Approximation Schemes for Minimizing the Maximum Lateness on a Single Machine with Release Times under Non-Availability or Deadline Constraints*

Imed Kacem*, Hans Kellerer+
* LCOMS EA 7306, Université de Lorraine, Metz, France
imed.kacem@univ-lorraine.fr
+ ISOR, Universität Graz, Graz, Austria
hans.kellerer@uni-graz.at

Keywords: Single machine scheduling; release times; lateness; deadlines; approximation algorithms.

1 Problem description

The problem we consider is the one-machine scheduling problem with release dates and delivery times. The objective is to minimize the maximum lateness. Formally, the problem is defined in the following way. We have to schedule a set $J = \{1, 2, \ldots, n\}$ of n jobs on a single machine. Each job $j \in J$ has a processing time p_j, a release time (or head) r_j and a delivery time (or tail) q_j. The machine can only perform one job at a given time. Preemption is not allowed. The problem is to find a sequence of jobs, with the objective of minimizing the maximum lateness $L_{\max} = \max_{1\leq j\leq n} \{C_j + q_j\}$ where C_j is the completion time of job j. We also define s_j as the starting time of job j, i.e., $C_j = s_j + p_j$, then $P = \sum_{j=1}^{n} p_j$ as the total processing time and $p(H) = \sum_{j\in H} p_j$ as the total processing time for a subset of jobs $H \subset J$. Four scenarios are considered:

- Scenario 1: every job should be completed before a deadline d, i.e., $C_{\max} = \max_{1\leq j\leq n}\{C_j\} \leq d$. This scenario is denoted as $1|r_j, C_{\max} \leq d|L_{\max}$ and it is abbreviated as Π_1.

*This work has been published under the reference [4].

- Scenario 2: we are given the one-machine bicriteria scheduling problem with the two objectives $L_{\max}$ and $C_{\max}$, denoted by $1|r_j|L_{\max}, C_{\max}$. We will speak shortly of problem Π_2.
- Scenario 3: the machine is not available during a given time interval $]T_1, T_2[$. This scenario is denoted by $1, h_1|r_j|L_{\max}$. We will speak shortly of problem Π_3. Here, $]T_1, T_2[$ is a machine non-availability (MNA) interval.
- Scenario 4: in this case, the non-availibility interval $]T_1, T_2[$ is related to the operator who is organizing the execution of jobs on the machine. An operator non-availability (ONA) period is an open time interval in which no job can start, and neither can complete. Using an extended notation as in [7], this scenario is denoted by $1, ONA|r_j|L_{\max}$ and abbreviated as Π_4.

2 Our contributions

All four presented scenarios are generalizations of the well-known fundamental problem $1|r_j, q_j|L_{max}$ which has been widely studied in the literature. For the sake of simplicity, this problem will be denoted by Π_0. It is well-known that problem $1|r_j, q_j|L_{max}$ is NP-hard in the strong sense [1]. Therefore, we are interested in the design of efficient approximation algorithms for our problems. For each of the four problems Π_1, Π_2, Π_3 and Π_4 we will present a polynomial time approximation scheme.

First, we consider problem Π_1 with a common deadline d for the jobs. The Schrage sequence shows whether a feasible solution for Π_1 exists or not. If $C_{\max}(\sigma_{Sc}(I)) > d$, then the problem has no feasible solution. Hence, we will assume in the following that $C_{\max}(\sigma_{Sc}(I)) \leq d$. Let $\varepsilon > 0$. A job j is called *large* if $p_j \geq \varepsilon L_{\max}(\sigma_{Sc}(I))/2$, otherwise it is called *small*. Let L be the subset of large jobs. Since $L_{\max}(\sigma_{Sc}(I)) \leq 2L^*_{\max}$ (see Carlier), it can be observed that $|L| \leq 2/\varepsilon$. Let $k = |L|$. We assume that jobs are indexed such that $L = \{1, 2, \ldots, k\}$. Our PTAS is based on the construction of a set of modified instances starting from the original instance I. Set $R = \{r_1, \ldots, r_n\}$ and $Q = \{q_1, \ldots, q_n\}$. Let us define the following sets of heads and tails: $R(i) = \{r_j \in R | r_j \geq r_i\}, \quad i = 1, 2, \ldots, k$, and $Q(i) = \{q_j \in Q | q_j \geq q_i\}, \quad i = 1, 2, \ldots, k$. Now, we define the set of all the combinations of couples $(r, q) \in R(i) \times Q(i)$ for every $i = 1, 2, \ldots, k$: $W = \{(\widetilde{r}_1, \widetilde{q}_1, \widetilde{r}_2, \widetilde{q}_2, \ldots, \widetilde{r}_k, \widetilde{q}_k) | \widetilde{r}_i \in R(i), \widetilde{q}_i \in Q(i), i = 1, 2, \ldots, k\}$ Clearly, the cardinality of W is bounded as follows: $|W| \leq n^{2k} = O(n^{4/\varepsilon})$. Let $w \in W$ with $w = (\widetilde{r}_1, \widetilde{q}_1, \widetilde{r}_2, \widetilde{q}_2, \ldots, \widetilde{r}_k, \widetilde{q}_k)$. Instance I_w is a slight modification of instance I and is defined as follows. I_w consists of large and small jobs. The k large jobs have modified release times $\widetilde{r}_1, \widetilde{r}_2, \ldots, \widetilde{r}_k$ and delivery times $\widetilde{q}_1, \widetilde{q}_2, \ldots, \widetilde{q}_k$. All processing times and the $n - k$ small jobs remain unchanged. Let $\mathcal{I}$ be the set of all possible instances I_w, i.e., $\mathcal{I} = \{I_w | w \in W\}$. It is clear that $|\mathcal{I}|$ is in $O(n^{4/\varepsilon})$. By the modification, the processing times of instances in $\mathcal{I}$ are not changed and release times and delivery times are not decreased.

