SAPIENZA
UNIVERSITÀ DI ROMA

A new mixed-integer modeling approach for capacity-constrained continuous-time scheduling problems

Scuola di Scienza e Tecnologia dell'informazione e delle comunicazioni

Dottorato di Ricerca in Automatica Bioingegneria e Ricerca Operativa – XXXI Ciclo

Candidate

Ludovica Maccarrone
ID number 1309929

Thesis Advisor

Prof. Stefano Lucidi

A thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Automatica Bioingegneria e Ricerca Operativa

October 2018

Thesis defended on 26 February 2019
in front of a Board of Examiners composed by:

Prof. Giancarlo Bigi, Università di Pisa (chairman)

Prof. Giovanni Sparacino, Università di Padova

Prof. Giovanni Ulivi, Università di Roma Tre

A new mixed-integer modeling approach for capacity-constrained continuous-time scheduling problems
Ph.D. thesis. Sapienza – University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: ludovica.maccarrone@uniroma1.it

Dedicated to my family.
For their love, support, and
constant encouragement.

Abstract

Nowadays, scheduling and resource management are increasingly important issues for organizations. Indeed, they do not only constitute an underlying necessity to make things work properly within the companies, but are and will always be more critical means to reduce costs and get competitive advantage in the market.

Different approaches have been typically employed for these problems during the years. Among the others, linear programming techniques represent a valid tool that, despite applicable only to instances of limited dimension, offers an extremely flexible modeling opportunity, able to produce either optimal or approximate solutions of certified quality. In this spirit, the definition of suitable indicator variables and the use of particular constraints are proposed in the present work, with the aim of providing a useful basis for different mathematical models, taking into account scarce resources and other potential limitations. More in detail, a very well-known problem from the literature, the Resource Constrained Project Scheduling Problem, is investigated, and a new mixed-integer linear formulation is introduced, which treats time as a continuous variable. The considered model presents several advantages from the computational point of view, that are deeply studied and compared with those of one of the best methods recently developed in the same field. Extensive experiments reveal the good performances achieved by the proposed formulation over all the KPIs included in the analysis, thus motivating further applications to derived problems, such as the workforce planning and scheduling framework presented at the end of this dissertation.

Acknowledgments

Firstly, I would like to express my gratitude to my advisor Prof. Stefano Lucidi for its guidance during all the last years, for his patience, motivation, and knowledge. He helped me from the earliest study to the writing of this thesis, supporting my work with the best suggestions and discussions.

I also sincerely thank ACT-OR and all its staff, for having encouraged me to start the PhD program and for having inspired this research project.

I wish to say thank you to my colleagues of DIAG, who shared with me this important experience, full of work days, full of thoughts night, and funny moments.

Last but not least, my special thanks go to Tommaso and to all my family and friends for their kind and precious support during the last period and during my life in general.

Contents

List of Figures

List of Tables

Introduction

It is well-known in the field of operations research and applied mathematics in general that the same problem can normally be modeled in several ways, and that distinct methods can often lead to significantly different performances of solution procedures. Considering this observation with reference to linear programming techniques implies to recognize that, assuming equivalent conditions and calculation times, different formulations can obtain very different results, in particular when integer or binary variables are included. This is one of the reasons why the literature on ILP and MILP approaches has been proliferating over the years, and undoubtedly constitutes a fundamental conception at the basis of the work proposed here.

Starting from these considerations, we have posed our attention on a specific branch of the theory on model building, focusing on the use of logical variables and indicator constraints, with special regard to their employment in scheduling problems, which is really widespread.

Indeed, scheduling problems imply the calculation of feasible timetables for groups of activities. In such situations, many "yes or no" decisions typically have to be taken. These in general represent the need to avoid conflicts potentially arising between contending operations, as for example to establish which activity should be processed first or to determine if a certain allocation is convenient. In particular, many decisions can be seen as answers related to the use of physical or conceptual resources that are available in limited quantities. This is for instance the case of machines, operators, time slots, energy, data storage capacity, computation time, and so on.

Looking at time management issues from this point of view, it is easy to understand the theoretical and practical interest behind the study of the so-called Resource Constrained Project Scheduling Problem (RCPSP) here considered. It is indeed a generalization of the Job-Shop, Flow-Shop and Open-Shop scheduling problems famous in production contexts, involving not only the restrictions imposed by precedence relationships, but also the constraining effects generated by the temporary requirement of additional scarce resources.

The RCPSP belongs to the class of NP-hard optimization problems. Nevertheless, it is studied by many researchers for its general structure and interesting applicability to several circumstances in production planning and software implementation. Moreover, since also human labor can be identified as a resource, it can naturally be thought as a model for applications in the fields of project and workforce management, which are increasingly emerging topics for today's organizations.

In this dissertation, we present a new continuous time MILP formulation for the RCPSP and compare it to one of the best methods in the related literature,

showing very promising results on a large number of test instances, including the most broadly used. Indeed, the proposed new model achieves better results in terms of both computational time spent and solution quality attained.

In the following, each of the above mentioned topics will be investigated in details. The work is organized as follows: in Chapter 1 the basic building blocks of any mathematical formulation including binary variables and indicator constraints are introduced. The aim of such mathematical tools is deeply investigated and some particular applications to activities scheduling are discussed. In Chapter 2 the analysis is focused on standard modeling techniques for scheduling problems and especially on their extension to represent more difficult settings where capacity constraints are present, by means of the basic building blocks introduced in Chapter 1. Chapter 3 constitute the main contribution of this dissertation. After a formal description of the RCPSP and a comprehensive literature review on the MILP solution approaches to it, a new formulation, based on the constraints and variables analyzed in Chapter 2, is proposed. The new formulation is then compared to some state-of-the-art works both from a formal and numerical point of view. Finally, in Chapter 4 a potential application of the above proposed new mathematical model to workforce planning and scheduling problems is presented. It is based on a black-box framework leveraging a simulation-optimization paradigm.

Chapter 1

Indicator variables and constraints

MILP approaches make frequent use of indicator variables and associated constraints. They indeed permit to easily model logical connections between different components of the formulation, thus allowing a great flexibility.

The aim of this chapter is to provide a comprehensive overview of the main modeling opportunities offered by the employment of these techniques. In particular, Section 1.1 introduces the basic notions, and Section 1.2 presents the methodologies to logically connect two general inequality constraints. Then, Section 1.3 links such logical relations to suitable binary variables. Finally, Section 1.4 focuses on some applications of previous arguments to scheduling problems.

1.1 Basic notions

In mathematics, inequalities represent conditions that variables, depending on their values, either satisfy or not. In particular, let us consider a simple optimization problem in which $x \in \mathbb{R}^n$ is the decision variable and $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function. If we include in the formulation a general less-than-or-equal constraint $g(x) \leq 0$, with $g : \mathbb{R}^n \to \mathbb{R}$, we get the following:

$$\min_{x \in \mathbb{R}^n} f(x)$$
$$g(x) \leq 0$$

Here, $g(x) \leq 0$ defines the feasible region, i.e. the values of x for which the inequality is satisfied. If a vector $\bar{x}$ is such that $g(\bar{x}) > 0$, then it can not be a solution for the problem since the purpose of the constraint is just to exclude it from the possible results.

In some cases, however, formulations involve inequalities not as explicit limitations of the feasible space, but rather with a role connected to the value of a binary variable, which is said indicator. Indicator variables have a dual role: on the one hand they allow to force particular relations to be true when some conditions are met, on the other they permit to detect if inequalities are satisfied for certain values of problem main variables.

For example, if in an optimization model we want to indicate that if $\gamma \in \{0, 1\}$ is "active" (i.e. $\gamma = 1$) then $g(x)$ has to be less than or equal to 0, we may use the so called big-M formulation and write:

$$g(x) \leq M(1 - \gamma),$$

where M is supposed to be a large enough constant such that

$$g(x) \leq M$$

is always satisfied in the problem under consideration. This idea has the advantage of preserving constraint linearity if g is an affine linear function, and allows to simply express the condition:

$$\gamma = 1 \Rightarrow g(x) \leq 0,$$

or equivalently:

$$g(x) > 0 \Rightarrow \gamma = 0.$$

The first implication tells that $\gamma = 1$ forces (and indeed it is sufficient to have) $g(x) \leq 0$. The second clarifies the idea behind the term "indicator variable" since γ "indicates" the status of $g(x)$, in particular taking value 0 when $g(x) \leq 0$ is not satisfied.

Similarly, we may use γ to control the opposite condition, that is:

$$g(x) \leq 0 \Rightarrow \gamma = 1,$$

which is identical to:

$$\gamma = 0 \Rightarrow g(x) > 0.$$

This can be expressed considering a very small positive parameter ϵ and a sufficiently negative constant m such that the inequality

$$m < g(x)$$

is always true in the particular problem under study. Indeed, we can impose (once again, without introducing any nonlinearity):

$$m\gamma + \varepsilon \leq g(x),$$

where ε is established in accordance with instances considered and has the role of avoiding issues on the strict inequality $g(x) > 0$, practically specifying a precision to be checked (e.g., 10^{-3}). In other words, it transforms the initial conditions in $g(x) \leq 0 \Rightarrow \gamma = 1$ and $\gamma = 0 \Rightarrow g(x) \geq \varepsilon$.

Thus, $\gamma = 1$ is necessary to have $g(x) \leq 0$ (it indicates that constraint is satisfied). If $\gamma = 0$, $g(x) \leq 0$ has to be false.

Notice that using both the above derived relations, i.e.

$$m\gamma + \varepsilon \leq g(x) \leq M(1 - \gamma),$$

we get a necessary and sufficient condition:

$$\gamma = 1 \Leftrightarrow g(x) \leq 0.$$

Condition	Equivalent condition	Constraints
$\gamma = 1 \Rightarrow g(x) \leq 0$	$g(x) > 0 \Rightarrow \gamma = 0$	$g(x) \leq M(1-\gamma)$
$g(x) \leq 0 \Rightarrow \gamma = 1$	$\gamma = 0 \Rightarrow g(x) > 0$	$m\gamma + \varepsilon \leq g(x)$
$\gamma = 1 \Leftrightarrow g(x) \leq 0$	$\gamma = 0 \Leftrightarrow g(x) > 0$	$m\gamma + \varepsilon \leq g(x) \leq M(1-\gamma)$
$\gamma = 1 \Rightarrow g(x) \geq 0$	$g(x) < 0 \Rightarrow \gamma = 0$	$m(1-\gamma) \leq g(x)$
$g(x) \geq 0 \Rightarrow \gamma = 1$	$\gamma = 0 \Rightarrow g(x) < 0$	$g(x) \leq M\gamma - \varepsilon$
$\gamma = 1 \Leftrightarrow g(x) \geq 0$	$\gamma = 0 \Leftrightarrow g(x) < 0$	$m(1-\gamma) \leq g(x) \leq M\gamma - \varepsilon$
$\gamma = 0 \Rightarrow g(x) \leq 0$	$g(x) > 0 \Rightarrow \gamma = 1$	$g(x) \leq M\gamma$
$g(x) \leq 0 \Rightarrow \gamma = 0$	$\gamma = 1 \Rightarrow g(x) > 0$	$m(1-\gamma) + \varepsilon \leq g(x)$
$\gamma = 0 \Leftrightarrow g(x) \leq 0$	$\gamma = 1 \Leftrightarrow g(x) > 0$	$m(1-\gamma) + \varepsilon \leq g(x) \leq M\gamma$
$\gamma = 0 \Rightarrow g(x) \geq 0$	$g(x) < 0 \Rightarrow \gamma = 1$	$m\gamma \leq g(x)$
$g(x) \geq 0 \Rightarrow \gamma = 0$	$\gamma = 1 \Rightarrow g(x) < 0$	$g(x) \leq M(1-\gamma) - \varepsilon$
$\gamma = 0 \Leftrightarrow g(x) \geq 0$	$\gamma = 1 \Leftrightarrow g(x) < 0$	$m\gamma \leq g(x) \leq M(1-\gamma) - \varepsilon$

Table 1.1. Single-constraint logical implications and corresponding formulation.

Other possible cases, together with those already described, are briefly summarized in Table 1.1, where for simplicity we consider $m + \varepsilon \leq g(x) \leq M - \varepsilon$.

This clarifies all the possible ways to link a binary variable to a general inequality relation. In some cases, however, it could make sense to extend this approach to consider more than one constraint like $g(x) \leq 0$. In particular, it is interesting to analyze what happens when the inequalities to be linked to the binary variable become two or more. A single relation as $g(x) \leq 0$, indeed, only admits two logical possibilities: either it is true, or it is false ($g(x) > 0$ equals to **NOT** $g(x) \leq 0$). When constraints are two, more options are available, paving the way to several boolean operations that can benefit of being examined. For this reason, in the following section we present the possible logical relations between couples of inequality constraints, while in Section 1.3 we introduce how to link the satisfaction or non-satisfaction of these relations to suitable indicator variables.

1.2 Logical relations between inequality constraints

Let us consider a variable $x \in \mathbb{R}^n$ and two general constraints $g(x) \leq 0$ and $h(x) \leq 0$, where $g : \mathbb{R}^n \to \mathbb{R}$ and $h : \mathbb{R}^n \to \mathbb{R}$. If these constraints must be satisfied at the same time, they can be simply included in the set of conditions defining the solution space of the problem by writing:

$$\begin{cases} g(x) \leq 0 \\ h(x) \leq 0 \end{cases}$$

This implies that a solution $x' \in \mathbb{R}^n$ is feasible if and only if both $g(x') \leq 0$ and $h(x') \leq 0$ are true. Using the appropriate boolean operator, we say that

$$g(x) \leq 0 \textbf{ AND } h(x) \leq 0$$

must hold. If we schematically represent this concept, we obtain the truth Table 1.2.

$g(x) \leq 0$	$h(x) \leq 0$	$g(x) \leq 0$ **AND** $h(x) \leq 0$
F	F	F
F	T	F
T	F	F
T	T	T

Table 1.2. $g(x) \leq 0$ **AND** $h(x) \leq 0$ truth table.

A different situation applies when the conditions expressed by the constraints may be alternative, i.e. when just one of the two inequalities must necessarily be satisfied (see truth Table 1.3). In this case, a binary variable θ is added to the problem and the solution space is defined as follows:

$$\begin{cases} g(x) \leq M_g \theta \\ h(x) \leq M_h (1 - \theta) \end{cases}$$

where M_g and M_h are two large constants that constitute the upper bounds respectively for $g(x)$ and $h(x)$. This formulation preserves linearity if g and h are linear functions, and allows to indicate that at least one of the two constraints must be met. Indeed, if θ is equal to 0, $g(x) \leq 0$ and the second constraint is relaxed, otherwise θ is equal to 1 and only $h(x) \leq 0$ influences the feasible region. In other words, this guarantees

$$g(x) \leq 0 \textbf{ OR } h(x) \leq 0$$

to be valid.

$g(x) \leq 0$	$h(x) \leq 0$	$g(x) \leq 0$ **OR** $h(x) \leq 0$
F	F	F
F	T	T
T	F	T
T	T	T

Table 1.3. $g(x) \leq 0$ **OR** $h(x) \leq 0$ truth table.

Notice that while the **AND** relation could be easily extended to deal with more than two constraints, simply including the additional inequalities in the formulation, this is not true for the **OR**, which would require to introduce other binary variables. However, for the purpose of this chapter, we will consider only the possible relations within a couple of constraints, trying to provide a theoretical basis to tackle more complex situations.

In this respect, we may analyze another logical function that is quite common between two conditions: the **XOR**, i.e. the exclusive OR, which forces one and

exactly one constraint to be true, as shown in Table 1.4. Assuming m_g and m_h to be lower bounds for $g(x)$ and $h(x)$,

$$g(x) \leq 0 \ \textbf{XOR} \ h(x) \leq 0$$

can be expressed by the following:

$$\begin{cases} g(x) \leq M_g \theta \\ h(x) \leq M_h (1 - \theta) \\ m_g (1 - \theta) + \varepsilon \leq g(x) \\ m_h \theta + \varepsilon \leq h(x) \end{cases}$$

Here, if θ is equal to 0, then $m_g < g(x) \leq 0$ and $0 < h(x) \leq M_h$; otherwise θ is 1, $0 < g(x) \leq M_g$ and $m_h < h(x) \leq 0$. In both cases one constraint is satisfied and the other is not.

$g(x) \leq 0$	$h(x) \leq 0$	$g(x) \leq 0$ **XOR** $h(x) \leq 0$
F	F	F
F	T	T
T	F	T
T	T	F

Table 1.4. $g(x) \leq 0$ **XOR** $h(x) \leq 0$ truth table.

Other common examples of logical relations existing between two conditions are the complements of those previously listed. In particular, the opposite of the **AND** operator is the **NAND**. Therefore

$$g(x) \leq 0 \ \textbf{NAND} \ h(x) \leq 0$$

indicates that at most one of the constraints has to be valid. It is implied by the expressions:

$$\begin{cases} m_g (1 - \theta) + \varepsilon \leq g(x) \\ m_h \theta + \varepsilon \leq h(x) \end{cases}$$

It may be either $g(x) > 0$, $h(x) > 0$ or both; however $g(x) \leq 0$ and $h(x) \leq 0$ is not possible, as also reported in Table 1.5.

$g(x) \leq 0$	$h(x) \leq 0$	$g(x) \leq 0$ **NAND** $h(x) \leq 0$
F	F	T
F	T	T
T	F	T
T	T	F

Table 1.5. $g(x) \leq 0$ **NAND** $h(x) \leq 0$ truth table.

The logical **NOR** is the inverse of the **OR** (see truth Table 1.6). This means condition

$$g(x) \leq 0 \textbf{ NOR } h(x) \leq 0$$

implies original constraints (i.e., $g(x) \leq 0$ and $h(x) \leq 0$) not to be satisfied. Therefore:

$$\begin{cases} \varepsilon \leq g(x) \\ \varepsilon \leq h(x) \end{cases}$$

$g(x) \leq 0$	$h(x) \leq 0$	$g(x) \leq 0$ **NOR** $h(x) \leq 0$
F	F	T
F	T	F
T	F	F
T	T	F

Table 1.6. $g(x) \leq 0$ **NOR** $h(x) \leq 0$ truth table.

The **XNOR** operator admits two possibilities: either constraints are both satisfied or both unsatisfied, which constitutes the opposite of the **XOR**. In order to express

$$g(x) \leq 0 \textbf{ XNOR } h(x) \leq 0,$$

the following are used:

$$\begin{cases} g(x) \leq M_g \theta \\ h(x) \leq M_h \theta \\ m_g(1-\theta) + \varepsilon \leq g(x) \\ m_h(1-\theta) + \varepsilon \leq h(x) \end{cases}$$

Indeed, if θ is 0, then $m_g < g(x) \leq 0$ and $m_h < h(x) \leq 0$; if $\theta = 1$, $0 < g(x) \leq M_g$ and $0 < h(x) \leq M_h$. **XNOR** truth table is given in Table 1.7.

$g(x) \leq 0$	$h(x) \leq 0$	$g(x) \leq 0$ **XNOR** $h(x) \leq 0$
F	F	T
F	T	F
T	F	F
T	T	T

Table 1.7. $g(x) \leq 0$ **XNOR** $h(x) \leq 0$ truth table.

Finally, we consider the last two non-trivial relations that can be imposed between two general inequalities, which respectively represent the concept of implication and its complement. When constraints are linked by a relation specifying that if the first is met then also the second must be satisfied, the **IF-THEN** operator is used. Practically, to express

$$\textbf{IF } g(x) \leq 0 \textbf{ THEN } h(x) \leq 0,$$

it is possible to introduce these inequalities:

$$\begin{cases} m_g\theta + \varepsilon \leq g(x) \\ h(x) \leq M_h(1-\theta) \end{cases}$$

When θ is 0, $g(x) > 0$ and no limitations affect $h(x)$. Otherwise, if $\theta = 1$, $h(x) \leq 0$ and $g(x)$ is free. Thus the satisfaction of the first constraint implies also the second to be met since it is not possible to have $g(x) \leq 0$ and $h(x) > 0$ at the same time (see Table 1.8).

$g(x) \leq 0$	$h(x) \leq 0$	**IF** $g(x) \leq 0$ **THEN** $h(x) \leq 0$
F	F	T
F	T	T
T	F	F
T	T	T

Table 1.8. IF $g(x) \leq 0$ **THEN** $h(x) \leq 0$ truth table.

The complement of this situation is realized when $g(x) \leq 0$ has to be true and $h(x) \leq 0$ false, as reported in Table 1.9. In this case we say that

$$g(x) \leq 0 \textbf{ BUT-NOT } h(x) \leq 0$$

must hold and simply write:

$$\begin{cases} g(x) \leq 0 \\ \varepsilon \leq h(x) \end{cases}$$

$g(x) \leq 0$	$h(x) \leq 0$	$g(x) \leq 0$ **BUT-NOT** $h(x) \leq 0$
F	F	F
F	T	F
T	F	T
T	T	F

Table 1.9. $g(x) \leq 0$ **BUT-NOT** $h(x) \leq 0$ truth table.

Notice that the **IF-THEN** and the **BUT-NOT** relations are both non-symmetric in the sense that constraints, in contrast to all other cases, assume different roles in defining the feasible region. For example, (**IF** a **THEN** b) does not coincide with (**IF** b **THEN** a). It follows that, in general, it would make sense to also describe the inverse implication. However the result can be banally obtained by substituting $g(x)$ and $h(x)$ in previous formulation (as well as m_g and m_h, and M_h and M_g).

In this section, we have only taken into account logical relations between two general less-than-or-equal constraints $g(x) \leq 0$ and $h(x) \leq 0$. Before concluding, let us consider the following remarks to reduce all other cases to those already addressed:

1. If a constraint is a greater-then-or-equal inequality, $g(x) \geq 0$, it is obviously possible to apply $g'(x) = -g(x)$ and consequently get $g'(x) \leq 0$.

2. If a constraint is a strict greater-than inequality, $g(x) > 0$, it's complement is a less-then-or-equal relation, i.e. $\mathbf{NOT}\,(g(x) > 0) = g(x) \leq 0$. Therefore, a different logical operation can be used. In particular:

   $g(x) > 0$ **AND** $h(x) \leq 0$) equals to $h(x) \leq 0$ **BUT-NOT** $g(x) \leq 0$;
   $g(x) > 0$ **AND** $h(x) > 0$) equals to $g(x) \leq 0$ **NOR** $h(x) \leq 0$;
   $g(x) > 0$ **OR** $h(x) \leq 0$) equals to **IF** $g(x) \leq 0$ **THEN** $h(x) \leq 0$;
   $g(x) > 0$ **OR** $h(x) > 0$) equals to $g(x) \leq 0$ **NAND** $h(x) \leq 0$;
   $g(x) > 0$ **XOR** $h(x) \leq 0$ equals to $g(x) \leq 0$ **XNOR** $h(x) \leq 0$;
   $g(x) > 0$ **XOR** $h(x) > 0$ equals to $g(x) \leq 0$ **XOR** $h(x) \leq 0$;
   $g(x) > 0$ **NAND** $h(x) \leq 0$ equals to **IF** $h(x) \leq 0$ **THEN** $g(x) \leq 0$;
   $g(x) > 0$ **NAND** $h(x) > 0$ equals to $g(x) \leq 0$ **OR** $h(x) \leq 0$;
   $g(x) > 0$ **NOR** $h(x) \leq 0$ equals to $g(x) \leq 0$ **BUT-NOT** $h(x) \leq 0$;
   $g(x) > 0$ **NOR** $h(x) > 0$ equals to $g(x) \leq 0$ **AND** $h(x) \leq 0$;
   $g(x) > 0$ **XNOR** $h(x) \leq 0$ equals to $g(x) \leq 0$ **XOR** $h(x) \leq 0$;
   $g(x) > 0$ **XNOR** $h(x) > 0$ equals to $g(x) \leq 0$ **XNOR** $h(x) \leq 0$;
   IF $g(x) > 0$ **THEN** $g(x) \leq 0$ equals to $g(x) \leq 0$ **OR** $h(x) \leq 0$;
   IF $g(x) \leq 0$ **THEN** $h(x) > 0$ equals to $g(x) \leq 0$ **NAND** $h(x) \leq 0$;
   IF $g(x) > 0$ **THEN** $h(x) > 0$ equals to **IF** $h(x) \leq 0$ **THEN** $g(x) \leq 0$;
   $g(x) > 0$ **BUT-NOT** $h(x) \leq 0$ equals to $g(x) \leq 0$ **NOR** $h(x) \leq 0$;
   $g(x) \leq 0$ **BUT-NOT** $h(x) > 0$ equals to $g(x) \leq 0$ **AND** $h(x) \leq 0$;
   $g(x) > 0$ **BUT-NOT** $h(x) > 0$ equals to $h(x) \leq 0$ **BUT-NOT** $g(x) \leq 0$.

3. If a constraint is a strict less-than inequality, $g(x) < 0$, both previous observations need to be considered.

1.3 Indicator variables for inequalities logical relations

In the last section we have seen that, given two generic inequalities, there are several ways in which they can be logically connected and that each connection can be easily forced by introducing a set of appropriate constraints.

In some cases, however, the problem is not to limit the feasible space due to an existing restriction given by different conditions, but rather to identify if a logical relation holds in order to use this information in other parts of the formulation, as in different constraints or in the objective function. In other words, it may happen that a particular relation between constraints $g(x) \leq 0$ and $h(x) \leq 0$ (such as the **AND**, **OR**, **XOR**, etc. previously described) has not to be forced, being optional, but just to be "monitored", linking it to a binary variable used elsewhere in the model.

In order to clarify this idea, we now introduce a simple example that we will later exploit for deriving a general formulation strategy.

1.3.1 Introductory example

Let us consider a small production environment where a single unit of a product p is manufactured by using 10 units of a raw material r. For technological reasons, there is a limit on the capacity per period equal to 400 parts. We assume that the supplier of r has a pricing policy based on the ordered quantity; in particular, we imagine he offers each unit of r at a cost of 5 dollars if a customer orders less than 1000 pieces, 4 dollars if the quantity is between 1000 and 3000, and only 3 dollars if the request is strictly higher than 3000 units. Furthermore we suppose the demand curve to be given by $y = 100 - \frac{1}{5}x$, where y and x are respectively the selling price and the quantity of p. Then, assuming to reason for a single period, the optimal choice for the manufacturer is given by the solution of the following integer quadratic problem with linear constraints:

$$\max_{x,\gamma_a,\gamma_b,\gamma_c} (100 - \frac{1}{5}x)x - 50x\gamma_a - 40x\gamma_b - 30x\gamma_c \tag{1.1}$$

$$\text{subject to: } m_a\gamma_a \leq 10x - 1000 \tag{1.2}$$

$$m_{b_l}\gamma_b + m_{b_l}\theta + \varepsilon \leq 1000 - 10x \tag{1.3}$$

$$m_{b_u}(1-\theta) + \varepsilon \leq 10x - 3000 \tag{1.4}$$

$$m_c\gamma_c \leq 3000 - 10x \tag{1.5}$$

$$x \leq 400 \tag{1.6}$$

$$x \in \mathbb{Z}_+ \tag{1.7}$$

$$\theta \in \{0,1\},\ \gamma_a \in \{0,1\},\ \gamma_b \in \{0,1\},\ \gamma_c \in \{0,1\} \tag{1.8}$$

where ε is a small parameter to avoid strict inequality constraints (practically here $\varepsilon \in (0, 10)$ because x is integer), $m_a \leq -1000$, $m_{b_l} \leq -3000 - \varepsilon$, $m_{b_u} \leq -3000 - \varepsilon$, and $m_c \leq -1000$.

In order to understand this formulation, notice first of all that solving it is exactly the same that finding the best solution of the three subproblems obtainable by limiting the ordered quantity between 0 and 1000, between 1000 and 3000, or between 3000 and the upper limit (at most 4000 units of r are required due to the capacity restriction), i.e.:

(a)	(b)	(c)
$\max_{x\in\mathbb{Z}_+} (100 - \frac{1}{5}x)x - 50x$	$\max_{x\in\mathbb{Z}_+} (100 - \frac{1}{5}x)x - 40x$	$\max_{x\in\mathbb{Z}_+} (100 - \frac{1}{5}x)x - 30x$
$0 \leq 10x < 1000$	$1000 \leq 10x \leq 3000$	$3000 < 10x \leq 4000$

It is easy to verify that the maximum profit can be reached when $x = 150$, that corresponds to the scenario in which each unit of raw material is bought at 4 dollars (b), while cases (a) and (c) are dominated.

Following this idea, we can define three binary variables γ_a, γ_b and γ_c such that:

$$0 \leq 10x < 1000 \Rightarrow \gamma_a = 1, \tag{1.9}$$

$$1000 \leq 10x \leq 3000 \Rightarrow \gamma_b = 1, \tag{1.10}$$

$$3000 < 10x \leq 4000 \Rightarrow \gamma_c = 1, \tag{1.11}$$

and build a unique formulation having (1.1) as objective function.

Since $0 \leq x \leq 400$ due to (1.6) and (1.7), inequality (1.2) is enough to provide (1.9), and (1.5) suffices to guarantee (1.11). Condition (1.10) is a little more complex, but is assured by relations (1.3) and (1.4), where the auxiliary binary variable θ is used. Indeed, if $100 \leq x \leq 300$, both right-hand sides of (1.3) and (1.4) become negative, implying $\theta = 0$ and $\gamma_b = 1$. In all other cases ($x < 100$ or $x > 300$) γ_b is set to 0 to minimize the objective function, while θ is adjusted to satisfy both constraints.

