Journal of the Korean Institute of Industrial Engineers
Vol. 36, No. 4, pp. 248-254, December 2010.

Parallel Machines Scheduling with GoS Eligibility Constraints : a Survey

Kyungkuk Lim†

Department of Industrial and Management Engineering, Pohang University of Science and Technology

GoS 상황에서의 스케줄링 문제 : 문헌 조사

임 경 국

포항공과대학교 산업경영공학과

In this paper, we survey the parallel machines scheduling problem with GoS eligibility constraints so as to minimize the makespan. Our survey covers off-line, online and semi-online scheduling problems. In the case of online scheduling, we only focus on online scheduling one by one. Hence we give an introduction to the problem and present important results of the problem.

Keywords: Grade of Service, Off-line, Online, Semi-online

1. Introduction

In practical situations, all machines may not be suitable for processing of jobs. The service provision problem in service industry is a good example of such cases (Hwang *et al*., 2004). In service industry, service provider provides differentiated services to the customers according to their promised grade of service (GoS) levels, resulting in an assignment restriction. As another examples, the load balancing with various server topologies arise a restricted assignment as well (Azar *et al*., 1994; Azar *et al*., 1995; Bar-Noy *et al*., 2001). The problem which deals with such situations is called *scheduling problem with eligibility constraints*, which have been studied extensively by computer scientists, operations researchers and mathematicians and so on (Leung and Li, 2008). In this problem, each job can be processed only by a machine in a job-dependent subset of the set of all machines, which is referred to as the *eligible set* of the job.

The scheduling problem with eligibility constraints can be stated as follows : We are given a set of n jobs denoted by $J=\{1, 2, \cdots, n\}$ to be processed on m identical parallel machines denoted by $M=1, 2, \cdots, \mathrm{m}$. Each job has a processing time and a set of machines $M_j \subseteq M$ to which it can be assigned. The objective is to minimize the maximum completion time (makespan). In this problem, there are some special eligibility constraints which is structured based on the relationship between jobs and machines. For example, Grade of Service (GoS) eligibility constraints, nested eligibility constraints, interval eligibility constraints and general eligibility constraints and so on (Azar *et al*., 1995; Bar-Noy *et al*., 2001; Glass and Kellerer, 2007; Huo and Leung, 2010; Hwang *et al*., 2004; Leung and Li, 2008; Lin and Li, 2004; Lim *et al*., 2010a). The *GoS eligibility constraints* have the property that for any two jobs j and k, either $M_j \subseteq M_k$ or $M_k \subseteq M_j$ (Bar-Noy *et al*., 2001; Hwang *et al*., 2004). There are some research on this eligibility with different name of *inclusive eligibility constraints*. Another eligibility is *nested eligibility constraints* which is mentioned by Pinedo (Pinedo, 1994). The nested eligibility implies that for pair M_j and M_k, either $M_j \cap M_k = \varnothing$, or $M_j \subseteq M_k$, or $M_k \subseteq M_j$ (Glass and Kellerer, 2007; Huo and Leung, 2010). For *the interval eligibility constraints*, eligible set of the job can be expressed by a set of consecutive machines in a linear ordering of machines

† Corresponding author : Ph.D Candidate Kyungkuk Lim, Department of Industrial and Management Engineering, Pohang University of Science and Technology, San 31 Hyoja-Dong, Pohang, Kyungbuk, 790-784, Republic of Korea,
Fax : +82-54-279-2870, E-mail : lkkuk@postech.ac.kr

Received October 18, 2010; Revision Received November 19, 2010; Accepted November 22, 2010.

(Bar-Noy *et al.*, 2001; Lin and Li, 2004). Moreover, *the general eligibility constraints* means that M_j is an arbitrary non-empty subset of M (Azar *et al.*, 1995; Lim *et al.*, 2010a).

In this paper, we only focus on *the parallel machines scheduling problem with GoS eligibility constraints* which is introduced by Hwang *et al.* (2004). Formally, we are given a set of n jobs to be scheduled on m machines. The i-th machine, j-th job and its required processing time are denoted by machine i, job j and p_j, respectively. Also, the grade of service assigned to job j is denoted by g_j, which is specified to indicate that the job may be processed by machine 1 through machine g_j. The objective is to assign each job to one of its eligible machines while minimizing the maximum completion time (makespan).

