Linear inequalities for neighborhood based dominance properties for the common due-date scheduling problem

Anne-Elisabeth Falq[1], Pierre Fouilhoux[1] and Safia Kedad-Sidhoum[2]

[1] Sorbonne Université, CNRS, LIP6, 4 place Jussieu, 75005 Paris, France
anne-elisabeth.falq@lip6.fr, pierre.fouilhoux@lip6.fr

[2] CNAM, CEDRIC, 292 rue Saint Martin, 75003 Paris, France safia.kedad_sidhoum@cnam.fr

Keywords : just-in-time scheduling, dominance properties, integer linear programming

We consider a set of n tasks J that have to be processed non-preemptively on a single machine around a common due-date d. Given for each task $j \in J$ a processing time p_j and a unitary earliness (resp. tardiness) penalty α_j (resp. β_j), the problem denoted $1 \,|\,|\, \sum \alpha_j\, [d–C_j]^+ + \beta_j\, [C_j–d]^+$, aims at finding a feasible schedule that minimizes the sum of earliness-tardiness penalties.

We assume that the due date is **unrestrictive**, *i.e.* $d \geq \sum p_j$. The problem is NP-hard [3]., even if the penalties are symmetric, *i.e.* if $\alpha_j = \beta_j$ for all $j \in J$. For the latter case, a dynamic programming algorithm is proposed in [3]. For arbitrary penalties, a heuristic method together with a benchmark is provided in [1]. These instances are efficiently solved by an exact method proposed in [4]. Our work also deals with arbitrary penalties, but for sake of brevity, we assume that the α-ratios α_j/p_j for $j \in J$ are different, as well as the β-ratios β_j/p_j.

1 A compact linear formulation

In a given schedule, a task is **early** (resp. **tardy**), if it completes before or at d (resp. after d), and a task is **on-time** if it completes exactly at time d. A schedule having an on-time task is said **V-shaped**, if early (resp. tardy) tasks are ordered by increasing α-ratios (resp. decreasing β-ratios). A schedule without idle time is called a **block**.

From [3], it is known that there exists an optimal solution which is a V-shaped block having an on-time task. Using this dominance property, we only consider this kind of schedules, which can be completely described by the partition between early and tardy tasks. Indeed, if the set of early tasks E is given, the set of tardy tasks $T = J \setminus E$ is also fixed, and the earliness e_u (resp. the tardiness t_u) of any task $u \in J$, can be deduced as follows :

$$e_u = \begin{cases} p(A(u) \cap E) & \text{if } u \in E \\ 0 & \text{otherwise} \end{cases} \qquad t_u = \begin{cases} p(B(u) \cap T) & \text{if } u \in T \\ 0 & \text{otherwise} \end{cases}$$

where $p(S) = \sum_{j \in S} p_j$ for any $S \subseteq J$, $A(u) = \left\{ j \in J \,|\, \frac{\alpha_j}{p_j} > \frac{\alpha_u}{p_u} \right\}$ and $B(u) = \left\{ j \in J \,|\, \frac{\beta_j}{p_j} > \frac{\beta_u}{p_u} \right\}$.

Note that, for each task u, the sets $A(u)$ and $B(u)$ can be pre-computed since they are defined from the instance parameters. We also define $\bar{A}(u) = J \setminus (A(u) \cup \{u\})$ and $\bar{B}(u) = J \setminus (B(u) \cup \{u\})$.

We proposed in [2] a compact integer linear model based on n boolean variables indicating if task j is early. *i.e.* on a vector $\delta \in \{0,1\}^J$ encoding the partition $(E = \{j \in J \,|\, \delta_j = 1\}$, $T = \{j \in J \,|\, \delta_j = 0\})$. This formulation, called (F), has a total of $n + n(n-1)/2$ boolean variables and $4n(n-1)/2$ inequalities. We propose in the next section a way to improve it.

2 Inequalities for neighborhood based dominance properties

It is common, in local search procedures, to slightly change a solution $\mathcal{S}$ to obtain a new one $\mathcal{S}'$, called a **neighbor** of $\mathcal{S}$. If the neighbor $\mathcal{S}'$ is better, (*i.e.* if it has a smaller total penalty in our case), we say that $\mathcal{S}$ is dominated (by $\mathcal{S}'$), it follows that $\mathcal{S}$ cannot be optimal.

