Ann Oper Res (2018) 271:1079–1085
https://doi.org/10.1007/s10479-018-2779-1

SHORT NOTE

A note: minimizing total absolute deviation of job completion times on unrelated machines with general position-dependent processing times and job-rejection

Baruch Mor[1] · Gur Mosheiov[2]

Published online: 2 February 2018

Abstract We study a scheduling problem with the objective of minimizing total absolute deviation of completion times (TADC). TADC is considered here in the most general form studied so far: the machine setting is that of parallel unrelated, job processing time are assumed to be position-dependent with no restrictions on the functional form, and the option of processing only a subset of the jobs (i.e., job-rejection) is allowed. We show that minimizing TADC in this very general form remains polynomially solvable in the number of jobs.

Keywords Scheduling · Unrelated machines · Total absolute deviation of completion times · Position-dependent processing times · Job-rejection

1 Introduction

The scheduling measure considered in this note is that of total absolute deviation of completion times (TADC). This measure arises in many service systems, where the main objective is to provide customers with identical or similar quality of service. It reflects their total time-in-system or total waiting time. TADC was introduced by Kanet (1981). He showed that minimizing TADC on a single machine can be solved in $O(n \log n)$, where n is the number of jobs. A number of extensions have been published: Mosheiov (2008) solved TADC on a single machine and on parallel identical machines with position-dependent job processing times. Oron (2008) and Li et al. (2009) focused on minimizing TADC on a single machine with simple linear deterioration of the job processing times. Koulamas and Kyparisis (2008) studied TADC with past-sequence-dependent setup times. Yang and Kuo (2009, 2010) considered TADC with both time-dependent increasing processing times (deterioration) and position-dependent decreasing processing times (learning). A further extension to parallel

✉ Gur Mosheiov
msomer@huji.ac.il

1 Department of Economics and Business Administration, Ariel University, 40700 Ariel, Israel

2 School of Business Administration, The Hebrew University, 91905 Jerusalem, Israel

identical machines and deteriorating jobs was studied by Huang and Wang (2011). Mani et al. (2011) also studied TADC with past-sequence-dependent setup times and learning, and focused on parametric analysis of the learning index. Mor and Mosheiov (2011) solved TADC on uniform and unrelated machines, and also studied a more general bi-criteria objective function (containing a linear combination of TADC and total completion time). Chen et al. (2015) studied the same bi-criteria objective with past-sequence-dependent setup times and a learning effect. Recently, Ben-Yehoshua et al. (2015) focused on minimizing TADC on a two-machine no-wait proportionate flowshop.

In this note we study a scheduling problem with the TADC objective in a more general setting. First, we allow the option of job-rejection, which became a popular topic among researchers in recent years. This is reflected in the recently published survey of Shabtay et al. (2013). Shabtay et al. claim that "in many practical cases, mostly in highly loaded make-to-order production systems, accepting all jobs may cause a delay in the completion of orders which in turn may lead to high inventory and tardiness cost. Thus, in such systems, the firm may wish to reject the processing of some jobs by either outsourcing them or rejecting them all together". Technically, in scheduling models considering rejection, the rejected jobs are penalized, and this cost becomes a factor in the objective function (in addition to the cost of a standard classical scheduling measure). Some of the very recent papers dealing with various models of scheduling with job-rejection are: Thevenin et al. (2015), Ou et al. (2015), Wang et al. (2016), Mor and Mosheiov (2016), Zhong et al. (2017) and Agnetis and Mosheiov (2017), Gerstl and Mosheiov (2017), Gerstl et al. (2017) and Mosheiov and Strusevich (2017).

Secondly, following another popular topic in scheduling research, we consider position-dependent job processing times. Scheduling problems with variable job processing times (either starting-time-dependent or position-dependent) have attracted many scheduling researchers in recent years; see e.g. Gawiejnowicz (2008) and more recently Rudek (2012), Sun et al. (2013), Agnetis et al. (2014), Gerstl et al. (2017) and Pei et al. (2017). In most cases, such extensions increase significantly the complexity of the problems, even if the processing times are restricted either to monotone functions of the job starting times or the job positions (reflecting aging or learning), or to specific functions (such as linear or exponential). In this study, we consider *general* position-dependent processing times, and thus no restrictions on their functional form are assumed. Finally, the machine setting assumed in this note is that of parallel unrelated. Thus, the processing time of a job is a function of: (1) the job itself, (2) the machine to which it is assigned, and (3) the position in the sequence processed on this machine.