An application of the scheduling with GoS eligibility constraints in the service industry is provided by Hwang *et al.* (2004). Another application of the problem is the assignment of computer programs to multiple processors with memory constraints, where a job can only be assigned to a processor with memory capacity no less than the job's memory requirement (Kafura and Shen, 1977). Ou *et al.* describe an application in loading and unloading cargoes of a vessel, where there are multiple non-identical loading/unloading cranes working in parallel (Ou *et al.*, 2008). The cranes have identical operating speed but different weight capacity limits. Each piece of cargo can be handled by any crane with a weight capacity limit no less than the weight of the cargo. Besides, Bar *et al.* mention diverse applications such as assigning classes of service to calls in communication networks, routing queries to hierarchical databases, signing documents by ranking executives, and upgrading classes of cars by car rental companies and so on (Bar-Noy *et al.*, 2001).

The rest of this paper is organized as follows. We survey off-line scheduling and online scheduling with GoS eligibility constraints in Section 2 and Section 3, respectively. In Section 3, we survey semi-online scheduling under GoS eligibility constraints. Finally, we suggest future directions of research.

2. Off-line Scheduling Problem

In this section, we survey off-line scheduling problem. A scheduling problem is referred to as *off-line scheduling problem* if full information of jobs to be scheduled is completely known in advance. In off-line scheduling, the quality of algorithm A is usually measured by its *worst-case performance ratio* defined as

$$R(A) = \sup\{\frac{z^A}{z^*}\}$$

, where z^A and z^* are the makespan for the algorithm A and the optimum makespan, respectively.

Since identical parallel machines scheduling is NP-hard in the strong sense, our problem is strongly NP-hard as well (Garey and Johnson, 1979). Hence many researchers have designed approximation algorithms for our problem with increasingly better worst-case ratios. Hwang *et al.* develop a "lowest grade-longest processing time first" (LG-LPT) algorithm for off-line scheduling under GoS eligibility constraints (Hwang *et al.*, 2004). The LG-LPT algorithm initially sorts the jobs such that increasing order of grade and then decreasing order of processing times when a tie happens. Then the algorithm assigns sorted jobs one after another to the least loaded machine among eligible machines. Although Hwang *et al.* do not state the time complexity of their algorithm in their paper, its time complexity is $O(n\log n + (n+m)\log m)$.

Theorem 1 : (Hwang *et al.*, 2004) For scheduling under GoS eligibility constraints, LG-LPT yields a schedule with its makespan.

(a) The algorithm has a tight bound of $\frac{5}{4}$ for two machines.

(b) The algorithm has a tight bound of $2-\frac{1}{m-1}$ for more than three machines.

In fact, our problem is same to the problem of assigning of computer programs to multiple processors with memory constrains which is introduced by Kafura and Shen (Kafura and Shen, 1977). Kafura and Shen develop a "largest-memory-time-first" (LMTF) algorithm, which is the same as the LG-LPT algorithm. They obtain the same results as Hwang *et al.*.

Later, Glass and Kellerer give an improved algorithm with a worst-case ratio at most $\frac{3}{2}$ for m machines (Glass and Kellerer, 2007). Their algorithm has a running time of $O(nm \log m)$. Ou *et al.* develop a $\frac{4}{3}+\epsilon$-approximation algorithm for our problem with a running time of $O((n+m)(\log n m)\log\frac{1}{\epsilon})$, where ϵ is a positive constant which can be set arbitrarily close to zero (Ou *et al.*, 2008). They also design a polynomial time approximation scheme (PTAS) for with a running time $O(mn^{\kappa(\epsilon)}\log P_{\mathrm{sum}} + m\log m)$, where $P_{\mathrm{sum}} = \sum_{j=1}^{n} p_j$ and $\kappa(\epsilon) = 2 + \frac{8}{\epsilon} log_2 \frac{4}{\epsilon}$. Recently, Li and Wang study a scheduling problem with GoS eligibility constraints and job release times (Li and Wang, 2010). They also develop a polynomial time approximation scheme (PTAS) for their problem, as well as a fully polynomial time approximation scheme (FPTAS) for the case in which the number of machines is fixed. Clearly, their PTAS and FPTAS can be applied to the scheduling under GoS eligibility constraints.

Theorem 2 : (Li and Wang, 2010; Ou *et al*., 2008) There exists a polynomial time approximation scheme (PTAS) for scheduling with GoS eligibility constraints.

