Single machine scheduling to minimise number of expedited jobs

S.S. Panwalkar

Carey Business School,
Johns Hopkins University,
100 International Drive,
Baltimore, MD 21202, USA
E-mail: panwalkar@jhu.edu

Abstract: We consider a static, deterministic, *n* jobs-single machine model in which jobs have release dates that can be expedited, if needed, to keep the machine busy without idle time. The objective is to minimise the number of expedited jobs. We propose a polynomially-bounded solution procedure [$O(n \log n)$] that is antithetic to a well-known procedure for minimising the number of tardy jobs.

Keywords: single machine; scheduling; expediting.

Reference to this paper should be made as follows: Panwalkar, S.S. (2013) 'Single machine scheduling to minimise number of expedited jobs', *Int. J. Planning and Scheduling*, Vol. 1, No. 4, pp.235–241.

Biographical notes: S.S. Panwalkar teaches operations management-related subjects in the Carey Business School. His major research interests are in deterministic scheduling and scheduling practices. He has published several papers on scheduling in *Management Science*, *Operations Research*, *European J. of Operational Research*, *Int. J. of Production Research*, *J. of Operational Res. Society*, *Naval Research Logistics* and many other journals.

1 Introduction

The origin of the deterministic single machine sequencing problems can be traced to Smith (1956) who proposed a simple solution to the problem involving minimisation of the mean flow time. Development of research in early years was somewhat slower; Moore (1968) developed an elegant algorithm to minimise number of tardy jobs (with modifications proposed by Hodgson in the same paper). Some of the work done in early seventies can be summarised by a more general model discussed by Garry and Johnson (1979) called 'sequencing to minimise tardy tasks' (problem SS2). Under this model there can be precedence constraints, tardy jobs may or may not be permitted (the latter case is defined as deadlines instead of due dates) , processing times can be randomly generated or equal (assumed to be one unit) and there may be individual release dates.

Of particular interest for this paper is the model considered by Moore and we will refer to Moore's algorithm as the MH algorithm. In the present paper, we will consider Moore's problem without due dates but with individual release dates that can be expedited by the scheduler and propose a polynomial time algorithm labelled 'EJ' (for expedited jobs).

A set of n jobs is to be scheduled for processing on a single machine. No preemption is allowed. Let $p_i > 0$ denote the processing time of job J_i with $P = \Sigma p_i$. Let $r_i \geq 0$ denote arrival (or release) time and let t_i represent the scheduled starting time. Position of a job in the sequence will be denoted by the square bracket [ ] around the subscript. If $t_i < r_i$, the job cannot be started unless the arrival time is expedited to the starting time (or earlier). The objective is to find a schedule that minimises the number of EJ. We will assume processing and release times to be integer-valued.

The motivation for the problem arises from the following situation. Prior to making a firm schedule, various customers have been given a tentative time (release time) when they can bring in the job with a promise that the job will be completed in the time window [0, P]. The single resource to be used is expensive and we do not wish to leave any idle time on the resource. Since a job cannot be started until it is released, it is possible to have idle time gaps in the schedule. We can expedite some jobs (and remove the idle time) by asking some customers to bring in their work earlier. There is some fixed cost associated with expediting (it can be in terms of fixed discount offered); hence the objective of minimising the number of EJ.

Single machine problems continue to generate considerable interest judging from the large number of publications appearing since nineteen nineties; we will list only a few involving minimisation of tardy jobs as well as a few involving release times.

Papers on number of tardy jobs include complexity issues (Chen and Bulfin, 1993), weighted tardiness objective (Sevaux and Dauzère-Pérès, 2003), additional objective functions or additional constraints. We will discuss a few of the latter two categories. Hoogeveen and T'kindt (2012) allow start time of the machine to be a variable and consider two criteria: minimising the number of tardy jobs and the cost of starting the machine at selected time. They propose an SPT-based algorithm. Shabtay (2008) considered different single machine due date assignment problems, one of which involves multiple costs including the cost due to number of tardy jobs. He et al. (2007) considered a single machine problem with controllable processing times and performance criteria involving cost of compression and number of tardy jobs. They analyse different cases involving discrete as well as continuous values for compression times and a case involving no tardiness allowed (by compressing each job as needed). Models involving additional constraints include downtime due to machine maintenance (Chen, 2009; Liao and Chen, 2003) and learning effects (Mosheiov and Sidney, 2005).

