipurpose machines. Journal of Combinatorial Optimization 26 (1), 198–222.

Karhi, S., Shabtay, D., 2014. Online scheduling of two job types on a set of multipurpose machines. International Journal of Production Economics 150, 155–162.

Kellerer, H., Leung, J.Y.-T., Li, C.-L., 2011. Multiple subset sum with inclusive assignment Set restrictions. Naval Research Logistics 58 (6), 546–563.

Lee, K., Hwang, H.-C., Lim, K., 2014. Semi-online scheduling with GoS eligibility constraints. International Journal of Production Economics 153, 204–214.

Lee, K., Leung, J.Y.-T., Pinedo, M.L., 2009a. A note on "An approximation algorithm for the load-balanced semi-matching problem in weighted bipartite graphs." Information Processing Letters 109 (12), 608–610.

Lee, K., Leung, J.Y.-T., Pinedo, M.L., 2009b. A note on graph balancing problems with restrictions. Information Processing Letters 110 (1), 24–29.

Lee, K., Leung, J.Y.-T., Pinedo, M.L., 2009c. Online scheduling on two uniform machines subject to eligibility constraints. Theoretical Computer Science 410 (38–40), 3975–3981.

Lee, K., Leung, J.Y.-T., Pinedo, M.L., 2010. Makespan minimization in online scheduling with machine eligibility. 4OR 8 (4), 331–364.

Lee, K., Leung, J.Y.-T., Pinedo, M.L., 2011a. Coordination mechanisms with hybrid local policies. Discrete Optimization 8 (4), 513–524.

Lee, K., Leung, J.Y.-T., Pinedo, M.L., 2011b. Scheduling jobs with equal processing times subject to machine eligibility constraints. Journal of Scheduling 14 (1), 27–38.

Lee, K., Leung, J.Y.-T., Pinedo, M.L., 2013. Makespan minimization in online scheduling with machine eligibility. Annals of Operations Research 204, 189–222.

Leung, J.Y.-T., Li, C.-L., 2008. Scheduling with processing set restrictions: A survey. International Journal of Production Economics 116 (2), 251–262.

Li, C.-L., Lee, K., 2016. A note on scheduling jobs with equal processing times and inclusive processing set restrictions. Journal of the Operational Research Society 67 (1), 83–86.

Li, C.-L., Li, Q., 2015. Scheduling jobs with release dates, equal processing times, and inclusive processing set restrictions. Journal of the Operational Research Society 66 (3), 516–523.

Li, C.-L., Wang, X., 2010. Scheduling parallel machines with inclusive processing set restrictions and job release times. European Journal of Operational Research 200 (3), 702–710.

Li, J., Li, W., Li, J., 2009. Polynomial approximation schemes for the max-min allocation problem under a grade of service provision. Discrete Mathematics, Algorithms and Applications 1 (3), 355–368.

Li, W., Li, J., Zhang, T., 2012. Two approximation schemes for scheduling on parallel machines under a grade of service provision. Asia-Pacific Journal of Operational Research 29 (5), 1250029 (12 pages).

Liao, L.-W., 2009. Scheduling jobs with service level requirements on parallel machines under availability and eligibility constraints. Journal of Information and Optimization Sciences 30 (6), 1157–1176.

Lim, K., 2010. Parallel machines scheduling with GoS eligibility constraints: A survey. Journal of the Korean Institute of Industrial Engineers 36 (4), 248–254.

Lim, K., Lee, K., Chang, S.Y., 2011. Improved bounds for online scheduling with eligibility constraints. Theoretical Computer Science 412 (39), 5211–5224.

Liu, M., Chu, C., Xu, Y., Zheng, F., 2011. Semi-online scheduling on 2 machines under a grade of service provision with bounded processing times. Journal of Combinatorial Optimization 21 (1), 138–149.