Ji and Cheng design FPTAS for the scheduling under a GoS provision where the number of machines is fixed (Ji and Cheng, 2008). The time complexity of their algorithm is $O(n^{m+1}L^{m+1}/\epsilon^m)$, where $L=\log(\max n,\ 1/\epsilon,\ p_{\max})$ and $p_{\max}$ is the largest processing time of jobs. Later, Woeginger give a simple alternative proof for the Ji and Cheng (Woeginger, 2009). More recently, Li *et al*. present a new FPTAS for the problem (Li *et al*., 2009).

Note that the identical parallel machines scheduling with GoS eligibility constraints is a special case of the unrelated parallel machines scheduling. We can set the processing time of job j on machine i to be p_j if $i \in M_j$ and ∞ otherwise. Horowitz and Sahni, Jansen and Porkolab, Efraimidis and Spirakis, and Fishkin *et al*. independently present different FPTASs for the unrelated parallel machines scheduling (Horowitz and Sahni, 1976; Jansen and Porkolab, 2001; Efraimidis and Spirakis, 2006; Fishkin *et al*., 2008).

Theorem 3 : (Ji and Cheng, 2008; Woeginger, 2009; Li *et al*., 2009) For scheduling with GoS eligibility constraints, there exists a fully polynomial-time approximation scheme (FPTAS) for the case where m is fixed.

Here, we give an introduction to a special case of the problem of scheduling with GoS eligibility constraints, that is *the scheduling problem with two GoS levels*. This problem is first proposed by Zhou *et al*. and can be stated as follows (Zhou *et al*., 2007) : We are given a set of n jobs and m identical machines. Each job j is labeled with the GoS level of g_j and each machine i is also labeled with the GoS level of $g(i)$. Moreover, $g(i)$ is equal to 1 or 2 for all $1 \leq i \leq m$ and g_j is also equal to 1 or 2 for all j. Without loss of generality, we assume that GoS levels of the first k machines and the last $m-k$ machines are 1 and 2, respectively. If $k=m$, then the problem can be reduced to the online scheduling on parallel identical machines.

For the off-line scheduling problem with two GoS levels, Zhou *et al*. present an algorithm with worst-case ratio of $\frac{4}{3}+(\frac{1}{2})^r$, where r is the desired number of iteration (Zhou *et al*., 2007)). Recently, Li *et al*. design a PTAS for scheduling with two GoS levels (Li *et al*., 2009).

3. Online Scheduling Problem

In this section, we survey online scheduling problem. Scheduling researchers are based on the assumption that knows all information on the jobs in advance. Hence, the majority of topic in scheduling problem is off-line scheduling problem. In practical situations, however, scheduler may start scheduling jobs with limited information of jobs in the beginning and more information is revealed later. For this reason, the online scheduling problem naturally have been arisen. In the literature on scheduling theory, online scheduling problems are classified as *online scheduling one by one* and *online scheduling over time* (Lee *et al*., 2010).

- **Online scheduling one by one**

In *online scheduling one by one*, all jobs can be scheduled at any time and arrive at time $t=0$, but are presented to the scheduler one at a time. The scheduler will schedule the jobs one at a time, and the scheduling decisions are irrevocable. Thus, the scheduler only knows the information of the current job while it does not know any information of the next job nor even its existence.

- **Online scheduling over time**

In *online scheduling over time*, the jobs have different release times, but the scheduler has no information about release times of the jobs. When a job arrives, the scheduler only knows about the job at its release time. Moreover, the scheduler does not have to schedule the job immediately once it is given to the scheduler and it may wait for the next job. Therefore, at any moment of time, the scheduler may have several unscheduled jobs.

In the study of the online scheduling problem, an online algorithm A is said to have competitive ratio of ρ (or to be ρ-competitive), if for every online problem instance it is guaranteed that the makespan of the schedule generated by the algorithm is never more than ρ times the optimum makespan of the off-line version of the same problem. Hence, ρ may well be referred to as the *upper bound* of the algorithm. On the other hand, we say that an online scheduling problem has a *lower bound* of ρ' if no deterministic online algorithm can have a competitive ratio smaller than ρ'. Then, an algorithm for an online scheduling problem is said to be *optimal* if the upper bound of the algorithm is equal to the lower bound of the online scheduling problem (i.e., $\rho=\rho'$).