Thus relations (1.1)-(1.8) allow to combine sub-problems (a), (b) and (c) in a single mathematical formulation providing the same optimal solution, i.e.:

$$x = 150,\ \theta = 0,\ \gamma_a = 0,\ \gamma_b = 1,\ \gamma_c = 0.$$

Notice here the different roles assumed by binary variables: we may observe that, once considered $x = 150$, θ and γ_b are forced to their value by inequalities (1.3)-(1.4). Differently, γ_a and γ_c do not become 0 due to constraining limitations, but simply as a consequence of the objective function effect. Indeed, also solution $x = 150,\ \theta = 0,\ \gamma_a = 1,\ \gamma_b = 1,\ \gamma_c = 1$ would be feasible for the problem, but it would provide a lower profit.

To motivate this fact, we point out that (1.2)-(1.5) constitute *necessary conditions* to activate γ_a, γ_b or γ_c when x satisfies the corresponding left-hand side among (1.9)-(1.11). However they are not *sufficient*, because they do not ensure the opposite implications.

Nevertheless, in some cases it could be needed to express also (or only) some conditions linking the activation of an indicator variable to the necessary satisfaction of particular constraints. For example, considering the same situation as before, let us now imagine there is another possibility for the manufacturer, who can also decide to order the overall quantity of raw material r from a different supplier. This distributor has a fixed price of 3.5 dollars per piece, but for contractual reasons, he can only conclude agreements for a number of units between 2000 and 3500. In this case, the problem can be formulated as follows:

$$\max_{x,\gamma_a,\gamma_b,\gamma_c,\delta} (100 - \frac{1}{5}x)x - (50x\gamma_a + 40x\gamma_b + 30x\gamma_c)(1-\delta) - 35x\delta \tag{1.12}$$

$$\text{subject to: } m_a\gamma_a \leq 10x - 1000 \tag{1.13}$$

$$m_{b_l}\gamma_b + m_{b_l}\theta + \varepsilon \leq 1000 - 10x \tag{1.14}$$

$$m_{b_u}(1-\theta) + \varepsilon \leq 10x - 3000 \tag{1.15}$$

$$m_c\gamma_c \leq 3000 - 10x \tag{1.16}$$

$$x \leq 400 \tag{1.17}$$

$$2000 - 10x \leq M_l(1-\delta) \tag{1.18}$$

$$10x - 3500 \leq M_u(1-\delta) \tag{1.19}$$

$$x \in \mathbb{Z}_+ \tag{1.20}$$

$$\theta \in \{0,1\},\ \gamma_a \in \{0,1\},\ \gamma_b \in \{0,1\},\ \gamma_c \in \{0,1\},\ \delta \in \{0,1\} \tag{1.21}$$

where $M_l \geq 2000$, $M_u \geq 500$, and variable δ represents the decision about the supplier; if $\delta = 1$ the new distributor is chosen, otherwise the initial option is selected

and $\delta = 0$. This is also confirmed by objective function (1.12), where costs for buying raw materials from the first or from the second vendor become alternative, being activated depending on the value of δ.

Linear constraints (1.13)-(1.17) are exactly the same as (1.2)-(1.5), while (1.18) and (1.19) allow to express condition:

$$\delta = 1 \Rightarrow 2000 \leq 10x \leq 3500. \tag{1.22}$$

Indeed, if $\delta = 1$ (i.e. if r is bought from the second supplier), right-hand sides of (1.18)-(1.19) become 0 and $200 \leq x \leq 350$; otherwise $\delta = 0$ and restrictions are relaxed.

It is possible to verify that an optimal solution for this problem is:

$$x = 200,\ \theta = 0,\ \gamma_a = 0,\ \gamma_b = 1,\ \gamma_c = 0,\ \delta = 1,$$

which involves to select the second distributor and to buy 2000 pieces of r for the production of 200 units of p. The same result could also be retrieved by solving a simple sub-problem representing a fourth scenario (d) to be compared with the previous three considered, that depicts the hypothetical situation in which only the second supplier is available, i.e.:

$$\begin{aligned} & & (d) \\ \max_{x \in \mathbb{Z}_+} \ & (100 - \frac{1}{5}x)x - 35x \\ & 2000 \leq 10x \leq 3500 \end{aligned}$$

However, notice once again the convenience of taking into account a single model formulation instead of several ones to be separately solved. This observation assumes even more value when the problem, by nature, can not be decomposed in different sub-problems and thus the use of indicator variables becomes the only way to find the real optimum. To imagine an example, consider what would happen if suppliers were not exclusive: in such a case the manufacturer would have to decide two quantities, y_1 and y_2, representing the amounts of raw material respectively provided by the first and the second distributors, subject to $y_1 + y_2 = 10x$. This would constraint the model, making it impossible to separate the problems.

This example should be enough to convince the reader about the utility of using indicator variables, either *forced by* or *forcing* particular inequalities. To continue our analysis, let us focus on relations (1.14)-(1.15) and (1.18)-(1.19). Referring to boolean operators introduced in the previous section, we may say they express the following:

$$1000 \leq 10x \ \textbf{AND} \ 10x \leq 3000 \Rightarrow \gamma_b = 1,$$
$$\delta = 1 \Rightarrow 2000 \leq 10x \ \textbf{AND} \ 10x \leq 3500,$$

or equivalently:

$$1000 - 10x \leq 0 \ \textbf{AND} \ 10x - 3000 \leq 0 \Rightarrow \gamma_b = 1,$$
$$\delta = 1 \Rightarrow 2000 - 10x \leq 0 \ \textbf{AND} \ 10x - 3500 \leq 0,$$

where γ_b is *forced* to be (i.e. is *necessarily*) 1 when the left-side **AND** condition is satisfied, while $\delta = 1$ *forces* (i.e. is *sufficient* to imply) its respective right-side **AND** expression to be true.

1.3.2 Important remarks on big-M formulations

In the previous example, we have not only introduced a strategy for modeling logical implications, but also implicitly suggested a possible way for choosing formulation parameters m_a, m_{b_l}, m_{b_u}, m_c, M_l, M_u and ε. It is very important to recall, indeed, that big-M formulations have very well-known advantages, disadvantages and characteristics that is worth to mention here:

- Linearity: as already noticed, big-M formulations provide a clean and flexible modeling tool to deal with nonlinearities and logical implications, offering the opportunity to classify models into common categories (MILP, MIQP) presenting only linear inequalities. This is really an important aspect from a practical point of view because linearly-constrained problems are typically the most spread and easy to tackle with standard solvers.

- Continuous relaxation: it is well-known that big-M constraints have a weak continuous relaxation. Indeed, depending on the values of coefficients M and m, relaxed indicator variables might assume fractional values very distant from their logical counterpart but still sufficient to satisfy problem constraints. As a consequence, the continuous relaxation value can significantly differ from the mixed integer optimum, thus disadvantaging branch-and-bound (and branch-and-cut) methods in their attempt to prune nodes of the search tree.

- Numerical stability: excessive values of indicator constraints constants (i.e. too large M and/or too small m) can influence the performance of solution methods and even compromise the validity of obtainable results. This occurs in particular when very small and very large numbers appear in the same constraint, possibly generating ratios below solvers standard precision. Sometimes the numerical tolerance of the tool can be adjusted, accepting longer solving times, but often this is not enough to prevent incorrect solutions.

These observations allow us to list some general remarks to be always considered when defining parameters of a big-M formulation:

1. The values selected should be realistic: in particular, it is a good practice to keep the M coefficients as low as possible and the m coefficients as high as possible.

2. A common mistake is to simply choose a unique positive number for each M and a unique negative value for each m in the model. However, they can be individually fixed depending on the constraint in which they are used, possibly obtaining a better linear relaxation.

3. Whenever a small parameter ε is introduced with the aim of avoiding unnecessary strict inequality constraints, it may generate problems related to its difference in order of magnitude from the M and m parameters absolute values. In general, it is worth to (i) select a reasonable value for ε, compatibly with the characteristics of the practical application under study, and (ii) investigate the possible interactions with model large constants, to avoid numerical issues involving tools limited precision.

This completes the description of the well-known aspects to take into account when modeling logical conditions through big-M formulations. From now on these concepts will be considered as given. Furthermore, in order to lighten the notation, we will conventionally use:

- $\varepsilon > 0$ to indicate a generic small parameter fixed according to representation requirements.
- $M > 0$ to identify a large enough generic parameter such that

  $$-M + \varepsilon \leq q(x) \leq M - \varepsilon$$

  for each possible inequality $q(x) \leq 0$ considered.

1.3.3 Indicator variables for the AND logical operation

The example in Section 1.3.1 permits to derive the following:

- If $g(x) \leq 0$ and $h(x) \leq 0$ are two general inequalities ($1000 - 10x \leq 0$ and $10x - 3000 \leq 0$ in the example), γ is a binary variable, and we need to express the *necessary condition*

  $$g(x) \leq 0 \textbf{ AND } h(x) \leq 0 \Rightarrow \gamma = 1,$$

  then two constraints can be used:

  $$-M\gamma - M\theta + \varepsilon \leq g(x), \tag{1.23}$$
  $$-M(1-\theta) + \varepsilon \leq h(x), \tag{1.24}$$

  where θ is an auxiliary binary variable.

- Differently, if $g(x) \leq 0$ and $h(x) \leq 0$ are two general inequalities ($2000 - 10x \leq 0$ and $10x - 3500 \leq 0$ in the example), γ is a binary variable, and we need to express the *sufficient condition*

  $$\gamma = 1 \Rightarrow g(x) \leq 0 \textbf{ AND } h(x) \leq 0,$$

  then we can use the following constraints:

  $$g(x) \leq M(1-\gamma), \tag{1.25}$$
  $$h(x) \leq M(1-\gamma). \tag{1.26}$$

Of course, if the double implication is required,

$$g(x) \leq 0 \textbf{ AND } h(x) \leq 0 \Leftrightarrow \gamma = 1,$$

it is possible to combine (1.23)-(1.24) and (1.25)-(1.26), and simply write:

$$-M\gamma - M\theta + \varepsilon \leq g(x) \leq M(1-\gamma),$$
$$-M(1-\theta) + \varepsilon \leq h(x) \leq M(1-\gamma).$$

Notice that, while inequalities (1.23)-(1.24) allow to only handle cases with two constraints (here $g(x) \leq 0$ and $h(x) \leq 0$), (1.25)-(1.26) clearly show the possibility of extension to consider more general situations. In particular, given k inequalities $g_1(x) \leq 0$, $g_2(x) \leq 0$, ..., $g_k(x) \leq 0$, it is possible to represent the condition

$$\gamma = 1 \Rightarrow g_1(x) \leq 0 \textbf{ AND } g_2(x) \leq 0 \textbf{ AND } ... \textbf{ AND } g_k(x) \leq 0$$

by using the following:

$$\begin{aligned} g_1(x) &\leq M(1-\gamma), \\ g_2(x) &\leq M(1-\gamma), \\ &\vdots \\ g_k(x) &\leq M(1-\gamma). \end{aligned}$$

1.3.4 Indicator variables for the OR logical operation

Proceeding in a similar way, we could extend previous methodology to consider how it applies to the **OR** operator. In some cases, indeed, it could be useful to link the satisfaction of one among two constraints to the activation of a binary variable. This can be generalized as follows:

- If $g(x) \leq 0$ and $h(x) \leq 0$ are two general inequalities, γ is a binary variable, and we want to express the *necessary condition*

  $$g(x) \leq 0 \textbf{ OR } h(x) \leq 0 \Rightarrow \gamma = 1,$$

  then we can use the following constraints:

  $$-M\gamma + \varepsilon \leq g(x), \tag{1.27}$$
  $$-M\gamma + \varepsilon \leq h(x). \tag{1.28}$$

- Differently, if $g(x) \leq 0$ and $h(x) \leq 0$ are two general inequalities, γ is a binary variable, and we need to express the *sufficient condition*

  $$\gamma = 1 \Rightarrow g(x) \leq 0 \textbf{ OR } h(x) \leq 0,$$

  then two constraints can be used:

  $$g(x) \leq M\theta + M(1-\gamma), \tag{1.29}$$
  $$h(x) \leq M(1-\theta), \tag{1.30}$$

  where θ is an auxiliary binary variable.

Clearly, to express the double implication

$$g(x) \leq 0 \textbf{ OR } h(x) \leq 0 \Leftrightarrow \gamma = 1,$$

(1.27)-(1.28) and (1.29)-(1.30) must be combined:

$$\begin{aligned} -M\gamma + \varepsilon \leq g(x) \leq M\theta + M(1-\gamma), \\ -M\gamma + \varepsilon \leq h(x) \leq M(1-\theta). \end{aligned}$$

The extension to represent the *necessary condition* including any number of inequalities, i.e.

$$g_1(x) \leq 0 \textbf{ OR } g_2(x) \leq 0 \textbf{ OR } ... \textbf{ OR } g_k(x) \leq 0 \Rightarrow \gamma = 1,$$

can be easily derived in this way:

$$\begin{aligned} -M\gamma + \varepsilon &\leq g_1(x), \\ -M\gamma + \varepsilon &\leq g_2(x), \\ &\vdots \\ -M\gamma + \varepsilon &\leq g_k(x). \end{aligned}$$

1.3.5 Indicator variables for the other logical operations

As well as seen for the **AND** and **OR** operators, it is of course possible to consider other types of indicator variables which are linked to the satisfaction of other logic relations (i.e. **XOR**, **NAND**, **NOR**, **XNOR**, **IF-THEN** and **BUT-NOT**). However, before entering into details, it is worth to mention a basic result deriving from boolean algebra.

First we notice that, given three binary variables α, β and γ, the following relations are always true:

$$(\gamma \Rightarrow \textbf{NOT } \alpha) \Longleftrightarrow \gamma \leq 1 - \alpha, \tag{1.31}$$

$$(\gamma \Leftarrow \textbf{NOT } \alpha) \Longleftrightarrow \gamma \geq 1 - \alpha, \tag{1.32}$$

$$(\gamma \Leftrightarrow \textbf{NOT } \alpha) \Longleftrightarrow \gamma = 1 - \alpha, \tag{1.33}$$

$$(\gamma \Rightarrow \alpha \textbf{ AND } \beta) \Longleftrightarrow \begin{cases} \gamma \leq \alpha, \\ \gamma \leq \beta, \end{cases} \tag{1.34}$$

$$(\gamma \Leftarrow \alpha \textbf{ AND } \beta) \Longleftrightarrow \gamma \geq \alpha + \beta - 1, \tag{1.35}$$

$$(\gamma \Leftrightarrow \alpha \textbf{ AND } \beta) \Longleftrightarrow \begin{cases} \gamma \leq \alpha, \\ \gamma \leq \beta, \\ \gamma \geq \alpha + \beta - 1, \end{cases} \tag{1.36}$$

$$(\gamma \Rightarrow \alpha \textbf{ OR } \beta) \Longleftrightarrow \gamma \leq \alpha + \beta, \tag{1.37}$$

$$(\gamma \Leftarrow \alpha \textbf{ OR } \beta) \Longleftrightarrow \begin{cases} \gamma \geq \alpha, \\ \gamma \geq \beta, \end{cases} \tag{1.38}$$

$$(\gamma \Leftrightarrow \alpha \textbf{ OR } \beta) \Longleftrightarrow \begin{cases} \gamma \geq \alpha, \\ \gamma \geq \beta, \\ \gamma \leq \alpha + \beta. \end{cases} \tag{1.39}$$

Secondly, we recall that sets $\{\textbf{AND}, \textbf{NOT}\}$ and $\{\textbf{OR}, \textbf{NOT}\}$ are *functionally complete*; this means that any boolean function can be constructed using only the elements in one of these sets.

By extension, also $\{\textbf{AND}, \textbf{OR}, \textbf{NOT}\}$ is functionally complete. It follows that whenever it is necessary to have a binary variable linked to a boolean function including k inequalities $g_1(x) \leq 0$, $g_2(x) \leq 0$, ..., $g_k(x) \leq 0$, it is possible to:

1. for each $i \in \{1, ..., k\}$, uniquely bind the satisfaction of constraint $g_i(x) \leq 0$ to the activation of a boolean variable γ_i (i.e. $g_i(x) \leq 0 \Leftrightarrow \gamma_i = 1$), using the relation:
$$-M\gamma_i + \varepsilon \leq g_i(x) \leq M(1 - \gamma_i);$$
2. express the boolean function as a combination of operators **AND**, **OR**, **NOT** and variables $\gamma_1, ..., \gamma_k$;
3. use (1.31)-(1.39) and suitable auxiliary binary variables to convert the **AND**, **OR**, **NOT** relations into constraints involving also the output binary variable.

For example, consider condition:

$$g(x) \leq 0 \textbf{ XOR } h(x) \leq 0 \Rightarrow \gamma = 1. \tag{1.40}$$

We can first define variables γ_g and γ_h, and include constraints:

$$-M\gamma_g + \varepsilon \leq g(x) \leq M(1 - \gamma_g), \tag{1.41}$$
$$-M\gamma_h + \varepsilon \leq h(x) \leq M(1 - \gamma_h). \tag{1.42}$$

These assure $\gamma_g = 1 \Leftrightarrow g(x) \leq 0$ and $\gamma_h = 1 \Leftrightarrow h(x) \leq 0$. We can write:

$$\begin{aligned} g(x) \leq 0 \textbf{ XOR } h(x) \leq 0 &\Leftrightarrow \gamma_g \textbf{ XOR } \gamma_h \\ &\Leftrightarrow (\gamma_g \textbf{ AND } (\textbf{NOT } \gamma_h)) \textbf{ OR } ((\textbf{NOT } \gamma_g) \textbf{ AND } \gamma_h) \\ &\Rightarrow (\gamma_g \textbf{ AND } \bar{\gamma}_h) \textbf{ OR } (\bar{\gamma}_g \textbf{ AND } \gamma_h) \\ &\Rightarrow \gamma_1 \textbf{ OR } \gamma_2 \\ &\Rightarrow \gamma, \end{aligned}$$

where $\bar{\gamma}_g$, $\bar{\gamma}_h$, γ_1 and γ_2 are binary variables to be introduced together with constraints:

$$\bar{\gamma}_g \geq 1 - \gamma_g, \tag{1.43}$$
$$\bar{\gamma}_h \geq 1 - \gamma_h, \tag{1.44}$$
$$\gamma_1 \geq \gamma_g + \bar{\gamma}_h - 1, \tag{1.45}$$
$$\gamma_2 \geq \bar{\gamma}_g + \gamma_h - 1, \tag{1.46}$$
$$\gamma \geq \gamma_1, \tag{1.47}$$
$$\gamma \geq \gamma_2. \tag{1.48}$$

Indeed, (1.43)-(1.44) guarantee (see (1.32)):

$$\textbf{NOT } \gamma_g \Rightarrow \bar{\gamma}_g,$$
$$\textbf{NOT } \gamma_h \Rightarrow \bar{\gamma}_h.$$

Inequalities (1.45) and (1.46) ensure (see (1.35)):

$$\gamma_g \textbf{ AND } \bar{\gamma}_h \Rightarrow \gamma_1,$$
$$\bar{\gamma}_g \textbf{ AND } \gamma_h \Rightarrow \bar{\gamma}_2.$$

Finally, (1.47)-(1.48) imply (see (1.38)):

$$\gamma_1 \textbf{ OR } \gamma_2 \Rightarrow \gamma.$$

Thus, formulation (1.41)-(1.48) allows to express condition (1.40). If instead of considering this, we needed to represent the opposite implication, i.e.

$$\gamma = 1 \Rightarrow g(x) \leq 0 \textbf{ XOR } h(x) \leq 0,$$

then we could use relations (1.41)-(1.42) along with the following:

$$\bar{\gamma}_g \leq 1 - \gamma_g, \tag{1.49}$$
$$\bar{\gamma}_h \leq 1 - \gamma_h, \tag{1.50}$$
$$\gamma_1 \leq \gamma_g, \tag{1.51}$$
$$\gamma_1 \leq \bar{\gamma}_h, \tag{1.52}$$
$$\gamma_2 \leq \bar{\gamma}_g, \tag{1.53}$$
$$\gamma_2 \leq \gamma_h, \tag{1.54}$$
$$\gamma \leq \gamma_1 + \gamma_2. \tag{1.55}$$

Here, (1.49) and (1.50) ensure (see (1.31)):

$$\bar{\gamma}_g \Rightarrow \textbf{NOT } \gamma_g,$$
$$\bar{\gamma}_h \Rightarrow \textbf{NOT } \gamma_h.$$

The four inequalities (1.51)-(1.54) imply (see (1.34)):

$$\gamma_1 \Rightarrow \gamma_g \textbf{ AND } \bar{\gamma}_h,$$
$$\bar{\gamma}_2 \Rightarrow \bar{\gamma}_g \textbf{ AND } \gamma_h,$$

while (1.55) guarantees (see (1.37)):

$$\gamma \Rightarrow \gamma_1 \textbf{ OR } \gamma_2.$$

Of course, if the **XOR** function and the variable γ had to be linked by a double implication, then the problem could be formulated using (1.41)-(1.55) all together.

This example should suggest that reformulating boolean functions through simpler operators, which belong to functionally complete sets, is always a possible strategy to express logical implications. In general, there are many different ways of modeling the same condition. To give an instance, consider another formulation for the **XOR** using the **AND**, **OR** and **NOT** as components:

$$g(x) \leq 0 \textbf{ XOR } h(x) \leq 0 \Leftrightarrow \gamma_g \textbf{ XOR } \gamma_h$$
$$\Leftrightarrow (\gamma_g \textbf{ OR } \gamma_h) \textbf{ AND } (\textbf{NOT } (\gamma_g \textbf{ AND } \gamma_h)).$$

The reader can verify that applying this transformation as described above one can obtain a different model requiring one binary variable less than previous approach.

However, it should be evident that also this second formulation does not necessarily ensure an efficient solution, both in terms of number of variables and constraints. This is due to the fact that intermediary binaries and expressions must be introduced to represent components **AND**, **OR** and **NOT** implications. To avoid this, let us now consider a different method to model the **XOR** function. In particular:

- If $g(x) \leq 0$ and $h(x) \leq 0$ are two general inequalities, γ and θ are two binary variables, and we want to express the *necessary condition*

$$g(x) \leq 0 \ \textbf{XOR} \ h(x) \leq 0 \Rightarrow \gamma = 1,$$

then the following constraints can be introduced:

$$h(x) \leq M\theta + M\gamma, \tag{1.56}$$
$$-M\gamma - M(1-\theta) + \varepsilon \leq h(x), \tag{1.57}$$
$$g(x) \leq M\theta, \tag{1.58}$$
$$-M(1-\theta) + \varepsilon \leq g(x). \tag{1.59}$$

- Otherwise, if $g(x) \leq 0$ and $h(x) \leq 0$ are two general inequalities, γ and θ are two binary variables, and we need to express the *sufficient condition*

$$\gamma = 1 \Rightarrow g(x) \leq 0 \ \textbf{XOR} \ h(x) \leq 0,$$

then constraints (1.58)-(1.59) may be used together with the following:

$$h(x) \leq M(1-\theta) + M(1-\gamma), \tag{1.60}$$
$$-M(1-\gamma) - M\theta + \varepsilon \leq h(x). \tag{1.61}$$

In order to compare this methodology to the other previously shown, consider the different possibilities for modeling the double implication:

$$g(x) \leq 0 \ \textbf{XOR} \ h(x) \leq 0 \Leftrightarrow \gamma = 1.$$

The latter approach would involve to introduce two binary variables, γ and θ, and six constraints, i.e. (1.56)-(1.61). The previous two alternatives would at least require twice the number of variables and constraints. However, the choice of the best modeling strategy is often a complex task related to different aspects of a MILP formulation, such as the properties of the continuous relaxation and other computational advantages. This idea will be clarified in Chapter 3, where a practical comparison between different models will be addressed.

Following similar arguments, we may describe other sets of constraints that allow to link **NAND**, **NOR**, **XNOR**, **IF-THEN** and **BUT-NOT** relations to binary variables. These are listed in Table 1.10 together with those already derived for the **AND**, **OR** and **XOR**, and are distinguished by the type of condition to represent (necessary or sufficient).

	Condition	Required constraints
AND	$g(x) \leq 0$ **AND** $h(x) \leq 0 \Rightarrow \gamma = 1$	$-M\gamma - M\theta + \varepsilon \leq g(x)$ $-M(1-\theta) + \varepsilon \leq h(x)$
	$g(x) \leq 0$ **AND** $h(x) \leq 0 \Leftarrow \gamma = 1$	$g(x) \leq M(1-\gamma)$ $h(x) \leq M(1-\gamma)$

OR	$g(x) \leq 0$ **OR** $h(x) \leq 0 \Rightarrow \gamma = 1$	$-M\gamma + \varepsilon \leq g(x)$ $-M\gamma + \varepsilon \leq h(x)$
	$g(x) \leq 0$ **OR** $h(x) \leq 0 \Leftarrow \gamma = 1$	$g(x) \leq M\theta + M(1-\gamma)$ $h(x) \leq M(1-\theta)$
XOR	$g(x) \leq 0$ **XOR** $h(x) \leq 0 \Rightarrow \gamma = 1$	$h(x) \leq M\theta + M\gamma$ $-M\gamma - M(1-\theta) + \varepsilon \leq h(x)$ $g(x) \leq M\theta$ $-M(1-\theta) + \varepsilon \leq g(x)$
	$g(x) \leq 0$ **XOR** $h(x) \leq 0 \Leftarrow \gamma = 1$	$h(x) \leq M(1-\theta) + M(1-\gamma)$ $-M(1-\gamma) - M\theta + \varepsilon \leq h(x)$ $g(x) \leq M\theta$ $-M(1-\theta) + \varepsilon \leq g(x)$
NAND	$g(x) \leq 0$ **NAND** $h(x) \leq 0 \Rightarrow \gamma = 1$	$g(x) \leq M\gamma$ $h(x) \leq M\gamma$
	$g(x) \leq 0$ **NAND** $h(x) \leq 0 \Leftarrow \gamma = 1$	$-M(1-\gamma) - M\theta + \varepsilon \leq g(x)$ $-M(1-\theta) + \varepsilon \leq h(x)$
NOR	$g(x) \leq 0$ **NOR** $h(x) \leq 0 \Rightarrow \gamma = 1$	$g(x) \leq M\theta + M\gamma$ $h(x) \leq M(1-\theta)$
	$g(x) \leq 0$ **NOR** $h(x) \leq 0 \Leftarrow \gamma = 1$	$-M(1-\gamma) + \varepsilon \leq g(x)$ $-M(1-\gamma) + \varepsilon \leq h(x)$
XNOR	$g(x) \leq 0$ **XNOR** $h(x) \leq 0 \Rightarrow \gamma = 1$	$h(x) \leq M(1-\theta) + M\gamma$ $-M\gamma - M\theta + \varepsilon \leq h(x)$ $g(x) \leq M\theta$ $-M(1-\theta) + \varepsilon \leq g(x)$
	$g(x) \leq 0$ **XNOR** $h(x) \leq 0 \Leftarrow \gamma = 1$	$h(x) \leq M\theta + M(1-\gamma)$ $-M(1-\gamma) - M(1-\theta) + \varepsilon \leq h(x)$ $g(x) \leq M\theta$ $-M(1-\theta) + \varepsilon \leq g(x)$
IF-THEN	**IF** $g(x) \leq 0$ **THEN** $h(x) \leq 0 \Rightarrow \gamma = 1$	$g(x) \leq M\gamma$ $-M\gamma + \varepsilon \leq h(x)$
	IF $g(x) \leq 0$ **THEN** $h(x) \leq 0 \Leftarrow \gamma = 1$	$h(x) \leq M(1-\theta)$ $-M(1-\gamma) - M\theta + \varepsilon \leq g(x)$

BUT-NOT	$g(x) \leq 0$ **BUT-NOT** $h(x) \leq 0 \Rightarrow \gamma = 1$	$h(x) \leq M(1-\theta)$ $-M\gamma - M\theta + \varepsilon \leq g(x)$
	$g(x) \leq 0$ **BUT-NOT** $h(x) \leq 0 \Leftarrow \gamma = 1$	$g(x) \leq M(1-\gamma)$ $-M(1-\gamma) + \varepsilon \leq h(x)$

Table 1.10. Logic conditions and corresponding constraints set.

1.4 Indicator variables, logical relations and scheduling

In the area of scheduling, the use of indicator variables is really widespread mainly due to the nature of restrictions that generally influence application contexts, such as manufacturing and logistic processes.

For this reason, techniques like those described in previous section often offer interesting modeling possibilities, allowing to introduce particular sets of inequalities and to represent in this way logical connections between different constraints or between feasible solutions and the objective function.

To give evidence of this fact and to provide some representative cases, we propose here several examples.

Common notations will be used from now on. In particular, given a set of m activities, let $A = \{1, ..., j, ..., m\}$ be the set of indexes for these activities. For each $j \in A$ we define:

$\boldsymbol{t_j}$: variable starting time of activity j, with $t_j \geq 0$;

$\boldsymbol{\tau_j}$: parametric positive quantity indicating processing time of activity j;

$\boldsymbol{\sigma_j}$: parametric positive quantity indicating setup time of activity j;

$\boldsymbol{r_j}$: parametric release date/time for activity j;

$\boldsymbol{d_j}$: parametric due date/time for activity j.

We will consider different possible settings and corresponding constraints. To generalize, the following types of restrictions will be often used:

Sequence / Precedence constraints: if two activities i and j are such that, for some reason, i has to be completed before j begins, we impose:

$$t_i + \sigma_i + \tau_i \leq t_j.$$

Of course, if no setup is defined, it becomes:

$$t_i + \tau_i \leq t_j,$$

where the left-hand side of the constraint clearly represents activity i completion time.

