

Erratum to "Johnson's algorithm: A key to solve optimally or approximately flowshop scheduling problems with unavailability periods" [International Journal of Production Economics 121 (2009) 81-87]

Christophe Rapine

► **To cite this version:**

Christophe Rapine. Erratum to "Johnson's algorithm: A key to solve optimally or approximately flowshop scheduling problems with unavailability periods" [International Journal of Production Economics 121 (2009) 81-87]. International Journal of Production Economics, Elsevier, 2013, 144 (2), pp.641-643. 10.1016/j.ijpe.2012.11.010 . hal-00697092

HAL Id: hal-00697092

<https://hal.archives-ouvertes.fr/hal-00697092>

Submitted on 14 May 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 “Johnson’s algorithm : A key to solve optimally or approximately flowshop scheduling problems with unavailability periods” [International Journal of Production Economics 121 (2009) 81 – 87]

C. Rapine^a

^aUniversité de Lorraine, Laboratoire LGIPM, Île du Saulcy, F-57045 Metz, France.

Abstract

In Allaoui & Artiba (2009) (Allaoui H., Artiba A. “Johnson’s algorithm : A key to solve optimally or approximately flowshop scheduling problems with unavailability periods” [International Journal of Production Economics 121 (2009) 81 – 87]) the authors propose optimality conditions for the Johnson sequence in presence of one unavailability period on the first machine and pretend for a performance guarantee of 2 when several unavailability periods may occur. We establish in this note that these conditions are false, and that minimizing the makespan in presence of more than one unavailability period is not in APX, unless $P = NP$. We also point out that the results given for the stochastic part of the problem are incorrect. Finally, we propose a simplified proof and a generalization for the two-stage hybrid flowshop.

Optimality condition for the Johnson’s Algorithm

Allaoui & Artiba (2009) give in Section 2 & 3 two conditions of optimality for the sequence delivered by Johnson’s Algorithm (JA) and the Modified Johnson’s Algorithm (MJA) in presence of one unavailability period on the first machine, under the non-resumable model. Proposition 1 claims that a null idle time I_{MP} is a sufficient condition for the JA sequence to be optimal. Unfortunately I_{MP} is simply introduced as the “idle time due to the maintenance period” and no precise definition is given. Nevertheless the same optimality conditions (as stated in Example 2 and Proposition 3) are detailed by the authors in a previous article and happen to be false, see Rapine (submitted). In fact even when *both* conditions are fulfilled, the JA sequence is not necessarily optimal.

JA as a heuristic

The authors state in Section 4 that the JA sequence has a performance guarantee of 2 for the non-resumable case when the first machine is subject to *several* unavailability periods. The result is incorrect, most probably

due to the amazing definition of h^* in the first paragraph. Indeed even for a single resource, minimizing the makespan subject to 2 unavailability periods can not be approximated within any constant guarantee, unless $P = NP$. We have the following result:

Lemma 1. *If $P = NP$, the two machine flowshop problem is not in APX in the non-resumable case if more than one unavailability period occurs on the first machine.*

Proof. The gap preserving reduction is immediate from PARTITION, which is one of the first problems proved NP-complete by Garey & Johnson (1979). An instance x of PARTITION is constituted of n integers a_1, \dots, a_n . It is asked whether there exists a partition $S \cup T$ of $[1, n]$ such that $\sum_{i \in S} a_i = \sum_{i \in T} a_i$. Without loss of generality we can restrict to instances such that $\sum_i a_i$ is even, and define $A = \sum_i a_i / 2$. Given a fixed value $\lambda \geq 1$, we transform the instance x into an instance $g(x)$ of the flowshop problem as follows: We have n jobs, each job i with processing time a_i on the first machine and, say, 1 on the second machine. Two unavailability periods occur on the first machine, respectively on time interval $[A, A + 1]$ and $[2A + 1, \lceil \lambda(2A + 2) \rceil]$. It is clear that if x is a positive instance of PARTITION, then the optimal makespan of $g(x)$ is equal to $2A + 2$, whereas if x is a negative instance, the optimal makespan of $g(x)$ is greater than $\lambda(2A + 2)$. As a consequence no polynomial time algorithm can have a performance guarantee lower