In this paper, we only survey online scheduling one by one. The online scheduling with GoS eligibility constraints is first analyzed by Bar-Noy *et al*. (2001). (Also in Crescenzi *et al*., 2004) They also propose an online algorithm with a constant competitive ratio, as stated in the next theorem :

Theorem 4 : (Bar-Noy *et al*., 2001) There is an online algorithm for scheduling with GoS eligibility constraints with a competitive ratio of $e+1 \approx 3.718$, where $e(\approx 2.718)$ is the base number of natural logarithm. The same algorithm gives a competitive ratio of $e \approx 2.718$ if all jobs have unit length.

For scheduling of two machines, Jiang *et al*. and Park *et al*. independently develop an optimal algorithm with a competitive ratio of $\frac{5}{3}$ (Jiang *et al*., 2006; Park *et al*., 2006). Recently, Lim *et al*. design an optimal algorithm which is yet another alternative which seems to be the most intuitive and simple (Lim *et al*., 2010b). In fact, this problem is relation to the GoS scheduling on uniform machines is introduced by Chassid and Epstein (2008). In the GoS scheduling on two uniform machines, machines are provided with different capability, i.e., the one with speed s can schedule all jobs, while the other one with speed 1 can only process partial jobs. Indeed, we have $g_j = 1$ if job j may run only on the machine 1 and $g_j = 2$ if it can run on machine 1 and machine 2. Liu *et al*. show that two lower bounds of the competitive ratio for different speeds s and propose two algorithms (Liu *et al*., 2009). However, their results contains an error which is discovered by Lee *et al*. (2009). Thus, Lee *et al*. provide tighter lower bounds and present algorithms with worst-case analysis for various ranges of machine speeds (See <Table 1>). Tan and Zhang propose an optimal algorithm with competitive ratio

$$\begin{cases} \min\left\{1+s,\ 1+\dfrac{1+s}{1+s+s^2}\right\} & \text{for } 0 < s < 1 \\ \min\left\{1+\dfrac{1}{s},\ 1+\dfrac{2s}{1+s+s^2}\right\} & \text{for } 1 < s \end{cases}$$

(Tan and Zhang, 2009). If both machines have the same speed, then the GoS scheduling on uniform machines is identical to the online scheduling of two machines with GoS eligibility constraints. Thus, we are able to confirm the result of the scheduling under GoS eligibility constraints with the results of the GoS scheduling on uniform machines (Liu *et al*., 2009; Lee *et al*., 2009; Tan and Zhang, 2009).

Lim *et al*. and Zhang *et al*. independently propose an algorithm for online problem on three machines and show that it is optimal with competitive ratio of 2 (Lim *et al*., 2010b; Zhang *et al*., 2008). Recently, Tan and Zhang present an improved algorithm with better competitive ratios of 2.333 and 2.610 for 4 machines and 5 machines, respectively (Tan and Zhang, 2009; Tan and Zhang, 2010b). <Table 2> summarizes all the results of online scheduling problem with GoS eligibility constraints.

Table 2. The results of online scheduling problem with GoS eligibility constraints

The number of machines	LB	UB	Ref.
2	$\frac{5}{3}$	$\frac{5}{3}$	Jiang *et al*. (2006), Park *et al*. (2006), Lim *et al*. (2010b)
3	2	2	Lim *et al*. (2010b), Zhang *et al*.(2008)
4	open	2.333	Tan and Zhang (2009), Tan and Zhang (2010b)
5	open	2.610	Tan and Zhang (2009), Tan and Zhang (2010b)
m	e	$e+1$	Bar-Noy *et al*. (2001)

For the online scheduling problem with two GoS levels, Jiang show that the lower bound when $k = 4$ and $m = 16$ is at least 2 (Jiang, 2008). He also present an approximation algorithm with competitive ratio $\frac{12+4\sqrt{2}}{7} \approx 2.522$. Recently, Zhang *et al*. present an online algorithm with a competitive ratio of $1+\frac{m^2-m}{m^2-km+k^2} < \frac{7}{3}$ for any m and k (Zhang *et al*., 2009). They also give results of upper and lower bounds for small m and k (See <Table 3>).

Table 3. Upper and lower bounds for small m and k (Zhang *et al*., 2009)

The number of machines	$k=1$		$k=2$	
	LB	UB	LB	UB
2	1.667	1.667		
3	1.824*	1.857*	1.801*	1.857*
4	1.848*	1.923*	1.907*	2.000*
5	1.848*	1.952*	1.907*	2.000*
6	1.829*	1.968*	1.907*	2.000*

Note) * : Results of Zhang *et al*..

