Complexity results for parallel machine scheduling problems with a server in computer systems

S. Guirchoun *, A. Souhkal * and P. Martineau *

* *Laboratoire d'Informatique - Ecole Polytechnique de l'Université de Tours - 64 av J. Portalis, 37200 Tours (France)*
tel: (+33) 247 361 423
fax: (+33) 247 361 422
samuel.guirchoun@univ-tours.fr

1 Introduction

We consider deterministic scheduling environnement of a computer system composed by a single server and m identical parallel machines. Each of n jobs must be loaded by the server and processed without preemption on one of the parallel machines. After a loading or setup, the server is available to perform another loading activity, that is to say a single server can handle only one job at a time and can be considered like a single machine. Moreover, the loading of a job must be immediately followed by its processing. In this environment, we consider several possibilities for the setup time which can require unit or an arbitrary equal time for all jobs. The aim is to find a feasible schedule in this environment with minimum total completion time.

Many results of the last few years are issued from parallel machines scheduling problems with server. [Koulamas, 1996] proposes a beam search heuristic algorithm for an environment with two parallel processors and a single server. [Kravchenko and Werner, 1997], [Hall et al., 2000] and [Brucker et al., 2002] given some complexity results. So a heuristic to minimize the sum of the completion times in the case of unit setup times and arbitrary processing times is proposed by [Kravchenko and Werner, 2001]. In the same way, [Abdekhodaee and Wirth, 2002] propose some heuristics to minimize the makespan in the case of equal length jobs. Implicitly, the scheduling problems tackled in these papers consider production environment where, the loading activity is usually considered like a multiprocessor task and requires simultaneously the server and the machine to be performed.

In a computer system, machines dispose of a communication coprocessor which allows them to receive server information at any time. So, the loading activity can be considered like a job that requires only the server to be performed. In this way, the computer system that we consider can be seen as a two-stage hybrid flow shop or multiprocessor flow shop, with a single machine at the first stage, which is the server and m parallel machines at the second stage, with a no-wait constraint between the two stages. These problems are known to be strongly NP-hard [Gupta, 1988] even for their preemptive version [Hoogeveen et al., 1996]. [Vignier et al., 1999] and [Linn and Zhang, 1999] propose a state-of-the-art survey on hybrid flow shop scheduling problems..

This article is organized as follow : after introduced the notations in Section 2, we show that two-stage hybrid flow shop problems are at least as difficult as the corresponding parallel machine scheduling problems with a single server by a polynomial reduction and we give one theorem and two corollaries for particular cases. Finally, an overview of complexity results for these problems are given for some objective functions.

2 Notations

We consider a set J of n jobs $\{i\}_{1 \leq i \leq n}$. We assume that preemption is not allowed and that each machine can process only one operation at a time. We denote by d_i the due date and r_i the release date of job i.

According to the notation introduced in [Vignier et al., 1999], the hybrid flow shop problem under consideration is denoted by $FH2, (1, Pm)|nowait|\gamma$. The first stage contains a single machine and the second stage m machines. Each job is composed by two operations: the operation $o_{i,1}$ processed at the first stage and the operation $o_{i,2}$ processed at the second stage. We denote by $p_{i,1}$ and $p_{i,2}$ the processing times of operation $o_{i,1}$ and operation $o_{i,2}$, respectively. The operations at the first stage correspond to the setup of jobs and the second operations at the second stage correspond to the process of jobs. *FH2-problems* refers to a two-stage hybrid flow shop problem.

In the same way, [Kravchenko and Werner, 1997] introduced a notation for identical parallel machine scheduling problems with a single server denoted by $Pm, S1|s_i|\gamma$. Each job is composed by two operations: the setup s processed by the server and a parallel machine and the process p performed by a parallel machine. We denote by s_i the setup times and p_i the processing times of job i. We note *PS-problems* the parallel machine problems with a single server and *FH2-problems* the two-stage hybrid flow shop problems.

For each job i of any schedule σ we define for both problems:

$C_i(\sigma)$ refers to the completion time of job i.

$T_i(\sigma) = \max\{C_i(\sigma) - d_i, 0\}$ the tardiness of job i

$L_i(\sigma) = C_i(\sigma) - d_i$, the lateness of job i.

$U_i(\sigma) = 1$ if job i is completed by its due date, 0 otherwise.

The objective functions contained in the field γ that we consider are :

$C_{max} = \max\{C_i\}$ the latest completion time or makespan of any jobs.

$\sum_{i=1}^{n} C_i$, the total completion time of jobs.

$\sum_{i=1}^{n} w_iC_i$, the total weighted completion time of jobs.

$\sum_{i=1}^{n} T_i$, the total tardiness of jobs.

$\sum_{i=1}^{n} w_iT_i$, the total weighted tardiness of jobs.

$L_{max} = \max\{L_i\}$, the maximum lateness of any jobs.

$\sum_{i=1}^{n} U_i$, the number of tardiness jobs.

$\sum_{i=1}^{n} w_iU_i$, the number weighted of tardiness jobs.