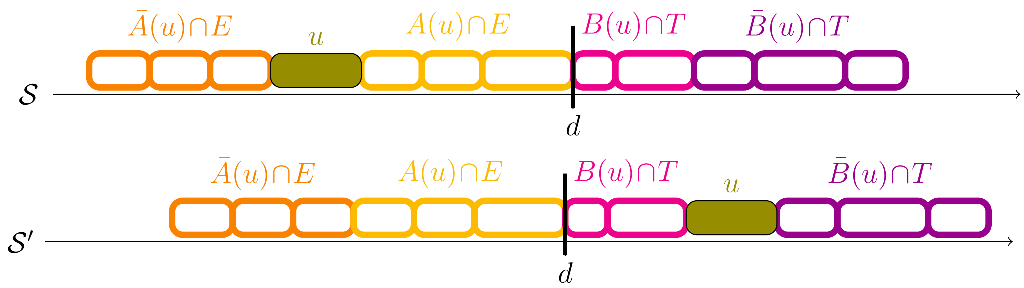

FIG. 1 – Insertion of an early task u on the tardy side of a schedule

This simple observation leads to a dominance property for any **neighborhood** $\mathcal{N}$ which associates to a solution a set of neighbors. A solution $\mathcal{S}$ is said **$\mathcal{N}$-dominated** if there exists $\mathcal{S}'\in\mathcal{N}(\mathcal{S})$ which is strictly better than $\mathcal{S}$. Hence the solutions which are not $\mathcal{N}$-dominated are dominant. In our work, as a schedule is encoded by a partition (E, T) between early and tardy tasks, we consider two kinds of operations providing a neighbor (E', T') :

- the **insertion** operation, which consists in inserting an early task on the tardy side of the schedule, *i.e.* $E'=E\backslash\{u\}$ and $T'=T\cup\{u\}$ for some $u\in E$ (see Figure 1), or conversely in inserting a tardy task on the early side *i.e.* $E'=E\cup\{u\}$ and $T'=T\backslash\{u\}$ for some $u\in T$,
- the **swap** operation, which consists in inserting an early task on the tardy side of the schedule, while a tardy task is inserted on the early side *i.e.* $E' = E\setminus\{u\}\cup\{v\}$ and $T'=T\setminus\{v\}\cup\{u\}$ for some $(u, v)\in E\times T$.

For each of these operations, we propose a **linear inequality translating the associated dominance**, which means that this inequality cuts exactly all schedules which are dominated by their neighbor obtained by the above mentioned operation. For example, for the insertion of a task $u\in J$ on the tardy side, the inequality proposed cuts all schedules in which u is early and which are dominated by the schedule obtained by inserting u on the tardy side, but is valid for any other schedule, in particular for all optimal schedules since they are non-dominated.

Note that these inequalities, called **insert** and **swap inequalities**, are not standard reinforcement inequalities. Classically, valid inequalities are added to cut extreme points which are not integer and then do not encode a feasible solution. On the contrary, these dominance inequalities cut integer points encoding feasible solutions because they correspond to dominated, and then non-optimal, schedules. By adding these inequality to (F), we obtain a new formulation (F') for our problem.

To assess the computational advantage of adding insert and swap inequalities to the compact formulation, we implement (F) and (F') using a linear solver (CPLEX version 12.6.3.0), and test them on the benchmark proposed by [1]. Under a time limit of one hour, formulation (F) using all CPLEX features allows to exactly solve instances up to 50 tasks, while formulation (F') without any CPLEX feature allows to exactly solve instances up to 150 tasks.

References

[1] D. Biskup and M. Feldmann. Benchmarks for scheduling on a single machine against restrictiveand unrestrictive common due dates. *Computers and operations research*, Vol 28 :787–801, 2001.

[2] A-E. Falq, P. Fouilhoux, and S. Kedad-Sidhoum. Mixed integer formulations using natural variables for single machine scheduling around a common due date. *CoRR*, abs/1901.06880, 2019.

[3] N.G. Hall and M.E. Posner. Earliness-tardiness sceduling problems, 1 : Weighted deviation of completion times about a common due date. *Operations Research*, Vol 39 :836–846, Sep-Oct 1991.

[4] F. Sourd. New exact algorithms for one-machine earliness-tardiness scheduling. *INFORMS Journal on Computing*, 21(1) :167–175, 2009.