Liu, M., Xu, Y., Chu, C., Zheng, F., 2009. Online scheduling on two uniform machines to minimize the makespan. Theoretical Computer Science 410 (21–23), 2099–2109.

Low, C.P., 2006. An approximation algorithm for the load-balanced semi-matching problem in weighted bipartite graphs. Information Processing Letters 100 (4) 154–161.

Lu, X., Liu, Z., 2013. Semi-online scheduling problems on two uniform machines under a grade of service provision. Theoretical Computer Science 489–490, 58–66.

Lu, X., Liu, Z., 2015. Online hierarchical scheduling on two uniform machines with bounded job sizes. Asia-Pacific Journal of Operational Research 32 (5), 1550032 (31 pages).

Luo, T., Xu, Y., 2014a. Semi-online hierarchical load balancing problem with bounded processing times. Lecture Notes in Computer Science 8546, 231–240.

Luo, T., Xu, Y., 2014b. Semi-online scheduling on two machines with GoS levels and partial information of processing time. The Scientific World Journal, Article ID 576234 (6 pages).

Luo, T., Xu, Y., 2016. Optimal algorithm for semi-online scheduling on two machines under GoS levels. Optimization Letters 10 (1), 207–213.

Luo, T., Xu, Y., Luo, L., He, C., 2014. Semi-online scheduling with two GoS levels and unit processing time. Theoretical Computer Science 521, 62–72.

Mandelbaum, M., Shabtay, D., 2011. Scheduling unit length jobs on parallel machines with lookahead information. Journal of Scheduling 14 (4), 335–350.

Muratore, G., Schwarz, U.M., Woeginger, G.J., 2010. Parallel machine scheduling with nested job assignment restrictions. Operations Research Letters, 38 (1), 47–50.

Ou, J., Leung, J.Y.-T., Li, C.-L., 2008. Scheduling parallel machines with inclusive processing set restrictions. Naval Research Logistics 55 (4), 328–338.

Pinedo, M., Reed, J., 2013. The "least flexible job first" rule in scheduling and in queueing. Operations Research Letters 41 (6), 618–621.

Recalde, D., Rutten, C., Schuurman, P., Vredeveld, T., 2010. Local search performance guarantees for restricted related parallel machine scheduling. Lecture Notes in Computer Science 6034, 108–119.

Rossi, A., 2010. A robustness measure of the configuration of multi-purpose machines. International Journal of Production Research 48 (4), 1013–1033.

Rossi, A., Aubry, A., Jacomino, M., 2011. A sensitivity analysis to assess the completion time deviation for multi-purpose machines facing demand uncertainty. Annals of Operations Research 191, 219–249.

Rutten, C., Recalde, D., Schuurman, P., Vredeveld, T., 2012. Performance guarantees of jump neighborhoods on restricted related parallel machines. Operations Research Letters 40 (4), 287–291.

Shabtay, D., Karhi, S., 2012a. An asymptotically optimal online algorithm to minimize the total completion time on two multipurpose machines with unit processing times. Discrete Optimization 9 (4), 241–248.

Shabtay, D., Karhi, S., 2012b. Online scheduling of two job types on a set of multipurpose machines with unit processing times. Computers & Operations Research 39 (2), 405–412.

Shabtay, D., Karhi, S., Oron, D., 2015. Multipurpose machine scheduling with rejection and identical job processing times. Journal of Scheduling 18 (1), 75–88.

Su, L.-H., Chang, W.-Y., Chou, F.-D., 2011. Minimizing maximum lateness on identical parallel machines with flexible resources and machine eligibility constraints. International Journal of Advanced Manufacturing Technology 56 (9), 1195–1204.

Tadayon, B., Salmasi, N., 2012. A flexible flowshop scheduling problem with machine eligibility constraint and two criteria objective function. World Academy of Science, Engineering and Technology 6 (2), 103–108.

Tadayon, B., Salmasi, N., 2013. A two-criteria objective function flexible flowshop scheduling problem with machine eligibility constraint. The International Journal of Advanced Manufacturing Technology 64 (5–8), 1001–1015.