Disjunctive / Non-overlapping constraints: scheduling problems typically require to model the fact that two operations can not be simultaneously executed. This can happen for several reasons, for instance because they use the same machine, the same unique resource or for other logical causes which exclude parallelization. In any case, to express the impossibility of overlapping, a common trick is used. Namely, given two activities i and j, an alternative is identified: either j starts after the end of i, so i precedes j, or the contrary is true, thus i follows j. This is probably the most popular example of **OR** (practically equivalent to a **XOR**, since alternatives are exclusive by nature) linking a couple of inequalities, i.e.:

$$t_i + \tau_i - t_j \leq 0 \textbf{ OR } t_j + \tau_j - t_i \leq 0.$$

Recalling also the arguments from previous sections, disjunctive (or non-overlapping) constraints are given by:

$$\begin{aligned} t_i + \tau_i - t_j &\leq M(1-\theta), \\ t_j + \tau_j - t_i &\leq M\theta, \end{aligned}$$

where θ is the binary variable specifying the sequence between the activities; if $\theta = 1$, then i precedes j and $t_j \geq t_i + \tau_i$, otherwise θ is equal to 0 and condition $t_i \geq t_j + \tau_j$ is implied (i is processed after j is finished). Similarly, when also setups are involved:

$$\begin{aligned} t_i + \sigma_i + \tau_i - t_j &\leq M(1-\theta), \\ t_j + \sigma_j + \tau_j - t_i &\leq M\theta. \end{aligned}$$

Release date constraints: when an activity j has a minimum starting time (i.e. release date) $r_j \geq 0$, we have:

$$t_j \geq r_j.$$

Otherwise $t_j \geq 0$ holds.

Due date / Deadline constraints: if activity j has to be completed before a maximum time (i.e. due date) $d_j \geq \tau_j$, the following must be satisfied:

$$t_j + \tau_j \leq d_j.$$

If a setup operation is executed between activity start and processing:

$$t_j + \sigma_j + \tau_j + \sigma_j \leq d_j.$$

We implemented all the models in AMPL and solved them using Gurobi optimizer under default configurations.

1.4.1 Example 1: Single machine scheduling for fresh products jobs

For the first example, let us take into consideration a single machine scheduling problem where different jobs must be processed to produce different types of parts. We suppose the set of indexes for jobs is J, the set of parts is P, and J_p indicates the subset of J allocated to $p \in P$ where:

$$\bigcup_{p \in P} J_p = J \text{ and } \bigcap_{p \in P} J_p = \emptyset$$

Each job $j \in J$ requires a processing time τ_j and a setup. In particular, the setup time depends on the sequence of parts, such that σ_{pq} indicates the amount of time required to setup the machine from producing part $p \in P$ to $q \in P$, and $\sigma_{pp} = 0$ for each $p \in P$.

We imagine that jobs are prepared for a single customer, who imposes a due date d_j^{max} for each $j \in J$, that is the time in which he agreed to pick up goods. Finished parts are fresh products: for this reason, they can not be completed too early with respect to d_j^{max}, otherwise they risk to expire. We thus assume that a due window $[d_j^{min}, d_j^{max}]$ is considered for the scheduling of each job completion.

We examine three different scenarios:

(a) Supposing d_j^{min} and d_j^{max} constitute soft restrictions, we minimize the overall weighted distance of jobs from their due window, i.e.

$$\sum_{j \in J} w_j \left((t_j + \tau_j - d_j^{max})^+ + (d_j^{min} - t_j - \tau_j)^+ \right),$$

where w_j represents the weight for job j.

(b) We imagine that early jobs (i.e. jobs completed before d_j^{min}) and tardy jobs (i.e. jobs finished after d_j^{max}) are stored in a refrigerated area and shipped in a successive planning period without risk of expiration. However they must be completed with a maximum tardiness given by a parameter $\bar{l}$. The objective is to maximize the overall number of jobs able to meet their due interval.

(c) This case is similar to scenario (b), with an additional restriction: we assume that, due to technological reasons, no more than $\bar{a}$ jobs for part $\bar{p}$ can be started during interval (a^{min}, a^{max}). The objective is again the maximization of on time jobs subject to maximum tardiness $\bar{l}$.

For this example, we will consider data of Table 1.11, where $|J| = 12$, $|P| = 3$, and $|J_p| = 4$ for each $p \in P$. Jobs processing times only depend on the part type, while weights are all equivalent. Setups are given in Table 1.12. We assume $\bar{l} = 120$, $\bar{p} = \text{Z}$, $\bar{a} = 2$, $a^{min} = 50$, and $a^{max} = 260$.

The first step to model this problem is to impose disjunctive constraints that avoid the execution of more than one job on the machine at the same time. Therefore, we define a binary variable ζ_{jk} with $j \in J, k \in J$ and $j < k$, such that $\zeta_{jk} = 1$ if job

Job	Part	τ_j	w_j	d_j^{min}	d_j^{max}
1	X	20	1	60	90
2	X	20	1	190	220
3	X	20	1	120	150
4	X	20	1	200	230
5	Y	16	1	30	50
6	Y	16	1	150	170
7	Y	16	1	190	210
8	Y	16	1	50	70
9	Z	24	1	70	90
10	Z	24	1	20	40
11	Z	24	1	120	140
12	Z	24	1	170	190

Table 1.11. Data for a particular instance with 12 jobs and 3 part types.

from \ to	X	Y	Z
X	0	5	7
Y	6	0	4
Z	5	6	0

Table 1.12. Change-dependent setup times for a particular instance with 3 different part types.

j precedes job k, and $\zeta_{jk} = 0$ otherwise. Then, for each p and $q \in P$ we write:

$$\begin{aligned} t_j + \tau_j + \sigma_{pq} - t_k &\leq M(1 - \zeta_{jk}) && j \in J_p,\ k \in J_q,\ j < k,\ p \neq q \\ t_k + \tau_k + \sigma_{qp} - t_j &\leq M\zeta_{jk} && j \in J_p,\ k \in J_q,\ j < k,\ p \neq q \\ t_j + \tau_j - t_k &\leq M(1 - \zeta_{jk}) && j \in J_p,\ k \in J_p,\ j < k \\ t_k + \tau_k - t_j &\leq M\zeta_{jk} && j \in J_p,\ k \in J_p,\ j < k \end{aligned}$$

The overall weighted distance of jobs from the due window corresponds to the weighted sum, for each $j \in J$, of job earliness, defined as $(d_j^{min} - t_j - \tau_j)^+$, plus job tardiness, given by $(t_j + \tau_j - d_j^{max})^+$. Introducing variables $e \in \mathbb{R}_+^{|J|}$ and $l \in \mathbb{R}_+^{|J|}$, we can formulate problem (a) by adding the following:

$$\begin{aligned} \min \sum_{j \in J} w_j(e_j + l_j) & \\ e_j \geq d_j^{min} - t_j - \tau_j & \qquad j \in J \\ l_j \geq t_j + \tau_j - d_j^{max} & \qquad j \in J \\ e_j \geq 0 & \qquad j \in J \\ l_j \geq 0 & \qquad j \in J \end{aligned}$$

If the objective is the maximization of the number of jobs completed during their due interval, as in case (b), a different approach can be used. A new variable γ_j may

be defined for each job $j \in J$, such that

$$\gamma_j = 1 \Rightarrow d_j^{min} \leq t_j + \tau_j \leq d_j^{max},$$

or equivalently:

$$\gamma_j = 1 \Rightarrow d_j^{min} - t_j - \tau_j \leq 0 \textbf{ AND } t_j + \tau_j - d_j^{max} \leq 0.$$

These conditions are expressed by constraints:

$$\begin{aligned} d^{min} - t_j - \tau_j &\leq M(1 - \gamma_j) \qquad j \in J \\ t_j + \tau_j - d^{max} &\leq M(1 - \gamma_j) \qquad j \in J \end{aligned}$$

The objective becomes:

$$\min \sum_{j \in J} \gamma_j.$$

Maximum tardiness limitation is enforced by using variable l_j (as defined before) and writing inequalities:

$$\begin{aligned} l_j &\geq t_j + \tau_j - d_j^{max} \qquad j \in J \\ l_j &\geq 0 \qquad j \in J \\ l_j &\leq \bar{l} \qquad j \in J \end{aligned}$$

Formulation of scenario (c) requires the additional restriction on some jobs starting time. For this reason we imagine to have a binary variable δ_j, $\forall\ j \in J_{\bar{p}}$, such that:

$$a^{min} \leq t_j \leq a^{max} \Rightarrow \delta_j = 1.$$

Then, to have no more than $\bar{a}$ jobs for part $\bar{p}$ starting between a^{min} and a^{max}, we write:

$$\sum_{j \in J_{\bar{p}}} \delta_j \leq \bar{a}.$$

Condition on δ_j, for each $j \in J_{\bar{p}}$, is equivalent to:

$$a^{min} - t_j \leq 0 \textbf{ AND } t_j - a^{max} \leq 0 \Rightarrow \delta_j = 1.$$

Thus, we can simply add the following constraints to the formulation:

$$\begin{aligned} -M\theta_j - M\delta_j + \varepsilon &\leq a^{min} - t_j \qquad j \in J \\ -M(1 - \theta_j) + \varepsilon &\leq t_j - a^{max} \qquad j \in J \end{aligned}$$

Notice here the difference between roles of variables γ_j, $\forall\ j \in J$, and δ_j, $\forall\ j \in J_{\bar{p}}$. The first assures that the objective function value can only be increased by satisfying jobs due windows, the second guarantees that the limits on the number of jobs starting in a given interval is respected. While γ_j is encouraged by the optimization process to become 1, δ_j can possibly be forced to 0 due to the effect of a constraint. So it is not surprising that conditions

$$\gamma_j = 1 \Rightarrow d_j^{min} \leq t_j + \tau_j \leq d_j^{max}$$

and

$$\delta_j = 1 \Leftarrow a^{min} \leq t_j \leq a^{max}$$

have opposite direction, the first expressing that the binary variable is *sufficient* to meet the time interval, the second being *necessary* for it, indeed:

$$\delta_j = 0 \Rightarrow t_j < a^{min} \textbf{ OR } t_j > a^{max}.$$

Figures 1.1, 1.2, and 1.3 show optimal solutions for problems (a), (b), and (c), using data of Table 1.11 and 1.12. Corresponding variables are reported in Table 1.13, where jobs completion times are indicated with symbol c_j.

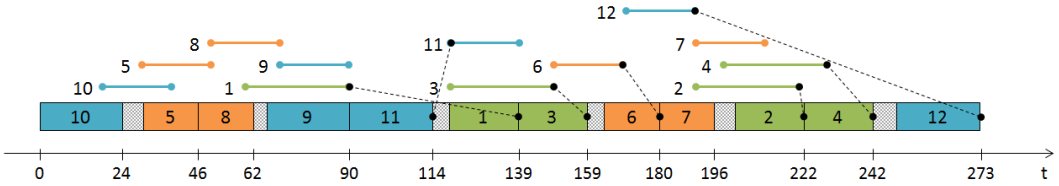

Figure 1.1. Graphical representation of an optimal solution to problem (a). Lines above the schedule illustrate jobs due intervals. Dashed connectors highlight earliness/tardiness issues.

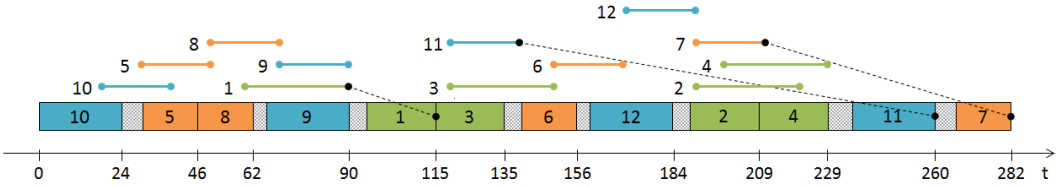

Figure 1.2. Graphical representation of an optimal solution to problem (b). Lines above the schedule illustrate jobs due intervals. Dashed connectors highlight earliness/tardiness issues.

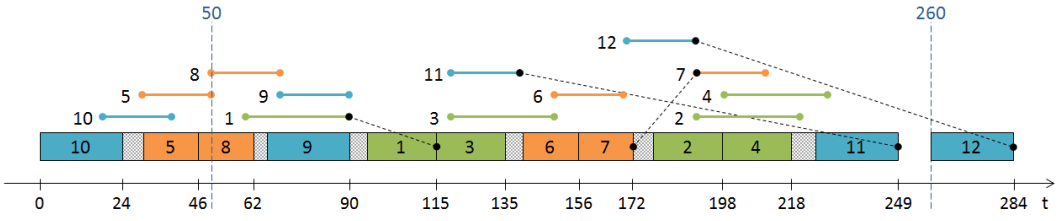

Figure 1.3. Graphical representation of an optimal solution to problem (c). Lines above the schedule illustrate jobs due intervals. Dashed connectors highlight earliness/tardiness issues.

In scenario (a) we have:

$$\sum_{j \in J} w_j \left(l_j^{(a)} + e_j^{(a)} \right) = 171,$$

j	$t_j^{(a)}$	$c_j^{(a)}$	$e_j^{(a)}$	$l_j^{(a)}$	$t_j^{(b)}$	$c_j^{(b)}$	$\gamma_j^{(b)}$	$t_j^{(c)}$	$c_j^{(c)}$	$\delta_j^{(c)}$	$\gamma_j^{(c)}$
1	119	139	0	49	95	115	0	95	115	-	0
2	202	222	0	2	189	209	1	178	198	-	1
3	139	159	0	9	115	135	1	115	135	-	1
4	222	242	0	12	209	229	1	198	218	-	1
5	30	46	0	0	30	46	1	30	46	-	1
6	164	180	0	10	140	156	1	140	156	-	1
7	180	196	0	0	266	282	0	156	172	-	0
8	46	62	0	0	46	62	1	46	62	-	1
9	66	90	0	0	66	90	1	66	90	1	1
10	0	24	0	0	0	24	1	0	24	0	1
11	90	114	6	0	236	260	0	225	249	1	0
12	249	273	0	83	160	184	1	260	284	0	0

Table 1.13. Optimal solutions for problems (a), (b) and (c).

with 6 tardy jobs, 1 early job and only 5 on time. In the second case (b) the number of jobs able to meet their due interval increases, indeed:

$$\sum_{j \in J} \gamma_j^{(b)} = 9.$$

However, the overall distance of completion times from due intervals gets worse, while satisfying the maximum tardiness admitted for each $j \in J$. Third scenario limits the number of jobs for part C that can be started in the interval (a^{min}, a^{max}) to 2. For this reason, last job is delayed until 260, becoming tardy. Job 7 is instead anticipated before its due window, completing too early. The optimal value now is:

$$\sum_{j \in J} \gamma_j^{(c)} = 8.$$

1.4.2 Example 2: Scheduling of timetable in a manufacturing cell

In this example we consider a job shop manufacturing system consisting of a finite set of stations S and a finite set of jobs $J = \{1, 2, ..., n\}$. Every job $i \in J$ is constituted by different operations O_{ih} to be performed on preassigned stations ($h \in S$), following a predefined technological sequence $E_i = \{(h, h') : h \in S, h' \in S, O_{ih} \text{ precedes } O_{ih'}\}$. Processing times for each job i and station h are given by parameters τ_{ih}.

We assume that the objective is to schedule all daily activities, thus generating a timetable which concentrates all operations within the eight-hour working day (from 8:00 to 16:00). We investigate three possible scenarios with different minimization goals:

(a) We assume that working hours are split into two shifts, the first from 8:00 to 12:00 and the second from 12:00 to 16:00. To avoid the transfer of tasks between operators of different shifts, it is important to minimize the number of operations which are in progress at shift change, i.e. at 12:00.

(b) Here we consider the same division of working hours as previous scenario but we imagine a different objective: instead of minimizing the number of activities spread over different shifts, we look at avoiding jobs splitting, thus trying to concentrate all operations for a job in only one of the two 4-hours slots.

(c) In this scenario we assume a different situation in which shifts are not fixed. We suppose that the manufacturer can decide in advance to call operators for two types of employment:

1 Full-time: continued from 8:00 to 16:00, paid a dollars.

2 Part-time: from 9:00 to 13:00, paid b dollars.

The type of employment to choose depends on the working time established for the station to which operators are assigned. Therefore, if a station h must be in action from 9:00 to 13:00 (or less) operators working on that station, which are r_h, are paid for a part-time shift, otherwise they are engaged for a full-time and remain available from 8:00 to 16:00.

Before entering into details of these three scenarios, we recall that typical job shop constraints can be formulated as follows:

$$\begin{aligned} t_{ih} + \tau_{ih} \leq t_{i'h} + M(1 - y_{ii'h}) \qquad & i, i' \in J,\ i < i',\ h \in S \\ t_{i'h} + \tau_{i'h} \leq t_{ih} + M y_{ii'h} \qquad & i, i' \in J,\ i < i',\ h \in S \\ t_{ih} + \tau_{ih} \leq t_{ih'} \qquad & i \in J,\ (h, h') \in E_i \end{aligned}$$

Where t_{ih} indicates the variable starting time of job i on station h and $y_{ii'h}$ is a binary variable that is equal to 1 if job i precedes job i' on station h and is 0 otherwise (job i' precedes job i on station h).

Furthermore we introduce release date and due date constraints:

$$\begin{aligned} t_{ih_i^f} \geq 8 \qquad & i \in J \\ t_{ih_i^l} + \tau_{ih_i^l} \leq 16 \qquad & i \in J \end{aligned}$$

where h_i^f and h_i^l respectively indicate the first and the last stations in job i predefined processing sequence.

This formulation can be extended to represent scenarios (a), (b) and (c). In scenario (a) we want generate a schedule which minimizes the number of operations that are in progress at 12:00. For this purpose, we define a variable $\gamma_{ih} \in \{0, 1\}$ for all $i \in J$ and $h \in S$, and express the objective as:

$$\min \sum_{i \in J} \sum_{h \in S} \gamma_{ih}.$$

Each operation starting time is linked to variable γ_{ih} through the condition:

$$t_{ih} < 12 \ \textbf{AND}\ t_{ih} + \tau_{ih} > 12 \Rightarrow \gamma_{ih} = 1,$$

i.e. γ_{ih} is necessarily activated if operation O_{ih} starts before and ends after time instant 12. The same condition could also be written as:

$$12 - t_{ih} \leq 0\ \textbf{NOR}\ t_{ih} + \tau_{ih} - 12 \leq 0 \Rightarrow \gamma_{ih} = 1.$$

Recalling results derived in Section 1.3, this can be expressed by the following constraints:

$$\begin{aligned} 12 - t_{ih} &\leq M\theta_{ih} + M\gamma_{ih} \qquad i \in J,\ h \in S \\ t_{ih} + \tau_{ih} - 12 &\leq M(1 - \theta_{ih}) \qquad i \in J,\ h \in S \end{aligned}$$

Where θ_{ih} is an auxiliary binary variable defined for each $i \in J$ and $h \in S$.

An optimal solution for a simple example with 6 jobs and 3 stations assuming data in Table 1.14 is given in Figure 1.4. The objective value is 0 since no operations are in progress when shift changes at 12:00.

job	operation 1		operation 2		operation 3
1	A (0.4)	$\longrightarrow$	B (2.5)	$\longrightarrow$	C (1.8)
2	B (1.2)				
3	C (2.4)	$\longrightarrow$	B (0.5)		
4	A (2.2)	$\longrightarrow$	B (1.8)		
5	C (1.6)	$\longrightarrow$	B (1.5)		
6	C (2.2)				

Table 1.14. Processing times and sequences for a particular instance with 6 jobs and 3 stations.

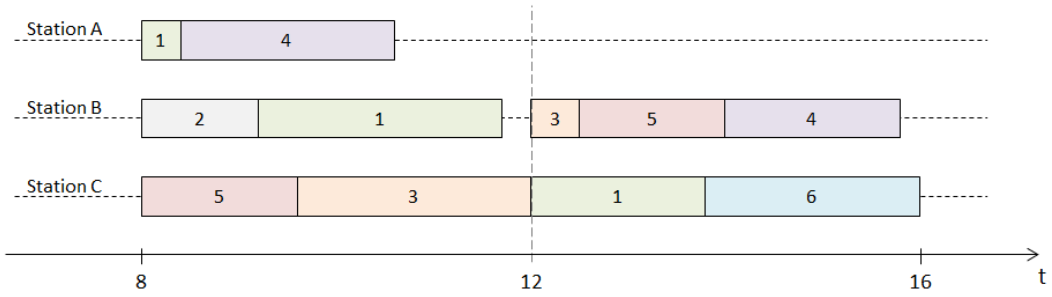

Figure 1.4. Optimal solution for scenario (a), assuming job shop problem data of Table 1.14.

Scenario (b) is similar to the previous one. However, in order to avoid dispersion of complete jobs over two shifts, only one binary variable for each job $i \in J$ is required. We call this variable δ_i and express condition

$$12 - t_{ih_i^f} \leq 0 \ \textbf{NOR}\ t_{ih_i^l} + \tau_{ih_i^l} - 12 \leq 0 \Rightarrow \delta_i = 1$$

by using constraints:

$$\begin{aligned} 12 - t_{ih_i^f} &\leq M\beta_i + M\delta_i \qquad i \in J \\ t_{ih_i^l} + \tau_{ih_i^l} - 12 &\leq M(1 - \beta_i) \qquad i \in J \end{aligned}$$

where β_i is an additional binary variable, for each $i \in J$.

The objective now becomes:

$$\min \sum_{i \in J} \delta_i.$$

Figure 1.5 shows an optimal solution for this problem, using the same data as before (see Table 1.14). The objective value is 2 due to jobs 1 and 3 which are split over different shifts. However, notice the improvement with respect to situation in Figure 1.4, where four jobs began before and finished after 12:00. By postponing the start of job 4 and anticipating the second operation of job 5, a better solution is found.

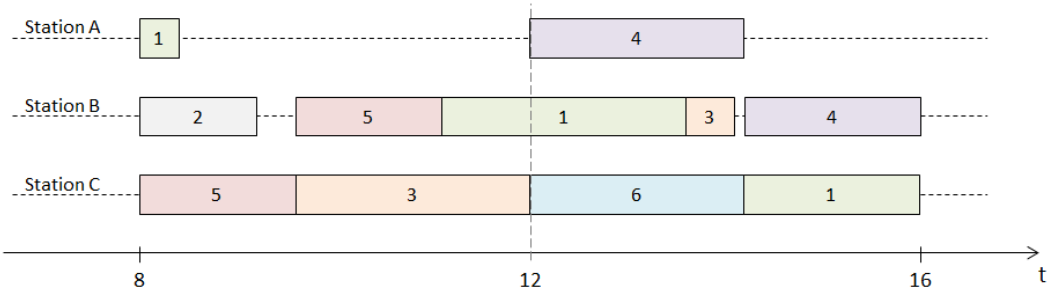

Figure 1.5. Optimal solution for scenario (b), assuming job shop problem data of Table 1.14.

Let us now consider scenario (c). In this case we want to investigate if it is possible to reduce the working time of particular stations – still respecting the due dates –, in order to reduce the employment costs. For this purpose we introduce a binary variable ζ_h that becomes equal to 1 if a full-time is selected for station $h \in S$. Then, we write the objective as follows:

$$\min \sum_{h \in S} r_h[a\zeta_h + b(1 - \zeta_h)].$$

To express the relation between variable ζ_h and operations times, we should consider if the first activity scheduled on station h starts before 9:00 or the last ends after 13:00 (in these cases we would enforce $\zeta_h = 1$ because a part-time shift is not feasible). However, we don't know a priori the sequence of operations on station h. More simply, we introduce a condition which depends on all the operations executed on station h:

$$\left.\begin{array}{l} t_{1h} < 9 \textbf{ OR } t_{2h} < 9 \textbf{ OR } \ldots \textbf{ OR } t_{nh} < 9 \textbf{ OR} \\ t_{1h} + \tau_{1h} > 13 \textbf{ OR } t_{2h} + \tau_{2h} > 13 \textbf{ OR } \ldots \textbf{ OR } t_{nh} + \tau_{nh} > 13 \end{array}\right\} \Rightarrow \zeta_h = 1.$$

In other terms, for each job $i \in J$ and station $h \in S$ we have:

$$9 - t_{ih} \leq 0 \textbf{ NAND } t_{ih} + \tau_{ih} - 13 \leq 0 \Rightarrow \zeta_h = 1.$$

These conditions can be expressed by the following constraints:

$$9 - t_{ih} \leq M\zeta_h \qquad i \in J,\ h \in S$$
$$t_{ih} + \tau_{ih} - 13 \leq M\zeta_h \qquad i \in J,\ h \in S$$

Considering once again data from Table 1.14, we provide in Figure 1.6 an example of solution having assumed:

$$a = 70,\ b = 40,\ r_A = r_B = r_C = 1.$$

In this case, of course, the result for stations B and C is trivial, since they have a work load for more than 4 hours. However, the effect of optimization is visible on station A where job 1 and job 4 are scheduled within the part-time hours 9:00-13:00. This brings to an overall cost of 180 dollars.

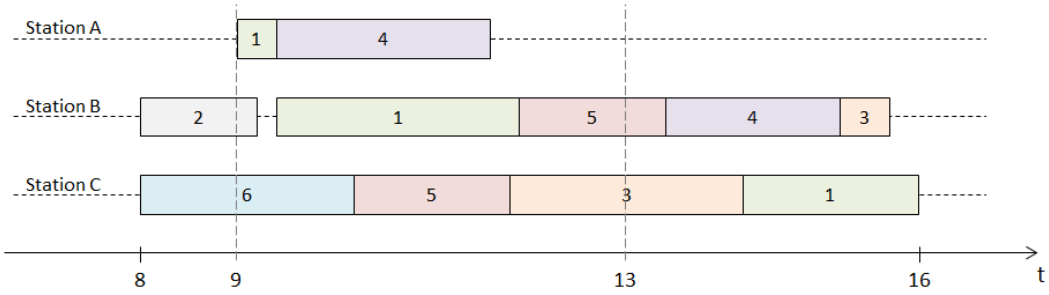

Figure 1.6. Optimal solution for scenario (c), assuming job shop problem data of Table 1.14.

1.4.3 Example 3: Costs minimization for two machines subject to electricity load peaks

We now consider a simple production center constituted by two identical parallel machines. Every time a job $\jmath$ flows into this center, it reaches one of the two stations and queues for being processed. When it comes its turn, a setup of $\sigma_\jmath$ minutes is executed by an operator and then the processing starts, taking $\tau_\jmath$ minutes in total, where:

- the first 10% of duration $\tau_\jmath$ is executed at a high level of energy,
- the remaining 90% of work is instead low demanding.

We suppose the enterprise pays a small penalty of p dollars every time it exceeds a given power threshold t, which happens only if both machineries execute high demanding operations at the same time.

Furthermore we assume the company may either decide to employ only one operator to execute setups on both stations or to engage another worker to parallelize processes, having one human assigned to each machine.

The objective is to minimize costs, taking into account that each operator must be paid s dollars and that jobs have urgent due dates $d_\jmath$ for $\jmath = 1, ..., m$.

To model the scheduling of jobs over two parallel machines we use the following:

$$\begin{aligned} t_{j^1} + \sigma_{j^1} + \tau_{j^1} - t_{j^2} &\leq M(1 - \alpha_{j^1 j^2}) && j^1, j^2 \in \{1, ..., m\},\ j^2 > j^1 \\ t_{j^2} + \sigma_{j^2} + \tau_{j^2} - t_{j^1} &\leq M\alpha_{j^1 j^2} + M\beta_{j^1 j^2} && j^1, j^2 \in \{1, ..., m\},\ j^2 > j^1 \\ \beta_{j^1 j^2} + \beta_{j^2 j^3} + \beta_{j^1 j^3} &\leq 2 && j^1, j^2, j^3 \in \{1, ..., m\},\ j^3 > j^2 > j^1 \end{aligned}$$

where α and β are additional binary variables indicating if two different jobs run in sequence (and the precedence is specified by α) or overlap (stated by β).

Notice that the first two constraints guarantee:

$$t_{j^1} + \sigma_{j^1} + \tau_{j^1} \leq t_{j^2} \textbf{ NOR } t_{j^2} + \sigma_{j^2} + \tau_{j^2} \leq t_{j^1} \Rightarrow \beta_{j^1 j^2} = 1.$$

In other terms, if j^1 and j^2 overlap, $\beta_{j^1 j^2}$ becomes 1.

Third inequality takes into account that only two machines are available. In particular, it limits the number of possible two-by-two overlappings in all sets of three jobs to be at maximum 2. This way no more than two activities can run in parallel at every time.