Models using release times include release times as decision variables (see for example, Wisner 1995). On the other hand, Cheng et al. (2006) and Shakhlevich and Strusevich (2006) consider changing release times in a different context. To the best of our knowledge, however, no paper involving release time expediting as an objective has been considered in the literature.

In Section 2, we will present the EJ algorithm with a numerical example followed by the proof of optimality in Section 3. While a connection between the objectives minimising number of tardy jobs and minimising number of EJ may not be obvious, the two solution procedures (MH and EJ) are in fact antithetic. We will discuss this and other points in Section 4.

2 Algorithm EJ with a numerical example

From logical point of view, we would not specify a release date larger than P (in fact, the largest value would be $(P - p_i)$. However, mathematically we could have $r_i \geq 0$. In that case, any job with $r_i > (P - p_i)$ must be expedited in every schedule. We take care of this in Step 0 of the proposed algorithm.

Step 0 Start with an initial sequence $\tau = \{\pi', \pi\}$ as follows. $J_i \in \pi'$ if $r_i > (P - p_i)$. Let the cardinality of π' be $|\pi'| = m$, $0 \leq m \leq n$. All jobs in π' are expedited; set the release date of all jobs in π' to zero and calculate start times for all jobs. If $(n - m) \leq 1$, go to Step 4; otherwise put the remaining jobs in π so that for the initial sequence, $r_{[i]} \leq r_{[i+1]}$ for $i = 1, 2, \ldots, n - m - 1$. Break ties with SPT. Set $k = n$.

Step 1 If $r_{[k]} \leq t_{[k]}$ go to Step 3.

Step 2 Remove the job with the largest processing time among positions k to n from π and put it in π' in position 1 and assign release date equal to 0. Recalculate the start times for all jobs and return to Step 1.

Step 3 Set $k = k - 1$. If $r_{[k]} > 0$, return to Step 1.

Step 4 Set $\{\pi', \pi\}$ now represents an optimal schedule.

2.1 Numerical example

Jobs in the example are already numbered in the ascending order of release dates.

Job	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
Pr. time	5	1	18	8	15	4	6	6	3
Release date	4	7	28	32	32	48	51	57	65
Start date	0	5	6	24	32	47	51	57	63

Step 0 Job 9 has a release date beyond $P = 63$; move it to π with release date equal to 0 $\{\pi' = 9, \pi = 1, 2, 3, 4, 5, 6, 7, 8\}$.

Job	*9*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Pr. time	3	5	1	18	8	15	4	6	6
Release date	0	4	7	28	32	32	48	51	57
Start date	0	3	8	9	27	35	50	54	60

Steps 1–3 Used number of times as follows:

- $t_4 < r_4$, move job 5 to position 1 $\{\pi' = 5, 9, \pi = 1, 2, 3, 4, 6, 7, 8\}$
- $t_3 < r_3$, move job 3 to position 1 $\{\pi' = 3, 5, 9, \pi = 1, 2, 4, 6, 7, 8\}$.

Step 4 No more start times before release times. The final schedule is shown below.

Job	*3*	*5*	*9*	*1*	*2*	*4*	*6*	*7*	*8*
Pr. time	18	15	3	5	1	8	4	6	6
Release date	0	0	0	4	7	32	48	51	57
Start date	0	18	33	36	41	42	50	54	60

3 Proof of optimality of the EJ algorithm

We will prove the optimality by converting the problem of minimising the number of tardy jobs and show that our procedure is the MH algorithm in reverse. Here we will assume that $r_i \leq P$ since jobs with larger values of release dates must be expedited in all schedules. Define a new problem identical to the proposed problem except that for each job J_i we define the due date $d_i = P - r_i$ and all jobs are available at time zero. The objective function is to minimise the number of tardy jobs. Obviously, we can use MH algorithm to obtain an optimal sequence for the modified problem. Note that in the MH algorithm, $r_i > P$ implies that the job is already late and it will be removed at the beginning. For the proof, we start with two figures.

