



HAL
open science

**Erratum to "Scheduling of a two-machine flowshop with
availability constraints on the first machine"
[International Journal of Production Economics 99
(2006)]**

Christophe Rapine

► **To cite this version:**

Christophe Rapine. Erratum to "Scheduling of a two-machine flowshop with availability constraints on the first machine" [International Journal of Production Economics 99 (2006)]. International Journal of Production Economics, 2013, 142, pp.211-212. 10.1016/j.ijpe.2004.12.003 . hal-00686067

HAL Id: hal-00686067

<https://hal.science/hal-00686067>

Submitted on 6 Apr 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Erratum to "Scheduling of a two-machine flowshop with availability constraints on the first machine" [International Journal of Production Economics 99 (2006) 16 – 27]

C. Rapine^a

^aUniversité de Lorraine, Laboratoire LGIPM, île du Saulcy, 57045 Metz Cedex 1, France.

Abstract

In Allaoui et al. (2006) (Allaoui H., Artiba A., Elmaghraby S.E., Riane F. Scheduling of a two-machine flowshop with availability constraints on the first machine. International Journal of Production Economics 99 (2006): 16 – 27) two conditions of optimality for the Johnson sequence are given. We establish in this note that these conditions are false, under the resumable and the non-resumable case. We also point out two incorrect proofs and give evidence that the time complexity of their dynamic approach is very unlikely to be correct.

Optimality condition for the Johnson order

Allaoui et al. (2006) give in their Section 3 & 4 two conditions of optimality for the Johnson order (JO) and the Modified Johnson order (MJO) in presence of an unavailability period $[s, t]$ on the first machine. Considering the first machine, let a be the first job completed after the unavailability period, and let B^H be the set of jobs completed before time s . The following propositions are stated in Allaoui et al. (2006), with the makespan minimization as objective:

Proposition 2. *If the completion time of set B^H on the second machine is larger than $t + p_{1,a}$, then JO is optimal.*

Proposition 3. *If no idle time appears on the first machine, then MJO is optimal.*

The following example proves that both propositions are false, under the resumable and the non-resumable case. The unavailability period on the first machine is $[5, 6]$ and we have 5 jobs to schedule, whose processing times are given below:

Jobs	1	2	3	4	5
$p_{1,i}$	1	2	2	3	4
$p_{2,i}$	4	2.1	2.1	3.1	3.9

The JO (and MJO) sequence is $(1, 2, 3, 4, 5)$ and is unique. The resulting makespan is 16.9. Observe that set $B^H = \{1, 2, 3\}$ completes exactly at the beginning of the unavailability period, at time $s = 5$. Thus the JO schedule fulfills the condition of Proposition 3. Also observe that set B^H completes at time 9.2 on the second machine, while job 4, which plays the role of job a , completes at time 9 on the first machine. Thus the JO schedule also fulfills the condition of Proposition 2. However this schedule is not optimal: sequence $(1, 5, 2, 3, 4)$ results in a makespan of 16.2. Notice that on the example we obtain the same JO schedule in the resumable and the non-resumable case.

This example shows that even if a JO schedule fulfills conditions of Proposition 2 **and** Proposition 3, it is not necessarily optimal. The following simpler example establishes that if a JO schedule satisfies only condition of Proposition 3, it can be asymptotically as bad as twice the optimal value, which is the worst possible performance ratio for a JO schedule ! Consider an unavailability period of length g ($g \geq 3$) starting at time 3, and 3 jobs to schedule, whose processing times are given below:

Jobs	1	2	3
$p_{1,i}$	1	2	3
$p_{2,i}$	2	3	$g + 1$

The JO (and MJO) sequence is $(1, 2, 3)$ and is unique. Notice that this is also the only optimal sequence relaxing the unavailability period. The makespan of the

Email address: christophe.rapine@univ-lorraine.fr
(C. Rapine)

resulting JO schedule is $2g + 7$, and clearly no idle time appears on the first machine, i.e. the condition of Proposition 3 is fulfilled. However sequence (3, 2, 1) has a makespan of only $g + 9$. The performance ratio tends to 2 when g tends to infinity.