In summary, in this note we consider a more general version (than those published so far) of a scheduling problem with a TADC objective. Specifically, we extend TADC in three aspects simultaneously: we minimize TADC (1) on *parallel unrelated* machines, with (2) *general position-dependent* job processing times, and (3) allowing the option of *job-rejection*. We show that the problem remains polynomially solvable in the number of jobs.

2 Formulation

We consider an n-job m-machine scheduling problem. Job processing times are assumed to be position-dependent, and the machines are unrelated. The set of jobs is denoted by N. The set of jobs assigned to machine i is denoted N_i, $i = 1, \ldots, m$. The number of jobs assigned to machine i is n_i $(= |N_i|)$. The scheduler may decide to process only a subset of the jobs and reject the others. Let P denote the set of the processed jobs (with $n_P = |P|$ denoting the number of processed jobs), and R denote the set of the rejected jobs (with $n_R = |R|$ denoting the number of rejected jobs). It follows that $N = P \cup R = (\bigcup_{i=1}^{m} N_i) \cup R$, and $\sum_{i=1}^{m} n_i + n_r = n_P + n_R = n$.

The processing time of job j if assigned to position r on machine i is denoted by p_{ijr}, $i = 1, \ldots, m$; $j = 1, \ldots, n$; $r = 1, \ldots, n_i$. For a given job schedule, C_j $(j \in P)$ denotes the completion time of job j. Total absolute deviation of job completion times on machine i is given by: $\sum_{k \in N_i} \sum_{l \in N_i, l \neq k} |C_k - C_l|$. $TADC$ on all m machines is given by $TADC = \sum_{i=1}^{m} \sum_{k \in N_i} \sum_{l \in N_i, l \neq k} |C_k - C_l|$. The rejection cost of job j is denoted by e_j, $j = 1, \ldots, n$. The total rejection cost is given by $TR = \sum_{j \in R} e_j$. The objective function considered in this note is the sum of both: $TADC + TR$. Using the standard three-field notation of scheduling problems, the problem studied here is:

$$R/rejection, p_{jr}/TADC + TR.$$

3 A polynomial time solution for $R/rejection, p_{jr}/TADC + TR$

In this section we introduce a polynomial time solution procedure, which is based on solving a sequence of linear assignment problems. The single machine TADC problem was solved by Kanet (1981), who introduced a solution based on matching job processing times to positions. The positional weight of position r is given by: $W_r = -r^2 + nr + 2r - n - 1$; $r = 1, \ldots, n$. Recall that the input to our problem contains the position-dependent job processing times on each machine and the job-dependent rejection costs. Thus, the input consists of m matrices of size $n \times n$ (each matrix contains the job-position processing times on one machine), and a vector of size n containing the rejection costs. Recall that n_i is the number of jobs assigned to machine i, $i = 1, ..., m$ $\left(n_P = \sum_{i=1}^{m} n_i\right)$, n_R is the number of rejected jobs, and $n_P + n_R = n$.

Assume first that the vector $(n_1, n_2, \ldots, n_m, n_r)$ is given. In order to build the input for a linear assignment problem, we first create a matrix based on the job-position processing times, consisting of $m+1$ blocks. Block i $(i = 1, \ldots, m)$ is of size $n \times n_i$, $i = 1, \ldots m$, and contains the processing times in the first n_i positions of machine i. Block $m+1$ (reflecting the rejected jobs) is of size $n \times \left(n - \sum_{i=1}^{m} n_i\right)$, and contains "1" in all entries. Block $m+1$ can be regarded as a "pseudo-rejection machine". For convenience we define $n_{m+1} = n_R = n - \sum_{i=1}^{m} n_i$: the number of columns in block $m + 1$is the number of rejected jobs. Note that the size of the matrix is $n \times n$. Thus, the processing time matrix, denoted , $PROC$, is the following (see Fig. 1 for the special case of $m = 2$):

$$PROC_{ijr} = \begin{cases} p_{ijr}, & i = 1, \ldots, m;\ j = 1, \ldots, n;\ r = 1, \ldots, n_i \\ 1, & i = m+1;\quad j = 1, \ldots, n;\ r = 1, \ldots, n_{m+1} \end{cases}$$