3 Complexity results

In this section, we will present a reduction which leads to complexity results for *FH2-problems*. We have show that two-stage hybrid flow shop problem with a single machine at the first stage, m identical parallel machines at the second stage, no-wait constraint between two stages and equal processing times at the first stage are at least as difficult as the m identical parallel machines problem with a single server and equal setup times.

Theorem 1. *for $\alpha \in \{m, \emptyset\}$, $\beta \in \{r_i, \emptyset\}$, setup times s_i all equal and processing times p_i, the problem $P\alpha, S1|s_i = s, \beta|\gamma$ reduces polynomially to $FH2, (1, P\alpha)|p_{i,1} = s, nowait| \sum$ for all objective functions $\gamma \in \{C_{max},\ L_{max},\ \sum C_i,\ \sum w_iC_i,\ \sum U_i,\ \sum w_iU_i,\ \sum T_i,\ \sum w_iT_i\}$.*

Let A the list where jobs are enumerated in a non decreasing order of their r_i.
while $A \neq \emptyset$ **do**
    Place the first job of A on the first available machine at the second stage.
    Remove this job of the list A.
end while
Return the solution $S = max\{C_{i,2}\}$.

Table 1: Algorithm for the $FH2,(1,P\alpha)|p_{i,1} = s,\ p_{i,2} = p,\ r_i,\ nowait|C_{max}$ problem

By this reduction, some more NP-Hardness results for two-stage hybrid flow shop problems can be derived. For example, $FH2,(1,P2)|p_{i,1} = p|\sum C_i$ is NP-hard, since the corresponding *PS-problems* is NP-hard [Hall et al., 2000]. In the same way, $Pm, S1|s_i = 1|\sum C_i$ is solved polynomially, since $FH2,(1,Pm)|p_{i,1} = 1|\sum C_i$ is solved polynomially [Guirchoun et al., 2004]. More recently, we have proved the complexity of some open problems with the theorem 2 and two corollaries for the makespan criterion.

For the $FH2,(1,P\alpha)|p_{i,1} = s,\ p_{i,2} = p,\ r_i,\ nowait|C_{max}$ problem we can see that the schedule where jobs have been placed in the non decreasing order of their release date is optimal.

Theorem 2. *The $FH2,(1,P\alpha)|p_{i,1} = s,\ p_{i,2} = p,\ r_i,\ nowait|C_{max}$ problem can be solved optimally in a polynomial time by the algorithm 1.*

With the theorem 2 and $r_i = 0$ for all jobs, we can see that any semi-actif schedule is an optimal solution.

Corollary 3. *The $FH2,(1,P\alpha)|p_{i,1} = s,\ p_{i,2} = p,\ nowait|C_{max}$ problem can be solved optimally in a polynomial time.*

We can also deduce the following corollary.

Corollary 4. *The $FH2,(1,P\alpha)|p_{i,1} = 1,\ p_{i,2} = 1,\ nowait|C_{max}$ problem can be solved optimally in a polynomial time.*

4 Synthesis of complexity results

In the end of this paper, Tables 2 presents the complexity results for *FH2-problems*. This table is divided into seven columns. The first column indicates if environment contains 2, 3 or undefined identical parallel machines. The next three columns relate to both processing times and to release date of jobs. The criterion is specified in the fifth column and the complexity of the problem in the sixth column. If the result of the complexity of a problem is deduced from the theorem 2 or 1, or from corollaries 3 or 4, it will be indicated in bold. Otherwise the references of the articles which give the complexity of the problem are specified.

Conclusion

In this paper, we consider a parallel machine scheduling problem with a server and a hybrid flow shop scheduling problem with a no-wait constraint. We propose a polynomial reduction between these two scheduling problems and we give an overview of complexity results. This polynomial reduction is the beginning of a contribution to show that the two-stage hybrid flow shop problems are at least as difficult as the parallel machine problems with a server for well-known objective functions.