Table 1. The results of online scheduling problem on uniform machine where $s_1 = 0.6180$, $s_2 = 0.8668$, $s_3 = 1.1059$ and $s_4 = 1.3247$ (Lee *et al*., 2009)

Range of s	$(0, s_1]$	$(s_1, s_2]$	$(s_2, 1)$	1	$(1, s_3)$	$[s_3, s_4)$	$[s_4, \infty)$
LB	$1+s^*$	$\frac{1+\sqrt{5}}{2}^*$	$1+\frac{s^2+s}{s^2+s+1}^*$	$\frac{5}{3}$	$1+\frac{2s}{s^2+s+1}^*$	$\frac{1+\sqrt{1+4s}}{2}^*$	$1+\frac{1}{s}^*$
UB	$1+s^*$	$1+\frac{2s}{s^2+s+1}^*$		$\frac{5}{3}$	$1+\frac{s^2+s}{s^2+s+1}^*$		$1+\frac{1}{s}^*$

Note) * : Results of Lee *et al*..

4. Semi-online Scheduling Problem

In this section, we survey semi-online scheduling problem. A online scheduling problem is referred to as *semi-online scheduling problem* if some partial information of all jobs is known in advance before scheduler construct a schedule. Hence, scheduler can obtain algorithms with better performance than online algorithms. Some kinds of partial information that have been studied the semi-online scheduling with GoS eligibility constraints as follows :

- sum : The total processing time of all jobs is known in advance.
- max : The largest processing time of all jobs is known in advance.
- opt : The optimal value of the instance is known in advance.
- group : All jobs have their processing times between p and rp (where, $p > 0$ and $r \geq 1$)

For the case of semi-online scheduling with the known total processing time, Park *et al.* study the semi-online scheduling problem on two machines and develop optimal algorithm with competitive ratio of $\frac{3}{2}$ (Park *et al.*, 2006). Recently, Lim *et al.* give a simple alternative proof for the result of Park *et al.* (Lim *et al.*, 2010b). They also study the scheduling problem on three machines. They prove that no algorithm has the competitive ratio less than $\frac{5}{3}$ and give an algorithm the competitive ratio of $\frac{5}{3}$. Next, we observe Theorem 2 in the results of Lim *et al.* (2010b). They present the lower bound for online scheduling on three machines as follow :

LOWER-BOUND-EXAMPLE-GENERATION
We generate three jobs with $p_1 = p_2 = p_3 = 1$ and $g_1 = g_2 = g_3 = 3$.

Case 1 : If two or more jobs are assigned to same machine, we stop generating jobs. Hence, $\frac{z^A}{z^*} \geq 2$.

Case 2 : If exactly one job is assigned to each machine, we generate job 4 with $p_4 = 1$ and $g_4 = 2$,

Case 2.1 : If job 4 is assigned to the first machine, we generate job 5 with $p_5 = 2$ and $g_5 = 1$. Thus, $z^* = 2$ and $\frac{z^A}{z^*} = \frac{4}{2} = 2$.

Case 2.2 : If job 4 is assigned to the second machine, we generate job 5 with $p_5 = 2$ and $g_5 = 2$.

Case 2.2.1 : If job 5 is assigned to the first machine, we generate job 6 with $p_6 = 3$ and $g_6 = 1$, implying $z^* = 3$ and $\frac{z^A}{z^*} = \frac{6}{3} = 2$.

Case 2.2.2 : If job 5 is assigned to the second machine, we stop generating jobs. Hence, $z^* = 2$ and $\frac{z^A}{z^*} = \frac{4}{2} = 2$.

Actually, the lower bound of 2 for the online scheduling on 3 machines can be extended into a lower bound of 2 for the semi-online scheduling on more than 9 machines. Simply declare the total processing time to be 9, and then continue as in the strategy described there (giving only jobs with $g_j \leq 3$). Then, at the end, extend each used instance with jobs with $p_j = 1$ and $g_j = 9$, so that the total processing time is 9 as promised. Hence, we can easily conclude that a lower bound of semi-online model for arbitrary number of machines is at least 2.