The cost matrix has an identical structure (of $m + 1$ blocks). Block i $(i = 1, ..., m)$ contains $W_{ijr} = W_{ir}$, $i = 1, \ldots, m$; $j = 1, \ldots, n$; $r = 1, \ldots n_i$. Thus, the weight of job j if assigned to position r on machine i (on which n_i jobs are processed) is: $W_{ir} = -r^2 + n_i r + 2r - n_i - 1$ Note that all lines in block are identical. Block $m+1$ contains $e_{ijr} = e_j$ for $i = 1, \ldots, n_{m+1}$, i.e., the rejection cost, which is machine- and position-independent, is identical for all entries of a given row in this block. Thus, the cost matrix, denoted $COST$, is the following (see Fig. 2 for the case of $m = 2$):

$$COST_{ijr} = \begin{cases} W_{ijr}, & i = 1, \ldots, m;\ j = 1, \ldots, n;\ r = 1, \ldots, n_i \\ e_j, & i = m+1;\quad j = 1, \ldots, n;\ r = 1, \ldots, n_{m+1} \end{cases}$$

$$
\begin{array}{ccccc|ccccc|ccccccccc}
\multicolumn{5}{c}{\text{Machine 1 block}} & \multicolumn{5}{c}{\text{Machine 2 block}} & \multicolumn{9}{c}{\text{Rejection block}} \\
\multicolumn{5}{c}{(n_1 \text{ jobs})} & \multicolumn{5}{c}{(n_2 \text{ jobs})} & \multicolumn{9}{c}{(n_{m+1} = n - n_1 - n_2 \text{ jobs})} \\
p_{111} & p_{112} & p_{113} & \cdots & p_{11n_1} & p_{211} & p_{212} & p_{213} & \cdots & p_{21n_2} & 1 & 1 & 1 & 1 & 1 & 1 & \ldots & 1 & 1 \\
p_{121} & p_{122} & p_{123} & \cdots & p_{12n_1} & p_{221} & p_{222} & p_{223} & \cdots & p_{22n_2} & 1 & 1 & 1 & 1 & 1 & 1 & \ldots & 1 & 1 \\
p_{131} & p_{132} & p_{133} & \cdots & p_{13n_1} & p_{231} & p_{232} & p_{233} & \cdots & p_{23n_2} & 1 & 1 & 1 & 1 & 1 & 1 & \ldots & 1 & 1 \\
p_{141} & p_{142} & p_{143} & \cdots & p_{14n_1} & p_{241} & p_{242} & p_{243} & \cdots & p_{24n_2} & 1 & 1 & 1 & 1 & 1 & 1 & \ldots & 1 & 1 \\
 & & \vdots & & & & & \vdots & & & & & & & \vdots & & & & \\
p_{1(n-1)1} & p_{1(n-1)2} & p_{(n-1)31} & \cdots & p_{1(n-1)n_1} & p_{2(n-1)1} & p_{2(n-1)2} & p_{2(n-1)3} & \cdots & p_{2(n-1)n_2} & 1 & 1 & 1 & 1 & 1 & 1 & \ldots & 1 & 1 \\
p_{1n1} & p_{1n2} & p_{n31} & \cdots & p_{1nn_1} & p_{2n1} & p_{2n2} & p_{2n3} & \cdots & p_{2nn_2} & 1 & 1 & 1 & 1 & 1 & 1 & \ldots & 1 & 1
\end{array}
$$