Figure 1 Starting sequence for MH

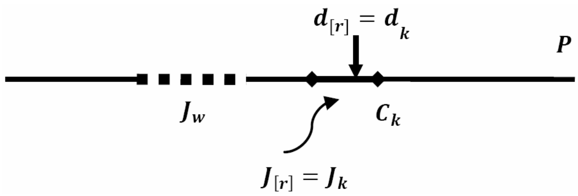
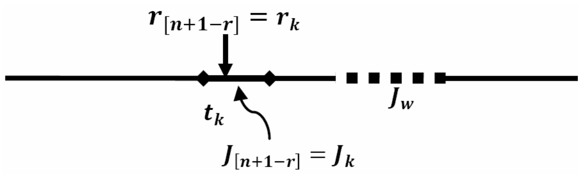


Figure 2 Starting sequence for EJ



In the MH algorithm, the starting point is an EDD sequence (σ); sequence τ for the EJ algorithm is exactly in reverse (the tiebreaking rule in EJ may have to be suitably modified to achieve the reverse sequence).

A job in position $[r]$ in σ will be in position $[n + 1 - r]$ in τ. Suppose this job is J_k. Starting time of J_k in σ will be equal to the completion time of the same job in τ and the completion time of J_k in σ will be equal to the starting time of the same job in τ. The position of d_k in σ will be equal to the position of r_k in τ so that only when J_k is tardy in σ its starting time is less than the release time in τ and the number of tardy jobs in σ is equal to number of jobs needing expediting in τ. Finally, in the MH algorithm, the procedure of removing the largest job up to the first tardy job in σ is the reverse of the job removal step (from π) in the EJ algorithm. (The largest job to be removed, job J_w, is shown with dotted lines in both figures).

4 Discussion

We will look at the EJ algorithm from two different perspectives; first from a theoretical point of view and then from a practical point of view. Conway et al. (1967, p.36) defined scheduling rules (or procedures) as antithetical if they produce opposite sequences in a

n / 1 problem. Rules R and R' are antithetical if a job is assigned to the position i by R is assigned to $(n - i + 1)$ by R'. They gave an example of the SPT vs. LPT rule; the former minimising the mean flow time and the latter maximising it.

Since the MH procedure and the EJ procedures produce opposite sequences, are they antithetical? Minimisation of number of tardy jobs (MH objective) and minimisation of EJ (EJ objective) appear to have no connection. Perhaps, we can view the minimisation of EJ from another perspective as discussed in the optimality proof.

Another objective, if not very common, is to minimise the number of early jobs. This objective can be important if the storage space for completed job is limited. This objective was considered by Oguz and Cheng (1995) in conjunction with deadlines (rather than due dates) and found it to be NP-hard. Huang and Yang (2007) considered a problem similar to Moore's problem except for the objective function (minimising number of early jobs instead of minimising number of tardy jobs) and they solved the problem by arranging the jobs in the ascending order of their slack values. Like the EJ algorithm, they scheduled the jobs starting from the back.

Would a procedure for minimising the number of early jobs be antithetic to the MH algorithm with suitable modifications? The answer is mixed, both yes and no. The answer is no because one can always schedule each job after its due date (leaving the machine idle) so that no jobs will be early. However, if we impose the restriction that the jobs must be scheduled from time zero to P continuously, one can add a pseudo-due date d_i' for job J_i equal to $d_i' = P - d_i + p_i$ and solve the new problem with the MH algorithm to minimise the number of tardy jobs. The sequence in reverse will be optimal for the problem of minimising the number of early jobs and we have the antithetic rules for two opposite performance measures.

We can also discuss the complexity of the solution procedure. It has been shown (see for example, Dekel and Sahni, 1983) that the MH algorithm has a complexity of $O(n \log n)$. Since the proposed algorithm mirrors the MH algorithm, the complexity of EJ will also be $O(n \log n)$. A more general problem (SS2) defined by Garry and Johnson (1979) can be can be NP hard even for a case involving unit processing times. On the other hand, the proposed problem, with unit processing times will have complexity of $O(n)$, since sorting for problems involving consistent due dates (release times in our case) can be done faster (see Monma, 1982).

Although the EJ algorithm gives an optimal solution, one may wish to 'improve' the solution from a practical point of view. We can introduce the secondary objective of reducing the sum of the differences in the release times of the EJ. We can make certain obvious modifications to the EJ algorithm to achieve this secondary goal as much as possible.

1 In Step 2, when removing the largest job from π, we can break ties by selecting the job with the smallest release value.