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

than λ unless $P = NP$. Since this holds for any constant λ , the problem can not belong to APX. \square

JA combines with other rule

In Section 5 the authors consider a two-stage hybrid flowshop composed of a single machine at the first stage and two machines on the second stage. They propose a heuristic combining the JA sequence with the LBM rule, and prove a performance guarantee of 3 for it. Their result is correct; however their Proposition 10 can be generalized to any list scheduling algorithm for the second stage, and its proof simplified. List scheduling algorithms, introduced by Graham (1969), are a very popular approach in scheduling theory. Its principle is to greedily allocate tasks to idle resources, breaking ties between tasks concurrently available using a priority list.

Proposition 2. *Using the JA sequence on the first stage, any list scheduling algorithm on the second stage leads to a performance guarantee of 3.*

Proof. Let C_{\max}^L be the makespan obtained by a list algorithm L , and let l be the last task of the schedule. For short let us denote by r_l the completion time of l on the first stage and by t_l its starting time on the second stage. By definition we have

$$C_{\max}^L = C_l = t_l + p_{l,2} = \left(r_l + \frac{1}{2} p_{l,2} \right) + \left(t_l - r_l + \frac{1}{2} p_{l,2} \right)$$

The first term is lower or equal to $2C_{\max}^*$ due to Lemma 9 in Allaoui & Artiba (2009). For the second term, the greedy allocation of list scheduling algorithms implies that both machines of the second stage are continuously busy on time interval $[r_l, t_l]$. Using the conservation of the work, we obtain that $2(t_l - r_l) + p_{l,2} \leq 2C_{\max}^*$. The result follows. \square

We can further generalize this result to the case of m parallel machines on the second stage. Lemma 9 remains valid considering the *splash* relaxation of the second stage to a single resource with processing times $p_{j,2}/m$, but this bound becomes poor when m is large. A more natural relaxation of the second stage is to consider an infinite number of machines. Observe that this relaxation is equivalent to minimizing the maximum lateness L_{\max} on a single resource. With no unavailability period, it is folklore that this problem is solved optimally by the Jackson sequence, where tasks are scheduled in nonincreasing order of their queue (here $p_{j,2}$).

Lemma 2. *Using the Jackson sequence on the first stage, any list scheduling algorithm leads to a performance guarantee of 3 whatever the number of machines on the second stage.*

Proof. As previously, let C_{\max}^L be the makespan obtained by a list algorithm L , and l be the last task of the schedule. The same analysis leads to

$$C_{\max}^L = C_l = (r_l + p_{l,2}) + (t_l - r_l) \leq L_{\max}^J + C_{\max}^*$$

where L_{\max}^J is the maximum lateness of the Jackson sequence for the relaxation. We have to prove that $L_{\max}^J \leq 2C_{\max}^*$. Let L_{\max}^* be the optimal lateness without considering the unavailability period. Clearly we have $L_{\max}^* \leq C_{\max}^*$. Since this lateness is obtained by the Jackson sequence and since the unavailability period $[s, e]$ delays at most of e time units the start of a task, we have $L_{\max}^J \leq L_{\max}^* + e$. If the optimal makespan is lower than e , it implies that any sequence completes before the unavailability period on the first stage, and in particular $L_{\max}^J = L_{\max}^* \leq C_{\max}^*$. Otherwise we have $e \leq C_{\max}^*$ and thus $L_{\max}^J \leq L_{\max}^* + C_{\max}^*$. The result follows. \square

An extension of Johnson's rule

In this last section the authors considered a stochastic version of the problem where each processing time is a random variable exponentially distributed. The first machine is subject to one unavailability period during time interval $[s, e]$. The authors study Talwar's sequence (TO), which is optimal in expectation when no unavailability period occurs. The results of this section are all highly questionable or incorrect due to many errors in the proofs or imprecisions in the definitions.