Tan, Z., Zhang, A., 2010. A note on hierarchical scheduling on two uniform machines. Journal of Combinatorial Optimization 20 (1), 85–95.

Tan, Z., Zhang, A., 2011. Online hierarchical scheduling: An approach using mathematical programming. Theoretical Computer Science 412 (3), 246–256.

Wang, I.-L., Yang, T., Chang, Y.-B., 2012. Scheduling two-stage hybrid flow shops with parallel batch, release time, and machine eligibility constraints. Journal of Intelligent Manufacturing 23 (6), 2271–2280.

Wang, J.-Q., J.Y.-T. Leung, 2014. Scheduling jobs with equal-processing-time on parallel machines with non-identical capacities to minimize makespan. International Journal of Production Economics 156, 325–331.

Wang, S., Su, H., Wan, G., 2015. Resource-constrained machine scheduling with machine eligibility restriction and its applications to surgical operations scheduling. Journal of Combinatorial Optimization, 30 (4), 982–995.

Wang, Z., Xing, W., 2010. Worst-case analysis for on-line service policies. Journal of Combinatorial Optimization 19 (1), 107–122.

Wang, Z., Xing, W., Chen, B., 2009. On-line service scheduling. Journal of Scheduling 12 (1), 31–43.

Woeginger, G.J., 2009. A comment on parallel-machine scheduling under a grade of service provision to minimize makespan. Information Processing Letters 109 (7), 341–342.

Wu, Y., Cheng, T.C.E., Ji, M., 2014. Optimal algorithms for semi-online machine covering on two hierarchical machines. Theoretical Computer Science 531, 37–46.

Wu, Y., Ji, M., Yang, Q., 2012a. Optimal semi-online scheduling algorithms on two parallel identical machines under a grade of service provision. International Journal of Production Economics 135 (1), 367–371.

Wu, Y., Ji, M., Yang, Q., 2012b. Semi-online machine covering under a grade of service provision. Applied Mechanics and Materials 101–102, 484–487.

Xu, D., Li, C.-L., Leung, J.Y.-T., 2012. Berth allocation with time-dependent physical limitations on vessels. European Journal of Operational Research 216 (1), 47–56.

Xu, J., Liu, Z., 2015. Online scheduling with equal processing times and machine eligibility constraints. Theoretical Computer Science 572, 58–65.

Zhang, A., 2015. On the optimality of the LP-based algorithm for online scheduling with GoS eligibility constraints. Operations Research Letters 43 (5), 522–525.

Zhang, A., Jiang, Y., Fan, L., Hu, J., 2015. Optimal online algorithms on two hierarchical machines with tightly-grouped processing times. Journal of Combinatorial Optimization 29 (4), 781–795.

Zhang, A., Jiang, Y., Tan, Z., 2009. Online parallel machines scheduling with two hierarchies. Theoretical Computer Science 410 (38–40), 3597–3605.

Table 1: Offline algorithms for inclusive processing set models

Problem	Algorithmic result	References
$P\|\mathcal{M}_j(inclusive), r_j\|C_{\max}$	PTAS	Li and Wang (2010)
$P\|\mathcal{M}_j(inclusive)\|C_{\max}$ with 1 or 2 GoS levels	EPTAS	Li *et al.* (2012)
$Pm\|\mathcal{M}_j(inclusive)\|C_{\max}$	FPTAS	Woeginger (2009); Li *et al.* (2012)
$P\|\mathcal{M}_j(inclusive), r_j, p_j = p\|C_{\max}$	$O(n^2 + mn \log n)$	Li and Li (2015)
	$O(\min\{m, \log n\} n \log n)$	Li and Lee (2016)
$Q\|\mathcal{M}_j(inclusive), r_j, p_j = p\|C_{\max}$	$O(mn^2 \log m)$	Li and Li (2015)
$Q\|\mathcal{M}_j(inclusive)\|C_{\max}$ (speed hierarchical model)	PTAS	Epstein and Levin (2011)
$P\|\mathcal{M}_j(inclusive), r_j, pmtn\|C_{\max}$	$O(nk \log P + mnk^2 + m^3k)$, where $k =$ no. of release dates and $P = \sum_j p_j$	Huo *et al.* (2009a)
$P\|\mathcal{M}_j(inclusive)\|C_{\min}$	PTAS	Li *et al.* (2009)
$Pm\|\mathcal{M}_j(inclusive)\|C_{\min}$	FPTAS	Li *et al.* (2009)

