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

Keywords: Single machine, 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* (Dempe et al. 2015). 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 (Caprara et al. 2016, Della Croce et al. 2019, Fischetti et al. 2017, Fischetti et al. 2018, Fischetti et al. 2019) 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 (Abass 2005, Karlof and Wangs 1996, Kis and Kovacs 2012). 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 when the leader can only modify data of the problem.

2 Adversarial bilevel single machine scheduling

2.1 Sum of completion times

It is assumed that, given a list of n jobs with processing times p_j^F, the follower is scheduling jobs so that their sum of completion times, denoted by $\sum_j C_j^F$, is minimum. This is doable in polynomial time by applying the so-called SPT rule (*Shortest Processing Times* first). 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 $q_j < 0$. In addition, the leader has a budget so that $\sum_j |q_j| \leq Q$, with $Q \in \mathbb{N}$ given. This problem is referred to as $1|ADV-p|\sum_j C_j^F$, with $ADV-p$ meaning that it concerns an adversarial bilevel problem in which the leader can only modify the processing time values.

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

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

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

2.2 Weighted sum of completion times

Now, let us assume that, in addition to the previous problem, jobs are also attached weights w_j^F and the follower is scheduling jobs so that their weighted sum of completion times, denoted by $\sum_j w_j^F C_j^F$, is minimum. Whenever the processing times are fixed, this is doable in polynomial time by applying the so-called WSPT rule (*Weighted Shortest Processing Times* first). Let be the initial processing times p_j so that $\frac{p_1}{w_1^F} \leq ... \leq \frac{p_n}{w_n^F}$. Again, 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 as worse as possible. Obviously, it is of no interest for the leader that some $q_j < 0$. This problem is referred to as $1|ADV - p|\sum_j w_j^F C_j^F$.

We first consider the relaxed version where $q_j \in \mathbb{R}, \forall j = 1..n$, denoted by $1|ADV - p, q_j \in \mathbb{R}|\sum_j w_j^F C_j^F$.

Theorem 2. *The* $1|ADV - p, q_j \in \mathbb{R}|\sum_j w_j^F C_j^F$ *problem can be solved in* $O(n\log(n))$ *time. The leader sets:*

- $q_j = \frac{(Q + \sum_{\ell=1}^{k_R} p_\ell) w_j^F}{\sum_{\ell=1}^{k_R} w_\ell^F} - p_j$, $\forall j = 1..k_R$
- $q_j = 0$, $\forall j = (k_R + 1)..n$,

with $R = \frac{Q - \sum_{j=1}^{k_R} p_j}{\sum_{j=1}^{k_R} w_j^F}$ *and* k_R *the job such that* $\frac{p_{k_R}}{w_{k_R}^F} \leq R < \frac{p_{k_R+1}}{w_{k_R+1}^F}$. *The follower applies the WSPT rule on* $p_j^F = p_j + q_j$ *and* $w_j^F = w_j$, $\forall j = 1..n$.

The optimal solution of the $1|ADV - p|\sum_j w_j^F C_j^F$ problem can be obtained by solving iteratively the relaxed version: first solve it with the initial Q value and round down the computed q_j's. Then, on the remaining quantity $Q' = (Q - \sum_j q_j)$ solve again the relaxed problem to modify processing times. This process is iterated until all initial budget Q is assigned to jobs. As there are at most n iterations, this leads to an exact algorithm than can be implemented in $O(n^2)$ time.

Let us turn to the other possible adversarial problem in which the leader can only modify the weights of the follower. So, for the follower's problem we set $p_j^F = p_j$ and $w_j^F = w_j + q_j$, $\forall j = 1..n$, with $q_j \in \mathbb{N}$. This problem is referred to as $1|ADV - w|\sum_j w_j^F C_j^F$ and as previously, $1|ADV - w, q_j \in \mathbb{R}|\sum_j w_j^F C_j^F$ refers to the relaxed version with real valued q_j's.