Fig. 1 The processing time matrix (PROC) for a 2-machine problem

$$
\begin{array}{ccccc|ccccc|ccccccc}
\multicolumn{5}{c}{\text{Machine 1 block}} & \multicolumn{5}{c}{\text{Machine 2 block}} & \multicolumn{7}{c}{\text{Rejection block}} \\
\multicolumn{5}{c}{(n_1 \text{ jobs})} & \multicolumn{5}{c}{(n_2 \text{ jobs})} & \multicolumn{7}{c}{(n_{m+1} = n - n_1 - n_2 \text{ jobs})} \\
w_{11} & w_{12} & w_{13} & \cdots & w_{1n_1} & w_{21} & w_{22} & w_{23} & \cdots & w_{2n_1} & e_1 & e_1 & e_1 & e_1 & \cdots & e_1 & e_1 \\
w_{11} & w_{12} & w_{13} & \cdots & w_{1n_1} & w_{21} & w_{22} & w_{23} & \cdots & w_{2n_1} & e_2 & e_2 & e_2 & e_2 & \cdots & e_2 & e_2 \\
w_{11} & w_{12} & w_{13} & \cdots & w_{1n_1} & w_{21} & w_{22} & w_{23} & \cdots & w_{2n_1} & e_3 & e_3 & e_3 & e_3 & \cdots & e_3 & e_3 \\
w_{11} & w_{12} & w_{13} & \cdots & w_{1n_1} & w_{21} & w_{22} & w_{23} & \cdots & w_{2n_1} & e_4 & e_4 & e_4 & e_4 & \cdots & e_4 & e_4 \\
 & & \vdots & & & & & \vdots & & & & & & \vdots & & & \\
w_{11} & w_{12} & w_{13} & \cdots & w_{1n_1} & w_{21} & w_{22} & w_{23} & \cdots & w_{2n_1} & e_{n-1} & e_{n-1} & e_{n-1} & e_{n-1} & \cdots & e_{n-1} & e_{n-1} \\
w_{11} & w_{12} & w_{13} & \cdots & w_{1n_1} & w_{21} & w_{22} & w_{23} & \cdots & w_{2n_1} & e_n & e_n & e_n & e_n & \cdots & e_n & e_n
\end{array}
$$

Fig. 2 The cost matrix (COST) for a 2-machine problem

We define the standard binary variables: $X_{ijr} = 1$ if job j is assigned to position r on machine i, or if job j is rejected (i.e., assigned to any position on the pseudo-rejection machine $m + 1$), and $X_{ijr} = 0$ otherwise, $i = 1, \ldots, m + 1$; $j = 1, \ldots, n$; $r = 1, \ldots, n_i$. The resulting assignment problem is the following:

$$
\begin{array}{lll}
MIN & \sum_{i=1}^{m+1} \sum_{j=1}^{n} \sum_{r=1}^{n_i} X_{ijr} PROC_{ijr} COST_{ijr} & \\
S.T. & & \\
 & \sum_{i=1}^{m+1} \sum_{r=1}^{n_i} X_{ijr} = 1 & j = 1, \ldots, n. \\
 & \sum_{j=1}^{n} X_{ijr} = 1, & i = 1, \ldots, m + 1;\ r = 1, \ldots, n_i \\
 & X_{jr}\ binary, & i = 1, \ldots, m + 1;\ j = 1, \ldots, n; r = 1, \ldots, n_i
\end{array}
$$

Recall that this assignment problem is defined for given numbers of jobs assigned to the m machines and a given number of rejected jobs. We denote this problem $AP\,(n_1, n_2, \ldots, n_m, n_{m+1} = n_r)$. This standard $(n \times n)$ assignment problem is known to be solved in $O\left(n^3\right)$. It is clear that this problem should be solved for all possible vectors $(n_1, n_2, \ldots, n_{m+1})$ (such that $\sum_{i=1}^{m+1} n_i = n$). The number of allocation vectors of n numbers into $m + 1$ sets is bounded by $\frac{(2n)^{m+1}}{m!}$; see Stirzaker (1994). In fact, if the number of processed jobs $n_P = \sum_{i=1}^{m} n_i$ is known, then the number of the (remaining) rejected jobs is determined. Thus, the latter expression can be reduced to $\frac{(2n)^m}{(m-1)!}$; see Ji and Cheng (2010). It follows that the number of assignment problems to be solved does not exceed $O\,(n^m)$, which is polynomial in the number of jobs for a given number of machines. We conclude that

Theorem 1 *For a given number of machines, problem $R/rejection, p_{jr}\,/TADC + TR$ can be solved in polynomial time in the number of jobs.*