Let $\widetilde{\mathcal{I}} \subseteq \mathcal{I}$. An instance $I' \in \widetilde{\mathcal{I}}$ is called *maximal* if there is no other instance $I'' \in \widetilde{\mathcal{I}}$ such that for every $i = 1, 2, \ldots, k$, we have $r'_i \leq r''_i$, $q'_i \leq q''_i$ and at least one inequality is strict. Here, r'_i, q'_i denote the heads and tails in I' and r''_i, q''_i the heads and tails in I'', respectively. Our procedure PTAS1 depends on the instance I, the deadline d and the accuracy ε. First, it runs Schrage's algorithm for all instances in $\mathcal{I}$. Then, it selects the best feasible solution, i.e., the best solution with $C_{\max} \leq d$. Finally, it applies the corresponding sequence to the original instance I.

Theorem 1 *Algorithm PTAS1 yields a* $(1+\varepsilon)$*-approximation for* Π_1 *and has polynomial running time for fixed* ε*.*

Based on Theorem 1 we can derive a PTAS for the second scenario $1|r_j|L_{\max}, C_{\max}$. The detail will be given in the presentation. Another important theorem will be explained:

Theorem 2 *Problem* Π_3 *admits a PTAS.*

By using the previous theorem and a similar transformation as proposed in [7], the existence of an PTAS for Scenario Π_4 $(1, ONA|r_j|L_{\max})$ can be proved.

Theorem 3 *Scenario* Π_4 *admits a PTAS.*

3 Conclusions

In this paper, we considered important single-machine scheduling problems under release times and tails assumptions, with the aim of minimizing the maximum lateness. Four variants have been studied under different scenarios. In the first scenario, all the jobs are constrained by a common deadline. The second scenario consists in finding the Pareto front where the considered criteria are the makespan and the maximum lateness. In the third scenario, a non-availability interval constraint is considered (breakdown period, fixed job or a planned maintenance duration). In the fourth scenario, the non-availibility interval is related to the operator who is organizing the execution of jobs on the machine, which implies that no job can start, and neither can complete during the operator non-availability period. For each of these four open problems, we establish the existence of a PTAS.

As a perspective, the study of min-sum scheduling problems in a similar context seems to be a challenging subject.

References

[1] Carlier J, Moukrim A, Hermes F, Ghedira K (2010) Exact resolution of the one-machine sequencing problem with no machine idle time. *Computers and Industrial Engineering* 59(2): 193-199.

[2] Hall LA, Shmoys DB (1992) Jackson's rule for single machine scheduling: making a good heuristic better. *Mathematics of Operations Research* 17: 22-35.

[3] Kacem I, Haouari M (2009) Approximation algorithms for single machine scheduling with one unavailability period. *4OR: A Quarterly Journal of Operations Research* 7(1): 79-92.

[4] Kacem I, Kellerer H (2018) Approximation Schemes for Minimizing the Maximum Lateness on a Single Machine with Release Times under Non-Availability or Deadline Constraints. *Algorithmica* (SPRINGER), 2018, https://doi.org/10.1007/s00453-018-0417-6.

[5] Kacem I, Kellerer H (2016) Semi-online scheduling on a single machine with unexpected breakdown, *Theoretical Computer Science* 646: 40-48.

[6] Kacem I, Kellerer H (2014) Approximation algorithms for no-idle time scheduling on a single machine with release times and delivery times. *Discrete Applied Mathematics* 164(1): 154-160.

[7] Kacem I, Kellerer H, Seifaddini M (2016) Efficient approximation schemes for the maximum lateness minimization on a single machine with a fixed operator or machine non-availability interval. *Journal of Combinatorial Optimization* 32: 970-981.