Due date constraints are also included, considering both setup and processing durations:

$$t_{j^1} + \sigma_{j^1} + \tau_{j^1} \leq d_{j^1} \qquad j^1 \in \{1, ..., m\}$$

Costs minimization is addressed by the introduction of an objective function composed of two terms:

$$\min \{s(1 + \gamma) + p \sum_{\substack{j^1 \in \{1,...,m\} \\ j^2 \in \{1,...,m\} \\ j^2 > j^1}} \zeta_{j^1 j^2}\}.$$

The first term accounts operators salaries, while the second considers penalties to be paid. Roles of variables γ and $\zeta_{j^1 j^2}$ are defined by the following constraints:

$$\begin{aligned} t_{j^1} + \sigma_{j^1} - t_{j^2} &\leq M(1 - \theta_{j^1 j^2}) && j^1, j^2 \in \{1, ..., m\},\ j^2 > j^1 \\ t_{j^2} + \sigma_{j^2} - t_{j^1} &\leq M\theta_{j^1 j^2} + M\gamma && j^1, j^2 \in \{1, ..., m\},\ j^2 > j^1 \\ t_{j^1} + \sigma_{j^1} + 0.1\tau_{j^1} - t_{j^2} - \sigma_{j^2} &\leq M(1 - \delta_{j^1 j^2}) && j^1, j^2 \in \{1, ..., m\},\ j^2 > j^1 \\ t_{j^2} + \sigma_{j^2} + 0.1\tau_{j^2} - t_{j^1} - \sigma_{j^1} &\leq M\delta_{j^1 j^2} + M\zeta_{j^1 j^2} && j^1, j^2 \in \{1, ..., m\},\ j^2 > j^1 \end{aligned}$$

where $\theta_{j^1 j^2}$ and $\delta_{j^1 j^2}$ are auxiliary binary variables.

The first two inequalities guarantee:

$$t_{j^1} + \sigma_{j^1} \leq t_{j^2} \textbf{ NOR } t_{j^2} + \sigma_{j^2} \leq t_{j^1} \Rightarrow \gamma = 1.$$

Therefore, if setups of two jobs overlap, $\gamma = 1$ and two operators are considered in the objective function.

The other two constraints ensure:

$$t_{j^1} + \sigma_{j^1} + 0.1\tau_{j^1} \leq t_{j^2} + \sigma_{j^2} \textbf{ NOR } t_{j^2} + \sigma_{j^2} + 0.1\tau_{j^2} \leq t_{j^1} + \sigma_{j^1} \Rightarrow \zeta_{j^1 j^2} = 1,$$

i.e. if two jobs j^1 and j^2 are parallelized during their high demanding phase, $\zeta_{j^1 j^2}$ becomes 1 and penalty is added to overall costs.

All necessary limitations have been formulated. Table 1.15 and Figure 1.7 reports data and results for a particular instance, where $p = 2$ and $s = 80$ are assumed. Objective function value is 164 since two operators and two penalties have to be paid ($\gamma = 1, \zeta_{12} = 1, \zeta_{56} = 1$).

j	τ_j	σ_j	d_j	$t_j^\star$	$t_j^\star + \sigma_j + \tau_j$
1	210	30	300	0	240
2	150	30	500	0	180
3	160	30	500	240	430
4	230	30	500	180	440
5	180	30	650	430	640
6	180	30	650	440	650
7	160	30	950	655	845
8	210	30	1100	845	1085
9	150	30	1100	640	820
10	180	30	1100	820	1030

Table 1.15. Data and optimal solution for a particular instance with $p = 2$ and $s = 80$.

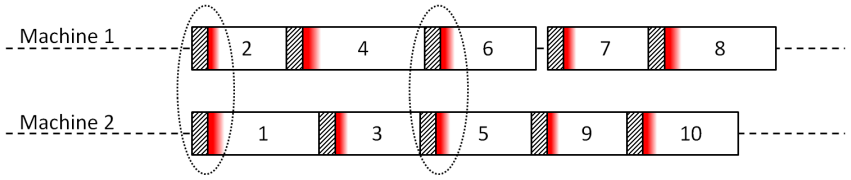

Figure 1.7. Optimal schedule for the two parallel machines.

1.4.4 Example 4: Scheduling of inspections in a continuous production system

Let us consider a continuous manufacturing environment which is subject to controls to verify that the system status does not deteriorate over time. To guarantee this, several checks must be arranged during the scheduling horizon, being sure to respect some limitations:

1. k controls have to be scheduled in the first N days of production (with $N \geq k$), each one requiring b hours.

2. Controls can not be executed while other activities are processing so they must fit in the intervals between different activity executions (activities do not run in parallel due to non-overlapping constraints that we consider already included in the formulation).

3. There are two possible control policies under evaluation. We assume they are "activated" depending on the value of a variable γ, used somewhere else in the model:

   (a) if $\gamma = 1$ the first policy is selected: it is based on the idea that it doesn't make sense to plan too close in time inspections, thus a minimum time of d hours must be guaranteed between the start of two consecutive controls, where $d > b$.

   (b) if $\gamma = 0$ the second policy is chosen: it establishes that at most one inspection can be started in each date. However, it admits controls to be less spaced in time over different days, satisfying a minimum distance of c hours between two starts, with $d > c > b$.

First requirement is clearly implicated by:

$$s_i + b \leq 24N \qquad i = 1, ..., k$$

where s_i represents the starting time of control i.

In order to ensure the k inspections not to be parallelized with the m activities, non-overlapping constraints are introduced:

$$\begin{aligned} t_j + \tau_j - s_i &\leq M(1 - \theta_{ji}) \qquad i = 1, ..., k,\ j = 1, ..., m \\ s_i + b - t_j &\leq M\theta_{ji} \qquad i = 1, ..., k,\ j = 1, ..., m \end{aligned}$$

where t_j is the starting time of activity j and θ_{ji} is an auxiliary binary variable to indicate the precedence between activity j and control i.

Let us now consider limitations linked to γ. If the first policy is active ($\gamma = 1$), every pair of controls $\{i, i'\}$ must be sufficiently spaced, i.e. either control i starts more than d hours before i' or the contrary should happen. Using the additional variable $\zeta_{ii'}$, we write constraints:

$$\begin{aligned} s_i + d - s_{i'} &\leq M(1 - \zeta_{ii'}) \qquad i = 1, ..., k,\ i' > i \\ s_{i'} + d - s_i &\leq M\zeta_{ii'} + M(1 - \gamma) \qquad i = 1, ..., k,\ i' > i \end{aligned}$$

These guarantee:

$$\gamma = 1 \Rightarrow s_i + d \leq s_{i'} \textbf{ OR } s_{i'} + d \leq s_i.$$

The second possibility is more complex to formulate. First of all we observe that, given a control i, it starts in day n (with $n \in 1, 2, ..., N$) if and only if:

$$24(n-1) \leq s_i < 24n.$$

We can therefore introduce binary variables δ_{in} such that, for each $i = 1, ..., k$ and $n = 1, ..., N$,

$$24(n-1) \leq s_i \textbf{ AND } s_i < 24n \Rightarrow \delta_{in} = 1,$$

which can also be written as:

$$24(n-1) - s_i \leq 0 \textbf{ BUT-NOT } 24n - s_i \leq 0 \Rightarrow \delta_{in} = 1.$$

To express these conditions we introduce constraints:

$$\begin{aligned} -M\delta_{in} - M\xi_{in} + \varepsilon \leq 24(n-1) - s_i \qquad & i = 1, ..., k,\ n = 1, ..., N \\ 24n - s_i \leq M(1 - \xi_{in}) \qquad & i = 1, ..., k,\ n = 1, ..., N \end{aligned}$$

where $\xi_{in} \in \{0, 1\}^{k \times N}$ are additional variables.

Then, to limit the amount of controls that can be arranged every day to 1 when the second policy is selected, we write:

$$\sum_{i=1}^{k} \delta_{in} \leq 1 + (k-1)\gamma \qquad n = 1, ..., N$$

Indeed, if γ is equal to 0, $\sum_{i=1}^{k} \delta_{in} \leq 1$ must hold in each day n, otherwise inequalities are relaxed (here $k-1$ acts as a big-M coefficient).

As a last step to complete defining this policy, we need to sufficiently separate inspections. Thus we have (again, $\beta_{ii'}$ is an auxiliary binary variable to represent precedences):

$$\begin{aligned} s_i + c - s_{i'} \leq M(1 - \beta_{ii'}) \qquad & i = 1, ..., k,\ i' > i \\ s_{i'} + c - s_i \leq M\beta_{ii'} \qquad & i = 1, ..., k,\ i' > i \end{aligned}$$

Notice it is not necessary to consider how to relax these constraints when γ is not 0, since first policy, as long as $d > c$, automatically satisfy these conditions. However, it could be easily taken into account by adding term $+M\gamma$ on the second constraints right-hand side.

Just to have a visual example of this problem, let us now consider a simple situation with 5 activities, each one with a processing time of 15 hours. The following data is assumed:

$$k = 3,\ N = 3,\ b = 3,\ d = 20,\ c = 8.$$

We imagine the function to be minimized is the sum of activities completion times and γ to be simply decided according to this objective.

An optimal solution is shown in Figure 1.8 where the first policy is selected, bringing an objective of 248 hours. White boxes represent activities while gray boxes correspond to inspections.

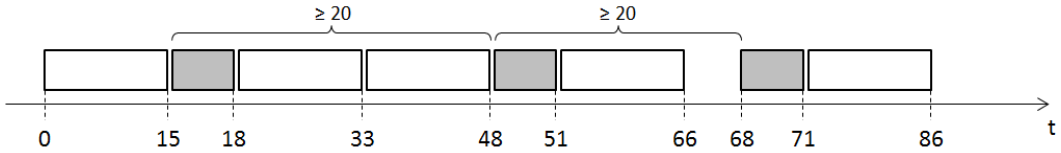

Figure 1.8. Optimal solution: first policy is used.

Differently, Figure 1.9 reports the optimal schedule that would be obtained if the second policy was forced by setting $\gamma = 0$. Its objective value is 249 hours.

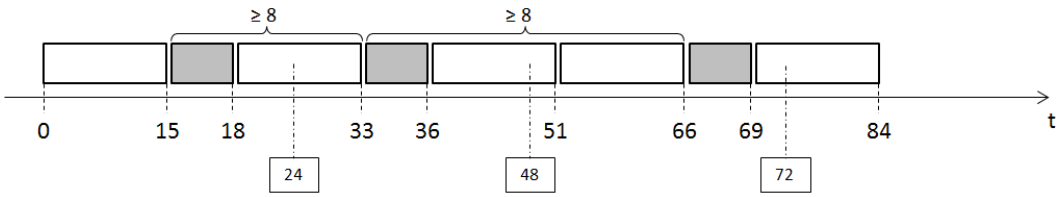

Figure 1.9. Optimal solution when second policy is forced.

Chapter 2

New inequalities for capacity-constrained scheduling formulations

In Chapter 1 we have seen in detail some possible ways to express logical conditions by introducing particular sets of big-M constraints and using suitable indicator variables. Furthermore, we provided some explicative examples, with the purpose of showing how these arguments have relevance in certain application fields and especially for scheduling problems. Following on from this point, in this chapter we focus on a specific use of the techniques previously described, in particular by addressing problems in which resources employed for processing are scarce and consequently the scheduling of activities is necessarily constrained by availability issues.

The chapter is organized as follows: in Section 2.1 we introduce the main concepts and variables used in our work, taking into account only two activities and showing the differences with standard approaches. In Section 2.2 we extend previous arguments to m operations, and focus on the modeling of capacity constraints. Finally, in Section 2.3 we present some modifications to improve the proposed formulation.

2.1 From disjunctive constraints to their extensions

As already noticed, disjunctive constraints are frequently used in scheduling formulations. Indeed, they allow to model situations in which activities have to be sequenced, without (or at least not completely) having a predefined chronological order to respect. This is the case, for example, of different jobs to be processed on the same machine: either one knows which is the working sequence of the machine, or a mathematical model with a suitable structure to define precedences and avoid times overlapping must be built. Practically, let A1 and A2 be two activities and τ_1 and τ_2 their respective durations; if a priori we have information that A1 must precede A2, we can easily express the limitation with:

$$t_1 + \tau_1 \leq t_2, \tag{2.1}$$

where t_1 and t_2 are the activities variable starting times and the left-hand side of the constraint represents the completion of A1. Vice versa, if A1 must follow A2, we obviously have:

$$t_2 + \tau_2 \leq t_1. \tag{2.2}$$

However, if the relative order between the activities is not given, disjunctive (or non-overlapping) constraints can be used, as described in Section 1.4:

$$t_1 + \tau_1 - t_2 \leq M(1 - \theta), \tag{2.3}$$

$$t_2 + \tau_2 - t_1 \leq M\theta, \tag{2.4}$$

with θ being an additional binary variable, such that $\theta = 1$ implicates (2.1) and $\theta = 0$ involves (2.2).

This idea can clearly be extended to consider more complex situations: let us now assume that in some cases it is possible to relax the non-parallelization requirement. So, similarly to what seen in Section 1.3, we may introduce a binary variable γ that, if equal to 1, allows to ignore scheduling constraints and to eventually execute activities simultaneously. The following have to be satisfied:

$$t_1 + \tau_1 - t_2 \leq M(1 - \theta), \tag{2.5}$$

$$t_2 + \tau_2 - t_1 \leq M\theta + M\gamma, \tag{2.6}$$

which express condition:

$$t_1 + \tau_1 - t_2 > 0 \textbf{ AND } t_2 + \tau_2 - t_1 > 0 \Rightarrow \gamma = 1 \ (\textbf{AND } \theta = 0),$$

or equivalently:

$$t_1 + \tau_1 - t_2 \leq 0 \textbf{ NOR } t_2 + \tau_2 - t_1 \leq 0 \Rightarrow \gamma = 1 \ (\textbf{AND } \theta = 0).$$

Variable γ represents the possibility to parallelize the activities. If A1 and A2 overlap (i.e. $t_2 < t_1 + \tau_1$ and $t_1 < t_2 + \tau_2$), then γ must become equal to 1. However, $\gamma = 0$ reduces (2.5) and (2.6) to disjunctive constraints (2.3) and (2.4).

Similar approaches can be considered to express other types of relations between activities. Notice that inequalities (2.5) and (2.6) admit activity parallelization thanks to γ, which is anyway unable to bring any information other than this. Now we want to introduce a finer concept by moving the focus from the fact that activities overlap to the way they do. Indeed, also considering only two activities with generic durations, there are many different modes they may be ordered in time. This is shown in Figure 2.1. Similarly, Figure 2.2 illustrates all possible cases of non-parallelization.

In order to explore some interesting properties, we suppose to be interested in knowing the relative position of an activity with respect to the other activity starting time. To address this topic, we first need to define the concepts of left-overlapping and right-overlapping:

Definition 1. *Given two activities A1 and A2, having processing times τ_1 and τ_2 and starting times t_1 and t_2, we say that activity A1* ***left-overlaps*** *A2 if and only if:*

$$t_1 \leq t_2 < t_1 + \tau_1.$$

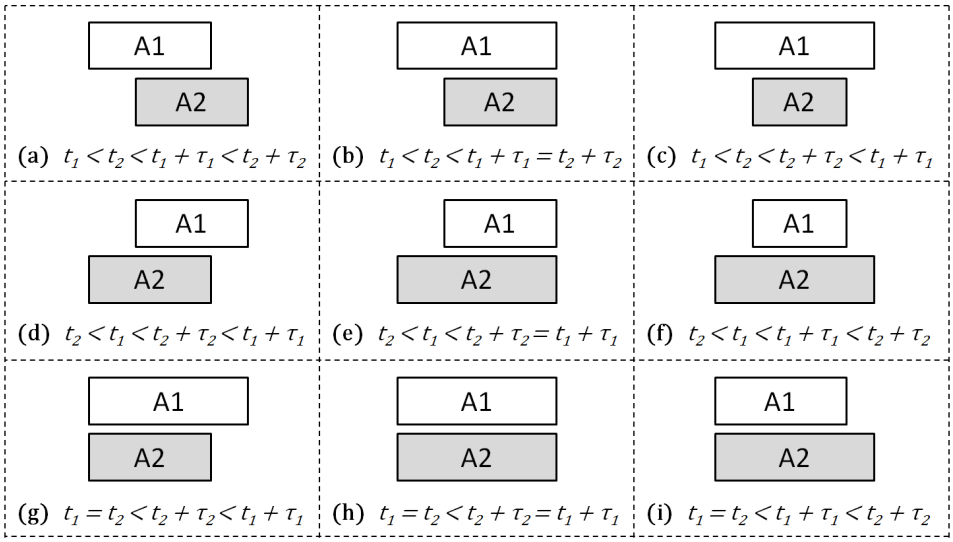

Figure 2.1. Possible modes of overlapping between two activities with generic processing times.

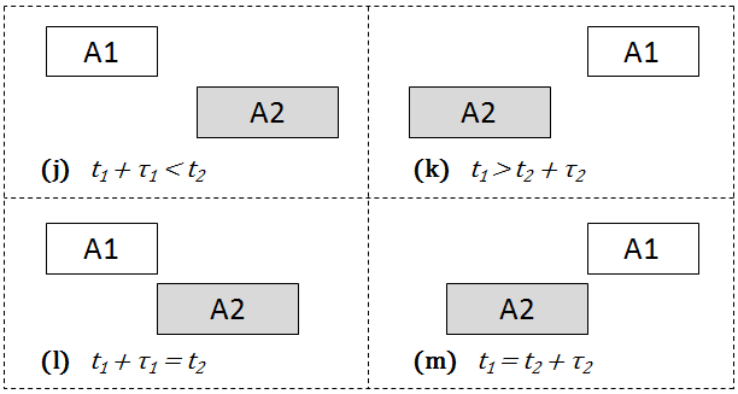

Figure 2.2. Possible dispositions of two activities that do not overlap.

Conversely, we say that activity A1 ***right-overlaps*** *A2 if and only if:*

$$t_2 \leq t_1 < t_2 + \tau_2.$$

Of course the two definitions are related: indeed, if activity A1 left-overlaps A2, then A2 right-overlaps A1. Furthermore, if activity A1 left-overlaps A2 and A2 left-overlaps A1, then it means that the two activities start at the same time. In a similar way, if activities A1 and A2 begin simultaneously, then it is possible to say that they both left- and right-overlap each other.

Referring to Figure 2.1 for a visual example, we have that A1 left-overlaps A2 in cases (a), (b), (c), (g), (h) and (i) and that A2 left-overlaps A1 in cases (d), (e), (f), (g), (h) and (i).

These concepts allow to analyze different modeling opportunities. Let us imagine to know that A2 can not start while A1 is processing (A1 can not left-overlap A2). In this situation, there are two possibilities: either A1 begins after t_2 (i.e. $t_1 > t_2$) or

A1 finishes before (or exactly at) t_2 (i.e. $t_1 + \tau_1 \leq t_2$). This alternative is expressed by the following two conditions:

$$t_1 + \tau_1 - t_2 \leq M(1 - \theta_{12}), \tag{2.7}$$

$$t_2 - t_1 \leq M\theta_{12} - \varepsilon, \tag{2.8}$$

where ε is a small parameter useful to remove the strict inequality and binary variable θ_{12} governs the choice between the two available options.

It is worth to notice the similarity between this couple of constraints and relations (2.3)-(2.4), where we used θ in place of θ_{12}. In particular, it can be trivially derived that, as far as τ_2 is a positive constant greater than ε (as we actually assume), (2.3) and (2.4) simply implicate (2.7) and (2.8). This makes sense also in more general terms, since avoiding overlapping (i.e. imposing constraints (2.3)-(2.4)) of course involves to avoid left-overlapping (i.e. (2.7)-(2.8)).

To further investigate this idea, we may once again consider variable γ as used in constraints (2.5)-(2.6), i.e. with the role of admitting/forbidding activity parallelization. Then, we can define variables γ_{12} and γ_{21} such that $\gamma_{12} = 1$ means that A1 can left-overlap A2 and vice versa for $\gamma_{21} = 1$. The following relations are conceptually true:

$$\gamma \geq \gamma_{12},$$

$$\gamma \geq \gamma_{21},$$

$$\gamma \leq \gamma_{12} + \gamma_{21}.$$

Indeed, few options are possible:

1. $\gamma_{12} = 0$, $\gamma_{21} = 0$, $\gamma = 0$: activities can not overlap.
2. $\gamma_{12} = 1$, $\gamma_{21} = 0$, $\gamma = 1$: activity A1 can left-overlap activity A2.
3. $\gamma_{12} = 0$, $\gamma_{21} = 1$, $\gamma = 1$: activity A2 can left-overlap activity A1 (A1 can right-overlap A2).
4. $\gamma_{12} = 1$, $\gamma_{21} = 1$, $\gamma = 1$: activities A1 and A2 can overlap and possibly start at the same time.

In order to practically express the relation between variable γ_{12} and the activity starting times, we need to modify constraints (2.7)-(2.8) by adding the term $M\gamma_{12}$, which allows their relaxation. So we can write:

$$t_1 + \tau_1 - t_2 \leq M(1 - \theta_{12}), \tag{2.9}$$

$$t_2 - t_1 \leq M\theta_{12} + M\gamma_{12} - \varepsilon. \tag{2.10}$$

It is easy to verify that here γ_{12} controls if activity A1 can left-overlap A2. Indeed, if $\gamma_{12} = 0$, A2 is forced to start before t_1 or after $t_1 + \tau_1$ (i.e., (2.9)-(2.10) are reduced again to (2.7)-(2.8)); otherwise both constraints can be relaxed by setting $\theta_{12} = 0$. Equivalently, if A1 does not left-overlap A2, then γ_{12} can take any value; in the opposite case it necessarily becomes equal to 1.

Considering exactly the same logic also for variable γ_{21}, it is possible to write the following:

$$t_2 + \tau_2 - t_1 \leq M(1 - \theta_{21}), \tag{2.11}$$

$$t_1 - t_2 \leq M\theta_{21} + M\gamma_{21} - \varepsilon, \tag{2.12}$$

where variable θ_{21}, similarly to θ_{12}, permits to identify the alternatives when left-overlapping is not admitted ($\gamma_{21} = 0$).

2.1.1 Overlapping variables: γ, γ_{12} and γ_{21}

An interesting observation about previous formulations is what is possible to express by using constraints (2.9)-(2.12) all together. Indeed, by introducing a set of four different linear inequalities, we may have information (through variables γ_{12} and γ_{21}) on the relative position of each activity with respect to the other. If, instead of (2.9)-(2.12), we had written just two constraints with variable γ, namely (2.5)-(2.6), we would have been able only to spot the mutual overlapping of activities without indications on their ordering in time.

This idea probably becomes clearer by extending the role of indicator variables γ, γ_{12} and γ_{21} from only being *necessarily* activated in case of overlapping to also be *sufficient* for this.

In particular, we saw that (2.5)-(2.6) express implication:

$$t_1 + \tau_1 - t_2 \leq 0 \textbf{ NOR } t_2 + \tau_2 - t_1 \leq 0 \Rightarrow \gamma = 1.$$

If we want to transform this condition in an if-and-only-if and guarantee

$$t_1 + \tau_1 - t_2 \leq 0 \textbf{ NOR } t_2 + \tau_2 - t_1 \leq 0 \Leftrightarrow \gamma = 1,$$

we need to add two inequalities:

$$-M(1 - \gamma) + \varepsilon \leq t_1 + \tau_1 - t_2, \tag{2.13}$$

$$-M(1 - \gamma) + \varepsilon \leq t_2 + \tau_2 - t_1. \tag{2.14}$$

Similarly, in (2.9)-(2.12), $\gamma_{12} + \gamma_{21} \geq 1$ is a *necessary* condition for overlapping. Indeed, two relations are true:

$$t_1 \leq t_2 < t_1 + \tau_1 \Rightarrow \gamma_{12} = 1,$$

$$t_2 \leq t_1 < t_2 + \tau_2 \Rightarrow \gamma_{21} = 1,$$

that is:

$$t_1 - t_2 \leq 0 \textbf{ BUT-NOT } t_1 + \tau_1 - t_2 \leq 0 \Rightarrow \gamma_{12} = 1,$$

$$t_2 - t_1 \leq 0 \textbf{ BUT-NOT } t_2 + \tau_2 - t_1 \leq 0 \Rightarrow \gamma_{21} = 1.$$

In order to make these conditions also *sufficient* and guarantee the double implications

$$t_1 - t_2 \leq 0 \textbf{ BUT-NOT } t_1 + \tau_1 - t_2 \leq 0 \Leftrightarrow \gamma_{12} = 1,$$

$$t_2 - t_1 \leq 0 \textbf{ BUT-NOT } t_2 + \tau_2 - t_1 \leq 0 \Leftrightarrow \gamma_{21} = 1,$$

Case	γ	γ_{12}	γ_{21}	Cases in Figures 2.1-2.2
activities do not overlap	0	0	0	(j),(k),(l),(m)
activities overlap and t_1 precedes t_2	1	1	0	(a),(b),(c)
activities overlap and t_2 precedes t_1	1	0	1	(d),(e),(f)
activities overlap and $t_1 = t_2$	1	1	1	(g),(h),(i)

Table 2.1. Possible cases and corresponding values for variables γ, γ_{12}, γ_{21}, as determined by constraints (2.5)-(2.6), (2.13)-(2.14) and (2.9)-(2.12), (2.15)-(2.18).

four constraints must be added to (2.9)-(2.12):

$$-M(1-\gamma_{12})+\varepsilon \leq t_1+\tau_1-t_2, \tag{2.15}$$

$$-M(1-\gamma_{12}) \leq t_2-t_1, \tag{2.16}$$

$$-M(1-\gamma_{21})+\varepsilon \leq t_2+\tau_2-t_1, \tag{2.17}$$

$$-M(1-\gamma_{21}) \leq t_1-t_2. \tag{2.18}$$

Now that all binary variables have been linked to activities starting times through necessary and sufficient conditions, it is possible to schematically collect our findings, as reported in Table 2.1. Here the differences between approaches in (2.5)-(2.6),(2.13)-(2.14) and in (2.9)-(2.12),(2.15)-(2.18) become evident. Indeed, paying the cost of having twice the number of constraints and binary variables of the first, the second formulation allows to distinguish between the four different cases of Table 2.1. This of course is not possible by using only variable γ which in fact assumes the same value every time there is overlapping of activities (last three rows in the table). Referring once again to Figure 2.1, this means that $\gamma = 1$ could be indistinctly associated to any case from (a) to (i).

We will soon clarify how the additional information brought by variables γ_{12} and γ_{21} can be leveraged to efficiently formulate some particular scheduling limitations. Before, however, it is worth to focus on another important aspect of previous models: the role of variables θ, θ_{12} and θ_{21}.

2.1.2 Sequencing variables: θ, θ_{12} and θ_{21}

Starting from formulation (2.5)-(2.6),(2.13)-(2.14), we notice that, as long as γ is equal to 1, θ has to be 0. Indeed, (2.5) and (2.13) together give:

$$-M(1-\gamma)+\varepsilon \leq M(1-\theta),$$

where the left-hand side of inequality is strictly positive.

A similar implication comes out also from relations (2.9)-(2.12),(2.15)-(2.18), where $\gamma_{12} = 1$ and $\gamma_{21} = 1$ respectively force θ_{12} and θ_{21} to 0 (see (2.9) and (2.15) for θ_{12} and (2.11) and (2.17) for θ_{21}).

Nevertheless, if in the first formulation γ is 0, θ specifies in which order the activities are executed ($\theta = 1$ if A1 precedes A2, $\theta = 0$ if A1 follows A2). In other

terms, variable θ indicates "in which direction" parallelization is avoided. Referring to θ_{12} and θ_{21}, they express an analogous idea with reference to the definition of left-overlapping. For example, if $\gamma_{12} = 0$ and $\theta_{12} = 0$, constraint (2.10) imposes:

$$t_2 < t_1,$$

so A1 can not left-overlap A2 because A2 starts before A1. Vice versa, if $\gamma_{12} = 0$ and $\theta_{12} = 1$, from (2.9) we have:

$$t_2 \geq t_1 + \tau_1$$

and left-overlapping can not happen because A2 waits at least until A1 is completed.

These observations seem to suggest that θ_{12} and θ_{21} can express a sort of ordering between the activities, although not explicit as the one provided by θ. So different questions may rise: if θ_{12} can tell if A2 follows A1 and θ_{21} is close to indicate the contrary, are they redundant? Can one of the two variables be simply expressed in terms of the other? In particular, does the transformation $\theta_{21} = 1 - \theta_{12}$ hold? Unfortunately the answer to all these questions is no, in the formulation considered. To prove this fact, consider relations (2.9) and (2.11) where we have replaced θ_{21} with $1 - \theta_{12}$. We get:

$$t_1 + \tau_1 - t_2 \leq M(1 - \theta_{12}),$$

$$t_2 + \tau_2 - t_1 \leq M\theta_{12}.$$

These relations express the alternative:

$$t_1 + \tau_1 \leq t_2 \textbf{ OR } t_2 + \tau_2 \leq t_1,$$

and this of course has to be avoided since it would again preclude overlapping, that is the opposite of our initial scope.

In the following, we will propose a trick to make the substitution $\theta_{21} = 1 - \theta_{12}$ feasible, giving the opportunity to remove a variable from the formulation and so modeling the problem in a more efficient way. For the moment, however, let us postpone this topic to Section 2.3.2 and conclude with some remarks:

- Constraints (2.9) and (2.11) guarantee:

  $$\theta_{12} = 1 \Rightarrow t_1 + \tau_1 \leq t_2,$$

  $$\theta_{21} = 1 \Rightarrow t_2 + \tau_2 \leq t_1.$$

  Thus, if θ_{12} or θ_{21} is equal to 1, parallelization is not possible, and consequently $\gamma_{12} = \gamma_{21} = 0$ (see sufficient relations (2.15)-(2.18)).

- Variables θ_{12} and θ_{21} can not be 1 at the same time.

- If $\theta_{12} = 0$ and $\theta_{21} = 0$, at least one between γ_{12} and γ_{21} has to be 1, implying the overlapping of A1 and A2 (see constraints (2.10) and (2.12)).

Table 2.2 summarizes these results.