Table 1 Job-position processing times: machine 1 (Example 1)

Job	Position									
	1	2	3	4	5	6	7	8	9	10
1	53	78	92	96	73	48	49	92	75	40
2	2	12	60	27	11	78	31	17	20	30
3	66	5	50	35	81	9	36	68	45	87
4	23	32	21	21	97	55	5	5	71	77
5	47	13	87	67	3	47	95	70	71	9
6	46	37	4	28	41	48	86	30	30	4
7	56	47	29	7	79	4	8	30	49	29
8	41	65	90	76	14	71	24	67	17	9
7	1	11	44	9	23	53	63	11	95	74
8	4	58	61	1	31	65	37	91	53	27

Table 2 Job-position processing times: machine 2 (Example 1)

Job	Position									
	1	2	3	4	5	6	7	8	9	10
1	72	83	47	2	53	64	20	27	39	78
2	25	5	61	37	37	67	62	94	55	66
3	18	73	46	88	17	22	89	16	70	35
4	29	56	27	7	4	59	43	73	80	66
5	95	43	38	98	36	85	38	50	56	10
6	18	9	16	97	4	47	11	21	91	92
7	16	81	22	51	74	61	100	53	59	16
8	39	59	43	99	16	35	98	54	7	42
7	62	68	18	27	12	16	2	31	17	88
8	59	42	1	8	34	38	90	17	86	96

Table 3 Job-dependent rejection costs (Example 1)

Job	Rejection cost
1	49
2	11
3	75
4	55
5	37
6	17
7	55
8	23
9	54
10	69

Numerical Example 1:
We solved a 10-job 2-machine problem. The job-position processing times on the two machines are given in Tables 1 and 2, respectively. The rejection costs are provided in Table 3. We solved $AP\ (n_1, n_2, \ldots, n_{m+1})$ for all possible allocation vectors. The resulting solution is the following: $n_1 = 4, n_2 = 5, n_r = 1$. The job sequence on machine 1: (9, 3, 6,

7). The job sequence on machine 2: (5, 2, 10, 1, 4). The rejected job: job 8. Total TADC of the processed jobs: 106. Total rejection cost: 23. Total cost: 129.

4 Conclusion

We solved a scheduling problem with the objective of minimizing total absolute deviations of job completion times (TADC). We extended the classical setting in three aspects simultaneously: (1) we considered parallel unrelated machines, (2) we assumed general position-dependent job processing times, and (3) we allowed the option of job-rejection. We showed that the problem in this very general form is solved in polynomial time in the number of jobs. Solving TADC with position- or time-dependent job processing time and job rejection, is a challenging topic for future research.

Acknowledgements This research was supported by the Israel Science Foundation (Grant No. 1286/14). The second author was also supported by the Charles I. Rosen Chair of Management, and by The Recanati Fund of The School of Business Administration, The Hebrew University, Jerusalem, Israel.

References

Agnetis, A., Billaut, J.-C., Gawiejnowicz, S., Pacciarelli, D., & Soukhal, A. (2014). *Multiagent scheduling: Models and algorithms*. Berlin: Springer.

Agnetis, A., & Mosheiov, G. (2017). Scheduling with job-rejection and position-dependent processing times on proportionate flowshops. *Optimization Letters*, *11*, 885–892.

Ben-Yehoshua, Y., Hariri, E., & Mosheiov, G. (2015). A note on minimising total absolute deviation of job completion times on a two-machine no-wait proportionate flowshop. *International Journal of Production Research*, *53*, 5717–5724.

Chen, S. H., Mani, V., & Chen, Y. H. (2015). Bi-criterion single machine scheduling problem with a past-sequence-dependent setup times and learning effect. In *Proceedings of the 2015 IEEE IEEM* (pp. 185–189).

Gawiejnowicz, S. (2008). *Time-dependent scheduling*. Berlin, Heidelberg: Springer.

Gerstl, E., Mor, B., & Mosheiov, G. (2017). Minmax scheduling with acceptable lead-times: Extensions to position-dependent processing times, due-window and job rejection. *Computers and Operations Research*, *83*, 150–156.