First Proposition 11 states an optimality condition similar to the one of Proposition 1 for the JO sequence, namely that the TO sequence is optimal if I_{gap} has a centered distribution. Once again the random variable I_{gap} (the counterpart of I_{MP} in Proposition 1), the "idle time due to the gap", is not properly defined. But whatever its definition, one can wonder how such a random variable could have a centered distribution (all processing times have exponential distribution) ? and most of all how such a random variable could take some negative values (implying a negative idle time) ? Thus Proposition 11 may be correct (the implication is true), but it is certainly useless (the condition is never verified).

Second, the validity of Theorem 12 is questionable due to several errors in the proof. In Equation 12, the density f_Z of the sum of 2 random variables X and Z

should be the integral on the interval $[0, z]$ (instead of $[0, +\infty]$). Nevertheless the derived expressions for $p_{1,1} + p_{1,2}$ and $p_{2,1} + p_{2,2}$ are correct, except for the case where processing times have the same distribution. For two random variables X and Y exponentially distributed with the same parameter λ , the correct expression is:

$$f_{X+Y}(z) = \lambda^2 e^{-\lambda z} z$$

The main problem of the proof comes from the expression used for the expectation of the min of 2 random variables X and Y . The authors write that:

$$E(\min(X, Y)) = \int_0^{+\infty} f_X(z) f_Y(z) dz$$

This is clearly false. For instance if variables X and Y follow a uniform distribution, respectively on $[0, 1]$ and $[1, 2]$, the expectation of $\min(X, Y)$ is clearly $E[X] = 0.5$ while the previous formula gives a null expectation. These calculus can certainly be repaired, but the argument of the proof itself is not convincing. As a matter of fact the authors compute for each sequence ($J_1 J_2$ and $J_2 J_1$) the conditionnal expectation of its makespan, knowing that only the first operation fits before the unavailability period. They argue, rightfully, that TO is optimal if both operations are scheduled after the unavailability period. However the probability of this event, that is the first operation of the sequence does not fit before the unavailability period, is by definition sequence dependent. Its probability for the sequence $J_1 J_2$ differs from the sequence $J_2 J_1$ if $\lambda_{1,1} \neq \lambda_{2,1}$. Hence establishing that $E[C_{\max}(J_1 J_2) | J_1 \text{ fits}] \leq E[C_{\max}(J_2 J_1) | J_2 \text{ fits}]$ is not sufficient to prove that the inequality holds in unconditional expectation.

Finally, the authors give in Proposition 13 a necessary condition for k jobs to fit surely (with probability 1) before the unavailability period. The expression given for the distribution of the sum of k random variables exponentially distributed is incorrect. In fact this quantity is well known as the k -Erlang distribution, and has the following expression, see Bertsekas & Tsitsiklis (2008):

$$f_k(x) = \frac{\lambda^k}{(k-1)!} x^{k-1} e^{-\lambda x}$$

This error jeopardizes Proposition 13. But besides that, such a result does not make sense. Indeed the exponential distribution has not a finite support: By definition for any value s , the probability for a single random variable X with exponential distribution λ to be greater than s is $P(X > s) = 1 - F_X(s) = e^{-\lambda s} > 0$. Thus there is no way to derive a condition for the first k jobs to fit with probability 1 before the unavailability period.

References

- Allaoui, H., & Artiba, A. (2009). Johnson's algorithm : A key to solve optimally or approximately flowshop scheduling problems with unavailability periods. *International Journal of Production Economics*, 121, 81 – 87.
- Bertsekas, D. P., & Tsitsiklis, J. N. (2008). *Introduction to Probability*. (2nd ed.). Athena Scientific.
- Garey, M., & Johnson, D. (1979). *Computers and Intractability : A Guide to the Theory of NP-completeness*. W.H. Freeman & Co.
- Graham, R. (1969). Bounds on multiprocessing timing anomalies. *SIAM Journal on Applied Mathematics*, 17.
- Rapine, C. (submitted). Erratum to "scheduling of a two-machine flowshop with availability constraints on the first machine". *International Journal of Production Economics*, 99, 16 – 27.