γ_{12}	γ_{21}	θ_{12}	θ_{21}	Meaning	Cases in Figures 2.1-2.2
0	0	0	1	A2 precedes A1	(k),(m)
		1	0	A1 precedes A2	(j),(l)
0	1	0	0	A2 left-overlaps A1	(d),(e),(f)
1	0	0	0	A1 left-overlaps A2	(a),(b),(c)
1	1	0	0	A1 and A2 start at the same time	(g),(h),(i)

Table 2.2. Possible values of variables $\gamma_{12}, \gamma_{21}, \theta_{12}$ and θ_{21} and corresponding meaning, as determined by constraints (2.9)-(2.12), (2.15)-(2.18).

2.2 Extension to m activities

In the previous sections we described the role of each inequality and variable in formulation (2.9)-(2.12) and (2.15)-(2.18). In order to practically exploit our findings, we now extend our approach to consider the case in which the activities that must be scheduled are m, with $m > 2$.

Formally, we can define a set $A = \{1, ..., m\}$ of indexes, such that each element $j \in A$ represents a particular non-preemptive activity characterized by a parametric duration $\tau_j > 0$ and a variable starting time $t_j > 0$.

In order to study the properties of overlapping in this situation, it still makes sense to take into account activities two by two. We thus introduce the set containing all possible activity pairs:

$$A^2 = \{\{j, c\} : j \in A, c \in A, c \neq j\}$$

Starting from this, we can write constraints (2.9)-(2.12) and (2.15)-(2.18) for every pair $\{j, c\} \in A^2$ as follows:

$$t_j + \tau_j - t_c \leq M(1 - \theta_{jc}), \tag{2.19}$$
$$t_c - t_j \leq M\theta_{jc} + M\gamma_{jc} - \varepsilon, \tag{2.20}$$
$$t_c + \tau_c - t_j \leq M(1 - \theta_{cj}), \tag{2.21}$$
$$t_j - t_c \leq M\theta_{cj} + M\gamma_{cj} - \varepsilon, \tag{2.22}$$
$$-M(1 - \gamma_{jc}) + \varepsilon \leq t_j + \tau_j - t_c, \tag{2.23}$$
$$-M(1 - \gamma_{jc}) \leq t_c - t_j, \tag{2.24}$$
$$-M(1 - \gamma_{cj}) + \varepsilon \leq t_c + \tau_c - t_j, \tag{2.25}$$
$$-M(1 - \gamma_{cj}) \leq t_j - t_c, \tag{2.26}$$

where (2.19)-(2.20) and (2.23)-(2.24) guarantee:

$$t_j \leq t_c < t_j + \tau_j \Leftrightarrow \gamma_{jc} = 1,$$

while (2.21)-(2.22) and (2.25)-(2.26) yield:

$$t_c \leq t_j < t_c + \tau_c \Leftrightarrow \gamma_{cj} = 1.$$

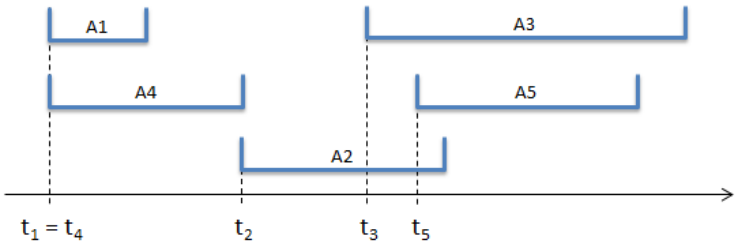

Figure 2.3. Example of a Gantt chart involving five activities.

2.2.1 Sharing of resources and capacity restrictions

Constraints (2.19)-(2.26) suggest a new way of interpreting variables γ_{jc} and γ_{cj}. We say that γ_{jc} is equal to 1 if and only if activity j left-overlaps activity c. So, considering a particular activity $\hat{\jmath}$, it would be possible to determine which are the activities $c \in A$ that start during its processing, simply observing variables $\gamma_{\hat{\jmath}c}$. But it is also interesting to look at the same definition from a different perspective: given an activity $\hat{c}$, we may want to take into account those activities $j \in A$ that are in progress at the instant $\hat{c}$ begins.

This allows to easily model scheduling problems with sharing of resources. Indeed, finding a feasible schedule means ordering operations and establishing the starting time for each activity involved. The output, formally represented by a vector $t \in \mathbb{R}^m_+$, can be naturally displayed in a Gantt chart. An example with five activities is reported in Figure 2.3 where the importance of variable $t \in \mathbb{R}^5_+$ is evident, defining the main instants, together with the activities ending times, in which the status of the system changes.

From a practical point of view, this can greatly help in checking the feasibility of a solution, because reduces the problem of satisfying some given constraints over time to that of controlling them in particular instants of the schedule. For future reference, consider the example of activities in Figure 2.3 and assume to have written inequalities (2.19)-(2.26) for each pair $\{j, c\} \in A^2$. The following hold:

- A1 and A4 overlap and start at the same time ($\gamma_{14} = 1$, $\gamma_{41} = 1$);
- A2 left-overlaps A3 ($\gamma_{23} = 1$, $\gamma_{32} = 0$);
- A2 left-overlaps A5 ($\gamma_{25} = 1$, $\gamma_{52} = 0$);
- A3 left-overlaps A5 ($\gamma_{35} = 1$, $\gamma_{53} = 0$);
- all the other pairs of activities $\{j, c\}$ do not overlap ($\gamma_{jc} = \gamma_{cj} = 0$).

Single exclusive resource

Figure 2.3 shows different intervals in which activities overlap. During $[t_1, t_1 + \tau_1], [t_3, t_5]$ and $[t_2 + \tau_2, t_5 + \tau_5]$ two operations run in parallel while in the interval $[t_5, t_2 + \tau_2]$ three activities (i.e. A2, A3 and A5) are simultaneously in process.

If it was necessary to completely avoid parallelization, it would be enough to impose $\gamma_{jc} = 0$ and $\gamma_{cj} = 0$ for each $\{j, c\} \in A^2$, or equivalently:

$$\sum_{\{j,c\}\in A^2} \gamma_{jc} + \gamma_{cj} = \sum_{j\in A} \sum_{\substack{c\in A \\ c\neq j}} \gamma_{jc} \leq 0. \tag{2.27}$$

Consequently, a feasible solution would sequentialize activities (e.g., $t_1 = 0$, $t_4 = t_1 + \tau_1$, $t_2 = t_4 + \tau_4$, $t_3 = t_2 + \tau_2$, $t_5 = t_3 + \tau_3$). This situation well represents the case in which the m activities make use of the same single resource which can be exclusively allocated to only one operation at a time. A typical example is a machine for different jobs.

Notice that if in place of formulation (2.19)-(2.26) plus (2.27) we used standard constraints like (2.5)-(2.6) and (2.13)-(2.14), we would be able to express the same limitations:

$$\begin{aligned}
t_j + \tau_j - t_c &\leq M(1 - \bar{\theta}_{jc}) && \forall\{j,c\} \in A^2, \\
t_c + \tau_c - t_j &\leq M\bar{\theta}_{jc} + M\bar{\gamma}_{jc} && \forall\{j,c\} \in A^2, \\
-M(1 - \bar{\gamma}_{jc}) + \varepsilon &\leq t_j + \tau_j - t_c && \forall\{j,c\} \in A^2, \\
-M(1 - \bar{\gamma}_{jc}) + \varepsilon &\leq t_c + \tau_c - t_j && \forall\{j,c\} \in A^2, \\
\sum_{\{j,c\}\in A^2} \bar{\gamma}_{jc} &\leq 0,
\end{aligned}$$

where $\bar{\gamma}_{jc}$ and $\bar{\theta}_{jc}$ respectively replace original γ and θ to take into account and represent every possible pair of activities.

Two resources

Let us now consider a more complex situation and assume to have a constraint specifying that no more than two activities can be parallelized, for instance because there is a resource with capacity equal to 2 that must be shared between different operations. In this case, we can complete formulation (2.19)-(2.26) with the m additional inequalities:

$$\sum_{\substack{j\in A \\ j\neq c}} \gamma_{jc} \leq 1 \qquad \forall c \in A.$$

These can be interpreted as follows: given the set of activities A, to verify there are no more than two activities in process at the same time, it is enough to make a check for each $c \in A$. Indeed, selected a particular c, the number of activities that left-overlap c is made available by binary variables γ_{jc}; to limit parallelization, it is possible to simply bound that amount, forcing the sum of γ_{jc} (with $j \neq c$) to be lower than the maximum admitted, i.e. the capacity minus 1 (one unit of capacity is considered occupied by activity c).

Since a check is performed for each activity (at every t_j, with $j \in A$), that is in each instant a new overlapping could be generated, this is sufficient to guarantee feasibility over time. Activities ending times play no role in these inequalities because they do not create issues in terms of capacity, only freeing resources previously occupied. Referring to the case of A1 and A2, this is the reason why it is not

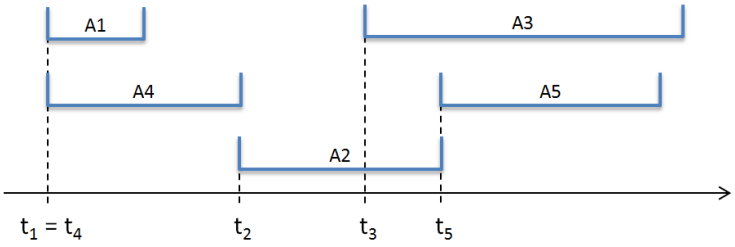

Figure 2.4. Example of a feasible solution subject to a capacity limit of 2 parallel activities.

necessary to distinguish, for instance, between situations (a),(b) and (c) of Figure 2.1. Indeed they all show the same behavior at t_2, equally implying $\gamma_{12} = 1$.

To have a graphic example of how the inclusion of the last constraints can avoid to simultaneously execute activities A2, A3 and A5 of Figure 2.3, look at Figure 2.4 where A5 has been postponed to satisfy capacity restriction. In particular, the condition for $c = 5$,

$$\gamma_{15} + \gamma_{25} + \gamma_{35} + \gamma_{45} \leq 1,$$

now holds ($\gamma_{15} = 0, \gamma_{25} = 0, \gamma_{35} = 1, \gamma_{45} = 0$). Before we had $\gamma_{15}+\gamma_{25}+\gamma_{35}+\gamma_{45} = 2$ since γ_{25} was equal to 1.

If we considered the same limitation using the standard scheduling variable $\bar{\gamma}_{jc}$, the formulation would become more elaborate: indeed, to avoid the parallelization of more than two operations, it would be necessary to enumerate all possible groups of three activities, for example A1, A2 and A3, and write for all of them a constraint of this type:

$$\bar{\gamma}_{12} + \bar{\gamma}_{23} + \bar{\gamma}_{13} \leq 2.$$

The disadvantage of this procedure lies in the fact that, as well as the capacity and m change, the number of inequalities to be included in the formulation may increase exponentially, making also impossible to write the model in a compact form. For example, if the activities are 5 and the capacity is 2, the model requires 10 constraints, i.e. one for each possible combination of 3 over 5 operations. For an equal capacity and $m = 10$ the situation worsens, involving 120 relations (in general, $\frac{m!}{3!(m-3)!}$). By using capacity limitations as defined in the first approach, only 10 inequalities would be needed, one for each $c \in A$.

General case: non unitary request and different resource types

To definitely extend formulation (2.19)-(2.26) to consider any possibility of restriction due to scarce resources, two last steps must be taken into account. The first is what happens if the request of capacity by an activity is not unitary and possibly not discrete, as for example in the case of energy, where each operation may ask any predefined quantity between 0 and the total available. The second is how to manage limitations if more than one type of resource influences the possibility to execute the activities.

We address both these issues at the same time by introducing a general group of inequalities, that we call *capacity constraints*. First of all we define a set $R = \{1, ..., n\}$

representing the indexes of different resource types. Each resource $i \in R$ has a fixed capacity $C_i > 0$ and each activity j requires r_{ij} units of i, where $0 \leq r_{ij} \leq C_i$. Then, capacity constraints can be written as follows:

$$r_{ic} + \sum_{\substack{j \in A \\ j \neq c}} r_{ij}\gamma_{jc} \leq C_i \qquad \forall c \in A, \forall i \in R. \tag{2.28}$$

This corresponds to check for each resource and in every activity starting time that the total availability is not exceeded. The desired limitations can thus be modeled paying the cost of adding $m \times n$ linear inequalities.

A graphic representation of a feasible solution involving 5 activities and 2 resources is show in Figure 2.5. Two charts are used, one for each $i \in R$. Activities have two significant dimensions: the width, representing the duration τ_j, and the height, indicating the resource requirement r_{ij}. Parameters considered are reported in Table 2.3.

Resource type i	C_i	r_{i1}	r_{i2}	r_{i3}	r_{i4}	r_{i5}
1	2	0	1	1	2	1
2	3	1	2	1	1	3

Table 2.3. Data used for the example representation of Figure 2.5.

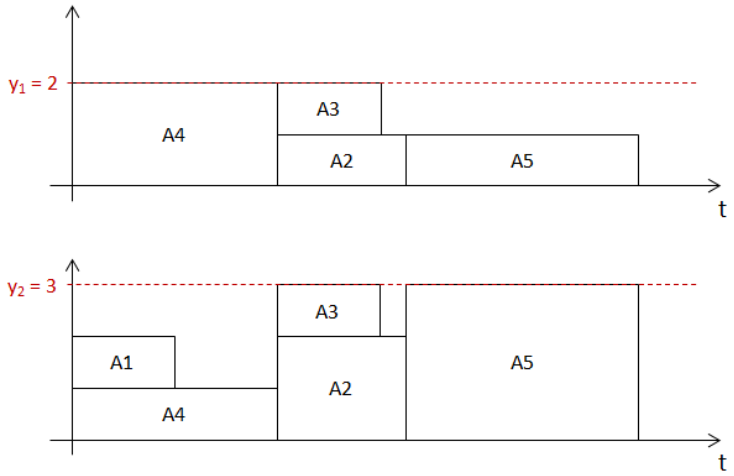

Figure 2.5. Example scheduling representation of 5 activities subject to 2 resource limitations.

2.3 Formulation improvements

In the previous sections we have shown the properties of our proposed formulation, in particular by highlighting its ability to deal with capacity limitations in a simple

way, allowing to model the problem in a compact form and keeping also a reasonable dimension.

Here, we continue on these arguments and present additional considerations with the aim of further improving constraints and better addressing the following chapters.

We will take into account the reduction of constraints (Section 2.3.1) and variables (Section 2.3.2), and the refinement of parameters (Section 2.3.3).

2.3.1 Constraints reduction

Taking a step back, we briefly recall the role of constraints (2.19)-(2.26) with respect to binary variables γ_{jc} and γ_{cj}, for each $\{j, c\} \in A^2$. We have:

- $t_j \leq t_c < t_j + \tau_j \Rightarrow \gamma_{jc} = 1$, due to (2.19)-(2.20);
- $t_c \leq t_j < t_c + \tau_c \Rightarrow \gamma_{cj} = 1$, due to (2.21)-(2.22);
- $\gamma_{jc} = 1 \Rightarrow t_j \leq t_c < t_j + \tau_j$, due to (2.23)-(2.24);
- $\gamma_{cj} = 1 \Rightarrow t_c \leq t_j < t_c + \tau_c$, due to (2.25)-(2.26).

The first two represent *necessary* relations, the third and fourth are instead *sufficient* implications.

Let us now consider capacity constraints (2.28): their aim is to limit the number of overlapping activities over time, such that the maximum availability of each resource type $i \in R$ is never exceeded. They may exclusively have a lowering effect on variables γ_{jc} and γ_{cj} since every coefficient r_{ij} is assumed nonnegative and the only way to satisfy conditions eventually unmet is by moving at least one binary variable from 1 to 0.

This suggests a simple observation: constraints (2.28) do not need the *sufficient* implications brought by relations (2.23)-(2.26), which can thus be removed from our formulation. As a result, we get a model constituted by $4m^2 + mn - 4m$ constraints: $4m(m-1)$ from (2.19)-(2.22) and mn from (2.28).

2.3.2 Variables reduction

We have already noticed that, as long as overlapping is not admitted, variables θ_{jc} and θ_{cj} assume a role in sequencing activities, indicating the precedence between operations j and c. We can observe that, if θ_{jc} is 1, constraint (2.19) implies $t_1 + \tau_1 \leq t_2$, and consequently $t_1 < t_2$. Moreover, if $\gamma_{jc} = 0$ and θ_{jc} is 0, it follows from (2.20) that $t_1 > t_2$. On the other hand, if θ_{cj} is 1, $t_2 + \tau_2 \leq t_1$ due to (2.21) (therefore $t_2 < t_1$), while $\gamma_{cj} = 0$ and $\theta_{cj} = 1$ involve $t_2 > t_1$ (see (2.22)). In addition, constraints (2.19) and (2.21) together exclude $\theta_{jc} = \theta_{cj} = 1$.

All these observations suggest two questions:

1. Do θ_{jc} and θ_{cj} indicate the sequence of activity j and c starting times?
2. Do the variables exactly express opposite meanings? Is it possible for example to substitute θ_{cj} with $1 - \theta_{jc}$?

The answer to these questions is no, but motivates some reasonings to improve our formulation. Proceeding step by step, we first notice that if we replaced θ_{cj} with $1 - \theta_{jc}$ in constraints (2.19) and (2.21), we would get an undesirable result:

$$t_j + \tau_j - t_c \leq M(1 - \theta_{jc}),$$

$$t_c + \tau_c - t_j \leq M\theta_{jc}.$$

Indeed, this would completely exclude overlapping, creating the alternative between $t_j + \tau_j \leq t_c$ (j precedes c) and $t_c + \tau_c \leq t_j$ (c precedes j).

In order to overcome this problem and still try to remove unnecessary variables from our formulation, we consider a trick. First of all, we add the terms $M\gamma_{jc}$ and $M\gamma_{cj}$ to inequalities (2.19) and (2.21) right-hand sides, and rewrite (2.19)-(2.22) as:

$$t_j + \tau_j - t_c \leq M(1 - \theta_{jc}) + M\gamma_{jc},$$

$$t_c - t_j \leq M\theta_{jc} + M\gamma_{jc} - \varepsilon,$$

$$t_c + \tau_c - t_j \leq M(1 - \theta_{cj}) + M\gamma_{cj},$$

$$t_j - t_c \leq M\theta_{cj} + M\gamma_{cj} - \varepsilon.$$

This does not alter the meaning of constraints since we have:

$$t_c \geq t_j \textbf{ BUT-NOT } t_c \geq t_j + \tau_j \Rightarrow \gamma_{jc} = 1;$$

$$t_j \geq t_c \textbf{ BUT-NOT } t_j \geq t_c + \tau_c \Rightarrow \gamma_{cj} = 1.$$

Using now the substitution $\theta_{cj} = 1 - \theta_{jc}$, we can write:

$$t_j + \tau_j - t_c \leq M(1 - \theta_{jc}) + M\gamma_{jc}, \tag{2.29}$$

$$t_c - t_j \leq M\theta_{jc} + M\gamma_{jc} - \varepsilon, \tag{2.30}$$

$$t_c + \tau_c - t_j \leq M\theta_{jc} + M\gamma_{cj}, \tag{2.31}$$

$$t_j - t_c \leq M(1 - \theta_{jc}) + M\gamma_{cj} - \varepsilon. \tag{2.32}$$

This formulation is coherent. To prove this, we distinguish case by case depending on the values of γ_{jc} and γ_{cj}:

- $\gamma_{jc} = 0$ and $\gamma_{cj} = 0$: if overlapping is not admitted, constraints (2.29)-(2.32) give:

  $$t_j + \tau_j - t_c \leq M(1 - \theta_{jc}),$$

  $$t_c - t_j \leq M\theta_{jc} - \varepsilon,$$

  $$t_c + \tau_c - t_j \leq M\theta_{jc},$$

  $$t_j - t_c \leq M(1 - \theta_{jc}) - \varepsilon.$$

  There are two possibilities:

  1. $\theta_{jc} = 0$: in this case the first and the last relations are relaxed, the third gives

     $$t_c + \tau_c \leq t_j,$$

     and since $\tau_c > \varepsilon$, also the second constraint is met.

2. $\theta_{jc} = 1$: the two central inequalities are relaxed, the first implicates

$$t_j + \tau_j \leq t_c,$$

while the fourth is automatically satisfied.

Thus, overlapping is not possible and θ_{jc} governs the precedence between activities j and c.

- $\gamma_{jc} = 0$ and $\gamma_{cj} = 1$: in this case constraints (2.29)-(2.32) become:

$$t_j + \tau_j - t_c \leq M(1 - \theta_{jc}),$$

$$t_c - t_j \leq M\theta_{jc} - \varepsilon,$$

$$t_c + \tau_c - t_j \leq M\theta_{jc} + M,$$

$$t_j - t_c \leq M(1 - \theta_{jc}) + M - \varepsilon.$$

The last two relations are always relaxed. The other create two alternatives:

1. if $\theta_{jc} = 0$, then $t_c < t_j$ due to the second constraint, and the first is relaxed.
2. if $\theta_{jc} = 1$, first inequality implies $t_c \geq t_j + \tau_j$, and the second is relaxed.

This confirms that j can not left-overlap c (while the contrary may happen), which is consistent with the values of γ_{jc} and γ_{cj}.

- $\gamma_{jc} = 1$ and $\gamma_{cj} = 0$: this case is specular to the previous one. Constraints (2.29)-(2.32) imply:

$$t_j + \tau_j - t_c \leq M(1 - \theta_{jc}) + M,$$

$$t_c - t_j \leq M\theta_{jc} + M - \varepsilon,$$

$$t_c + \tau_c - t_j \leq M\theta_{jc},$$

$$t_j - t_c \leq M(1 - \theta_{jc}) - \varepsilon.$$

The first two relations are relaxed, while the others implicate:

1. $t_j \geq t_c + \tau_c$, if $\theta_{jc} = 0$.
2. $t_j < t_c$, if $\theta_{jc} = 1$.

As expected, given $\gamma_{cj} = 0$, it is not possible for activity c to left-overlap j.

- $\gamma_{jc} = 1$ and $\gamma_{cj} = 1$: all constraints are relaxed in such situation, being banally satisfied for every value of θ_{jc}. Solutions with $t_j = t_c$ are feasible, eventually admitting j to both left- and right-overlap c.

γ_{jc}	γ_{cj}	θ_{jc}	Meaning
0	0	0	no overlapping, c precedes j
		1	no overlapping, j precedes c
0	1	0	c starts before j
		1	no overlapping, j precedes c
1	0	0	no overlapping, c precedes j
		1	j starts before c
1	1	0	-
		1	-

Table 2.4. Possible cases and corresponding values of variables γ_{jc}, γ_{cj} and θ_{jc}, on the basis of constraints (2.29)-(2.32).

Thus, formulation (2.29)-(2.32) is equivalent to (2.19)-(2.22), but allows to remove variable θ_{cj} for each $\{j, c\} \in A^2$, resulting in a model with $m(m-1)$ less binary variables. From the computational point of view, this can generally contribute to make the solution process more efficient.

Table 2.4 collects our results. With the exception of the last rows, in which γ_{jc} and γ_{cj} are equal to 1 and $t_j = t_c$ is possible, the following holds:

$$t_j < t_c \Rightarrow \theta_{jc} = 1,$$

$$t_c < t_j \Rightarrow \theta_{jc} = 0.$$

This is the first step to start looking at θ_{jc} as the variable that may govern the start-to-start sequence between operations j and c, that is the natural extension of the basic concept of precedence to situations in which activities can be parallelized.

New formulation and capacity constraints

We have just shown a method to make feasible the substitution $\theta_{cj} = 1 - \theta_{jc}$, thus permitting to replace formulation (2.19)-(2.22) with (2.29)-(2.32) for each ordered pair of activities $\{j, c\}$. To complete this topic, we now consider also the capacity constraints and highlight some interesting aspects deriving from the combined use of inequalities (2.29)-(2.32) $\forall\{j, c\} \in A^2$ and (2.28).

We start by noticing that this new formulation constitutes a generalization of inequalities (2.3)-(2.4) for operations A1 and A2. Indeed, if we consider just one type of resource ($R = \{1\}$) with unitary capacity ($C_1 = 1$) and unitary request by each activity ($r_{11} = 1$, $r_{12} = 1$), (2.28) becomes:

$$1 + \gamma_{12} \leq 1,$$

$$1 + \gamma_{21} \leq 1.$$

This implies $\gamma_{12} = 0$ and $\gamma_{21} = 0$. Constraints (2.29)-(2.32) thus give:

$$t_1 + \tau_1 - t_2 \leq M(1 - \theta_{12}),$$
$$t_2 - t_1 \leq M\theta_{12} - \varepsilon,$$
$$t_2 + \tau_2 - t_1 \leq M\theta_{12},$$
$$t_1 - t_2 \leq M(1 - \theta_{12}) - \varepsilon.$$

It is evident that the first and third relations coincide with (2.3)-(2.4) where θ is replaced by θ_{12}, while the second and fourth inequalities become redundant. This confirms our idea.

Another interesting aspect of constraints (2.28), used together with (2.29)-(2.32), is that they explain the important role played by ε in inequalities (2.30) and (2.32). To understand this, let us first observe that if $t_j < t_c$ (equivalently $t_j \leq t_c - \varepsilon$) or $t_j > t_c$ ($t_j \geq t_c + \varepsilon$) then (2.30) and (2.32) are not influenced by the presence of terms $-\varepsilon$ and variables θ_{jc}, γ_{jc} and γ_{cj} are fixed according to t_j and t_c.

This means that the only situation in which ε becomes decisive is when $t_j = t_c$ (formally $|t_j - t_c| \leq \varepsilon$). Indeed, if t_j and t_c are equal, there are two possibilities depending on the value of θ_{jc}:

1. $\theta_{jc} = 0$: in this case constraint (2.30) imposes $\gamma_{jc} = 1$ and (2.31) $\gamma_{cj} = 1$.
2. $\theta_{jc} = 1$: (2.32) forces $\gamma_{cj} = 1$ and (2.29) implies $\gamma_{jc} = 1$.

Thus ε assures the following:

$$t_1 = t_2 \Rightarrow \begin{cases} \gamma_{jc} = 1, \\ \gamma_{cj} = 1. \end{cases}$$

On the contrary, if the term $-\varepsilon$ was removed from (2.30) and (2.32), then it would hold:

$$t_1 = t_2 \Rightarrow \gamma_{jc} + \gamma_{cj} \geq 1$$

(see constraints (2.29) and (2.31)).

This result would undermine the representativity of capacity constraints (2.28). To prove this fact, let us consider the case of activities in Figure 2.6, having all the same starting time ($t_1 = t_2 = t_3$). We suppose that there is only one shared resource with a total availability of 5 units. Furthermore we assume that each activity has a request equal to 2.

If we write formulation (2.28)-(2.32) for all involved pairs, we get the following:

$$t_1 + \tau_1 - t_2 \leq M(1 - \theta_{12}) + M\gamma_{12},$$
$$t_2 - t_1 \leq M\theta_{12} + M\gamma_{12} - \varepsilon,$$
$$t_2 + \tau_2 - t_1 \leq M\theta_{12} + M\gamma_{21},$$
$$t_1 - t_2 \leq M(1 - \theta_{12}) + M\gamma_{21} - \varepsilon,$$
$$t_1 + \tau_1 - t_3 \leq M(1 - \theta_{13}) + M\gamma_{13},$$
$$t_3 - t_1 \leq M\theta_{13} + M\gamma_{13} - \varepsilon,$$

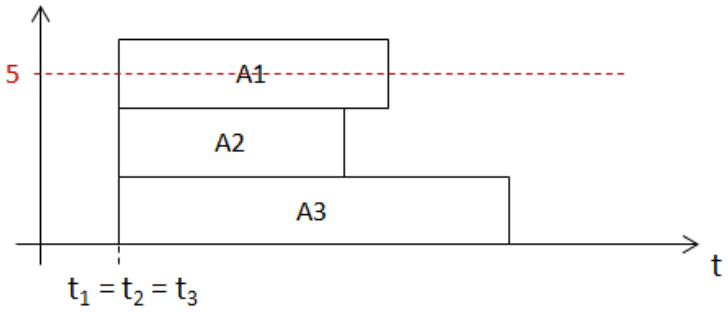

Figure 2.6. Example of three activities with the same starting time.

$$t_3 + \tau_3 - t_1 \leq M\theta_{13} + M\gamma_{31},$$

$$t_1 - t_3 \leq M(1 - \theta_{13}) + M\gamma_{31} - \varepsilon,$$

$$t_2 + \tau_2 - t_3 \leq M(1 - \theta_{23}) + M\gamma_{23},$$

$$t_3 - t_2 \leq M\theta_{23} + M\gamma_{23} - \varepsilon,$$

$$t_3 + \tau_3 - t_2 \leq M\theta_{23} + M\gamma_{32},$$

$$t_2 - t_3 \leq M(1 - \theta_{23}) + M\gamma_{32} - \varepsilon,$$

$$2 + 2\gamma_{21} + 2\gamma_{31} \leq 5,$$

$$2 + 2\gamma_{12} + 2\gamma_{32} \leq 5,$$

$$2 + 2\gamma_{13} + 2\gamma_{23} \leq 5.$$

Observe that solution of Figure 2.6 is not feasible, because the first twelve inequalities imply $\gamma_{jc} = 1$ and $\gamma_{cj} = 1$ for each $\{j, c\} \in A^2$ and therefore capacity constraints are not satisfied $(2 + 2 + 2 = 6 > 5)$.

However, if we had assumed $\varepsilon = 0$, then the following solution would have met all inequalities, with $t_1 = t_2 = t_3$: $(\gamma_{12} = 1,\ \gamma_{21} = 0,\ \gamma_{13} = 0,\ \gamma_{31} = 1,\ \gamma_{23} = 1,\ \gamma_{32} = 0,\ \theta_{12} = 1,\ \theta_{13} = 0,\ \theta_{23} = 1)$. But this would be wrong. This once again confirms that ε must be a small parameter arbitrarily greater than 0.