Gerstl, E., & Mosheiov, G. (2017). Single machine scheduling problems with generalized due-dates and job-rejection. *International Journal of Operations Research*, *55*, 3164–3172.

Huang, X., & Wang, M. Z. (2011). Parallel identical machines scheduling with deteriorating jobs and total absolute differences penalties. *Applied Mathematical Modeling*, *35*, 1349–1353.

Ji, M., & Cheng, T. C. E. (2010). Scheduling with job-dependent learning effects and multiple rate-modifying activities. *Information Processing Letters*, *110*, 460–463.

Kanet, J. (1981). Minimizing variation of flow time in single machine systems. *Management Science*, *27*, 1453–1459.

Koulamas, C., & Kyparisis, G. J. (2008). Single machine scheduling problems with past-sequence-dependent setup times. *European Journal of Operational Research*, *187*, 1045–1049.

Li, Y., Li, G., Sun, L., & Xu, Z. (2009). Single machine scheduling of deteriorating jobs to minimize total absolute differences in completion times. *International Journal of Production Economics*, *118*, 424–429.

Mani, V., Chang, P. C., & Chen, S. H. (2011). Single-machine scheduling with past-sequence-dependent setup times and learning effects: A parametric analysis. *International Journal of Systems Science*, *42*, 2097–2102.

Mor, B., & Mosheiov, G. (2011). Total absolute deviation of job completion times on uniform and unrelated machines. *Computers and Operations Research*, *38*, 660–665.

Mor, B., & Mosheiov, G. (2016). Minimizing maximum cost on a single machine with two competing agents and job rejection. *Journal of The Operational Research Society*, *67*, 1524–1531.

Mosheiov, G. (2008). Minimizing total absolute deviation of job completion times: Extensions to position-dependent processing times and parallel identical machines. *Journal of the Operational Research Society*, *59*, 1422–1424.

Mosheiov, G., & Strusevich, V. (2017). Determining optimal sizes of bounded batches with rejection via quadratic min-cost flow. *Naval Research Logistics*, *64*, 217–224.

Oron, D. (2008). Single machine scheduling with simple linear deterioration to minimize total absolute deviation of completion times. *Computers and Operations Research*, *35*, 2071–2078.

Ou, J., Zhong, X., & Wang, G. (2015). An improved heuristic for parallel machine scheduling with rejection. *European Journal of Operational Research*, *241*, 653–661.

Pei, J., Cheng, B., Liu, X., Pardalos, P. M., & Kong, M. (2017). Single-machine and parallel-machine serial-batching scheduling problems with position-based learning effect and linear setup time. *Annals of Operations Research*. https://doi.org/10.1007/s10479-017-2481-8.

Rudek, R. (2012). Scheduling problems with position dependent job processing times: Computational complexity results. *Annals of Operations Research*, *196*, 491–516.

Shabtay, D., Gaspar, N., & Kaspi, M. (2013). A survey on offline scheduling with rejection. *Journal of Scheduling*, *16*, 3–28.

Stirzaker, D. (1994). *Elementary probability*. Cambridge: Cambridge University Press.

Sun, L.-H., Cui, K., Chen, J.-H., Wang, J., & He, X.-C. (2013). Research on permutation flow shop scheduling problems with general position-dependent learning effects. *Annals of Operations Research*, *211*, 473–480.

Thevenin, S., Zufferey, N., & Widmer, M. (2015). Metaheuristics for a scheduling problem with rejection and tardiness penalties. *Journal of scheduling*, *18*, 89–105.

Wang, D. J., Yin, Y., & Liu, M. (2016). Bicriteria scheduling problems involving job rejection, controllable processing times and rate-modifying activity. *International Journal of Production Research*, *54*, 3691–3705.

Yang, D. L., & Kuo, W. H. (2009). Single machine scheduling with both deterioration and learning effects. *Annals of OR*, *172*, 315–327.

Yang, D. L., & Kuo, W. H. (2010). Some scheduling problems with deteriorating jobs and learning effects. *Computers & Industrial Engineering*, *58*, 25–28.

Zhong, X., Pan, Z., & Jiang, D. (2017). Scheduling with release times and rejection on two parallel machines. *Journal of combinatorial Optimization*, *33*, 934–944.