A compact MIP formulation for single machine scheduling to minimize a piecewise linear objective function

Philippe Baptiste **Ruslan Sadykov**

LIX, Ecole Polytechnique, France

Optimeo'08

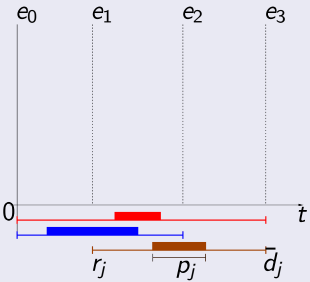
The problem PWL (I)

Data

1 machine, m intervals
$I_u = [e_{u-1}, e_u]$, $u \in M$;
for each job $j \in N = \{1, \dots, n\}$

- r_j - release date,
- p_j - processing time,
- $\bar{d}_j$ - deadline,

such that $\{r_j, \bar{d}_j\}_{j \in M} \subseteq \{e_u\}_{u \in M}$.



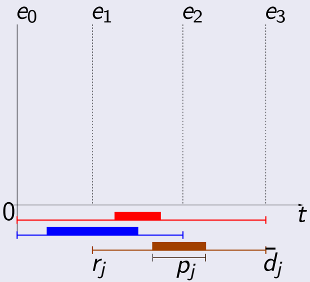
The problem PWL (I)

Data

1 machine, m intervals
$I_u = [e_{u-1}, e_u]$, $u \in M$;
for each job $j \in N = \{1, \ldots, n\}$

- r_j - release date,
- p_j - processing time,
- $\bar{d}_j$ - deadline,

such that $\{r_j, \bar{d}_j\}_{j \in M} \subseteq \{e_u\}_{u \in M}$.



Constraints

- the machine can process only one job at a time,
- preemtion is not allowed,

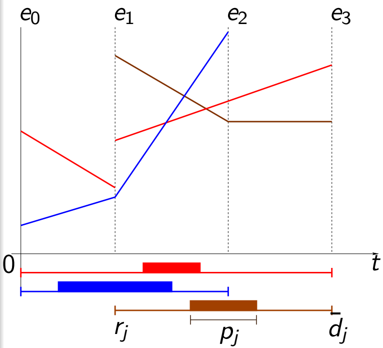
The problem PWL (II)

Objective

$\forall j \in N$, the cost function $F_j(C_j)$ is linear in each interval :

$$\text{if } e_{u-1} < C_j \leq e_u \text{ then}$$

$$F_j(C_j) = f_j^u + w_j^u \cdot (C_j - e_{u-1}).$$



The problem PWL (II)

Objective

$\forall j \in N$, the cost function $F_j(C_j)$ is linear in each interval :

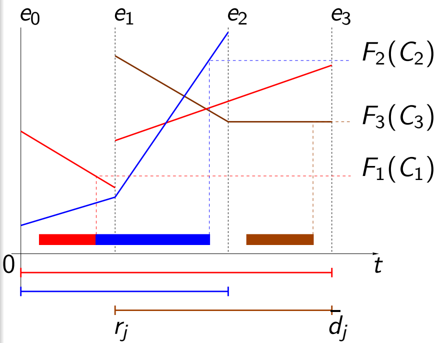
$$\text{if } e_{u-1} < C_j \leq e_u \text{ then}$$

$$F_j(C_j) = f_j^u + w_j^u \cdot (C_j - e_{u-1}).$$

We minimize the total cost :

$$\sum_{j \in N} F_j(C_j)$$

.



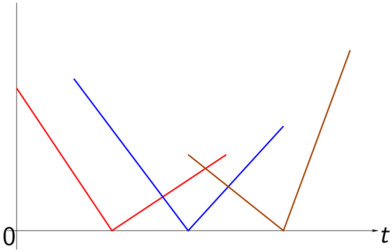
Generality of the problem PWL

All classical non-preemptive scheduling problems can be formulated as the problem PWL, for example :

Generality of the problem PWL

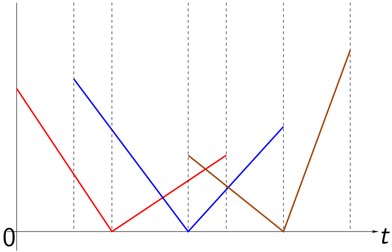
All classical non-preemptive scheduling problems can be formulated as the problem PWL, for example :

Earliness-tardiness scheduling



All classical non-preemptive scheduling problems can be formulated as the problem PWL, for example :