2.3.3 Parameters refinement

We have already mentioned the problems which are generally linked to the use of big-M formulations and the best practices commonly employed in the attempt of avoiding undesirable side effects. Among them, two are the main guidelines to follow: (i) keep model coefficients as low as possible and (ii) adapt them to associated constraints, eventually choosing different constants to appear in different places of different inequalities.

Taking into account these aspects, we split our discussion into two paragraphs to separately consider the coefficients of variables γ_{jc} or γ_{cj}, and those multiplying θ_{jc} or $(1 - \theta_{jc})$.

Coefficients for variables γ_{jc} and γ_{cj}

We start from the coefficients of overlapping variables, assuming the constants multiplying θ_{jc} and $(1-\theta_{jc})$ maintain generic large values – still referred as M – able to relax constraints. This is justified by a simple observation: if $\gamma_{jc}=0$ and $\gamma_{cj}=0$, inequalities (2.29)-(2.32) take the form of standard disjunctive relations. This imposes the coefficients of θ_{jc} and $(1-\theta_{jc})$ to be big enough to guarantee feasibility both if $\theta_{jc}=0$ and if $\theta_{jc}=1$ (so they may have to be very large to span long time horizons).

Let us introduce the following new formulation:

$$t_j+\tau_j-t_c\le M(1-\theta_{jc})+(\tau_j-\varepsilon)\gamma_{jc} \qquad \forall\{j,c\}\in A^2, \tag{2.33}$$

$$t_c-t_j\le M\theta_{jc}+\varepsilon\gamma_{jc}-\varepsilon \qquad \forall\{j,c\}\in A^2, \tag{2.34}$$

$$t_c+\tau_c-t_j\le M\theta_{jc}+\tau_c\gamma_{cj} \qquad \forall\{j,c\}\in A^2, \tag{2.35}$$

$$t_j-t_c\le M(1-\theta_{jc})-\varepsilon \qquad \forall\{j,c\}\in A^2. \tag{2.36}$$

It is possible to show that formulation (2.33)-(2.36) is equivalent to (2.29)-(2.32) for each $\{j,c\}\in A^2$. To prove this, we consider constraints (2.29)-(2.32), where four different parameters a_1, a_2, a_3, and a_4 are used in place of the original M multiplying γ_{jc} and γ_{cj}:

$$t_j+\tau_j-t_c\le M(1-\theta_{jc})+a_1\gamma_{jc}, \tag{2.37}$$

$$t_c-t_j\le M\theta_{jc}+a_2\gamma_{jc}-\varepsilon, \tag{2.38}$$

$$t_c+\tau_c-t_j\le M\theta_{jc}+a_3\gamma_{cj}, \tag{2.39}$$

$$t_j-t_c\le M(1-\theta_{jc})+a_4\gamma_{cj}-\varepsilon. \tag{2.40}$$

If $\gamma_{jc}=0$ and $\gamma_{cj}=0$ the result is trivial, taking us back to our initial observation: the terms containing new coefficients disappear from constraints (2.37)-(2.40), which simply reduce to standard disjunctions, exactly as done by (2.29)-(2.32). It follows that either j precedes c or c precedes j.

Therefore, we take into account all other cases in which at least one between γ_{jc} and γ_{cj} is different from 0:

- $\gamma_{jc}=0$ and $\gamma_{cj}=1$: since j can not left-overlap c ($\gamma_{jc}=0$) we would like to express the alternative between the following two:

$$t_c<t_j, \tag{2.41}$$

$$t_c\ge t_j+\tau_j. \tag{2.42}$$

  Depending on the value of θ_{jc}, we have:

  1. $\theta_{jc}=0$: (2.37) and (2.40) are relaxed. (2.38) and (2.39) give:

$$t_c-t_j\le-\varepsilon,$$

$$t_c-t_j\le a_3-\tau_c.$$

     The first is equivalent to (2.41). To be sure the second does not influence this condition, we must have $a_3-\tau_c\ge-\varepsilon$, thus:

$$a_3\ge\tau_c-\varepsilon. \tag{2.43}$$

2. $\theta_{jc} = 1$: (2.38) and (2.39) are relaxed. (2.37) and (2.40) respectively imply:

$$t_j - t_c \leq -\tau_j,$$

$$t_j - t_c \leq a_4 - \varepsilon.$$

The first is equivalent to (2.42). To allow this alternative without ties, a_4 must satisfy $a_4 - \varepsilon \geq -\tau_j$. Therefore:

$$a_4 \geq \varepsilon - \tau_j. \tag{2.44}$$

Summarizing, if (2.43) and (2.44) hold, $\gamma_{jc} = 0$, and $\gamma_{cj} = 1$, θ_{jc} expresses which of the two alternatives in (2.41)-(2.42) is true.

- $\gamma_{jc} = 1$ and $\gamma_{cj} = 0$: two alternatives should be made feasible to avoid c left-overlaps j:

$$t_j < t_c, \tag{2.45}$$

$$t_j \geq t_c + \tau_c. \tag{2.46}$$

We have the following, depending on the value of θ_{jc}:

1. $\theta_{jc} = 0$: (2.37) and (2.40) are relaxed. (2.38) and (2.39) give:

$$t_c - t_j \leq a_2 - \varepsilon,$$

$$t_c - t_j \leq -\tau_c.$$

The second is equivalent to (2.46). In order to avoid influences on this condition, it should be $a_2 - \varepsilon \geq -\tau_c$, thus:

$$a_2 \geq \varepsilon - \tau_c. \tag{2.47}$$

2. $\theta_{jc} = 1$: (2.38) and (2.39) are relaxed. (2.37) and (2.40) become:

$$t_j - t_c \leq a_1 - \tau_j,$$

$$t_j - t_c \leq -\varepsilon.$$

The second is equivalent to (2.45), and a_1 must satisfy $a_1 - \tau_j \geq -\varepsilon$, i.e.:

$$a_1 \geq \tau_j - \varepsilon. \tag{2.48}$$

Therefore, if (2.47) and (2.48) hold, $\gamma_{jc} = 1$, and $\gamma_{cj} = 0$, θ_{jc} permits to distinguish between the two alternatives in (2.45)-(2.46).

- $\gamma_{jc} = 1$ and $\gamma_{cj} = 1$: in this case overlapping is admitted and so there should not be constraints on activities starting times. We notice that, if $\theta_{jc} = 0$:
  - (2.37) is relaxed;
  - (2.38) gives: $t_c - t_j \leq a_2 - \varepsilon$;
  - (2.39) gives: $t_c - t_j \leq a_3 - \tau_c$;
  - (2.40) is relaxed;

thus:

$$t_c - t_j \leq \min(a_2 - \varepsilon, a_3 - \tau_c).$$

Correspondingly, if $\theta_{jc} = 1$:

- (2.37) gives: $t_c - t_j \geq \tau_j - a_1$;
- (2.38) is relaxed;
- (2.39) is relaxed;
- (2.40) gives: $t_c - t_j \geq \varepsilon - a_4$;

thus:

$$t_c - t_j \geq \max(\tau_j - a_1, \varepsilon - a_4),$$

or alternatively:

$$t_c - t_j > \max(\tau_j - a_1, \varepsilon - a_4) - \varepsilon.$$

Since we want $t_c - t_j$ to be allowed to vary over $\mathbb{R}$ (no constraints on starting times, considering both cases with $\theta_{jc} = 0$ and $\theta_{jc} = 1$), we can impose:

$$\min(a_2 - \varepsilon, a_3 - \tau_c) \geq \max(\tau_j - a_1, \varepsilon - a_4) - \varepsilon. \tag{2.49}$$

Combining condition (2.49) with (2.43),(2.44),(2.47) and (2.48), it is possible to write a system of inequalities in variables a_1, a_2, a_3, and a_4:

$$\begin{cases} a_1 \geq \tau_j - \varepsilon \\ a_2 \geq \varepsilon - \tau_c \\ a_3 \geq \tau_c - \varepsilon \\ a_4 \geq \varepsilon - \tau_j \\ a_1 + a_2 \geq \tau_j \\ a_2 + a_4 \geq \varepsilon \\ a_1 + a_3 \geq \tau_j + \tau_c - \varepsilon \\ a_3 + a_4 \geq \tau_c \end{cases}$$

This system admits infinitely many feasible solutions. Two particular possibilities are:

1. $a_1 = M,\ a_2 = M,\ a_3 = M,\ a_4 = M$;
2. $a_1 = \tau_j - \varepsilon,\ a_2 = \varepsilon,\ a_3 = \tau_c,\ a_4 = 0$.

They respectively constitute the combinations of coefficients used in formulation (2.29)-(2.32) and in (2.33)-(2.36). Notice that the second satisfies:

$$\min(a_2 - \varepsilon, a_3 - \tau_c) = \max(\tau_j - a_1, \varepsilon - a_4) - \varepsilon = 0.$$

Consequently, the following is guaranteed for all possible values of γ_{jc} and γ_{cj} (see also inequalities (2.34) and (2.36)):

$$t_c \leq t_j \Leftrightarrow \theta_{jc} = 0.$$

γ_{jc}	γ_{cj}	θ_{jc}	Meaning
0	0	0	no overlapping, c precedes j
		1	no overlapping, j precedes c
0	1	0	c starts before j
		1	no overlapping, j precedes c
1	0	0	no overlapping, c precedes j
		1	j starts before c
1	1	0	c starts before or together with j
		1	j starts before c

Table 2.5. Possible cases and corresponding values of variables γ_{jc}, γ_{cj} and θ_{jc}, on the basis of constraints (2.33)-(2.36).

Therefore θ_{jc} definitely assumes the role of controlling the start-to-start sequence between activities j and c. Table 2.5 extends Table 2.4 and reports updated results on the meaning of variables when new parameters are employed.

It is important to underline that the use of the reduced coefficients just presented can greatly contribute to speed up classical solvers procedures. We will give evidence of this fact in the next chapter, where computational results will provide a measure to substantiate our idea.

Coefficients for θ_{jc} and $(1 - \theta_{jc})$

Let us now try to improve the constants multiplying θ_{jc} and $(1 - \theta_{jc})$, assuming the other parameters fixed as in formulation (2.33)-(2.36). We introduce b_1, b_2, b_3 and b_4 in place of M, and consider the following:

$$t_j + \tau_j - t_c \leq b_1(1 - \theta_{jc}) + (\tau_1 - \varepsilon)\gamma_{jc}, \tag{2.50}$$
$$t_c - t_j \leq b_2\theta_{jc} + \varepsilon\gamma_{jc} - \varepsilon, \tag{2.51}$$
$$t_c + \tau_c - t_j \leq b_3\theta_{jc} + \tau_2\gamma_{cj}, \tag{2.52}$$
$$t_j - t_c \leq b_4(1 - \theta_{jc}) - \varepsilon. \tag{2.53}$$

We observe that:

1. When $\gamma_{jc} = 0$ and $\gamma_{cj} = 0$, θ_{jc} decides which activity precedes the other. Both possibilities must be feasible, therefore:

$$b_1 \geq t_j + \tau_j - t_c, \tag{2.54}$$
$$b_2 \geq t_c - t_j + \varepsilon, \tag{2.55}$$
$$b_3 \geq t_c + \tau_c - t_j, \tag{2.56}$$
$$b_4 \geq t_j - t_c + \varepsilon, \tag{2.57}$$

for all values of t_j and t_c.

2. Coefficients used for γ_{jc} and γ_{cj} are all nonnegative. Thus, $\gamma_{jc} = 1$ and/or $\gamma_{cj} = 1$ can only increase the right-hand sides of constraints (2.50)-(2.53). It follows that no additional conditions may tighten requirements (2.54)-(2.57), which indeed are the only ones to satisfy.

In order to reduce the values of b_1, b_2, b_3 and b_4 from the generic large number M, we consider the following options, presented in increasing order of detail:

- If a maximum time horizon H is given for the problem, since $t_j \geq 0 \; \forall j \in A$, it is possible to use:

$$\begin{aligned} b_1 &= H, \\ b_2 &= H - \tau_c + \varepsilon, \\ b_3 &= H, \\ b_4 &= H - \tau_j + \varepsilon. \end{aligned}$$

- If activities j and c have release dates r_j and r_c (nonnegative, 0 by default) and due dates d_j and d_c (satisfying $d_j \geq max(r_j, \tau_j)$ and $d_c \geq max(r_c, \tau_c)$, possibly set to H or M), one can employ:

$$\begin{aligned} b_1 &= d_j - r_c, \\ b_2 &= d_c - \tau_c - r_j + \varepsilon, \\ b_3 &= d_c - r_j, \\ b_4 &= d_j - \tau_j - r_c + \varepsilon. \end{aligned}$$

- In some cases, activities release times and due dates are not specified. However, it is possible that a different information is available. For example, if we know that a certain activity A2 has to necessarily follow another activity A1, we may assert that its starting time will not surely be smaller than τ_1. Similarly, if A3 comes after A2, it will of course satisfy: $t_3 > \tau_1 + \tau_2$. This reasoning is valid also the other way around. If for instance we are aware that the maximum time horizon is H, and that a given activity A4 precedes another activity A5, which must be processed before A6 and A7 can start, we may certainly affirm: $t_4 \leq H - \max\{\tau_6, \tau_7\} - \tau_5 - \tau_4$.
  We will soon see how this idea is easily generalized to take into account all the activities and all existing precedence relationships among them. For the moment, let us only define for every activity j the earliest start time ES_j and the latest start time LS_j, such that $[ES_j, LS_j]$ represents the time window for activity j to begin (i.e. $ES_j \leq t_j \leq LS_j$). The activity earliest finish time EF_j and its latest finish time LF_j can then be simply derived by adding the quantity τ_j respectively to ES_j and to LS_j.
  Since $ES_j \leq t_j \leq LS_j$ and $EF_j \leq t_j + \tau_j \leq LF_j$ for each $j \in A$, parameters multiplying θ_{jc} and $(1 - \theta_{jc})$ can be fixed in this way:

$$\begin{aligned} b_1 &= LF_j - ES_c, \\ b_2 &= LS_c - ES_j + \varepsilon, \\ b_3 &= LF_c - ES_j, \\ b_4 &= LS_j - ES_c + \varepsilon. \end{aligned}$$

These techniques are the most commonly used to moderate the sensitivity of MILP formulations to the large dimensions of big-M parameters in disjunctive scheduling constraints. Sometimes they are mixed, improved or adapted to be also valid for particular problems presenting different of the above characteristics or having special structures, both in terms of constraints and objective function.

In the following chapter we will come back to this topic, presenting an additional preprocessing step that may further contribute to tight the activities time windows. It is partially related to the characteristics of the specific problem that we will consider and for this reason we postpone the discussion until a proper formal introduction will be given. For the moment, in order to keep the notation simple and avoid over-detailed models, we generally use time horizon H in place of all parameters b_1, b_2, b_3 and b_4.

Chapter 3

A new formulation for the RCPSP

The resource constrained project scheduling problem (RCPSP) is one of the most widely studied problems in the context of scheduling (Artigues et al. (2013)). It consists of determining the starting times for the activities of a project considering both precedence relations and limited availabily of resources. The objective is to minimize the total duration of activities, i.e. the project makespan.

Formally, the RCPSP can be described as follows. Given m different activities and n distinct resources, let $A = \{1, ..., m\}$ and $R = \{1, ..., n\}$ represent the corresponding sets of indexes. The duration (or processing time) of a generic activity $j \in A$ is known and is denoted by τ_j. Once started, activities can not be interrupted (preemption is not allowed). So, if t_j represents the variable starting time of j, $t_j + \tau_j$ uniquely identifies the activity completion. Sequence constraints are considered to model technological finish-to-start precedence relations between couples of activities. These are given by the set Q of ordered index pairs, such that $(j_1, j_2) \in Q$ means that activity j_2 can not be started before j_1 is completed.

Activities require predefined amounts of resources to be processed. Each resource $i \in R$ is said renewable due to the fact that its full capacity, namely C_i, is assumed constantly available over time. The resource requirements are parameters of the problem, such that r_{ij} represents the number of units of resource i to be occupied by activity j during its processing, where $0 \leq r_{ij} \leq c_i$.

In the standard version of the RCPSP all inputs are considered deterministic integer-valued. Furthermore, a maximum time horizon H is supposed as given, being eventually set to the value of a precomputed upper bound or, by default, to the sum of all tasks durations. The output of the problem is a project schedule, i.e., a vector of starting times for the activities, such that all precedence relationships are respected, the total required quantity of each resource does not exceed its prescribed capacity at any point in time, and the overall makespan is minimized.

Graphically, the project is often associated to an *activity-on-node* directed network whose nodes correspond to activities and arcs represent precedence relations contained in Q. Two dummy nodes, namely 0 and $m + 1$, are generally added to the graph, representing the milestones ““project start”” and ““project end””. These dummy nodes can be seen as activities with zero duration and no resource

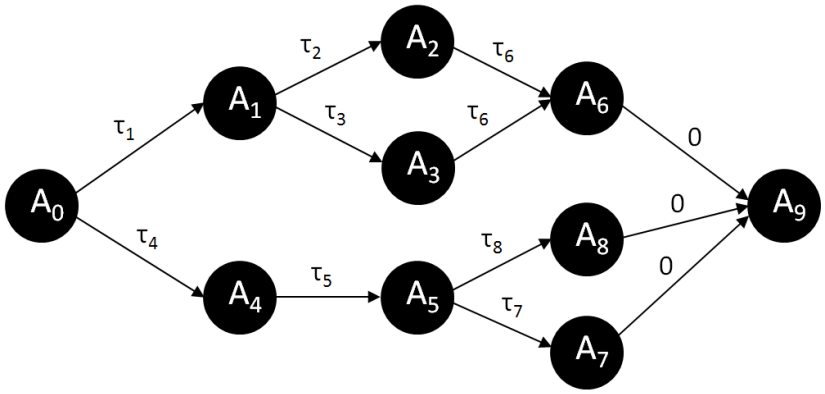

Figure 3.1. Example of a precedence activity-on-node graph.

requirements, connected to other tasks through additional arcs that make them the overall start and end points for the project. Since precedence constraints in input are supposed to be consistent, the network is considered acyclic by nature. As a result, the minimum project completion time can be determined as the length of the critical path, i.e., the longest path from 0 to $m + 1$, assuming each arc valued with its destination node processing time. A visual example involving 8 activities is shown in Figure 3.1, where A_0 and A_9 are the additional dummy tasks.

Notice that the algorithms for longest path calculation allow to polynomially compute not only the minimum project makespan, but also the earliest start time ES_j, for each activity $j \in A$. Indeed, every ES_j can be set to the length of the longest path from the start node 0 to j. Furthermore, integrating this method with one of the many existing heuristics for the RCPSP, it is possible to determine for each j the latest start time LS_j as the makespan upper bound value minus the length of the longest path from j to the end fictitious node. Then, the activities earliest finish time EF_j and latest finish time LF_j are simply derived, bringing to suitable time-windows that can be used to significantly improve the speed of solution procedures.

In this chapter we will focus on the mathematical formulation of the RCPSP. In particular, in Section 3.1 we review the existing literature, with a special attention to the linear programming approaches proposed over the years. In 3.2 we introduce a new formulation for the Resource Constrained Project Scheduling Problem based on the variables and constraints presented in the previous chapter. Section 3.3 recalls two known approaches in order to use them as a basis for comparison. A critical analysis on the characteristics of all the considered models is proposed in Section 3.4. Finally, experimental results are provided in 3.5.

3.1 Literature review

The RCPSP is an important and challenging problem both in scheduling research and in practice. Its applications involve many real-life situations, such as make-to-order production, construction projects, software development, service centers planning, etc.

The origin of the problem can be found in the field of project management, and in particular in the Critical Path Method. Indeed, the CPM is one of the first quantitative approaches to planning and scheduling. It takes into account only the durations of the activities, implicitly assuming the resources are unlimited. Extending such methodology, the RCPSP addresses similar issues, but considers the more realistic scenario in which resources are scarce, negatively influencing project execution time.

Blazewicz et al. (1983) demonstrated that the RCPSP belongs to the class of the strongly NP-hard problems, being a generalization of the classical job shop scheduling problem. Their theoretical conclusion is also confirmed by numerical evidence. In fact, according to Herroelen (2005), computational results indicate that exact methods applied to standard test instances (Kolisch and Sprecher (1997)) are unable to solve problems dealing with more than 60 activities. Even more recent works, such as the one by Kopanos et al. (2014), endorse this idea, showing considerable performance deterioration when addressing instances of increasing dimension, e.g. from 30 to 60 or 90 activities.

Due to its complexity, the RCPSP has motivated the study of solution techniques in different directions. Both approximate and exact methods have been extensively documented in the literature. On the one hand approximate approaches like heuristics and metaheuristics are able to obtain relatively good results with few computational requirements, but do not guarantee solutions optimality; on the other, exact techniques such as linear programming and ad-hoc branch and bound algorithms are successfully applied to problem instances of limited size, providing optimal or nearly-optimal certified solutions in a reasonable amount of time.

For an extensive comparison of computational results obtainable by approximate methods we refer to Hartmann and Kolisch (2000) and Kolisch and Hartmann (2006). More recent works following different approaches were also proposed (see for example Chen (2011), Jia and Seo (2013), Zamani (2013), Zheng and Wang (2015), and Chand et al. (2017)). For the exact solution strategies, specialized branch and bound procedures were described by Christofides et al. (1987), Demeulemeester and Herroelen (1992), Brucker et al. (1998), Baptiste and Pape (2000), Dorndorf et al. (2000), and Laborie (2005), while Koné et al. (2011), Bianco and Caramia (2013) and Kopanos et al. (2014) presented a detailed analysis on the performances of many well known linear programming models.

Although mathematical formulations in general cannot compete with the results of heuristics, metaheuristics and branch and bound algorithms, which are specifically built to exploit the structure of the problem during the solving steps, they are particularly interesting for different practical and theoretical reasons. First of all, compact mathematical models can be simply passed to standard optimization solvers, which are often the only software available to industrial practitioners, being consequently subject to continuous improvements. Secondly, they can contribute to the exploration of new lower bounds and the development of new hybrid approaches, involving effective constraint programming techniques (see Demassey et al. (2005)). Moreover, it is important to point out that there are some specific hard instances of the RCPSP for which linear programming formulations provided on average better results than specialized search algorithms.

Linear programming formulations for the RCPSP

Many authors proposed mathematical formulations for the RCPSP, all believing that how a model is formulated significantly influences obtainable results, in particular when integer variables are involved. In the literature, both ILP and MILP approaches are present in a variety of forms, depending on the different strategies chosen to express the objective function, the precedence constraints, and the resource constraints. These linear models can be classified according to two main characteristics:

Time granularity: There are two methods for considering time in RCPSP formulations. These allow to distinguish between *Discrete-Time* (DT) and *Continuous-Time* (CT) models. In DT models, the planning horizon is uniformly divided into a limited set of time slots, such that activities can start only at the beginning of one predefined slot. They are often said *Time-Indexed* due to the nature of binary variables involved in the formulations. Differently, in CT models, tasks may be scheduled at any point in time during the interval of interest, allowing much more modeling flexibility. This is a great advantage when dealing with RCPSP instances having long or very differentiated activity durations (see Koné et al. (2011)). However, when resources are very scarce, Continuous-Time approaches generally perform worse than Discrete-Time formulations due to their poor LP-relaxation bounds.

Number of variables and constraints: In general, there exist mathematical models for the RCPSP involving exponential, pseudo-polynomial or polynomial amounts of variables/constraints. *Exponential* formulations are based on the idea of using preprocessing procedures to generate subsets of activities that can be feasibly or unfeasibly parallelized according to their resource requirements. Such information can then be exploited to introduce suitable inequalities and/or binary decision variables associated to the calculated sets. These models generally lead to strong relaxations. However, since the number of sets resulting from the first phase in theory depends exponentially on m, the applicability of this type of formulation is restricted to those problem instances which are highly disjunctive (i.e. where many groups of activities cannot overlap, as observed by Baptiste and Pape (2000)) or involving relatively few activities.

In contrast to this type of approach, the other proposed methods do not require preprocessing logics, thus allowing the solution of problems directly through standard MILP solver. Among them, there are the formulations requiring a *pseudo-polynomial* number of variables and constraints, generally deriving from Discrete-Time strategies. Indeed, DT models involve time-indexed decision variables, such that the generic binary x_{jt} indicates a particular status of activity j at the start (or end) of time slot t. Consequently, the number of 0-1 variables increases with the amount of time intervals considered, and indirectly with the durations of the particular activities to schedule. In a similar way, also the amount of constraints is pseudo-polynomial, requiring to guarantee resource usage feasibility at every time t, with $t \in \{1, 2, ..., H\}$.

In the attempt to minimize the dimensions of models, and better address problem instances with long time horizons, different authors proposed compact

mathematical formulations for the RCPSP, involving a *polynomial* number of variables and constraints. Their modeling approach, which excludes the use of time-indexed variables, is the point that brought to the origin of the Continuous-Time models known in the literature.

All the works that have been proposed over the years to solve the RCPSP through linear programming techniques can be classified according to the above characteristics, including the first model, formulated by Pritsker et al. in 1969, and originally addressed to multi-project scheduling. In their Discrete-Time formulation, the authors introduce a single type of binary variables x_{jt}, such that $x_{jt} = 1$ if and only if activity j finishes at time t. Specific constraints guarantee that exactly one period is selected for the completion of each activity j. An upper limit H on project maximum duration is assumed available as input. Constraints and variables are both defined according to the time index t with $t \in \{1, 2, ..., H\}$, leading to a pure ILP formulation of pseudo-polynomial size. The objective function can essentially be written as $\sum_{t=1,...,H} t x_{m+1,t}$, with $x_{m+1,t}$, $t = 1, ..., H$, being the variables associated to the dummy activity indicating project end.

Following the same approach, Christofides et al. (1987) presented a similar model, and only reformulated the precedence constraints in a disaggregated form, which strongly improved relaxation results.

Alvarez-Valdés and Tamarit (1993) proposed a MILP Continuous-Time formulation of the RCPSP, making use of the concept of minimal incompatible sets. These are the smallest subsets of activities, not linked by any precedence relation, which could not be parallelized without exceeding the maximum amount of resources available. Minimal incompatible sets are computed in preprocessing. Then, binary variables y_{jk} are associated to pairs of activities (j, k), which equal 1 whenever j is processed before k in the scheduling. In order to avoid resource conflicts, many inequalities are introduced, forcing at least one precedence relationship (binary equal to 1) between a pair of activities in every incompatible set I (that is $\sum_{j \in I, k \in I} y_{jk} \geq 1$ for each I). This possibly generates an exponential number of constraints. Technological sequence restrictions are simply modeled by means of natural starting time continuous variables t_j $\forall j \in \{0, ..., m + 1\}$, and the makespan is optimized by minimizing t_{m+1}. Variables y_{jk} and t_j are linked through big-M constraints such that $y_{jk} = 1 \Rightarrow t_k \geq t_j + \tau_j$.

In a complementary way, Mingozzi et al. (1998) proposed a Discrete-Time formulation that, instead of considering forbidden sets of activities, exploits preprocessing to list all feasible overlapping combinations, i.e. all possible subsets of activities $F_1, F_2, ..., F_k$ that can be simultaneously executed without violating resource or precedence constraints. Boolean variables z_{ht} are defined, assuming the value of 1 to indicate that subset F_h is in progress at time t. Suitable constraints ensure that at most one set is selected for every $t \in \{1, ..., H\}$, and that each activity j is processed for exactly τ_j periods. The objective function and the precedence constraints are modeled similarly to Pritsker et al. (1969), using the additional variables x_{jt}. An exponential number of variables and constraints are required, depending on the amount of calculated feasible subsets. This limits the applicability of the formulation, that however produces very strong lower bounds on the makespan.

Different approaches were used by Klein (2000), who introduced three pseudo-

polynomial Discrete-Time formulations. The first one is a variant of the model proposed by Kaplan (1988), originally developed for the case of preemptive activities. The second and the third are instead based on step binary variables specifying if an activity j starts (completes) at time period t or earlier (after). The last formulation is weaker than the others, but presents the advantage of letting the τ_j eventually be decision variables and not necessarily fixed parameters.

Artigues et al. (2003) were the first to formulate the RCPSP using a polynomial number of variables and constraints. Their Continuous-Time model is built on the idea of resource flow, represented by the variables f_{jki}, which indicate the quantity of resource i that is directly transferred from activity j to activity k for processing. Dummy start and end nodes are respectively treated as unique source and sink for the flow, that is $r_{i,0} = r_{i,m+1} = C_i$. Other two types of variables are defined, similarly to Alvarez-Valdés and Tamarit (1993): the continuous natural starting time variables t_j and the sequential binary variables y_{jk}, equal to 1 whenever activity j precedes activity k. This formulation has a poor relaxation due to the presence of big-M constraints, but is compact and therefore adequate in case of large time horizons.

The work of Artigues et al. (2003) was studied by Koné et al. (2011), who compared different Discrete-Time and Continuous-Time approaches from the literature, including two new CT event-based models. Events are significant points in the schedule which can be selected as start (or end) time for one or more activities. Their position over the planning horizon is unknown a priori, so a continuous variable t_e is introduced to represent the date of each event e, considering that at most $m + 1$ variables of this type are needed.

