

## Single Machine Schedule Problem with a Fixed Unavailability Interval and Rejection

Xie Xie<sup>\*1, a</sup>, Xiangyu Kong<sup>2, b</sup> and Yongyue Zheng<sup>3, c</sup>

<sup>1 2</sup> Key Laboratory of Manufacturing Industrial and Integrated Automation, Shenyang University, China

<sup>3</sup> Liaoning Institute of standardization, Shenyang, China

<sup>a</sup>xiexie8118@gmail.com, <sup>b</sup>623069905@qq.com, <sup>c</sup>zhengyongyue@163.com

**Keywords:** Scheduling, Unavailability interval, Rejection, Heuristic.

**Abstract.** We investigate a single machine scheduling problem with a fixed unavailability interval on machine and rejection job. The objective is to minimize the sum of the flow-time of the accepted jobs and penalties of rejected jobs. For the demonstrated NP-hard problem, we first propose a dynamic programming algorithm for solving small scale problem optimally. Further we propose an approximation algorithm to obtain a better worst case bound.

### Introduction

In most classical scheduling problems, all jobs must be produced on the machines, and rejection is not allowed for the jobs. However, in order to reduce manufacturing costs and obtain maximal profits, the manufacturers often rejected some jobs which have long producing time and bring relatively small profits. Rejecting jobs frequently lead to low prestige for the manufacturer. Most scheduling problems literatures studied all machines are available, but in industry exist some reasons that make the machines may not produce all the time. Based on this phenomenon more and more scholars are interested the scheduling problem with unavailability interval. This problem is very practical for the situation, regularly happened in the industry. Such as manufacturers want to decrease the completing time and cut down the cost, they chose to reject some jobs with long producing time, accordingly, which case a penalty has to be paid. While some reasons can cause unavailability interval, if the scheduling problem happened for maintenance we call it deterministic, or the problem happen by breakdown we call the situation is stochastic. In this paper we consider a single machine scheduling problem with a fixed unavailability interval on machine and rejection job.

The machine scheduling problem with rejection was first introduced by Bartal et al. [1] and the objective is minimizing the sum of the makespan for the accepted jobs plus the total rejection penalties of the rejected jobs. Considered the situations of on-line and off-line, respectively proposed on-line algorithm and a full polynomial time approximation scheme(FPTAS). Numerous references studied the similar problems, they not only proposed a heuristic, but also given a full polynomial time approximation scheme, such as Zhang et al. [2] considered the single machine scheduling with release dates and rejection to minimize the sum of makespan of accepted jobs plus the total penalties of the rejected jobs. They proposed a 2- approximation algorithm, and designed a full polynomial time approximation scheme.

The machine scheduling problem with unavailability interval was first proposed by Adiri [7], the objective is to minimize the completion time. The first approved the problem is NP-hard, and by SPT algorithm to obtain the objective value, the worst case is  $2/7$ . He et al. [3] considered the non-resumable case of single machine scheduling problem with a deterministic unavailability interval, the objective is to minimize the completion time of all jobs, and they modified the SPT algorithm obtain an heuristic algorithm, and approved the worst case is  $1 + 2/5 + 2\sqrt{2k+8}$ . Kacem and Mahjoub [4] proposed a faster FPTAS for the weighted flow-time minimization on single machine with a deterministic non-availability interval, and the running time of this algorithm is

$O(\varepsilon^2/n^2)$ . Kacem and Kellerer. [5] studied single machine scheduling problem with releases dates, unavailability interval, for the problem is NP-hard, they propose a 2-approximation algorithm. Kacem and Chu [6] studied the single machine scheduling problem with unavailability problem, the objective is to minimize the completion time, and they assumed that the job non-resumable, last proposed both WSPT1 and MWSPT2 rules.

Based on the literatures above, most scholars only considered one subject, just a part of scholars considered two subjects that rejection and unavailability interval in scheduling problem. Although Zhang and Luo [9] studied the same problem of ours and they proposed a heuristic to obtain a approximate value, and approved the worst case is 4, in this paper, we improve the heuristic to obtain the approximation value for this problem, but the worst case is  $2 + 4/5 + 2\sqrt{2k+8}$  (where  $k$  is the number of the iterations).