m	$p_{i,1}$	$p_{i,2}$	r_i	Criteria	Complexity	References
2	1	any	no	C_{max}	$\mathcal{NP}$-Hard	**theorem 1**
2	s	$\geq s$	no	C_{max}	$\mathcal{NP}$-Hard	**theorem 1**
$\emptyset$	1	any	no	C_{max}	$\mathcal{NP}$-Hard	**theorem 1**
$\emptyset$	1	1	no	C_{max}	$\mathbf{O(n \log n)}$	**corollary 3**
$\emptyset$	s	1	no	C_{max}	$\mathbf{O(n \log n)}$	**corollary 4**
$\emptyset$	s	p	yes	C_{max}	$\mathbf{O(n \log n)}$	**theorem 2**
2	1	any	no	L_{max}	$\mathcal{NP}$-Hard	**theorem 1**
$\emptyset$	s	1	no	L_{max}	Open	[Hall et al., 2000]
2	1	any	no	$\sum C_i$	$O(n \log n)$	[Guirchoun et al., 2005]
2	s	any	no	$\sum C_i$	$\mathcal{NP}$-Hard	**theorem 1**
3	1	any	no	$\sum C_i$	$O(n \log n)$	[Guirchoun, 2004]
$\emptyset$	1	any	no	$\sum C_i$	$\mathcal{NP}$-Hard	**theorem 1**
$\emptyset$	1	1	no	$\sum C_i$	$O(n)$	[Guirchoun, 2004]
$\emptyset$	s	1	no	$\sum C_i$	$O(n)$	[Guirchoun, 2004]
$\emptyset$	1	$\geq m-2$	no	$\sum C_i$	Open	[Brucker et al., 2002]
$\emptyset$	1	$\geq m-1$	no	$\sum C_i$	$O(n \log n)$	[Guirchoun, 2004]
$\emptyset$	s	p	yes	$\sum C_i$	Open	[Brucker et al., 2002]
2	1	any	no	$\sum w_iC_i$	$\mathcal{NP}$-Hard	**theorem 1**
$\emptyset$	1	any	no	$\sum w_iC_i$	$\mathcal{NP}$-Hard	**theorem 1**
$\emptyset$	s	p	yes	$\sum w_iC_i$	Open	[Brucker et al., 2002]
2	1	any	no	$\sum T_i$	$\mathcal{NP}$-Hard	**theorem 1**
$\emptyset$	s	1	no	$\sum T_i$	Open	[Hall et al., 2000]
$\emptyset$	s	p	yes	$\sum T_i$	Open	[Brucker et al., 2002]
2	1	any	no	$\sum w_iT_i$	$\mathcal{NP}$-Hard	**theorem 1**
$\emptyset$	s	1	no	$\sum w_iT_i$	Open	[Hall et al., 2000]
$\emptyset$	s	p	no	$\sum w_iT_i$	Open	[Brucker et al., 2002]
2	1	any	no	$\sum U_i$	$\mathcal{NP}$-Hard	**theorem 1**
$\emptyset$	s	1	no	$\sum U_i$	Open	[Hall et al., 2000]
$\emptyset$	s	1	no	$\sum w_iU_i$	Open	[Hall et al., 2000]
$\emptyset$	s	p	no	$\sum w_iU_i$	Open	[Brucker et al., 2002]
$\emptyset$	s	p	yes	$\sum w_iU_i$	Open	[Brucker et al., 2002]

Table 2: Synthesis of complexity results for FH2 problems

References

[Abdekhodaee and Wirth, 2002] Abdekhodaee, A. H. and Wirth, A. (2002). Scheduling parallel machines with a single server : some solvable cases and heuristics. *Computers and Operations Research*, 29:295–315.

[Abdekhodaee et al., 2003] Abdekhodaee, A. H., Wirth, A., and Gan, H. S. (2003). Equal processing and equal setup time cases of scheduling parallel machines with a single server. *Computers and Operations Research*.

[Brucker et al., 2002] Brucker, P., Dhaenens-Flipo, C., Knust, S., Kravchenko, S. A., and Werner, F. (2002). Complexity results for parallel machine problems with a single server. *Journal of Scheduling*, 5(6):429–457.

[Guirchoun, 2004] Guirchoun, S. (2004). *Ordonnancement à machines parallèles avec serveur. (in french)*. Thèse de doctorat specialité informatique, Ecole polytechnique de l'Université de TOURS, Tours (France).

[Guirchoun et al., 2005] Guirchoun, S., Martineau, P., and Billaut, J.-C. (2005). Total completion time minimization in a computer system with a server and two parallel processors. *Computers and Operations Research*, (32):599–611.

[Guirchoun et al., 2004] Guirchoun, S., Martineau, P., and Soukhal, A. (2004). Computer system with a server and parallel processors : some solvable cases and heuristics. In *Proceedings of the PMS'2004*, pages 234–237, Nancy.

[Gupta, 1988] Gupta, J. N. D. (1988). Two-stage, hybrid flowshop scheduling problem. *Operational Research Society*, 39(4):359–364.

[Hall et al., 2000] Hall, N. G., Potts, C. N., and Sriskandarajah, C. (2000). Parallel machine scheduling with a common server. *Discrete Applied Mathematics*, 102:223–243.

[Hoogeveen et al., 1996] Hoogeveen, J. A., Lenstra, J. K., and Veltman, B. (1996). Preemptive scheduling in a two-stage multiprocessor flowshop is np-hard. *European Journal of Operational Research*, 89:172–175.

[Koulamas, 1996] Koulamas, C. P. (1996). Scheduling two parallel semiautomatic machines to minimize machine interference. *Computers and Operations Research*, 23(10):945–956.

[Kravchenko and Werner, 1997] Kravchenko, S. A. and Werner, F. (1997). Parallel machine scheduling problems with a single server. *Mathl. Comput. Modelling*, 26:1–11.

[Kravchenko and Werner, 2001] Kravchenko, S. A. and Werner, F. (2001). A heuristic algorithm for minimizing mean flow time with unit setups. *Information Processing Letters*, 79:291–296.

[Linn and Zhang, 1999] Linn, R. and Zhang, W. (1999). Hybrid flowshop scheduling : a survey. *Computers and Industrial Engineering*, (37):57–61.

[Vignier et al., 1999] Vignier, A., Billaut, J.-C., and Proust, C. (1999). Les flowshop hybrides : état de l'art. *RAIRO-RO*, 2(33):117–183.