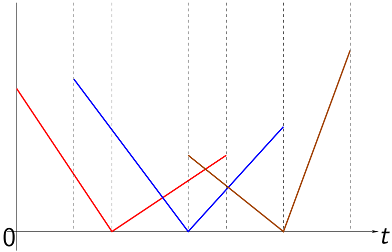
Earliness-tardiness scheduling



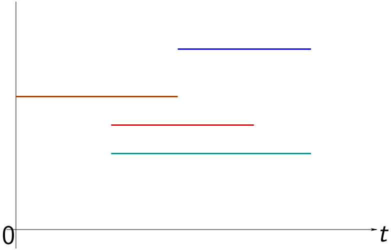
Generality of the problem PWL

All classical non-preemptive scheduling problems can be formulated as the problem PWL, for example :

Earliness-tardiness scheduling



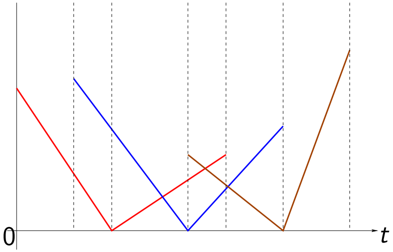
Maximizing weighted number of on-time jobs



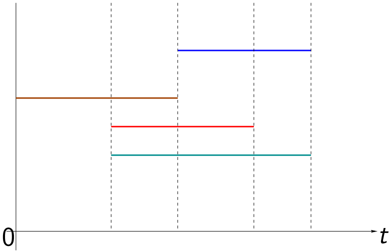
Generality of the problem PWL

All classical non-preemptive scheduling problems can be formulated as the problem PWL, for example :

Earliness-tardiness scheduling



Maximizing weighted number of on-time jobs



Time-indexed formulation

Data is integer $\rightarrow$ Time-indexed formulation [Sousa, Wolsey, 92]

$X_{jt} \in \{0, 1\}$, $X_{jt} = 1$ iff job j is started at time moment t

$$\min \quad \sum_{t=0}^{T} F_j(t + p_j) X_{jt}$$

$$s.t. \quad \sum_{t=r_j}^{\overline{d}_j - p_j} X_{jt} = 1, \quad j \in N,$$

$$\sum_{j \in N} \sum_{s=t-p_j+1}^{t} X_{js} \leq 1, \quad t \in [0, T),$$

Time-indexed formulation

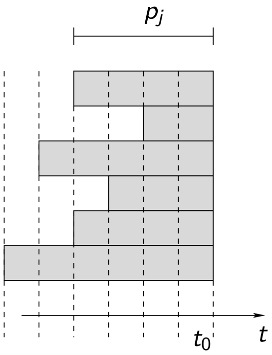
Data is integer → Time-indexed formulation [Sousa, Wolsey, 92]

$X_{jt} \in \{0,1\}$, $X_{jt} = 1$ iff job j is started at time moment t

$$\min \quad \sum_{t=0}^{T} F_j(t + p_j) X_{jt}$$

$$s.t. \quad \sum_{t=r_j}^{\bar{d}_j - p_j} X_{jt} = 1, \quad j \in N,$$

$$\sum_{j \in N} \sum_{s=t-p_j+1}^{t} X_{js} \leq 1, \quad t \in [0, T),$$



Interval decomposition

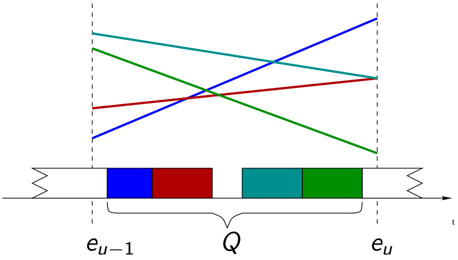
The cost functions are linear in an interval.

The cost functions are linear in an interval.

Property. If all jobs in a set Q are started and completed in the same interval $[e_{u-1}, e_u]$, it is optimal to process them according to the Smith rule :

$$i \to j \text{ if } \frac{w_i^u}{p_i} > \frac{w_j^u}{p_j}.$$

Idle time between jobs with positive and negative weights w_j^u



Interval-indexed formulation

Variables

$X_j^u, Y_j^u \in \{0,1\}$ — whether job j is started (completed) before e_u
W_u — length of the idle time in I_u
F_j^u — difference between the completion time and the border of I_u

Interval-indexed formulation

Variables

$X_j^u, Y_j^u \in \{0,1\}$ — whether job j is started (completed) before e_u
W_u — length of the idle time in I_u
F_j^u — difference between the completion time and the border of I_u