### Problem Formulation

We denote the scheduling problem is  $P$ , at the same time the machine can process only one job. Have a set of jobs are  $j = \{1, 2, \dots, n\}$  that process on the single machine, corresponding the job  $J_j$  processing time is  $p_j$  and the penalty is  $w_j$ ,  $[T_1, T_2]$  is the unavailability interval, in this interval any jobs do not be processed. We denote the problem is  $1, h_1 | reject | \sum_{j \in A} C_j + \sum_{j \in R} w_j$ , and  $A$  is the set of all accepted jobs,  $R$  is the set of all rejected jobs,  $\sum_{j \in A} C_j$  denotes the flow-time of all accepted jobs,  $\sum_{j \in R} w_j$  is the total penalties of all rejected jobs. Our objective is minimum the total flow-time of accepted jobs and the sum of penalties rejected jobs.

As Adiri et al. [7] and Lee and Liman [8] have proved the problem  $1|h_1|\sum_{j \in A} C_j$  is NP-hard, obviously, at least our problem is NP-hard. First we index all of jobs by *SPT* algorithm, (i.e  $p_1 \leq p_2 \leq \dots \leq p_n$ ). If all the accepted jobs can finish before  $T_1$ , the optimal solution obtained by the *SPT* rule (Smith, 1956), so we assume that at least one job finished after  $T_2$ .

### Dynamic Programming Algorithm

Due to the dominance of the *SPT* order, an optimal schedule is composed of two sequences of accepted jobs scheduled in non-decreasing order. So first we index all jobs by *SPT* rule. We can obtain an optimally solved by applying the dynamic programming algorithm. When we consider jobs  $J_1, \dots, J_j$ , denote  $f_j^{(V)}$  as the optimal value of the front of  $j$  jobs, and the total rejection penalty is exactly  $V$ . For the job  $J_j$  we have two cases, accept or reject.

**Step 1.** When  $j = 1$ , we have:

$$f_1^{(V)} = \begin{cases} w_1 \\ p_1 \\ T_2 + p_1 \end{cases}.$$

**Step 2.** When  $j = 2, \dots, n$ , and job  $J_j$  has been rejected, we have:

$$f_j^{(V)} = f_{j-1}^{(V-w_j)} + w_j.$$

**Step 3.** When job  $J_j$  has been accepted, we have two cases:

**Case 1.**  $J_j$  finished before  $T_1$ , and the optimal value of  $J_1, \dots, J_{j-1}$  is  $f_{j-1}^{(V)}$ , the total flow-time is  $f_{j-1}^{(V)} - V$ . So we have:

$$f_j^{(V)} = \sum_{i=1}^j p_i + f_{j-1}^{(V)}.$$

**Case 2.** When job  $J_i$  finished after  $T_2$ , we have:

$$\begin{aligned} f_j^{(V)} &= \max \{ f_{j-1}^{(V)} - V, T_2 \} + p_i + f_{j-1}^{(V)} - V + V \\ &= \max \{ f_{j-1}^{(V)} - V, T_2 \} + p_i + f_{j-1}^{(V)}. \end{aligned}$$

**Step 4.** The optimal value is  $\min \{ f_n^{(V)} : 0 \leq V \leq W \}$  ( $W = \sum_{j \in R} w_j$ ).

Algorithm  $D$  always obtains an optimal value in  $O(nV)$  time.

### Algorithm $H$

We propose a heuristic solution this problem, and denote algorithm  $H$ . It based on the following algorithm, which extends the one proposed by He et al.[3]. First we denote index all jobs as  $SPT$  (shortest processing time first) rule,  $A$  is the set of jobs scheduled before or at  $T_1$ , and by  $B$  the set of remaining jobs after  $T_2$ . Let  $a \leq |A|$ ,  $b \leq |B|$  and satisfying  $k \geq a \geq b$ , under the constraint that the total processing times of  $a$  jobs in  $A$  plus  $\delta$  is no less than the total processing times of  $b$  jobs in  $B$ , denote  $\delta$  the idle time on the machine before the maintenance.

**Step 1.** Process all the jobs according to the  $SPT$  rule.

**Step 2.** Given a positive integer  $k$ , try all  $k$ , do  $k$ -exchange procedure is an exchange of  $a$  jobs in  $A$  with  $b$  jobs in  $B$  generate new schedule.

**Step 3.** Choose the minimum one from the schedules in steps 1 and 2, and denote the schedule is  $\Pi_H$ , the minimum value  $\lambda_H = \sum C_j$ .

**Step 4.** Let the jobs in  $\Pi_H$ , and if the job  $J_i$  satisfied  $C_i \leq w_i$  process on the machine, otherwise the job is rejected, at the same time we denote the schedule is  $\Pi'_H$ ,  $\varphi_H(P) = \sum_{J_i \in \Pi'_H} C_j + \sum_{J_i \in R} w_j$ .