The first formulation, said *Start/End* or *SEE*, is a variant of the event-based formulation proposed by Zapata et al. (2008). It uses two binary variables to indicate if activities start or complete at given event points, and two continuous variables: one is t_e, while the other specifies the amount of resources required immediately after each event. The second model, known as *On/Off* or *OOE*, outperforms the first one, involving also a smaller number of variables. These are given by the continuous event dates plus m^2 booleans, which equal 1 if an activity j starts, ends or is in process at event e, and are 0 otherwise.

Since the number of necessary events is limited, these formulations are compact, and show, as expected, the best results on instances involving long scheduling horizons or high processing time range. However, they have two disadvantages: the first is that, although not including big-M constraints, these models provide low-quality linear relaxations. The second instead concerns the dimension, involving $O(m^3)$ constraints, that makes the solution impractical for high values of m.

Bianco and Caramia (2013) proposed a Discrete-Time formulation, successively detailed by Naber et al. (2014). Their model is based on the definition of step binary variables similar to those introduced by Klein (2000) plus additional continuous variables representing the percentage of activity j that has been performed until each time interval t. Before starting the activity percentage is set to 0, and becomes definitely 1 after completion. Authors extensively compared their approach with others in the literature, showing that their formulation generally outperformed the others in terms of solution time and quality. This once again proved that the way constraints and additional redundant variables are introduced and formulated do

have significant effects on the efficiency of exact methods.

Inspired by the work of Hartmann (2000), who focused on the similarity between packing problems and project scheduling models, Jia and Seo (2013) proposed a Continuous-Time formulation applying the facility layout concept to RCPSP. In this approach, activities are viewed as rectangular facilities to be arranged in a limited space which is $(n + 1)$-dimensional (n for different resource types and 1 for time), without violating precedence relations and capacity constraints. The presented formulation is obtained by introducing many different variables which allow to linearize disjunctive inequalities. A further step of constraint reduction is necessarily added, due to the significant dimensions reached by the model, involving also big-M inequalities.

One of the most recent comparative studies among MILP and ILP formulations for the RCPSP was proposed by Kopanos et al. (2014), who evaluated the performance of several approaches from the literature, including four new models. Among these four, two are Discrete-Time formulations which use binary variables similar to those introduced by Pritsker et al. (1969) and Kaplan (1988), with the second one considering a disaggregated version of precedence constraints, and two are Continuous-Time models, exploiting the concept of activity overlapping. Different preprocessing logics are applied, with the aim of simplifying constraints and avoiding the use of large big-M constants. A detailed computational analysis shows the results of the approaches considered, and underlines the efficiency of the latter two CT models, which can either be solved to optimality faster or, if not optimal, attain better solutions.

Recently, a Continuous-Time formulation for the RCPSP with generalized temporal constraints (i.e. with minimal and maximal time lags between pairs of activities) was presented by Varakantham et al. (2016), which employed it as a basis to solve nondeterministic instances with durational uncertainty. Their theoretical model is similar to the CT approaches introduced by Kopanos et al. (2014), using the same starting time and binary indicator variables, connected with big-M inequalities. Capacity constraints are formulated in a different way through the definition of additional variables identifying the resource consumption by an activity at the start of another activity. No computational results are reported for the solution of the problem in deterministic conditions utilizing standard solvers.

Finally, Rihm and Trautmann (2017) proposed a Continuous-Time formulation with three minor variants. Their approach consists in modeling resource constraints through the use of explicit assignment variables that specify which individual resource units are allocated to each activity. Natural starting time variables t_j and sequencing binary variables y_{jk} are also introduced (linked by big-M inequalities), allowing to express that if two activities j and k employ the same resource unit, then either j precedes k ($y_{jk} = 1$) or the contrary happens ($y_{kj} = 1$). Modifications include the elimination of some symmetric solutions from the search space and the simplification of sequencing constraints for pairs of activities that cannot be processed in parallel. A comparative analysis with the formulations of Pritsker et al. (1969) and Artigues et al. (2003) is performed, showing positive results specially for instances of the RCPSP with very low resource capacities.

This completes the list of works constituting the state of the art for MILP and ILP approaches to the classical Resource Constrained Project Scheduling Problem. It is

important to observe, however, that many variants and extensions of the RCPSP have been proposed in the literature over the years (see Brucker et al. (1999), Hartmann and Briskorn (2010), and Schwindt et al. (2015)). Among the others, we can cite the multi-mode RCPSP (MRCPSP, surveyed by Weglarz et al. (2011)), where activities can be processed with several possible modes requiring different resource quantities, the preemptive RCPSP (PRCPSP, introduced by Kaplan (1988)), where activity preemption is allowed, the stochastic RCPSP (SRCPSP, Herroelen and Leus (2005)), the RCPSP with generalized precedence constraints (RCPSP/max, Bartusch et al. (1988)), with consumption and production of resources (RCPSP/CPR, Koné et al. (2013)), with flexible resource profiles (FRCPSP, Naber and Kolisch (2014)) and the multi-objective RCPSP (MORCPSP, Ballestín and Blanco (2011)).

Problem instance sets for the RCPSP and its extensions

In line with the high volume of works regarding the RCPSP and the associated solution methods, a large number of standard instances were proposed in the literature. Among the most broadly used, there are those that we will consider for our computational studies and that we rapidly describe here:

PSPLib: these problem instances are described in Kolisch and Sprecher (1997) and publicly reachable through a website (http://www.om-db.wi.tum.de/psplib/). They are divided into four sets, varying on the number of activities involved, i.e., 30, 60, 90 or 120. Each of the first three sets (respectively named j30, j60 and j90) contains 480 instances, randomly generated in groups of 10, by considering all the combinations of three indicators: The Network Complexity $NC \in \{1.5, 1.8, 2.1\}$, the Resource Factor $RF \in \{0.25, 0.50, 0.75, 1\}$, and the Resource Strength $RS \in \{0.2, 0.5, 0.7, 1\}$. NC represents the average quantity of non-redundant precedence relationships per activity, RF is the normalized average number of required resources, and RS indicates the tightness of resource capacity constraints, with $0 \leq RS \leq 1$. The set j120, which considers 120 activities and 600 instances, is structured in a similar way, but involves different levels of RS (in detail, $RS \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$). Four resource types are taken into account in all problems. Today, PSPLib is the most famous test library for the RCPSP.

RanGen2: there are 1800 instances in this set, generated by Vanhoucke et al. (2008) using the combinations of six topological indicators, different from those of PSPLib. They are formally split into five groups, containing respectively 900, 180, 240, 240, and 240 test problems. Material is available at http://www.projectmanagement.ugent.be/research/data/RanGen. Despite having the same dimensions of j30 (i.e, 30 activities and 4 resource types), these instances are generally considered more difficult to be solved.

In our computational analysis we will use set RanGen2 and sets j30 and j60 of PSPLib, performing experiments on a total of 2760 problems.

Additional instances are available at the mentioned websites, involving also larger test cases and more complicated versions.

3.2 Our solution approach

As illustrated, many MILP and ILP models for the RCPSP were proposed in the literature, providing interesting basis for comparison. The extensive experiments performed by Koné et al. (2011) and Kopanos et al. (2014), in particular, showed the encouraging results obtained by Continuous-Time models. On the one hand, Koné et al. (2011) compared their two CT models with the formulations of Pritsker et al. (1969), Christofides et al. (1987) and Artigues et al. (2003), underlining the advantages of continuous approaches in solving problem instances with long time horizons. On the other, Kopanos et al. (2014) evaluated their two discrete-time and two continuous-time models, using as benchmark the formulations of Koné et al. (2011) and those previously introduced by Pritsker et al. (1969), Christofides et al. (1987), Kaplan (1988) (extended by Klein (2000)), Klein (2000) and Artigues et al. (2003). Collected computational results show their continuous-time models to perform better than other approaches, further motivating the research in this direction.

In addition to these considerations, we notice another important practical aspect. In discrete-time systems, activities may only start and complete at predefined time points. As a result, processing times are implicitly forced to be discrete, according to the granularity established for the scheduling horizon. In many cases this assumption is appropriate, allowing to well represent the actual problem, but there are some situations for which this setting would lead to unrealistic hypothesis. As an example, we can cite the industrial application presented by Pinto and Grossmann (1996), where activity durations vary from 0.783 to 11.250 days. In order to tackle this class of problems with DT approaches, only two possibilities are available: either one rounds the processing times of activities, consequently accepting approximate solutions, or significantly reduces the dimension of time slots, possibly obtaining a formulation with an impractical number of variables and constraints. By contrast, CT approaches easily overcome these issues, allowing the definition of compact models without increments in size.

Considering these observations, we propose in the following a new Continuous-Time formulation for the RCPSP based on the concepts presented in previous chapters. Some remarks on the computational behavior of this model are then analyzed in 3.2.2.

3.2.1 A new continuous-time formulation for the RCPSP

We consider m non-preemptive activities and n different resource types, respectively represented by the sets of indexes $A = \{1, ..., m\}$ and $R = \{1, ..., n\}$. Let τ_j be the duration of a generic activity $j \in A$, C_i be the capacity available for resource $i \in R$, and r_{ij} be the number of units of i required by j to be executed. We define the set of activity pairs $A^2 = \{\{j, c\} : j \in A, c \in A, c \neq j\}$. Precedence relations are given as input and considered in the set $Q \subset A \times A$. The time horizon is an upper bound for project makespan and is set by default to the sum of activities processing time ($H = \sum_{j \in A} \tau_j$). We introduce four types of variables:

- t_j : starting time of activity j, with $t_j \geq 0, \ \forall \ j \in A$;

- w : continuous project makespan, $w \geq 0$;
- θ_{jc} : binary sequencing variable, $\forall \{j,c\} \in A^2$;
- γ_{jc}, γ_{cj} : binary left-overlapping variables, $\forall \{j,c\} \in A^2$;

Our MILP Continuous-Time formulation is given by the following (F1):

(F1) :

$$\min \ w \tag{3.1}$$

$$w \geq t_j + \tau_j \qquad \forall j \in A, \tag{3.2}$$

$$t_c \geq t_j + \tau_j \qquad \forall (j,c) \in Q, \tag{3.3}$$

$$t_j + \tau_j - t_c \leq H(1-\theta_{jc}) + (\tau_j - \varepsilon)\gamma_{jc} \qquad \forall \{j,c\} \in A^2, \tag{3.4}$$

$$t_c - t_j \leq H\theta_{jc} + \varepsilon\gamma_{jc} - \varepsilon \qquad \forall \{j,c\} \in A^2, \tag{3.5}$$

$$t_c + \tau_c - t_j \leq H\theta_{jc} + \tau_c\gamma_{cj} \qquad \forall \{j,c\} \in A^2, \tag{3.6}$$

$$t_j - t_c \leq H(1-\theta_{jc}) - \varepsilon \qquad \forall \{j,c\} \in A^2, \tag{3.7}$$

$$r_{ic} + \sum_{\substack{j \in A \\ j \neq c}} r_{ij}\gamma_{jc} \leq C_i \qquad \forall c \in A, \forall i \in R, \tag{3.8}$$

$$\theta_{jc} \in \{0,1\} \qquad \forall \{j,c\} \in A^2, \tag{3.9}$$

$$\gamma_{jc} \in \{0,1\} \qquad \forall \{j,c\} \in A^2, \tag{3.10}$$

$$\gamma_{cj} \in \{0,1\} \qquad \forall \{j,c\} \in A^2, \tag{3.11}$$

$$t_j \geq 0 \qquad \forall j \in A, \tag{3.12}$$

$$w \geq 0, \tag{3.13}$$

where ε is a small positive parameter satisfying $\varepsilon < \tau_j \ \forall j \in A$. In particular, being the durations integer parameters, we may use $\varepsilon = 0.1$.

The objective (3.1) is the minimization of project makespan w, which is defined by inequalities (3.2). Precedence relations (3.3) assure the activities to be properly sequenced. Constraints (3.4) and (3.5) together guarantee $t_j \leq t_c < t_j + \tau_j \Rightarrow \gamma_{jc} = 1$, while (3.6) and (3.7) provide $t_c \leq t_j < t_c + \tau_c \Rightarrow \gamma_{cj} = 1$. Capacity constraints are given by (3.8), and assure that resource availability limits are never exceeded over time (i.e. at every activity start). Finally, (3.9)-(3.13) set variables domain.

Observe that inequalities (3.4)-(3.7) make use of the reduced parameters analyzed in Section 2.3.3, i.e. $\tau_j - \varepsilon$, ε, and τ_c.

Notice, moreover, that no dummy activities are considered in the formulation, which involves $\frac{3}{2}m(m-1)$ binary variables, $m+1$ continuous variables, and $2m(m-1) + mn + 2m + |Q| + 1$ constraints.

3.2.2 Computational considerations

Formulation (3.1)-(3.13) presents some characteristics able to justify interesting computational advantages. To show this fact, we take a step back and consider two

additional models, that do not take into account respectively constraints reduction and parameters refinement proposed in Sections 2.3.1 and 2.3.3. They are given by:

(F2) :

$$\min\ w \tag{3.14}$$

$$w \geq t_j + \tau_j \qquad \forall j \in A, \tag{3.15}$$

$$t_c \geq t_j + \tau_j \qquad \forall (j,c) \in Q, \tag{3.16}$$

$$t_j + \tau_j - t_c \leq H(1-\theta_{jc}) + (\tau_j - \varepsilon)\gamma_{jc} \qquad \forall \{j,c\} \in A^2, \tag{3.17}$$

$$t_c - t_j \leq H\theta_{jc} + \varepsilon\gamma_{jc} - \varepsilon \qquad \forall \{j,c\} \in A^2, \tag{3.18}$$

$$t_c + \tau_c - t_j \leq H\theta_{jc} + \tau_c\gamma_{cj} \qquad \forall \{j,c\} \in A^2, \tag{3.19}$$

$$t_j - t_c \leq H(1-\theta_{jc}) - \varepsilon \qquad \forall \{j,c\} \in A^2, \tag{3.20}$$

$$-H(1-\gamma_{jc}) + \varepsilon \leq t_j + \tau_j - t_c \qquad \forall \{j,c\} \in A^2, \tag{3.21}$$

$$-H(1-\gamma_{jc}) \leq t_c - t_j \qquad \forall \{j,c\} \in A^2, \tag{3.22}$$

$$-H(1-\gamma_{cj}) + \varepsilon \leq t_c + \tau_c - t_j \qquad \forall \{j,c\} \in A^2, \tag{3.23}$$

$$-H(1-\gamma_{cj}) \leq t_j - t_c \qquad \forall \{j,c\} \in A^2, \tag{3.24}$$

$$r_{ic} + \sum_{\substack{j \in A \\ j \neq c}} r_{ij}\gamma_{jc} \leq C_i \qquad \forall c \in A, \forall i \in R, \tag{3.25}$$

$$\theta_{jc} \in \{0,1\} \qquad \forall \{j,c\} \in A^2, \tag{3.26}$$

$$\gamma_{jc} \in \{0,1\} \qquad \forall \{j,c\} \in A^2, \tag{3.27}$$

$$\gamma_{cj} \in \{0,1\} \qquad \forall \{j,c\} \in A^2, \tag{3.28}$$

$$t_j \geq 0 \qquad \forall j \in A, \tag{3.29}$$

$$w \geq 0. \tag{3.30}$$

(F3) :

$$\min\ w \tag{3.31}$$

$$w \geq t_j + \tau_j \qquad \forall j \in A, \tag{3.32}$$

$$t_c \geq t_j + \tau_j \qquad \forall (j,c) \in Q, \tag{3.33}$$

$$t_j + \tau_j - t_c \leq H(1-\theta_{jc}) + H\gamma_{jc} \qquad \forall \{j,c\} \in A^2, \tag{3.34}$$

$$t_c - t_j \leq H\theta_{jc} + H\gamma_{jc} - \varepsilon \qquad \forall \{j,c\} \in A^2, \tag{3.35}$$

$$t_c + \tau_c - t_j \leq H\theta_{jc} + H\gamma_{cj} \qquad \forall \{j,c\} \in A^2, \tag{3.36}$$

$$t_j - t_c \leq H(1-\theta_{jc}) + H\gamma_{cj} - \varepsilon \qquad \forall \{j,c\} \in A^2, \tag{3.37}$$

$$r_{ic} + \sum_{\substack{j \in A \\ j \neq c}} r_{ij}\gamma_{jc} \leq C_i \qquad \forall c \in A, \forall i \in R, \tag{3.38}$$

$$\theta_{jc} \in \{0,1\} \qquad \forall\{j,c\} \in A^2, \tag{3.39}$$

$$\gamma_{jc} \in \{0,1\} \qquad \forall\{j,c\} \in A^2, \tag{3.40}$$

$$\gamma_{cj} \in \{0,1\} \qquad \forall\{j,c\} \in A^2, \tag{3.41}$$

$$t_j \geq 0 \qquad \forall j \in A, \tag{3.42}$$

$$w \geq 0, \tag{3.43}$$

Formulation F2 is equivalent to F1, but involves the additional sufficient constraints (3.21)-(3.24), such that double implications hold:

$$t_j \leq t_c < t_j + \tau_j \Leftrightarrow \gamma_{jc} = 1,$$

$$t_c \leq t_j < t_c + \tau_c \Leftrightarrow \gamma_{cj} = 1.$$

Formulation F3 is identical to F1 where inequalities (3.4)-(3.7) have been substituted with (3.34)-(3.37). This way the time horizon H is used as multiplier of γ_{jc} and γ_{cj} in place of reduced parameters $\tau_j - \varepsilon,\ \varepsilon,\ \tau_c,\ 0$.

Although F1, F2 and F3 provide the same optimal value, they can admit different solutions. In general, given the minimal makespan $w^* \leq H$, the number of distinct vectors $t \in \mathbb{R}^m$ such that all constraints of F1, F2, or F3 are satisfied, is equal for all formulations. However, this is not true for the integer variables. Indeed, if we indicate with $I_{Fx}(t,w)$ the set of feasible values for binary variables γ and θ for some fixed t and w in a generic formulation Fx (i.e. F1, F2 or F3), we can write the following:

$$I_{F2}(t,w) \subseteq I_{F1}(t,w) \subseteq I_{F3}(t,w).$$

This can be easily shown by ignoring for a moment resource constraints (3.8), (3.25) and (3.38), and focusing on a single pair of activities $\{j,c\}$. If j and c are not linked by precedence relations (either explicit or induced by the other ordered pairs in Q), they can generally be scheduled in different ways. We consider the following alternatives, also depicted in Figure 3.2:

(a) $t_j + \tau_j \leq t_c$;

(b) $t_j < t_c < t_j + \tau_j$;

(c) $t_j = t_c$;

(d) $t_c < t_j < t_c + \tau_c$;

(e) $t_c + \tau_c \leq t_j$.

Assuming t_j and t_c are fixed, only one of these situations can be true. Table 3.1 reports feasible values of γ_{jc}, γ_{cj} and θ_{jc} for each case and each formulation, supposing capacity constraints have no effect.

Thus, if we consider all pairs $\{j,c\} \in A^2$, assume a given vector $t \in \mathbb{R}^m$, and ignore resource availability limits, we may observe that:

1. F1 admits fewer solutions than F3, due to the use of reduced parameters.

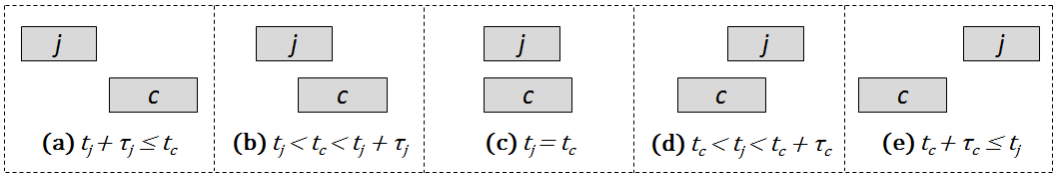

Figure 3.2. Possible disposition alternatives for two generic activities j and c.

$(\gamma_{jc}, \gamma_{cj}, \theta_{jc})$	(a)			(b)			(c)			(d)			(e)		
	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3
(0, 0, 0)													•	•	•
(0, 0, 1)	•	•	•												
(0, 1, 0)										•	•	•	•		•
(0, 1, 1)	•		•												
(1, 0, 0)													•		•
(1, 0, 1)	•		•	•	•	•									
(1, 1, 0)			•			•	•	•	•	•		•	•		•
(1, 1, 1)	•		•	•		•			•			•			•

Table 3.1. Feasible values (•) of γ_{jc}, γ_{cj} and θ_{jc} for all starting times alternatives and all formulations, ignoring resource constraints.

2. F2 admits fewer solutions than F1 and F3, thanks to the existence of necessary and sufficient conditions.

3. F2 admits exactly one feasible solution, for the combined effect of reduced parameters and sufficient constraints. This unique solution is always feasible also for F1 and F3.

4. F3 possible solutions are the same of F1 plus some additional ones with $\gamma_{jc} = 1, \gamma_{cj} = 1$ and different θ_{jc}, for at least one pair $\{j, c\} \in A^2$.

5. If a solution is feasible for F1 but not for F2, it certainly has the same value of θ of the unique F2 solution and at least one pair $\{j, c\} \in A^2$ for which γ_{jc}, γ_{cj}, or both are set to 1 instead of 0.

These observations must be reconsidered to take into account resource capacity limits. We notice that (3.8), (3.25) and (3.38) involve only variables γ, which appear on the left-hand sides of constraints and are multiplied by nonnegative parameters. Consequently, the only effect generated from these inequalities can be to lower the values of γ_{jc} and/or γ_{cj}, for at least one pair $\{j, c\} \in A^2$. It follows that:

$$I_{F2}(t, w) \subseteq I_{F1}(t, w) \subseteq I_{F3}(t, w),$$

and $I_{F2}(t, w) = I_{F1}(t, w) = I_{F3}(t, w)$ for very tight resource constraints.

Furthermore, indicating with D_{Fx} the feasible space of the generic formulation Fx, we can write:

$$D_{F2} \subseteq D_{F1} \subseteq D_{F3}.$$

Indeed, if $s = (t, w, \gamma, \theta)$ is a feasible solution for F2, then $s \in D_{F1}$, since all constraints in F1 are obviously met. Moreover, $s \in D_{F1}$ implicates $s \in D_{F3}$, because conditions (3.34)-(3.37) are banally satisfied for s respecting $(3.4) - (3.7)$ and $H \geq \tau_j \ \forall j \in A$.

Experimental results

In general, the fact that the feasible space of a MILP model is tighter than others, is an advantage for exact solving techniques, which are based on integer variables branching schemes. However, we can notice that the narrowest formulation, i.e. F2, involves $2m(m-1)$ constraints more than F1 and F3, and this can possibly compromise expected efficiency. The algorithm will indeed have to inspect a lower number of nodes, but the time spent to solve the linear relaxation of each node will probably increase with the amount of inequalities. As a consequence, it is generally not obvious to understand which is the best choice. For this reason, we have tested all the proposed formulations on the 480 instances contained in the set j30 of PSPLib.

The code has been implemented in JAVA using ILOG CPLEX Concert Technology 12.8.0. The experiments are performed under standard configurations on an Intel Core i7 CPU 5600U 2.60 GHz with 16GB RAM. We have set the time limit for each problem to 300 CPU seconds. Table 3.2 summarizes our computational results, where:

Feas.% indicates the percentage of instances that provided a feasible integer solution (optimal, suboptimal, or non-proven optimal) within the established maximum time.

Opt.% represents the percentage of instances for which the optimal solution could be found within the time limit.

Out.Gap% is the average gap of the integer non-proven optimal solutions from the best lower bounds calculated by the solver during the branching (as output by CPLEX at computation end).

Act.Gap% specifies the average gap of the integer non-proven optimal solutions from the actual optimal solutions (available for set j30 of PSPLib).

Time(sec.) is the average number of CPU seconds required by problem instances solved to proven optimality.

Formulation	Feas.%	Opt.%	Out.Gap%	Act.Gap%	Time(sec.)
F1	100.00	96.88	14.27	0.96	9.64
F2	100.00	96.25	14.97	0.79	10.88
F3	100.00	91.04	9.67	0.80	11.43

Table 3.2. Computational results for F1, F2 and F3 on j30 instances.

As shown, F1 has the best performances in terms of optimal solutions found and computation time spent for the search, while F3 seems to be the worst. Percentage gaps are different, but are influenced by the results of solutions which could be proven as optimal by one model and not by another. The same is true also for the average seconds. Thus, in order to better investigate the differences between formulations, we propose the detailed analysis of Tables 3.3 and 3.4, where F1 is independently compared with F2 and F3. The tables are split into four parts to separately consider the subsets of instances that could be solved to optimality by both models, by none of the two, or by one formulation and not by the other. The bottom left corner is always empty because neither F2 nor F3 was able to find the optimal value of any problem that was not solved by F1.

		F2	
		Optimal	**Sub/Non-proven optimal**
F1	**Optimal**	Instances: 462 (96.25%) F1: Out.Gap%: 0.00, Act.Gap%: 0.00, Time(sec.): 8.26, Avg.Nodes: 8864 F2: Out.Gap%: 0.00, Act.Gap%: 0.00, Time(sec.): 10.88, Avg.Nodes: 7550	Instances: 3 (0.62%) F1: Out.Gap%: 0.00, Act.Gap%: 0.00, Time(sec.): 221.70, Avg.Nodes: 335873 F2: Out.Gap%: 10.17, Act.Gap%: 0.00, Time(sec.): 300.00, Avg.Nodes: 239877
	Sub/Non-proven optimal	Instances: 0 (0.00%) –	Instances: 15 (3.13%) F1: Out.Gap%: 14.27, Act.Gap%: 0.96, Time(sec.): 300.00, Avg.Nodes: 261134 F2: Out.Gap%: 15.93, Act.Gap%: 0.95, Time(sec.): 300.00, Avg.Nodes: 181207

Table 3.3. Results comparison between formulations F1 and F2

Looking firstly at Table 3.3, we can notice again that formulation F1 reaches the best results, solving three instances more than F2. When both F1 and F2 find the optimum, F1 is on average faster than F2 (8.26 vs. 10.88 seconds). When both methods reach the time limit without proving the optimality, F1 returns a lower output gap, while the actual gaps are almost the same for F1 and F2. The three instances solved by F1 but not by F2 prove to be hard: F1 employs on average 221.70 seconds to find the optimal solutions, while F2 achieves the same values

without closing the algorithm gap (indeed Out.Gap% is 10.17, but Act.Gap% is 0.00). Another interesting observation is provided by the average number of nodes visited during the branching (Avg.Nodes), which is always higher for formulation F1 than for F2. On the one hand this is in accord with the fact that F2 faces a feasible space of lower dimension (when both models find the optimum, Avg.Nodes of F1 is 8864 and Avg.Nodes of F2 is 7550), on the other this evidences that the simplex solution at each node is faster for F1 (as expected, since the number of constraints is much smaller). Considering the 15 instances not solved by both methods, F1 visits about the 150% of nodes visited by F2 in the same amount of time of 300 seconds: this is probably due to the combined effect of a more efficient branching and a faster node relaxation computation.

Table 3.4 reports the result of the direct comparison between formulation F1 and F3. F1 is significantly better than F3, finding the optimum for 28 instances more. When both methods reach optimality, F3 is considerably slower than F1, visiting about twice the number of nodes and spending on average the 168% of seconds more than F1. For the instances which could not be solved by any model the result is similar, with F1 terminating with better percentage gaps than F3 (14.27 vs. 17.73 for the output gap and 0.96 vs. 1.60 for the actual gap).

		F3	
		Optimal	**Sub/Non-proven optimal**
F1	**Optimal**	Instances: 437 (91.04%) F1: Out.Gap%: 0.00, Act.Gap%: 0.00, Time(sec.): 4.26, Avg.Nodes: 5466 F3: Out.Gap%: 0.00, Act.Gap%: 0.00, Time(sec.): 11.43, Avg.Nodes: 10983	Instances: 28 (5.83%) F1: Out.Gap%: 0.00, Act.Gap%: 0.00, Time(sec.): 93.49, Avg.Nodes: 96931 F3: Out.Gap%: 5.35, Act.Gap%: 0.37, Time(sec.): 300.00, Avg.Nodes: 231913
	Sub/Non-proven optimal	Instances: 0 (0.00%) – –	Instances: 15 (3.13%) F1: Out.Gap%: 14.27, Act.Gap%: 0.96, Time(sec.): 300.00, Avg.Nodes: 261134 F3: Out.Gap%: 17.73, Act.Gap%: 1.60, Time(sec.): 300.00, Avg.Nodes: 214334

Table 3.4. Results comparison between formulations F1 and F3

Further investigation of performances

Summarizing what said until now, the comparisons provided in Tables 3.3 and 3.4 evidenced the effectiveness of formulation F1 with respect to the alternative models F2 and F3. In particular, F2 performed slightly worse than F1 due to the trade-off between the reduction of nodes to be visited and the higher computational requirement at each node. F3 was instead outperformed by F1 on several aspects. This suggests that using big-M parameters as small as possible (i.e., $\tau_j - \varepsilon$, ε, τ_c, instead of the large H) brings a remarkable effect, that is probably not justified by the only exclusion of some isomorphic solutions from the feasible region.

To better comprehend the reason of such proved advantage, we further investigated the behavior of F1 and F3 during the solving procedure. We ran some additional experiments disabling all the advanced methods provided by CPLEX (i.e., preprocessing presolve, heuristics, generation of cuts and dynamic search) in order to implement classical branch & bound logics. Table 3.5 collects our results that, although less efficient, still evidence the good performances of the first formulation with respect to the third one (F1 solves 65 instances more than F3, spending on average more than 10 seconds less). Also F2 KPIs are reported for completeness.