Table 2: Online algorithms for inclusive processing set models

Problem	Competitive ratio	References
$P\|\mathcal{M}_j(inclusive)\|C_{\max}$	$e+1 \approx 3.718$	Bar-Noy *et al.* (2001)
$P4\|\mathcal{M}_j(inclusive)\|C_{\max}$	2.333	Tan and Zhang (2011)
	2.294	Lim *et al.* (2011)
$P5\|\mathcal{M}_j(inclusive)\|C_{\max}$	2.610	Tan and Zhang (2011)
	2.501	Lim *et al.* (2011)
$P\|\mathcal{M}_j(inclusive)\|C_{\max}$ with two GoS levels	$1+\frac{m^2-m}{m^2-km+k^2}<\frac{7}{3}$, where k = no. of machines that can process all jobs	Zhang *et al.* (2009)
$P2\|\mathcal{M}_j(inclusive), r_j, p_j=p\|C_{\max}$ (online over time)	$\sqrt{2}$	Lee *et al.* (2011b)
$P\|\mathcal{M}_j(inclusive), r_j, p_j=p\|C_{\max}$ (online over time)	$\sqrt{2}$	Xu and Liu (2015)
$P2\|\mathcal{M}_j(inclusive), p_j=1\|\sum C_j$	$\rho_{LB}+O(\frac{1}{n})$, where ρ_{LB} is a lower bound of any online algorithm	Shabtay and Karhi (2012a)
$Q\|\mathcal{M}_j(inclusive)\|C_{\max}$ with two speeds and GoS levels	See Section 2.2	Hou and Kang (2012)
$Q2\|\mathcal{M}_j(inclusive)\|C_{\max}$	See Section 2.2	Lee *et al.* (2009c); Tan and Zhang (2010)
$Q2\|\mathcal{M}_j(inclusive), split\|C_{\min}$	$\frac{2s+1}{s+1}$, where s is the speed of M_1 which can process all jobs, and M_2 has speed 1	Chassid and Epstein (2008)
$Q\|\mathcal{M}_j(inclusive), split\|C_{\max}$ with two speeds and GoS levels	$\frac{(ks+m-k)^2}{k^2s^2+ks(m-k)+(m-k)^2}$, where k machines have speed s and can process all jobs; other machines have speed 1	Hou and Kang (2011)
$Q\|\mathcal{M}_j(inclusive), split\|C_{\min}$ with two speeds and GoS levels	$\frac{2ks+m-k}{ks+m-k}$, where k machines have speed s and can process all jobs; other machines have speed 1	Hou and Kang (2011)

Table 3: Offline algorithms for nested, interval, tree-hierarchical, and arbitrary processing set models

