Exact approaches for the single machine scheduling problem with distinct time windows

Bruno Rosa[1]*, Philippe Michelon[2], Zacharie Ales[2], Marcone Souza[3], Sérgio de Souza[1]

[1] Federal Center of Technological Education of Minas Gerais (CEFET-MG), Belo Horizonte, Brazil
brunorosa@div.cefetmg.br, sergio@dppg.cefetmg.br

[2] Laboratoire d'Informatique d'Avignon, Université d'Avignon et des Pays du Vaucluse, 84911 Avignon, France
philippe.michelon@univ-avignon.fr, zacharie.ales@univ-avignon.fr

[3] Department of Computer Science, Federal University of Ouro Preto (UFOP), Ouro Preto, Brazil
marcone@iceb.ufop.br

Keywords: *job scheduling, mathematical programming, constraint programming.*

1 Introduction

The Single Machine Scheduling Problem discussed in the present paper consists in determining the time in which a set of jobs must be performed to minimize the weighted sum of their earliness and tardiness penalties. To each job is associated a time window rather than a due date. Thus, a job generates an earliness penalty if it is completed before its time window and a tardiness penalty if the job finishes after.

The difficulty of the The Single Machine Scheduling Problem with Distinct Time Windows (SMSPTW) and its numerous applications in Just-in-Time manufacturing, chemical processing, Video on Demand services, among others, have motivated the development of many efficient resolution algorithms [1]. However, to our knowledge, no exact method has currently been proposed to tackle it.

The SMSPTW has the following characteristics: (*i*) A single machine should process a set I of n jobs; (*ii*) To each job $x \in I$ is associated a processing time P_x and a time window $[E_x, T_x]$, in which the job x should preferably be completed. E_x and T_x respectively represent the earliest and the tardiest due date; (*iii*) If the job x is finalized before E_x, there is a cost α_x per unit of earliness time. Similarly, a cost β_x is associated to the tardiness. Jobs that are completed within their time windows do not generate costs; (*iv*) The machine can perform only one job at a time and once the process is initiated, it can not be interrupted; (*v*) All jobs are available for processing on date 0; and (*vi*) Idle time between the execution of two consecutive jobs is allowed. The starting date of job $x \in I$ is represented by s_x, whereas the earliness and tardiness times of x are represented by $e_x = \max(0,\ E_x - s_x - P_x)$ and $t_x = \max(0, s_x + P_x - T_x)$, respectively.

The objective is to determine the starting dates of the jobs so that the weighted sum of their earliness and tardiness is minimized, i.e.,

$$\min \sum_{x \in I} (\alpha_x e_x + \beta_x t_x). \tag{1}$$

We present a Mathematical Programming model and a Constraint Programming model for the SMSPTW. The efficiency of the models are compared over sets of instances.

*Sponsored by CNPq - Brazil.

2 Mathematical Programming model - MP

The Mathematical Programming model of [2] can be adapted to formulate the SMSPTW. For two distinct jobs x and y, let w_{xy} be binary variables equal to 1 if job x precedes job y, and 0, otherwise. Given a sufficiently large number M, the SMSPTW can be formulated as follows:

$$\min \quad \sum_{x \in I} (\alpha_x e_x + \beta_x t_x) \tag{2}$$

$$\text{s.t.} \quad s_x + P_x + M(w_{xy} - 1) \leq s_y \quad \forall\, x, y \in I \quad \text{and} \quad x \neq y \tag{3}$$

$$w_{xy} + w_{yx} = 1 \quad \forall\, x, y \in I \quad \text{and} \quad x \neq y \tag{4}$$

$$s_x + P_x + e_x \geq E_x \quad \forall\, x \in I \tag{5}$$

$$s_x + P_x - t_x \leq T_x \quad \forall\, x \in I \tag{6}$$

$$s_x \geq 0 \quad \forall\, x \in I \tag{7}$$

$$e_x \geq 0 \quad \forall\, x \in I \tag{8}$$

$$t_x \geq 0 \quad \forall\, x \in I \tag{9}$$

$$w_{xy} \in \{0, 1\} \quad \forall\, x, y \in I \tag{10}$$

3 Constraint Programming model - CP

The set of modeling objects IBM ILOG Concert Technology [3] are used to formulate a Constraint Programming model of the SMSPTW. In this model, each job $x \in I$ is associated to an interval decision variable of size P_x. As only one job can be performed at a time the specialized constraint `IloNoOverlap` are considered. The tardiness and the earliness of each job, as well as the objective function, are modeled using the `IloEndOf` function.

4 Computational results

Table 1 presents a comparison of MP and CP models. For each number of jobs, 16 instances are considered.

# jobs		10	11	12	13	14	15	16	17	18
Average	**MP**	1.79	5.78	12.27	28.27	-	-	-	-	-
time (s)	**CP**	0.40	0.81	0.75	1.88	5.08	15.64	81.44	-	-

TAB. 1: Average times required by CPLEX to solve to optimality 16 instances with the MP and CP models. Character '-' is used whenever a model fails at solving at least one of the 16 instances.

Table 1 shows that the average computation time obtained with CP model is always significantly higher than the one obtained with MP.

References

[1] Adam Janiak, Władysław A. Janiak, Tomasz Krysiak, and Tomasz Kwiatkowski. A survey on scheduling problems with due windows. *European Journal of Operational Research*, 242(2):347 – 357, 2015.

[2] A. S. Manne. On the job-shop scheduling problem. *Operations Research*, 8:219–223, 1960.

[3] IBM ILOG CPLEX Optimization Studio. *CP Optimizer User's Manual*, 2015.