Johnson rule as heuristic

The authors state in Lemma 6 that the makespan of the JO schedule in the resumable case is at most the optimal makespan without the unavailability period plus the duration g of the period. In the notations of the paper, $C_{\max}^H(G) \leq C_{\max}^H(\emptyset) + g$. The result is correct, but not the proof given. Let b be the last job that completes on the first machine before the unavailability period. In case 1, the authors claim that if a condition very closed to Proposition 2 is fulfilled, namely $C_{2,b}^H \geq C_{1,a}^H$, then the equality $C_{\max}^H(G) = C_{\max}^H(\emptyset)$ holds. This claim is false, as proved by the following example: We have an availability period $[1, 2]$ and 3 jobs to schedule:

Jobs	1	2	3
$p_{1,i}$	1	2	4
$p_{2,i}$	3	3	1

The (unique) JO sequence is (1, 2, 3). The resulting makespan without the unavailability period is 8. With the availability period, jobs 1 plays the role of job b and job 2 the role of job a . It is easy to check that the condition is fulfilled, since $C_{2,1}^H = C_{1,2}^H = 4$. However the makespan of the JO schedule is now 9.

We give a short correct proof. Let $G^H(\emptyset)$ be the disjunctive graph oriented according to the JO sequence. It is folklore that the longest path in $G^H(\emptyset)$ is the makespan $C_{\max}^H(\emptyset)$ of the schedule. We define accordingly the execution graph $G^H(G)$, where the operation of a on the first machine is replaced by an operation $o'_{1,a}$ of duration $p_{1,a} + g$. Now consider the longest path of $G^H(G)$, whose length is clearly equal to $C_{\max}^H(G)$. This path may or may not pass through $o'_{1,a}$ (i.e. the unavailability period is or is not on the critical path), but in any case its length is at most g plus its length in graph $G^H(\emptyset)$. By definition this latter quantity is bounded by the length of the longest path in $G^H(\emptyset)$. The result follows.

Section 8 presents Lemma 9, the counter part of Lemma 6 for the non-resumable case. The proof is also incorrect: the same example as for Lemma 6 shows that the argument in Case 1 does not hold. The correct proof of Lemma 6 given above can be easily adapted by increasing the processing time of job a on the first ma-

chine by $g + \delta$, with δ the idle occurring on the first machine before the unavailability period.

A dynamic programming model

In Section 5, Allaoui et al. (2006) propose a dynamic programming approach to solve the problem in the non-resumable case. They claim to obtain a time complexity in $O(2^k n \log n)$, with k the maximal number of jobs that can be completed before the unavailability period in any schedule. As the authors noticed, we can assume that k is not larger than $n/2$. The details of the approach are hard to understand due to loose definitions, but the final result is quite surprising. Indeed the SubSetSum problem is a particular case of this flowshop scheduling problem, considering instances with zero processing time on the second machine. The time complexity of the algorithm in $O(2^{n/2} n \log n)$ challenges the best exact algorithm known for the SubSetSum problem, see Woeginger (2003) and references within. Moreover, consider the k -SubSetSum problem, a variant of the SubSetSum problem with the additional cardinality constraint to pick exactly k items. The DP proposed by Allaoui et al. (2006) can solve it to optimality in time complexity $O(2^k n \log n)$. This time complexity would demonstrate that k -SubSetSum is by definition fixed parameter tractable, with k the parameter of the problem (for an introduction to parametrized complexity, see Downey & Fellows (1995)). However Downey & Fellows (1995) establish that problem k -SubSetSum is hard for class $W[1]$, see their Theorem 4.3. Thus the DP of Allaoui et al. (2006) would involve that $FTP = W[1]$ and the collapse of the parameterized complexity hierarchy, which is very unlikely to happen.

References

- Allaoui, H., Artiba, A., Elmaghraby, S., & Riane, F. (2006). Scheduling of a two-machine flowshop with availability constraints on the first machine. *International Journal of Production Economics*, 99, 16 – 27.
- Downey, R. G., & Fellows, M. R. (1995). Fixed-parameter tractability and completeness II: On completeness for $W[1]$. *Theoretical Computer Science*, 141, 109 – 131.
- Woeginger, G. (2003). Exact algorithms for NP-Hard problems: A survey. In M. Junger, G. Reinelt, & G. Rinaldi (Eds.), *Combinatorial Optimization Eureka, You Shrink!* (pp. 185–207). Springer Berlin / Heidelberg volume 2570 of *Lecture Notes in Computer Science*.