Variables constraints

$$\begin{aligned}
\forall u \in M,\ \forall j \in N, \quad & Y_j^{u-1} \le Y_j^u,\ X_j^{u-1} \le X_j^u,\ Y_j^u \le X_j^u \\
\forall u \in M,\ \forall j \in NS_u, \quad & X_j^{u-1} \le Y_j^u \\
\forall u \in M,\ \forall j \in NB_u, \quad & Y_j^u \le X_j^{u-1} \\
\forall j \in N,\ r_j = e_u, \quad & X_j^u = 0 \\
\forall j \in N,\ \bar{d}_j = e_u, \quad & Y_j^u = 1
\end{aligned}$$

Interval-indexed formulation

Variables

$X_j^u, Y_j^u \in \{0,1\}$ — whether job j is started (completed) before e_u
W_u — length of the idle time in I_u
F_j^u — difference between the completion time and the border of I_u

Knapsack constraints

$$\forall u \in M, \qquad \sum_{j \in N} p_j Y_j^u + \sum_{v=1}^{u} W_v \leq e_u$$

$$\forall u \in M, \qquad \sum_{j \in N} p_j X_j^u + \sum_{v=1}^{u} W_v \geq e_u$$

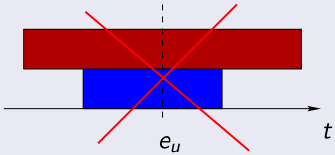
Interval-indexed formulation

Variables

$X_j^u, Y_j^u \in \{0,1\}$ — whether job j is started (completed) before e_u
W_u — length of the idle time in I_u
F_j^u — difference between the completion time and the border of I_u

Additional constraints I

$$\forall u \in M, \qquad \sum_{i \in N} (X_i^u - Y_i^u) \leq 1$$



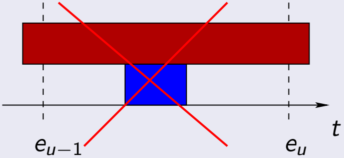
Interval-indexed formulation

Variables

$X_j^u, Y_j^u \in \{0,1\}$ — whether job j is started (completed) before e_u
W_u — length of the idle time in I_u
F_j^u — difference between the completion time and the border of I_u
NS_u, NB_u — set of "small" jobs and "big" jobs for I_u

Additional constraints II

$$\forall u \in M,\ \forall j \in NS_u, \qquad \sum_{i \in NB_u} (X_i^{u-1} - Y_i^u) + Y_j^u - X_j^{u-1} \leq 1$$



Interval-indexed formulation

Variables

$X_j^u, Y_j^u \in \{0,1\}$ — whether job j is started (completed) before e_u
W_u — length of the idle time in I_u
F_j^u — difference between the completion time and the border of I_u

Theorem

Vector (X, Y, W) *satisfies the constraints presented*
$\Leftrightarrow$ *the corresponding schedule is feasible*

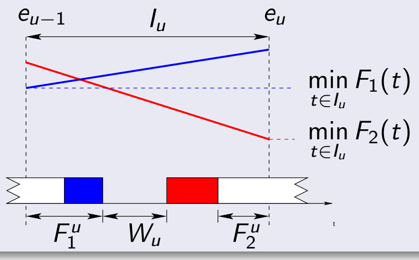
Interval-indexed formulation

Variables

$X_j^u, Y_j^u \in \{0,1\}$ — whether job j is started (completed) before e_u
W_u — length of the idle time in I_u
F_j^u — difference between the completion time and the border of I_u

Objective

$$\min \sum_{j\in N}\sum_{u\in M} \mid w_j^u \mid F_j^u + \sum_{j\in N}\sum_{u\in M} \min_{t\in I_u}\{F_j(t)\}\cdot(Y_j^u - Y_j^{u-1})$$



Interval-indexed formulation

Variables

$X_j^u, Y_j^u \in \{0,1\}$ — whether job j is started (completed) before e_u
W_u — length of the idle time in I_u
F_j^u — difference between the completion time and the border of I_u
NS_u, NB_u — set of "small" jobs and "big" jobs for I_u

Objective

$$\min \sum_{j \in N} \sum_{u \in M} \mid w_j^u \mid F_j^u + \sum_{j \in N} \sum_{u \in M} \min_{t \in I_u} \{F_j(t)\} \cdot (Y_j^u - Y_j^{u-1})$$

$$\forall u \in M,\ \forall j \in N, \quad F_j^u \geq \alpha X + \beta Y + \gamma W$$

Tightening interval-indexed formulation

- Redundant constraintes.
  Example : $p_j \leq e_u - e_v$: $C_j \leq e_v \Rightarrow C_j \leq e_u$ $(Y_j^u \geq X_j^u)$
- Appropriate partition of the time horizon
  Known order of jobs completed but not necessarily started in the same interval.