For the case of semi-online scheduling with prior information of the largest processing time, Lim *et al.* and Wu and Yang independently study semi-online scheduling on two machines (Lim *et al.*, 2010b; Wu and Yang, 2010). They also develop an optimal algorithm with the competitive ratio of an irrational number widely known as the *golden ratio*, which is $\frac{1+\sqrt{5}}{2} \approx 1.6180$. In particular, Lim *et al.* give the lower bound and upper bound which is equivalent to but much simpler than the results of Wu and Yang (Lim *et al.*, 2010b). They also design the lower bound of $\sqrt{3} \approx 1.7321$ for the competitive ratio of any algorithm for the case of three machines. For scheduling m machines, Bar-Noy *et al.* develop the lower bound of $e \approx 2.718$ for the competitive ratio (Bar-Noy *et al.*, 2001).

In addition, Wu and Yang study the semi-online with the known the optimal value and design an optimal algorithm with competitive ratio of $\frac{3}{2}$ (Wu and Yang, 2010). Liu *et al.* study two semi-online scheduling problems on two machines with jobs' processing times bounded by an interval $[p, rp]$, where $p > 0$ and $r \geq 1$ are two constant numbers (Liu *et al.*, 2010). For two problems, they show some lower bounds of the competitive ratio of semi-online algorithms for different values of r. They also propose an algorithm and prove that the algorithm is optimal for some situations.

For the semi-online scheduling under known the total and largest processing times, Lim *et al.* investigate the same results as the semi-online scheduling with only the known the total processing time (Lim *et al.*, 2010b). In other words, *the knowledge of the largest processing time cannot improve the performance of any algorithm when the total processing time is already known*. <Table 4> summarizes all the results of semi-online scheduling problem.

5. Conclusions

In this paper we have surveyed the parallel machines sched-

Table 4. The results of semi-online scheduling problem with GoS eligiblity constraints

	Sum			Max			Sum and Max			Opt		
	LB	UB	Ref.	LB	UB	Ref.	LB	UB	Ref.	LB	UB	Ref.
2	1.500	1.500	Park *et al.* (2006) Lim *et al.* (2010b)	1.618	1.618	Lim *et al.* (2010b)	1.500	1.500	Park *et al.* (2006) Lim *et al.* (2010b)	1.500	1.500	Wu and ang (2010)
3	1.667	1.667	Lim *et al.* (2010b)	1.732	open	Lim *et al.* (2010b)	1.667	1.667	Lim *et al.* (2010b)	open	open	
m	2.000*	open		2.718	open	Bar-Noy *et al.* (2001)	open	open		open	open	

Note) * : given in this paper.

uling problem with GoS eligibility constraints in order to minimize the makespan. Although much work has been done on this problem, yet there are many interesting open problems. Below is a list of open problems.

1) Bar-Noy study the online scheduling for m machines with GoS eligibility constraints (Bar-Noy *et al.*, 2001). They also show that lower and upper bounds are e and $e+1$, respectively, where e is a base of natural logarithm. Therefore, it is one of promising future research directions to find the exact or tighter competitive ratio for both eligibility cases.

2) Although many researcher study the semi-online scheduling with GoS eligibility constraints, maybe there are still some semi-online variant that could still not be solved (Bar-Noy *et al.*, 2001; Lim *et al.*, 2010b; Park *et al.*, 2006; Wu and Yang, 2010). Therefore, to find new semi-online scheduling with GoS eligibility constraints may be further work.

3) Jiang and Zhang *et al.* are study the online scheduling problem on m parallel machines with two GoS levels (Jiang, 2008; Zhang *et al.*, 2009) However, there exists the gap between the lower and upper bounds. Therefore, an interesting open problem is to close the gap between the lower and upper bounds.

References

Azar, Y., Broder, A. and Karlin, A. (1994), On-line load balancing, *Theoretical Computer Science*, **130**, 73-84.

Azar, Y., Naor, J. and Rom, R. (1995), The competitiveness of online assignments, *Journal of Algorithms*, **18**, 221-237.

Bar-Noy, A., Freund, A., and Naor, J. (2001), Online load balancing in a hierarchical server topology, *SIAM Journal on Computing*, **31**, 527-549.

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

Crescenzi, P., Gambosi, G., and Penna, P. (2004), On-line algorithms for the channel assignment problem in cellular networks, *Discrete Applied Mathematics*, **137**, 237-266.

