New Fully Polynomial Time Approximation Scheme for the makespan minimization with positive tails on a single machine with a fixed non-availability interval

Imed Kacem

LITA, Universit Paul Verlaine–Metz, Ile du Saulcy. Metz 57000 cedex, France.
(Email address: kacem@univ-metz.fr).

Key words:
scheduling, non-availability constraint, approximation, makespan

1 Introduction

The studied problem $(\mathcal{P})$ can be formulated as follows. We have to schedule a set J of n jobs on a single machine, where every job j has a processing time p_j and a tail q_j. The machine can process at most one job at a time and it is unavailable between T_1 and T_2 (i.e., $[T_1, T_2)$ is a forbidden interval). Preemption of jobs is not allowed (jobs have to be performed under the non-resumable scenario). All jobs are ready to be performed at time 0. With no loss of generality, we consider that all data are integers and that jobs are indexed according to Jackson's rule [1] (i.e., jobs are indexed in nonincreasing order of tails). Therefore, we assume that $q_1 \geq q_2 \geq ... \geq q_n$. The consideration of tails is motivated by the large set of scheduling problems such that jobs have delivery times after their processing. As an example, it is well-known that the minimization of makespan with tails is equivalent to the minimization of the maximum lateness with due dates [2]. Let $C_j(S)$ denote the completion time of job j in a feasible schedule S for the problem and let $\varphi_S(\mathcal{P})$ be the makespan yielded by schedule S for instance $\mathcal{I}$ of $(\mathcal{P})$:

$$\varphi_S(\mathcal{I}) = \max_{1 \leq j \leq n} \left(C_j(S) + q_j\right) \tag{1}$$

The aim is to find a feasible schedule S by minimizing the makespan. Due to the dominance of Jackson's order, an optimal schedule is composed of two sequences of jobs scheduled in nondecreasing order of their indexes.

If all the jobs can be inserted before T_1, the instance studied ($\mathcal{I}$) has obviously a trivial optimal solution obtained by Jackson's rule. We therefore consider only the problems in which all the jobs cannot be scheduled before T_1.

In the remainder of this paper $\varphi^*(\mathcal{I})$ denotes the minimal makespan for instance $\mathcal{I}$.

This type of problems has been studied in the literature under various criteria (a sample of these works includes Lee [7], Kacem [4], Kubzin and Strusevich [6], Qi *et al.* [8]-[9], Schmidt [10], He *et al.* [3]). However, few papers studied the problem we consider in this paper. Lee [7] explored the Jackson's sequence JS and proved that its deviation to the optimal makespan cannot exceed $\max_{1\leq j\leq n}(p_j)$, which is equivalent to state that $\varphi_{JS}(\mathcal{I}) \leq 2\varphi^*(\mathcal{I})$. Recently, Yuan et al. developed an interesting PTAS for the studied problem [11]. That is why this paper is a good attempt to design more efficient approximation heuristics and approximation schemes to solve the studied problem.

2 New FPTAS

Now, let describe our FPTAS. It uses a simplification technique based on merging small jobs [5]. First, we simplify the instance $\mathcal{I}$ as follows. Given an arbitrary $\varepsilon > 0$, we split the interval $[0, \max_{j\in J}\{q_j\}]$ in $1/\varepsilon$ equal lenght intervals and we round up every tail q_j to the next multiple of $\varepsilon\overline{q}$ ($\overline{q} = \max_{j\in J}\{q_j\}$). Then, we obtain a new instance $\mathcal{I}'$ with no $(1+\varepsilon)$-loss. Thus, J can be divided into $1/\varepsilon$ subsets $J(k)$ ($1 \leq k \leq 1/\varepsilon$) where jobs in $J(k)$ have identical tails of $k\varepsilon\overline{q}$. The second modification consists in reducing the number of small jobs in every subset $J(k)$. Small jobs are those having processing times $< \varepsilon P/2$ where $P = p_1 + p_2 + ... + p_n$. The reduction is done by merging the small jobs in each $J(k)$ so that we obtain new greater jobs having processing times between $\varepsilon P/2$ and εP. At most, for every subset $J(k)$, a single small job remains. We show that this reduction cannot increase the optimal solution value by more than $(1+\varepsilon)$-factor. We re-index jobs according to nondecreasing order of their tails. The new instance we obtain is denoted as $\mathcal{I}''$. Clearly, the number of jobs remaining in the simplified instance $\mathcal{I}''$ is less than $3/\varepsilon$.

Our FPTAS is based on two steps. First, we use the Jackson's sequence JS obtained for the initial instance $\mathcal{I}$. Then, we apply a modified dynamic programming algorithm APS'_ε on instance $\mathcal{I}''$. The main idea of APS'_ε is to remove a special part of the states generated by a dynamic programming algorithm (See Kacem [4]). Therefore, the modified algorithm becomes faster and yields an approximate solution instead of the optimal schedule.

Given an arbitrary $\varepsilon > 0$, we define $\overline{n} = \min\{n, 3/\varepsilon\}$, $\omega_1 = \left\lceil \frac{4\overline{n}}{\varepsilon} \right\rceil$, $\omega_2 = \left\lceil \frac{2\overline{n}^2}{\varepsilon} \right\rceil$,

$\delta_1 = \frac{\varphi_{JS}(\mathcal{I})}{\omega_1}$and $\delta_2 = \frac{T_1}{\omega_2}$.