Theorem 3. *The* $1|ADV - w, q_j \in \mathbb{R}|\sum_j w_j^F C_j^F$ *problem can be solved in* $O(n\log(n))$ *time. The leader sets:*

- $q_j = \frac{(Q + \sum_{\ell=k_R}^{n} w_\ell) p_j^F}{\sum_{\ell=k_R}^{n} p_\ell^F} - w_j$, $\forall j = k_R..n$
- $q_j = 0$, $\forall j = 1..(k_R - 1)$,

with $R = \frac{\sum_{j=k_R}^{n} p_j^F}{Q+\sum_{j=k_R}^{n} w_j}$ *and* k_R *the job such that* $\frac{p_{k_R-1}^F}{w_{k_R-1}} < R \leq \frac{p_{k_R}^F}{w_{k_R}}$. *The follower applies the WSPT rule on* $p_j^F = p_j$ *and* $w_j^F = w_j + q_j$, $\forall j = 1.n$.

The $1|ADV-w|\sum_j w_j^F C_j^F$ problem can be solved by iteratively solving the relaxation with real valued q_j's to dispatch the initial leader's budget Q. Again, this leads to an $O(n^2)$ optimal algorithm.

2.3 Maximum lateness

Assume that each job j is defined by a processing time p_j and a due date d_j. The aim, for the follower, is to schedule jobs so as to minimize the maximum lateness, defined by $L_{max}^F = \max_{j=1..n}(C_j^F - d_j^F)$. The leader can modify either the processing times or the due dates. Without loss of generality, let us assume that $d_1 \leq ... \leq d_n$.

We first focus on the problem where the leader can only modify the processing times, which is referred to as $1|ADV-p|L_{max}^F$. As the due dates remain unchanged, we set $d_j^F = d_j$, $\forall j = 1..n$. Besides, $p_j^F = p_j + q_j$ is the processing time value of the follower's problem. It is trivial to show that $q_j \in \mathbb{N}$ in order to make increasing the optimal solution value of the follower's problem. Besides, it is known that the $1||L_{max}$ problem is solved to optimality by the EDD rule (Earliest Due Dates first). So the follower builds the optimal sequence by sorting jobs by non decreasing values of the d_j^F's which is not impacted by any variations in the processing time values. Consequently, the $1|ADV-p|L_{max}^F$ problem can be solved in $O(n\log(n))$ time by sorting jobs according to EDD rule and then set:

- $q_k = Q$ with k the earliest job having $(C_k - d_k) = L_{max}^*$ and L_{max}^* the L_{max} value of the EDD schedule,
- $q_j = 0$, $\forall j = 1..n$, $j \neq k$.

Let us consider the problem in which the leader can only modify the due dates, which is referred to as $1|ADV-d|L_{max}^F$. Then, we set $p_j^F = p_j$ and $d_j^F = d_j + q_j$, $\forall j = 1..n$.

Theorem 4. *The* $1|ADV-d|L_{max}^F$ *problem can be solved in* $O(n\log(n))$ *time. The leader sets:*

- $q_\ell = D - d_\ell \leq 0$, $\forall \ell \in \cup_{j\in\mathcal{T}} B_j \cup \mathcal{T}$,
- *and* $q_\ell = 0$, *otherwise.*

with:

- $\mathcal{T} = \{j / C_j^F - d_j^F = L_{max}^*\}$, *with* L_{max}^* *the value of the initial EDD sequence,*
- $B_j = \{k < j | \nexists \ell \in \mathcal{T}$, *with* $k < \ell < j\}$, $\forall j \in \mathcal{T}$,
- $\alpha_\ell = (d_\ell - d_j) \leq 0$ *and* $[\ell]$ *is the* ℓ*-th* α_u *value when sorted by non decreasing values, i.e.* $\alpha_{[1]} \leq ... \leq \alpha_{[n']}$ *with* $n' = |\cup_{j\in\mathcal{T}} B_j|$,
- k *such that* $(|\mathcal{T}| + k)\alpha_{[k]} - \sum_{\ell=1}^{k} \alpha_{[\ell]} \leq Q \leq (|\mathcal{T}| + k + 1)\alpha_{[k+1]} - \sum_{\ell=1}^{k+1} \alpha_{[\ell]}$,
- *and* $D = \lfloor \frac{Q+\sum_{\ell=1}^{k} \alpha_{[\ell]}}{|\mathcal{T}|+k} \rfloor$.

The follower applies the EDD rule on $p_j^F = p_j$ *and* $d_j^F = d_j + q_j$, $\forall j = 1.n$.

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

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

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

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)

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

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)

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)

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

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).

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