Efraimidis, P. S. and Spirakis, P. G. (2006), Approximation schemes for scheduling and covering on unrelated machines, *Theoretical Computer Science*, **359**, 400-417.

Fishkin, A. V., Jansen, K. and Mastrolilli, M. (2008), Grouping techniques for scheduling problems : Simpler and faster, *Algorithmica*, **51**, 183-199.

Garey, M. R. and Johnson, D. S. (1979), Computers and intractability : A guide to the theory of NP-completeness, Freeman, New York.

Glass, G. A. and Kellerer, H. (2007), Parallel machine scheduling with job assignment restrictions, *Naval Research Logistics*, **54**, 250-257.

Horowitz, E. and Sahni, S. (1976), Exact and approximate algorithms for scheduling nonidentical processors, *Journal of the Association for Computing Machinery*, **23**, 317-327.

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

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

Jansen, K. and Porkolab, L. (2001), Improved approximation schemes for scheduling unrelated parallel machines, *Mathematics of Operations Research*, **26**, 324-338.

Ji, M. and Cheng, T. C. E. (2008), An FPTAS for parallel-machine scheduling under a grade of service provision to minimize makespan, *Information Processing Letters*, **108**, 171-174.

Jiang, Y. (2006), Online scheduling on parallel machines with two GoS levels, *Journal of Combinatorial Optimization*, **16**, 28-38. Also in : *Lecture Notes in Computer Science*, **4041**, 11-21.

Jiang, Y., He, Y., and Tang, C. (2006), Optimal online algorithms for scheduling on two identical machines under a grade of service, *Journal of Zhejiang University SCIENCE A*, **7**, 309-314.

Kafura, D. G. and Shen, V. Y. (1977), Task scheduling on a multiprocessor system with independent memories, *SIAM Journal of Computing*, **6**, 167-187.

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

Lee, K., Leung, J. Y-T. and Pinedo, M. L. (2010), Scheduling jobs with equal processing times subject to machine eilibility constraints, *Journal of Scheduling*, DOI 10.1007/s10951-010-0190-0.

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

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

Li, W., Li, J. and Zhang, T. (2009), Approximation schemes for scheduling on parallel machines with GoS Levels, *Lecture Notes in Operations Research-Operations Research and Its Applications*, **10**, 53-60.

Lin, Y. and Li, W. (2004), Parallel machine scheduling of machine-dependent jobs with unit-length, *European Journal of Operational Research*, **156**, 261-266.

Lim, K., Lee, K., and Chang, Soo Y. (2010a), On optimality of a greedy approach to the online scheduling under eligibility constraints, submitted for publication.

Lim, K., Chang, Soo Y., Park, J., and Lee, K. (2010b), Online and semi-online scheduling of two and three machines under a GoS provision, submitted for publication.

Liu, M., Chu, C., Xu, Y., and Zheng, F. (2010), Semi-online scheduling on 2 machines under a grade of service provision with bounded processing times, *Journal of Combinatorial Optimization*, DOI : 10.1007/s10878-009-9231-z.

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

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

Park, J., Chang, Soo Y., and Lee, K. (2006), Online and semi-online scheduling of two machines under a grade of service provision, *Operations Research Letters*, **34**, 692-696.

Pinedo, M. (1994), Scheduling : Theory, algorithms, and systems, New York : Prentice Hall.

Tan, Z. and Zhang, A. (2009), A mathematical programming approach for online hierarchical scheduling, *Lecture Notes in Computer Science*, **5573**, 438-450.

Tan, Z. and Zhang, A. (2010a), A note on hierarchical scheduling on two uniform machines, *Journal of Combinatorial Optimization*, **20**, 85-95.

Tan, Z. and Zhang, A. (2010b), Online hierarchical scheduling : An approach using mathematical programming, *Theoretical Computer Science*, In press, Corrected Proof.

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

Wu, Y. and Yang, Q. (2010), Optimal semi-online scheduling algorithms on two parallel identical machines under a grade of service provision, *Lecture Notes in Computer Science*, **6124**, 261-270.

Zhang, A., Jiang, Y., and Tan, Z. (2008), Optimal algorithms for online hierarchical scheduling on parallel machines, Manuscript.

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

Zhou, P., Jiang, Y., and He, Y. (2007), Parallel machine scheduling problem with two GoS levels}, *Applied Mathematics : A journal of Chinese Universities (Series A)*, **22**, 275-284, (in Chinese).