Tightening interval-indexed formulation

- Redundant constraintes.
  Example : $p_j \leq e_u - e_v$: $C_j \leq e_v \Rightarrow C_j \leq e_u$ $(Y_j^u \geq X_j^u)$
- Appropriate partition of the time horizon
  Known order of jobs completed but not necessarily started in the same interval.

Numerical experiments

Cplex 10.0, standard settings and only standard cuts,
1000 seconds time limit

- **Minimizing total weighted earliness and tardiness with release dates** $(1 \mid r_j \mid \alpha_j E_j + \beta_j T_j)$ — some 15 jobs instances could not be solved (all are solved if number of intervals is small).
- **Minimizing total weighted earliness and tardiness** $(1 \mid\mid \alpha_j E_j + \beta_j T_j)$ — all 15 jobs instances are solved (majority of 20 jobs instances are not solved).
- **Minimizing total weighted tardiness** $(1 \mid\mid w_j T_j)$ — all 20 jobs instances are solved (some 30 jobs instances are not solved). Here we need only Y and F variables.

Numerical experiments

Cplex 10.0, standard settings and only standard cuts,
1000 seconds time limit

- **Minimizing total weighted earliness and tardiness with release dates** $(1 \mid r_j \mid \alpha_j E_j + \beta_j T_j)$ — some 15 jobs instances could not be solved (all are solved if number of intervals is small).
- **Minimizing total weighted earliness and tardiness** $(1 \mid\mid \alpha_j E_j + \beta_j T_j)$ — all 15 jobs instances are solved (majority of 20 jobs instances are not solved).
- **Minimizing total weighted tardiness** $(1 \mid\mid w_j T_j)$ — all 20 jobs instances are solved (some 30 jobs instances are not solved). Here we need only Y and F variables.

Numerical experiments

Cplex 10.0, standard settings and only standard cuts,
1000 seconds time limit

- **Minimizing total weighted earliness and tardiness with release dates** $(1 \mid r_j \mid \alpha_j E_j + \beta_j T_j)$ — some 15 jobs instances could not be solved (all are solved if number of intervals is small).
- **Minimizing total weighted earliness and tardiness** $(1 \mid\mid \alpha_j E_j + \beta_j T_j)$ — all 15 jobs instances are solved (majority of 20 jobs instances are not solved).
- **Minimizing total weighted tardiness** $(1 \mid\mid w_j T_j)$ — all 20 jobs instances are solved (some 30 jobs instances are not solved). Here we need only Y and F variables.

Numerical experiments

Cplex 10.0, standard settings and only standard cuts,
1000 seconds time limit

- **Minimizing total weighted earliness and tardiness with release dates** $(1 \mid r_j \mid \alpha_j E_j + \beta_j T_j)$ — some 15 jobs instances could not be solved (all are solved if number of intervals is small).
- **Minimizing total weighted earliness and tardiness** $(1 \mid\mid \alpha_j E_j + \beta_j T_j)$ — all 15 jobs instances are solved (majority of 20 jobs instances are not solved).
- **Minimizing total weighted tardiness** $(1 \mid\mid w_j T_j)$ — all 20 jobs instances are solved (some 30 jobs instances are not solved). Here we need only Y and F variables.

Numerical experiments

Cplex 10.0, standard settings and only standard cuts,
1000 seconds time limit

- **Minimizing total weighted earliness and tardiness with release dates** $(1 \mid r_j \mid \alpha_j E_j + \beta_j T_j)$ — some 15 jobs instances could not be solved (all are solved if number of intervals is small).
- **Minimizing total weighted earliness and tardiness** $(1 \mid\mid \alpha_j E_j + \beta_j T_j)$ — all 15 jobs instances are solved (majority of 20 jobs instances are not solved).
- **Minimizing total weighted tardiness** $(1 \mid\mid w_j T_j)$ — all 20 jobs instances are solved (some 30 jobs instances are not solved). Here we need only Y and F variables.

Similar performance with the Time-indexed formulation

Conclusions

- A new compact MIP formulation for a general machine scheduling problem (size is $O(nm) \times O(nm)$)
- More efficient in practical situations (few different release/due dates)
- 4 open practical instances from ILOG [Le Pape, Robert, 07] has been solved (up to 25 jobs)
- (—) Applicable to small instances only

Conclusions

- A new compact MIP formulation for a general machine scheduling problem (size is $O(nm) \times O(nm)$)
- More efficient in practical situations (few different release/due dates)
- 4 open practical instances from ILOG [Le Pape, Robert, 07] has been solved (up to 25 jobs)
- (—) Applicable to small instances only