We split $[0, \varphi_{JS}(\mathcal{I}))$ into ω_1 equal subintervals $I^1_m = [(m-1)\delta_1, m\delta_1)_{1\le m\le\omega_1}$. We also split $[0, T_1)$ into ω_2 equal subintervals $I^2_s = [(s-1)\delta_2, s\delta_2)_{1\le s\le\omega_2}$ of length δ_2. Moreover, we define the two singletons $I^1_{\omega_1+1} = \{\varphi_{JS}(\mathcal{I})\}$ and $I^2_{\omega_2+1} = \{T_1\}$. Our algorithm APS'_ε generates reduced sets $\mathcal{X}^{\#}_k$ of states $[t, f]$ where t is the total processing time of jobs assigned before T_1 in the associated partial schedule and f is the makespan of the same partial schedule. It can be described as follows:

Algorithm APS'_ε

(i). Set $\mathcal{X}^{\#}_0 = \{[0, 0]\}$.

(ii). For $k \in \{1, 2, 3, ..., \overline{n}\}$,

For every state $[t, f]$ in $\mathcal{X}^{\#}_{k-1}$:

1) Put $\left[t, \max\left(f, T_2 + \sum_{i=1}^{k} p_i - t + q_k\right)\right]$ in $\mathcal{X}^{\#}_k$

2) Put $[t + p_k, \max(f, t + p_k + q_k)]$ in $\mathcal{X}^{\#}_k$ if $t + p_k \le T_1$

Remove $\mathcal{X}^{\#}_{k-1}$

Let $[t, f]_{m,s}$ be the state in $\mathcal{X}^{\#}_k$ such that $f \in I^1_m$ and $t \in I^2_s$ with the smallest possible t (ties are broken by choosing the state of the smallest f).

Set $\mathcal{X}^{\#}_k = \left\{[t, f]_{m,s} | 1 \le m \le \omega_1 + 1, 1 \le s \le \omega_2 + 1\right\}$.

(iii). $\varphi_{APS'_\varepsilon}(\mathcal{I}) = \min_{[t,f]\in\mathcal{X}^{\#}_n}\{f\}$.

Theorem 1 *Given an arbitrary $\varepsilon > 0$, algorithm APS'_ε yields an output $\varphi_{APS'_\varepsilon}(\mathcal{I}'')$ such that:*

$$\varphi_{APS'_\varepsilon}(\mathcal{I}'') - \varphi^*(\mathcal{I}'') \le \varepsilon\varphi^*(\mathcal{I}''). \tag{2}$$

The proof will be presented at the conference.

Lemma 2 *Given an arbitrary $\varepsilon > 0$, algorithm APS'_ε can be implemented in $O(n \log n + \min\{n, 1/\varepsilon\}^4/\varepsilon^2)$ time.*

The schedule obtained by APS'_ε for instance $\mathcal{I}''$ can be easily converted into a feasible one for instance $\mathcal{I}$. This can be done in $O(n)$ time. ¿From the previous lemma and theorem, the main result is proved and the following corollary holds.

Corollary 3 *Algorithm APS'_ε is an FPTAS and it can be implemented in $O(n \log n + \min\{n, 1/\varepsilon\}^4/\varepsilon^2)$ time.*

3 Conclusion

In this paper, we considered the non-resumable case of the single machine scheduling problem with a fixed non-availability interval. Our aim is to minimize the makespan when every job has a positive tail. We showed that the problem has an FPTAS (Fully Polynomial Time Approximation Scheme). Such an FPTAS is strongly polynomial. The obtained results outperform the previous polynomial approximation algorithms for this problem.

References

[1] Carlier, J., 1982. The one-machine sequencing problem. *European Journal of Operational Research* 11, 42-47.

[2] Dessouky, M.I., Margenthaler, C.R., 1972. The one-machine sequencing problem with early starts and due dates. *AIIE Transactions* 4:3, 214-222.

[3] He, Y., Zhong, W., Gu, H., 2006. Improved algorithms for two single machine scheduling problems. *Theoretical Computer Science* 363, 257-265.

[4] Kacem, I., 2009. Approximation algorithms for the makespan minimization with positive tails on a single machine with a fixed non-availability interval. *Journal of Combinatorial Optimization* 17:2, 117-133.

[5] Kacem, I., Kellerer, H., 2010. No-Wait Scheduling of a Single-Machine to Minimize the Maximum Lateness, *Proceedings of the 24^{th} Annual Conference of the Belgian Operational Society*, Liège, January 28-29.

[6] Kubzin, M.A., Strusevich, V.A., 2006. Planning machine maintenance in two machine shop scheduling. *Operations Research* 54, 789-800.

[7] Lee, C.Y., 1996. Machine scheduling with an availability constraints. *Journal of Global Optimization* 9, 363-384.

[8] Qi, X., 2007. A note on worst-case performance of heuristics for maintenance scheduling problems. *Discrete Applied Mathematics* 155, 416-422.

[9] Qi, X., Chen, T., Tu, F., 1999. Scheduling the maintenance on a single machine. *Journal of the Operational Research Society* 50, 1071-1078.

[10] Schmidt, G., 2000. Scheduling with limited machine availability. *European Journal of Operational Research* 121, 1-15.

[11] Yuan, J.J., Shi, L., and Ou, J.W., 2008. Single machine scheduling with forbidden intervals and job delivery times. *Asia-Pacific Journal of Operational Research* 25:3, 317-325.