2 For jobs in π' in the final sequence, set release time to the actual start time.

3 We can use the following heuristic [see the difference between a heuristic and a priority rule in Gere (1966)]. Move any jobs in π' and manually insert it in π in an appropriate position in the final sequence without increasing the number of EJ. For example, a revised schedule (with jobs 5 and 9 inserted in new positions) is shown below.

Job	*3*	*1*	*2*	*5*	*4*	*9*	*6*	*7*	*8*
Pr. time	18	5	1	15	8	3	4	6	6
Release date	0	4	7	24	32	47	48	51	57
Start date	0	18	23	24	39	47	50	54	60

While these steps do not guarantee that the secondary objective is achieved mathematically, it will be a substantial improvement over the original solution.

References

Chen, C.L. and Bulfin, R.L. (1993) 'Complexity of single machine, multi-criteria scheduling problems', *European Journal of Operational Research*, Vol. 70, No. 1, pp.115–125.

Chen, W.J. (2009) 'Minimizing number of tardy jobs on a single machine subject to periodic maintenance', *Omega*, Vol. 37, No. 3, pp.301–309.

Cheng, T.C.E., Kovalyov, M.Y. and Shakhlevich, N.V. (2006) 'Scheduling with controllable release dates and processing times: total completion time minimization', *European Journal of Operational Research*, Vol. 175, No. 2, pp.769–781.

Conway, R.W., Maxwell, W.L. and Miller, L.W. (1967) *Theory of Scheduling*, Addison-Wesley Publishing Co., Reading, MA.

Dekel, E. and Sahni, S. (1983) 'Parallel scheduling algorithms', *Operations Research*, Vol. 31, No. 1, pp.24–49.

Garry, M.R. and Johnson, D.S. (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman Press, San Francisco.

Gere, W.S. (1966) 'Heuristics in job shop scheduling', *Management Science*, Vol. 13, No. 3, pp.167–190.

He, Y., Wei, Q. and Cheng, T.C.E. (2007) 'Single-machine scheduling with trade-off between number of tardy jobs and compression cost', *Journal of Scheduling*, Vol. 10, Nos. 4–5, pp.303–310.

Hoogeveen, H. and T'kindt, V. (2012) 'Minimizing the number of late jobs when the start time of the machine is variable', *Operations Research Letters*, Vol. 40, No. 5, pp.353–355.

Huang, R.H. and Yang, C.L. (2007) 'Single-machine scheduling to minimize the number of early jobs', *IEEE International Conference on Industrial Engineering and Engineering Management*, pp.955–957.

Liao, C.J. and Chen, W.J. (2003) 'Single-machine scheduling with periodic maintenance and non-resumable jobs', *Computers & Operations Research*, Vol. 30, No. 9, pp.1335–1347.

Monma, C.L. (1982) 'Linear time algorithms on parallel processors', *Operations Research*, Vol. 30, No. 1, pp.116–124.

Moore, J.M. (1968) 'An n job, one machine sequencing algorithm for minimizing the number of late jobs', *Management Science*, Vol. 15, No. 1, pp.102–109.

Mosheiov, G. and Sidney, J.B. (2005) 'Note on scheduling with general learning curves to minimize the number of tardy jobs', *Journal of the Operational Research Society*, Vol. 56, No. 1, pp.110–112.

Oguz, C. and Cheng, T.C.E. (1995) 'Minimizing number of early jobs on a single machine', Paper presented at the *INFORMS Fall Conference*, 29 October to 1 November, New Orleans, LA.

Sevaux, M. and Dauzère-Pérès, S. (2003) 'Genetic algorithms to minimize the weighted number of late jobs on a single machine', *European Journal of Operational Research*, Vol. 151, No. 2, pp.296–306.

Shabtay, D. (2008) 'Due date assignments and scheduling a single machine with a general earliness/tardiness cost function', *Computers and Operations Research*, Vol. 35, No. 5, pp.1539–1545.

Shakhlevich, N.V. and Strusevich, V.A. (2006) 'Single machine scheduling with controllable release and processing parameters', *Discrete Applied Mathematics*, Vol. 154, No. 15, pp.2178–2199.

Smith, W.E. (1956) 'Various optimizers for single-stage production', *Naval Research Logistics Quarterly*, Vol. 3, Nos. 1–2, pp.159–162.

Wisner, J.D. (1995) 'A review of the order release policy research', *International Journal of Operations & Production Management*, Vol. 15, No. 6, pp.25–40.