Problem	Algorithmic result	References
$P\|\mathcal{M}_j(nested)\|C_{\max}$	Worst-case bound*: 7/4	Huo and Leung (2010b)
	Worst-case bound*: 5/3	Huo and Leung (2010a)
	PTAS	Muratore *et al.* (2010); Epstein and Levin (2011)
$P\|\mathcal{M}_j(nested), p_j \in \{1, 2, \ldots, 2^k\}\|C_{\max}$	$O(ke \log n)$, where $e = \sum_{j=1}^{n} \|\mathcal{M}_j\|$	Biró and McDermid (2014)
$P\|\mathcal{M}_j(nested), pmtn\|C_{\max}$	$O(n \log n)$	Huo *et al.* (2009b)
$P\|\mathcal{M}_j(tree)\|C_{\max}$	Worst-case bound*: 4/3	Huo and Leung (2010a)
	PTAS	Epstein and Levin (2011)
$P\|\mathcal{M}_j(tree), r_j, p_j = p\|C_{\max}$	$O(n^2 + mn \log n)$	Li and Li (2015)
	$O(mn \log n)$	Li and Lee (2016)
$Q\|\mathcal{M}_j(tree), r_j, p_j = p\|C_{\max}$	$O(mn^2 + \log m)$	Li and Li (2015)
$P\|\mathcal{M}_j\|C_{\max}$	Worst-case bound: $1/2 + \sqrt{m - 3/4}$	Recalde *et al.* (2010)
$Q\|\mathcal{M}_j\|C_{\max}$	Worst-case bound: $1/2 + \sqrt{1/4 + (m-1)\tilde{s}}$, where $\tilde{s}$ is the ratio of maximum vs. minimum machine speeds	Recalde *et al.* (2010)
$P\|\mathcal{M}_j, \|\mathcal{M}_j\| \leq 2\|C_{\max}$	Worst-case bound*: 1.75	Ebenlendr *et al.* (2014)
$P\|\mathcal{M}_j, r_j, p_j = p\|C_{\max}$	$O(m^{3/2} n^{5/2} \log n)$	Lee *et al.* (2011b)
$Q\|\mathcal{M}_j, r_j, p_j = p\|C_{\max}$	$O(m^{3/2} n^{5/2} \log nm)$	Lee *et al.* (2011b)
$P\|\mathcal{M}_j, p_j \in \{1, 2, \ldots, 2^k\}\|C_{\max}$	Worst-case bound*: $2 - \frac{1}{2^k}$	Biró and McDermid (2014)

*Worst-case bound obtained by polynomial-time algorithm.

Table 4: Online algorithms for nested, interval, tree-hierarchical, and arbitrary processing set models

Problem	Competitive ratio	References
$P\|\mathcal{M}_j(nested), r_j, p_j = 1\|C_{\max}$ (online over time)	1	Lee *et al.* (2011b)
$P\|\mathcal{M}_j(nested), r_j, p_j = p\|C_{\max}$ (online over time)	$(\sqrt{5}+1)/2$	Xu and Liu (2015)
$P\|\mathcal{M}_j(interval), p_j = 1\|C_{\max}$ with two job types and processing sets $\{M_1, \ldots, M_k\}$, $\{M_{s+1}, \ldots, M_m\}$	$\frac{mk}{(m-s)k+s^2}$	Shabtay and Karhi (2012b)
$P\|\mathcal{M}_j(interval)\|C_{\max}$ with two job types and processing sets $\{M_1, \ldots, M_k\}$, $\{M_{s+1}, \ldots, M_m\}$	$1 + \frac{(m-1)k}{(m-s)k+s^2}$	Shabtay and Karhi (2014)
$P3\|\mathcal{M}_j(tree), r_j, p_j = p\|C_{\max}$ (online over time)	$3/2$	Xu and Liu (2015)
$P\|\mathcal{M}_j\|C_{\max}$	$\lfloor \log_2 m \rfloor + \frac{m}{2^{\lfloor \log_2 m \rfloor}}$	Lim *et al.* (2011)
$Q2\|\mathcal{M}_j\|C_{\max}$	$1 + \min\{s, \frac{1}{s}\}$, where M_1 has speed 1, and M_2 has speed s	Lee *et al.* (2009c)
$P2\|\mathcal{M}_j, r_j, p_j = p\|C_{\max}$ (online over time)	$(\sqrt{5}+1)/2$	Lee *et al.* (2011b)