Adversarial bilevel scheduling on a single machine

Della Croce F.[1] and T'kindt V.[2]

[1] DIGEP, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy, CNR, IEIIT, Torino, Italy.
federico.dellacroce_a@polito.it

[2] Université de Tours, Laboratoire d'Informatique Fondamentale et Appliquée (EA 6300), ERL CNRS 7002 ROOT, Tours, France,
and
DIGEP, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy,
tkindt@univ-tours.fr

Mots-clés : *Scheduling, bilevel optimization.*

1 Introduction

In this contribution we focus on a particular setting in which two agents are concerned by the scheduling of a set of n jobs. The first agent, called the *leader*, can take some decisions before providing the jobset to the second agent, called the *follower*, who then takes the remaining decisions to solve the problem. As an example, the leader could select a subset of $n' \leq n$ jobs that the follower has to schedule. Notice that the decisions the agents can take are exclusive : in this example, the follower cannot decide the jobs to schedule and the leader cannot schedule the jobs. This setting falls into the category of *bilevel optimization* [4]. In such problems it is assumed that the leader and the follower follow their own objectives which can be contradictory, so leading to very hard optimization problems. Recently, many papers on bilevel combinatorial optimization appeared, here we refer to [2, 3, 6, 7, 5] just to mention a few. On the other hand, to the authors knowledge, the literature on bilevel scheduling is much more limited. We refer here to [1, 8, 9]. We focus in the following on single machine scheduling under the adversarial framework where the goal of the leader is to make the follower solution as bad as possible and provide several exact polynomial time algorithms for different objective functions.

2 Adversarial bilevel single machine scheduling

It is assumed that n jobs are to be scheduled on a single disjunctive machine. Eeach job j is defined by a processing time p_j and, depending on the problem, a weight w_j or a due date d_j. The follower is scheduling jobs so that its objective function $f^F \in \{\sum_j C_j^F, \sum_j w_j^F C_j^F, L_{max}^F\}$ is minimized. However, before the follower sequences the n jobs, the leader can decide how to fix quantities q_j so that the processing times or the weigths or the due dates are modified. The leader has a given budget $Q \in \mathbb{N}$ so that $\sum_j |q_j| \leq Q$. Considering the three-field notation for scheduling problems, we will denote by $ADV - p$ the problems in which the leader modifies only the processing times. Similarly, $ADV - w$ (resp. $ADV - d$) refers to the problems in which only the weights (resp. the due dates) are modified.

We show that the following list of problems can be solved in polynomial time : $1|ADV - p|\sum_j C_j^F$, $1|ADV - p|\sum_j w_j^F C_j^F$, $1|ADV - w|\sum_j w_j^F C_j^F$, $1|ADV - p|L_{max}^F$ and $1|ADV - d|L_{max}^F$.

Let us develop the solution of the $1|ADV - p|\sum_j C_j^F$ problem. Let be the initial processing times p_j so that $p_1 \leq ... \leq p_n$. Then, the leader has to decide how to fix quantities q_j so that, with $p_j^F = p_j + q_j$, the follower optimal solution is the worst possible. Obviously, it is of no interest for the leader that some $q_j < 0$.

Theorem 1 *The* $1|ADV - p|\sum_j C_j^F$ *problem can be solved in* $O(n \log(n))$ *time. The leader sets :*

- $q_j = P - p_j$, $\forall j = 1..(k_P - Q - k_P P + \sum_{i=1}^{k_P} p_i)$,
- $q_j = P - p_j + 1$, $\forall j = (k_P - Q - k_P P + \sum_{i=1}^{k_P} p_i)..k_P$,
- $q_j = 0$, $\forall j = k_P + 1..n$,

with $P = argmax_{0 \leq t \leq \sum_j p_j}((kt - \sum_{j=1}^{k} p_j) \leq Q | p_1 \leq ... \leq p_k \leq t \text{ and } p_{k+1} > t)$, *and* k_P *the job such that* $p_{k_P} \leq P < p_{k_P+1}$. *The follower applies the Shorter Processing Times first (SPT) rule on the* $p_j^F = p_j + q_j$*'s.*

Roughly, this result states that the leader uses all the budget to increase the smallest jobs that appear first in the SPT order as this rule is used by the follower to optimally sequence the jobs.

Acknowledgements

This work has been partially supported by "Ministero dell'Istruzione, dell'Università e della Ricerca" Award "TESUN-83486178370409 finanziamento dipartimenti di eccellenza CAP. 1694 TIT. 232 ART. 6".

References

[1] Abass S.A. : Bilevel programming approach applied to the flow shop scheduling problem under fuzziness. Computational Management Science. 2 : 279–293 (2005)

[2] Caprara, A., Carvalho, M., Lodi, A., Woeginger, G. : Bilevel Knapsack with Interdiction Constraints. INFORMS Journal on Computing. 28, 319–333 (2016)

[3] Della Croce, F., Scatamacchia, R. : Lower Bounds and a New Exact Approach for the Bilevel Knapsack with Interdiction Constraints. In : Lodi A., Nagarajan V. (eds) Integer Programming and Combinatorial Optimization. IPCO 2019. Lecture Notes in Computer Science, vol 11480, 155–167. Springer International Publishing (2019)

[4] Dempe, S., Kalashnikov, V. Perez-Valdes, G.A., Kalashnikova, N., 2015, "Bilevel programming problems", *Springer*.

[5] Fischetti, M., Ljubić, I., Monaci, M., Sinnl, M. : Interdiction Games and Monotonicity, with Application to Knapsack Problems. INFORMS Journal on Computing. 31, 390–410 (2019)

[6] Fischetti, M., Ljubić, I., Monaci, M., Sinnl, M. : A New General-Purpose Algorithm for Mixed-Integer Bilevel Linear Programs. Operations Research. 65, 1615–1637 (2017)

[7] Fischetti, M., Ljubić, I., Monaci, M., Sinnl, M. : On the use of intersection cuts for bilevel optimization. Mathematical Programming. 172, 77–103 (2018)

[8] J.K. Karlof, J.K., Wang, W. : Bilevel programming applied to the flow shop scheduling problem. Computers and Operations Research. 23 :5, 443–451 (1996).

[9] Kis, T., Kovacs, A. : On bilevel machine scheduling problems. OR Spectrum. 34, 43–68 (2012)