**Theorem 1.**  $\varphi_H(P) \leq 2 + 4/5 + 2\sqrt{2k+8} \varphi^*(P)$ .

**Proof.** We can obtain an objective value  $\varphi_H(P)$  of algorithm  $H$ . The step 1 to step 3 of algorithm  $H$  can obtain  $\lambda_H$ , and about problem  $1|h| \sum C_j$ , He et al[3] has propose a worst case analysis, corresponding the optimal schedule is  $\Pi^*$ , and the optimal value is  $\lambda^*$ , satisfied  $\lambda_H \leq 1 + \frac{2}{5 + 2\sqrt{2k+8}} \lambda^*$ . An optimal scheduling of our problem is  $\Pi^*$ , some jobs can be rejected,

then the remaining jobs propose a scheduling is  $\Pi^{**}$ . We have:

$$\lambda_H = \sum_{J_i \in \Pi_H} C_j = \sum_{J_i \in \Pi'_H} C_j + \sum_{J_i \in R} C_j \geq \sum_{J_i \in \Pi'_H} C_j + \sum_{J_i \in R} w_j = \varphi_H(P).$$

$$\lambda_H = \sum_{J_i \in \Pi_H} C_j \leq 1 + \frac{2}{5 + 2\sqrt{2k+8}} \lambda^* = 1 + \frac{2}{5 + 2\sqrt{2k+8}} \left( \sum_{J_i \in \Pi^*} C_j \right)$$

$$= 1 + \frac{2}{5 + 2\sqrt{2k+8}} \left( \sum_{J_i \in \Pi^*} C_j + \sum_{J_i \in R'} C_j \right)$$

$$\leq 1 + \frac{2}{5 + 2\sqrt{2k+8}} \left( \sum_{J_i \in \Pi^*} C_j + \sum_{J_i \in R'} \max \{ C_j, w_j \} \right)$$

$$\leq 1 + \frac{2}{5 + 2\sqrt{2k+8}} (\varphi^*(P) + \varphi^*(P))$$

$$= 2 + \frac{4}{5 + 2\sqrt{2k+8}} \varphi^*(P)$$

So  $\varphi_H(P) \leq 2 + 4/5 + 2\sqrt{2k+8} \varphi^*(P)$ . It is obvious that the algorithm  $H$  runs in time  $O(n^{2k+1})$ .

## Conclusions

In this paper, we consider the single machine scheduling problem with rejection and a fix unavailability interval. Since the problem is NP-hard we proposed a dynamic programming and improved a proposed heuristic algorithm with the worst-case ratio 4. The worst case of our algorithm is demonstrated as  $2 + 4/5 + 2\sqrt{2k+8}$  (where  $k$  is the number of the iterations).

## Acknowledgement

This research is supported by National Natural Science Foundation of China (Grant No. 71201104). The colleges and universities of Liaoning Province outstanding talent support plan (LJQ2014133).

## References

- [1] Y. Bartal, S. Leonardi, A.M. Spaccamela, J. Sgall, L. Stougie, Multi-processor scheduling with rejection, *SIAM Journal on Discrete Mathematics*, 13 (2000), 64-78.
- [2] L.Q. Zhang, L. F, J.J. Yuan, Single machine scheduling with release dates and rejection, *European Journal of Operation Research*, 198 (2009), 975-978.
- [3] Yong He, Weiya Zhong, Huikun Gu, Improved algorithms for two single machine scheduling problems, *Theoretical Computer Science*, 363 (2006), 257-265.
- [4] I. Kacem, A.R. Mahjoub, Fully polynomial time approximation scheme for the weighted flow-time minimization on a single machine with a fixed non-availability interval, *Computers and Industrial Engineering*, 56 (2009), 1708-1712.
- [5] I. Kacem, H. Kellerer, Fast approximation algorithms to minimize a special weighted flow-time criterion on a single machine with a non-availability interval and release dates, *Journal of Scheduling*, 14 (2011), 257-265.
- [6] I. Kacem, C. Chu, Worst-case analysis of the WSPT and MWSPT rules for single machine scheduling with one planned setup period, *European Journal of Operational Research*, 187 (2008), 1080-1089.
- [7] I. Adiri, J. Bruno, E. Frostig, A.H.G. Rinnooy Kan, Single machine flow-time scheduling with a single breakdown, *Acta Inform*, 26 (1989), 679-696.
- [8] C.Y. Lee, S.D. Liman, Single machine flow-time scheduling with scheduled maintenance, *Acta Inform*, 29 (1992), 375-382.
- [9] M. Zhang, C. Luo, Single machine scheduling problem with job rejection and machine unavailability interval. Natural Science Edition, *Journal of Chongqing Normal University*, 29 (2012) 5-19.