Formulation	Feas.%	Opt.%	Out.Gap%	Act.Gap%	Time(sec.)
F1	100.00	91.67	12.83	2.04	15.87
F2	100.00	89.38	12.30	1.89	17.53
F3	100.00	78.13	7.06	1.78	26.14

Table 3.5. Computational results for F1, F2 and F3 on j30 instances with CPLEX advanced methods disabled.

Once obtained this confirmation, we have analyzed in detail the branching steps executed by the algorithm when F1 is used. Two are the main observations to report, which can justify its advantage:

1. When relaxation is solved, in particular at the first node but in general at every point of the procedure, there is a preference in assigning fractional values to variables θ rather than to the γ. This is due to the fact that variables γ are multiplied by small parameters, being at the same time tightened by capacity constraints (3.8). Differently, variables θ have as coefficients the relative big number H. It follows that in most cases the linear relaxation can return a solution which, satisfying inequalities (3.2) and (3.3), has many non-integer values for the θ_{jc} and many zeros among the γ_{jc} and γ_{cj}. This way the resource constraints are easily met, and (3.4)-(3.7) are also respected (with several active) without moving the overlapping variables from 0. It must be considered indeed that the contribution which could be provided by γ_{jc} and γ_{cj} to satisfy (3.4)-(3.7), eventually assuming values close to 1, is often marginal, due to the influence of reduced multipliers.

2. The above effect is progressively mitigated as the search becomes deeper and more binaries are fixed (the feasible region gradually shrinks). However, it

plays an important role, particularly in the first phases. Indeed, having a typical relaxation solution the great majority of θ_{jc} set to fractional values and almost all the γ_{jc} and γ_{cj} equal to 0, the algorithm is automatically driven to branch first on the sequencing variables. This is advantageous because, thanks to the use of small coefficients, it contributes to rapidly reduce the size of the feasible space. For example, if, for a certain pair $\{j, c\}$, θ_{jc} is fixed to 1, the big parameter H disappears from the corresponding inequalities like (3.4) and (3.7); if instead θ_{jc} is set to 0, relations (3.5) and (3.6) are drastically tightened for the considered couple of activities. Thus, two of the four inequalities regarding pair $\{j, c\}$ are narrowed at each step, giving a direct indication to the continuous variables t_j and t_c. This helps the branching procedure to find improved lower bounds and to prune suboptimal nodes from the search. Moreover, assuming the branching on a given θ_{jc} happens at a certain distance from the root node, the corresponding decisions for the γ_{jc} and γ_{cj} will probably take place at deeper levels of the tree, where they will be actually pushed to take a value different from 0 only in case of overlapping (which is generally possible but less probable that simple sequencing, at least for not too cumulative instances). As a consequence, additional branchings to round fractional values of γ can frequently be avoided.

Notice that these considerations do not apply in the same way to formulation F3. Indeed, being the overlapping variables multiplied by H, it is simpler for the linear relaxation procedure to find some values for γ which at the same time are *(i)* not uniquely zero, *(ii)* high enough to "help" the θ in satisfying (3.17)-(3.20), and *(iii)* sufficiently low to meet resource constraints (3.21). Furthermore, also forcing the algorithm to branch on θ with higher priority, it is not possible to get the same tightening effect as for formulation F1 (small parameters are replaced by H), thus losing the associated benefits.

Figure 3.3 shows the branching tree explored by the algorithm, using formulation F1, to solve a very simple but representative example involving five activities and one resource type. Problem data and an optimal solution are reported in Table 3.6. Nodes in the tree are distinguished by color. The root is gray while children can either be black or white. Black nodes are obtained from their parent by fixing (to 0 or 1) the value of variable θ_{jc}, for a certain pair $\{j, c\} \in A^2$. White nodes are instead associated to branchings on variables γ_{jc} and γ_{cj}. Four numbers appear next to each node: the first two indicate the amounts of components in vectors θ and γ fixed until the considered point. The others specify the percentages of fractional values resulting from the solution of linear relaxation respectively associated to θ or γ. For instance, the node on the extreme left of the figure has 4 variables fixed, all belonging to the sequencing vector θ. The relaxation is solved considering only the other variables, which returns a solution composed of 84% integers and 16% fractional values. Among these, the 8% are non-integer θ_{jc}, and 8% non-integer γ_{jc} or γ_{cj}. Lower is the percentage of fractional overlapping variables with respect to the others, higher in general the probability that a θ_{jc} is selected for branching (ignoring selection strategies).

It is evident from this representation that the search is driven by variables θ in the first decisions (see the black nodes close to the root). Then, as the algorithm

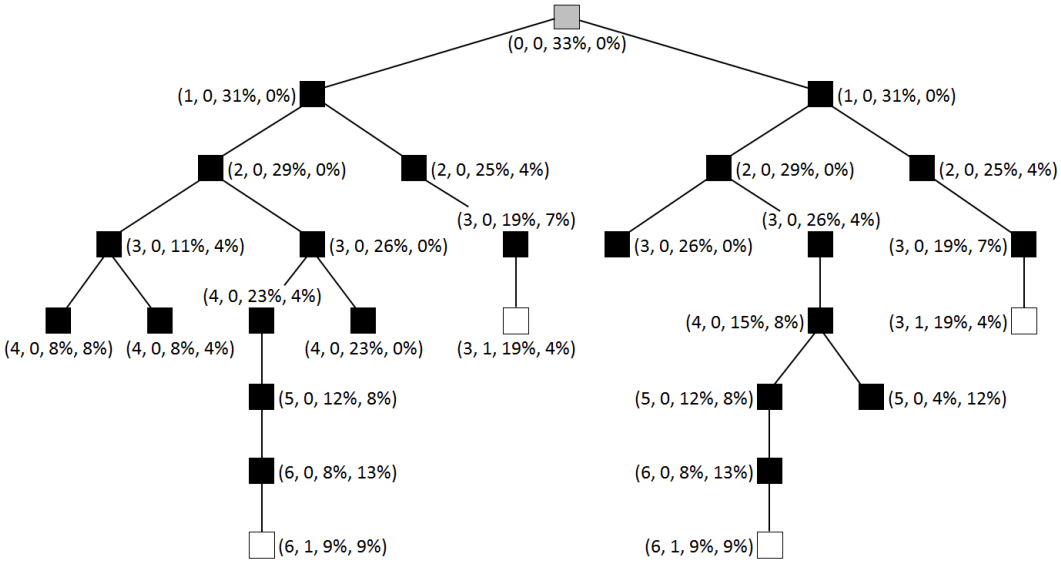

Figure 3.3. Example of a branching tree generated while solving the simple instance of Table 3.6 using formulation F1. Black and white colors respectively indicate if the node was created by branching on a θ or a γ variable. Node labels follow the format: (# fixed θ, # fixed γ, $\frac{\text{\# fractional } \theta \text{ from relaxation}}{\text{\# variables - \# fixed } \theta \text{ - \# fixed } \gamma}$, $\frac{\text{\# fractional } \gamma \text{ from relaxation}}{\text{\# variables - \# fixed } \theta \text{ - \# fixed } \gamma}$).

moves toward deeper levels, also variables γ are used to explore the space (look at white nodes at the end of branches), rapidly leading to fully integer solutions.

In the following, we will come back to this topic to show that the branching properties of our model constitute a factor of advantage also for comparisons with the existing literature. We introduce for this purpose the CT formulations proposed by Kopanos et al. (2014).

3.3 The approaches of Kopanos, Kyriakidis, and Georgiadis

As already mentioned in Section 3.1, in 2014, Kopanos, Kyriakidis, and Georgiadis proposed two Discrete-Time (namely Kop-DT1 and Kop-DT2) and two Continuous-Time (Kop-CT1 and Kop-CT2) formulations for the RCPSP. They compared their models with several existing ILP and MILP approaches, underlining the particular capacity of Kop-CT1 and Kop-CT2 to produce better results and find high quality solutions, either intended as optimal or very close to optimality.

In order to address a valid comparison with these two CT models, we present now all the details of their formulation.

Formulation Kop-CT1

Formulation Kop-CT1 uses two types of continuous variables and binary variables. They are:

- t_c : starting time of activity c, with $t_c \geq 0,\ \forall\ c \in (A \cup \{m+1\})$.

$Q = \{(1,2)\}$; $C_1 = 2$;

Activity j	τ_j	r_{1j}	t_j^*	θ_{j2}^*	θ_{j3}^*	θ_{j4}^*	θ_{j5}^*	γ_{j1}^*	γ_{j2}^*	γ_{j3}^*	γ_{j4}^*	γ_{j5}^*
1	1	1	9	1	0	0	0	-	0	1	0	0
2	2	1	10	-	0	0	0	0	-	0	0	0
3	3	1	9	-	-	0	0	1	1	-	0	0
4	4	2	5	-	-	-	0	0	0	0	-	0
5	5	2	0	-	-	-	-	0	0	0	0	-

Table 3.6. Example data and solution for a simple instance with five activities and one resource type.

- f_c : finishing time of activity c, with $f_c \geq 0,\ \forall\ c \in (A \cup \{m+1\})$.
- x_{cj} : binary variable to define the sequence between any pair of activities (c, j). If c and j cannot be processed in parallel, then $x_{cj} = 1 \Leftrightarrow f_c \leq t_j$. Otherwise, $x_{cj} = 1 \Leftrightarrow t_c \leq t_j$ with $c < j$, and $x_{cj} = 1 \Leftrightarrow t_c < t_j$ with $c > j$.
- z_{jc} : binary variable indicating the relation between the finishing time of activity j and the start of activity c, such that: $f_j > t_c \Rightarrow z_{jc} = 1$. If $f_j \leq t_c$, then z_{jc} can either be 0 or 1. The z_{jc} were originally called "overlapping" variables by Kopanos, Kyriakidis, and Georgiadis, but we avoid this name in order to avoid confusion with our variable γ.

The sets for the indexes of variables x_{cj} and z_{jc} are different and are determined after a preprocessing phase. They are described below the formulation, which is given by the following:

(Kop-CT1) :

$$\min\ f_{m+1} \tag{3.44}$$

$$f_c = t_c + \tau_c \qquad \forall c \in (A \cup \{m+1\}), \tag{3.45}$$

$$f_c \leq t_j \qquad \forall (c,j) \in (A \times (A \cup \{m+1\})) : (c,j) \in K, \tag{3.46}$$

$$f_c \leq t_j + (LF_c - ES_j)x_{jc} \qquad \forall (c,j) \in S : c \neq j, \tag{3.47}$$

$$x_{cj} + x_{jc} = 1 \qquad \forall (c,j) \in (B \setminus K') : c > j, \tag{3.48}$$

$$t_j \leq t_c + (LS_j - ES_c)x_{cj} \qquad \forall (c,j) \in P : c > j, \tag{3.49}$$

$$t_c + \lambda \leq t_j + (LS_c - ES_j + \lambda)x_{jc} \qquad \forall (c,j) \in P : c > j, \tag{3.50}$$

$$f_j - t_c \leq (LF_j - ES_c)z_{jc} \qquad \forall (c,j) \in P : c \neq j, \tag{3.51}$$

$$r_{ic} + \sum_{\substack{j \neq c\ :\ r_{ij} > 0 \\ (c,j) \in P}} r_{ij}(z_{jc} - x_{cj}) \leq C_i \qquad \forall c \in A, \forall i \in R : r_{ic} > 0, \tag{3.52}$$

$$x_{cj} \leq z_{jc} \qquad \forall (c,j) \in P : c \neq j, \tag{3.53}$$

$$x_{jc} = 1 \qquad \forall (c,j) \in P, c \neq j : LS_j < ES_c, \tag{3.54}$$

$$t_c \geq 0 \qquad \forall c \in (A \cup \{m+1\}), \quad (3.55)$$

$$f_c \geq 0 \qquad \forall c \in (A \cup \{m+1\}), \quad (3.56)$$

$$x_{cj} \in \{0,1\} \qquad \forall (c,j) \in (B \setminus K') : c \neq j, \quad (3.57)$$

$$z_{jc} \in \{0,1\} \qquad \forall (c,j) \in P : c \neq j, \quad (3.58)$$

where:

- ES_c, LS_c, EF_c, and LF_c are parameters representing the earliest and latest start and end times (generically said time-windows) for each activity $c \in A$. They are computed in preprocessing using the critical path method and a parallel scheduling heuristic with two competing rules for upper bound calculation.
- λ is a small positive parameter, with $0 < \lambda < 1$ for integer activity durations.
- K, K', S, P, B, and G are sets of activities pairs to be calculated before running the model. In particular:

  B contains pairs of activities $(c,j) \in A \times A$ which share at least one resource of the same type.

  G is composed of pairs of activities $(c,j) \in B$ that cannot overlap due to resource limitations, i.e. because $r_{ic} + r_{ij} > C_i$ for at least one resource type $i \in R$.

  K consists of pairs of activities $(c,j) \in A \times A$ for which precedence is known because (c,j) belongs to the transitive closure of Q or because $LF_c \leq ES_j$.

  K' includes the pairs of activities $(c,j) \in K$ and their symmetrical (j,c). No binary variables x_{cj}, z_{cj}, and z_{jc} are considered for pairs in this set.

  S is given by $G \setminus K'$. It contains pairs of activities that cannot be executed in parallel due to their excessive resource requirements, excluding those with known sequence relations. Variables z_{cj} and z_{jc} are not defined for pairs in this set.

  P can be calculated as $B \setminus (G \cup K')$. It includes pairs of activities that can overlap. All binary variables (x_{cj}, z_{cj}, and z_{jc}) are defined for pairs $(c,j) \in P$.

In formulation Kop-CT1, the makespan is optimized by minimizing the finishing time of the dummy activity $m+1$ (see objective (3.44)), assumed to be a successor for all the other activities of the project. Equations (3.45) set the connection between activities finishing and starting time. Inequalities (3.46) impose precedence relations, while (3.47) are standard disjunctive constraints for activities that cannot be processed in parallel. Equations (3.48) state that one and exactly one among x_{cj} and x_{jc} must be 1. Constraints (3.49)-(3.51) are defined for all pairs of activities subject to possible overlapping and contribute to specify the meaning of the considered binary variables. If $t_j > t_c$, then $x_{cj} = 1$, and if $f_j > t_c$, then $z_{jc} = 1$. Moreover, if two activities start at the same time ($t_c = t_j$), parameter λ assures that the precedence is arbitrarily given to the activity with the lower index (i.e., $x_{jc} = 1$ and $x_{cj} = 0$

for $j < c$). Resource capacity limitations for every resource $i \in R$ are modeled by inequalities (3.52), where the nonnegative term $(z_{jc} - x_{cj})$ is used to identify the activities j which are in progress at time instant t_c, for each activity $c \in A$. Indeed, $z_{jc} - x_{cj} = 1$ if and only if $z_{jc} = 1$ and $x_{cj} = 0$. This is *necessarily* implicated by $t_j < t_c < f_j$ when $c < j$, and by $t_j \leq t_c < f_j$ when $c > j$. It follows that resource conflicts involving two or more activities that start at the same time are actually caught by the inequalities corresponding to the equivalent-start activity having the higher index. Conditions (3.53) and (3.54) are already implicated by (3.45) and (3.49)-(3.51), but help in tightening the feasible region. Finally, (3.55)-(3.58) define variables domain.

Formulation Kop-CT2

The second formulation proposed by Kopanos et al. (2014) can be obtained from Kop-CT1 using the substitution $x_{cj} = 1 - x_{jc}$ for $c > j$, previously considered by constraint (3.48). In this way, the number of precedence variables x is reduced by one half. Employing the same notation as before, Kop-CT2 can be written as follows:

(Kop-CT2) :

$$\min \ f_{m+1} \tag{3.59}$$

$$f_j = t_j + \tau_j \qquad \forall j \in (A \cup \{m+1\}), \tag{3.60}$$

$$f_j \leq t_c \qquad \forall (j,c) \in (A \times (A \cup \{m+1\})) : (j,c) \in K, \tag{3.61}$$

$$f_j \leq t_c + (LF_j - ES_c)(1 - x_{jc}) \qquad \forall (c,j) \in S : c > j, \tag{3.62}$$

$$f_c \leq t_j + (LF_c - ES_j)x_{jc} \qquad \forall (c,j) \in S : c > j, \tag{3.63}$$

$$t_j \leq t_c + (LS_j - ES_c)(1 - x_{jc}) \qquad \forall (c,j) \in P : c > j, \tag{3.64}$$

$$t_c + \lambda \leq t_j + (LS_c - ES_j + \lambda)x_{jc} \qquad \forall (c,j) \in P : c > j, \tag{3.65}$$

$$f_j - t_c \leq (LF_j - ES_c)z_{jc} \qquad \forall (c,j) \in P : c \neq j, \tag{3.66}$$

$$r_{ic} + \sum_{\substack{j<c \,:\, r_{ij}>0 \\ (c,j)\in P}} r_{ij}(z_{jc} + x_{jc} - 1) + \sum_{\substack{j>c \,:\, r_{ij}>0 \\ (c,j)\in P}} r_{ij}(z_{jc} - x_{cj}) \leq C_i \qquad \forall c \in A, \forall i \in R : r_{ic} > 0, \tag{3.67}$$

$$x_{jc} \leq z_{cj} \qquad \forall (c,j) \in P : c > j, \tag{3.68}$$

$$1 - x_{jc} \leq z_{jc} \qquad \forall (c,j) \in P : c > j, \tag{3.69}$$

$$x_{jc} = 1 \qquad \forall (c,j) \in P, c > j : LS_j < ES_c, \tag{3.70}$$

$$t_c \geq 0 \qquad \forall c \in (A \cup \{m+1\}), \tag{3.71}$$

$$f_c \geq 0 \qquad \forall c \in (A \cup \{m+1\}), \tag{3.72}$$

$$x_{jc} \in \{0,1\} \qquad \forall (c,j) \in (B \setminus K') : c > j, \tag{3.73}$$

$$z_{jc} \in \{0,1\} \qquad \forall (c,j) \in P : c \neq j, \tag{3.74}$$

where:

- (3.59)-(3.61) are exactly the same as (3.44)-(3.46).
- (3.62)-(3.63) are equivalent to (3.47), with the substitution $x_{cj} = 1 - x_{jc}$.
- (3.64) corresponds to (3.49), with $x_{cj} = 1 - x_{jc}$.
- (3.65)-(3.66) are the same as (3.50)-(3.51).
- (3.67) is equivalent to (3.52), written in a different way to apply the substitution $x_{cj} = 1 - x_{jc}$.
- (3.68)-(3.69) correspond to (3.53), with $x_{cj} = 1 - x_{jc}$.
- (3.70) imposes the same condition of (3.54) for the remaining variables (having $c > j$).

3.4 Models comparison

The aim of this section is to give evidence of the efficiency of our new proposed model, also providing a valid comparison with the existing literature. In particular, our study tries to look at the problem from different points of view, finally analyzing the computational results obtained by our formulation when compared with the approaches of Kopanos et al. (2014) just introduced.

In general, there are several reasons that motivate our choice to use models Kop-CT1 and Kop-CT2 as a basis for comparison. First of all, they are Continuous-Time formulations, as the one proposed in this work. Consequently, they can theoretically be applied in the same contexts, having the common advantages of allowing to model situations in which the activity durations are arbitrarily large and possibly non-integer valued, without incurring into excessive problem dimensions and approximation errors. Moreover, in the study of Kopanos et al. (2014), authors extensively compared their results with those of almost all existing CT and DT models from the literature, thus proving the good performances obtained by their approaches, and in particular by formulations Kop-CT1 and Kop-CT2. They were the latest to present such a comprehensive computational analysis for the RCPSP and also for this reason we refer to their work as a valid benchmark for comparison.

Among the more recent proposals, only Rihm and Trautmann (2017) experimentally validated their MILP approach, reporting the differences of results with the two (one DT and one CT) state-of-the-art models of Pritsker et al. (1969) and Artigues et al. (2003). Their assignment-based formulation shows good performances for instances of the RCPSP with long time horizons or particularly tight resource limitations, but is generally overcome by the model of Pritsker et al. (1969) on the standard j30 and j60 problem sets of PSPLib. This is probably caused by the relative high number of decision variables required to represent the explicit assignment of individual resources to activities.

Indeed, it must be noticed that formulations involving a lower amount of variables often (but not always) reveal also better performances, due to the reduced size of the tree to be potentially explored by the branching procedures. Considering problem dimensions, models Kop-CT1 and Kop-CT2 undoubtedly constitute a valid option

compared to the other Continuous-Time compact formulations from the literature, as briefly summarized in Table 3.7.

Formulation	Number of constraints	Continuous variables	Binary variables
Artigues et al. (2003)	$\mathcal{O}(m^3 + m^2 n)$	$\mathcal{O}(m^2 n)$	$\mathcal{O}(m^2)$
Koné et al. (2011)	$\mathcal{O}(m^3 + mn)$	$\mathcal{O}(m)$	$\mathcal{O}(m^2)$
Jia and Seo (2013)	$\mathcal{O}(m^2 n)$	$\mathcal{O}(m^2 n)$	$\mathcal{O}(m^2 n)$
Kopanos et al. (2014)	$\mathcal{O}(m^2 + mn)$	$\mathcal{O}(m)$	$\mathcal{O}(m^2)$
Varakantham et al. (2016)	$\mathcal{O}(m^2 n)$	$\mathcal{O}(m^2 n)$	$\mathcal{O}(m^2)$
Rihm and Trautmann (2017)	$\mathcal{O}(lm^2 n)$	$\mathcal{O}(m)$	$\mathcal{O}(m^2 + lmn)$

Table 3.7. Approximated amounts of variables and constraints for all the Continuous-Time compact formulations proposed in the literature, considering m activities, n different resource types, and an average number of individual resources for each type equal to l.

In addition, we conclude by observing the last two interesting aspects of addressing a comparison with the work of Kopanos et al. (2014). The first is the theoretical possibility to consider the case of continuously divisible resources, as for example energy, liquids, raw materials, etc... Formulations Kop-CT1 and Kop-CT2 would in general be applicable to these situations, but the same would not be true for the assignment-based model of Rihm and Trautmann (2017). Secondly, we dispel a doubt which has maybe caught the attention of the reader: our new formulation and those proposed by Kopanos et al. have some similarities. We thus try to exploit this fact to compare the models in detail, show the main differences, and evidence the strengths of our proposal.

This section is articulated in two parts: the first (3.4.1) is mainly focused on the theoretical aspects, the second (3.4.2) on the practical implications.

3.4.1 Similarities and differences between formulations

The first evident difference between our proposed formulation and those of Kopanos et al. is that F1 does not require any initial procedure and can thus be directly passed to a standard solver for solution, while Kop-CT1 and Kop-CT2 involve two preprocessing phases respectively for the calculation of time-windows and the generation of suitable sets of activity pairs. Although these techniques may significantly contribute to improve the achievable performances, reducing the number of variables, constraints and the feasible region of the problem, on the other hand they make the solution process more complex, involving the development of several pieces of code (i.e., for critical path method, upper bounding heuristics, and sets computation) in addition to the classical model formulation.

The main scope of our work is to present a new general mathematical model for the RCPSP to be easily passed to commercial solvers for solution. In this spirit, we think that the best way to provide a valid comparison with the literature is

to consider the absence of preprocessing logics as a common requirement. For this reason, we propose two modifications to the formulations of Kopanos et al., namely Kop-CT1-m and Kop-CT2-m, which exclude time-windows and specific sets computations. We will use these simplified models in place of the original ones for our theoretical and practical analysis.

However, before starting a detailed comparison, we add the last important consideration. It is possible to verify that both the preprocessing approaches proposed by Kopanos et al., could be equivalently applied to our formulation, offering an opportunity of further improvements. In particular, the use of precalculated sets of pairs could reduce the number of variables and constraints in formulations F1 proportionally to Kop-CT1 and Kop-CT2. Activities time-windows, instead, could be used to reduce the coefficients H in F1 as they actually avoid big-M parameters in the models of Kopanos et al.. In this case, however, there is not an obvious expectation of the different impacts on the models under study. For this reason and for further addressing a valid comparison, in Section 3.4.2 we will also provide the results of some runs executed for all models considering the activities time-windows (i.e., the quantities ES_c, LS_c, EF_c, and LF_c, for each activity $c \in A$) as known.

Modified formulations Kop-CT1-m and Kop-CT2-m

In order to simplify the original Kop-CT1 and Kop-CT2 and propose two compact self-contained formulations, we make different assumptions. These can be interpreted as "worst case" adaptations of the models, since they correspond to the conditions which would eventually be met when preprocessing procedures failed to provide any consistent contribution to reduce the problem size. In particular, we consider the following:

- $B = A \times A$.
- $G = \emptyset$.
- $K = Q \cup (A \times \{m+1\})$.
- $K' = \emptyset$.
- $S = \emptyset$.
- $P = A \times A$.
- $LS_{\jmath^1} - ES_{\jmath^2} = H,\ \forall(\jmath^1, \jmath^2) \in A \times A$.
- $LS_{\jmath^1} - ES_{\jmath^2} + \lambda = H,\ \forall(\jmath^1, \jmath^2) \in A \times A$.
- $LF_{\jmath^1} - ES_{\jmath^2} = H,\ \forall(\jmath^1, \jmath^2) \in A \times A$.
- In general $r_{ic} \geq 0$, with few/no cases satisfying $r_{ic} = 0,\ \forall c \in A, i \in R$.

Formulations Kop-CT1-m and Kop-CT2-m consequently become:

(Kop-CT1-m) :

$$\min \ f_{m+1} \tag{3.75}$$

$$f_c = t_c + \tau_c \qquad \forall c \in (A \cup \{m+1\}), \tag{3.76}$$

$$f_c \leq t_j \qquad \forall (c,j) \in Q \cup (A \times \{m+1\}), \tag{3.77}$$

$$x_{cj} + x_{jc} = 1 \qquad \forall (c,j) \in (A \times A), c > j, \tag{3.78}$$

$$t_j \leq t_c + H x_{cj} \qquad \forall (c,j) \in (A \times A) : c > j, \tag{3.79}$$

$$t_c + \lambda \leq t_j + H x_{jc} \qquad \forall (c,j) \in (A \times A) : c > j, \tag{3.80}$$

$$f_j - t_c \leq H z_{jc} \qquad \forall (c,j) \in (A \times A) : c \neq j, \tag{3.81}$$

$$r_{ic} + \sum_{\substack{j \in A \\ j \neq c}} r_{ij}(z_{jc} - x_{cj}) \leq C_i \qquad \forall c \in A, \forall i \in R, \tag{3.82}$$

$$x_{cj} \leq z_{jc} \qquad \forall (c,j) \in (A \times A) : c \neq j, \tag{3.83}$$

$$t_c \geq 0 \qquad \forall c \in (A \cup \{m+1\}), \tag{3.84}$$

$$f_c \geq 0 \qquad \forall c \in (A \cup \{m+1\}), \tag{3.85}$$

$$x_{cj} \in \{0,1\} \qquad \forall (c,j) \in (A \times A) : c \neq j, \tag{3.86}$$

$$z_{jc} \in \{0,1\} \qquad \forall (c,j) \in (A \times A) : c \neq j. \tag{3.87}$$

(Kop-CT2-m) :

$$\min \ f_{m+1} \tag{3.88}$$

$$f_j = t_j + \tau_j \qquad \forall j \in (A \cup \{m+1\}), \tag{3.89}$$

$$f_j \leq t_c \qquad \forall (j,c) \in Q \cup (A \times \{m+1\}), \tag{3.90}$$

$$t_j \leq t_c + H(1 - x_{jc}) \qquad \forall (c,j) \in (A \times A) : c > j, \tag{3.91}$$

$$t_c + \lambda \leq t_j + H x_{jc} \qquad \forall (c,j) \in (A \times A) : c > j, \tag{3.92}$$

$$f_j - t_c \leq H z_{jc} \qquad \forall (c,j) \in (A \times A) : c \neq j, \tag{3.93}$$

$$r_{ic} + \sum_{\substack{j \in A \\ j < c}} r_{ij}(z_{jc} + x_{jc} - 1) + \sum_{\substack{j \in A \\ j > c}} r_{ij}(z_{jc} - x_{cj}) \leq C_i \qquad \forall c \in A, \forall i \in R, \tag{3.94}$$

$$x_{jc} \leq z_{cj} \qquad \forall (c,j) \in (A \times A) : c > j, \tag{3.95}$$

$$1 - x_{jc} \leq z_{jc} \qquad \forall (c,j) \in (A \times A) : c > j, \tag{3.96}$$

$$t_c \geq 0 \qquad \forall c \in (A \cup \{m+1\}), \tag{3.97}$$

$$f_c \geq 0 \qquad \forall c \in (A \cup \{m+1\}), \tag{3.98}$$

$$x_{jc} \in \{0,1\} \qquad \forall (c,j) \in (A \times A) : c > j, \tag{3.99}$$

$$z_{jc} \in \{0,1\} \qquad \forall (c,j) \in (A \times A) : c \neq j. \tag{3.100}$$

where we consider $H = \sum_{j \in A} \tau_j$. Observe once again the relation between Kop-CT1-m and Kop-CT2-m, where the second is essentially a transformation of the first one where equation (3.78) is used to eliminate $\frac{m(m-1)}{2}$ variables from the problem.

Connections between variables

Looking at formulation F1, Kop-CT1-m, and Kop-CT2-m together, we can immediately notice the similarity between the resource capacity constraints (3.8), (3.82), and (3.94). Writing these inequalities for a generic activity $\jmath^2 \in A$ and resource type $i \in R$, we respectively get:

$$r_{i\jmath^2} + \sum_{\substack{\jmath^1 \in A \\ \jmath^1 \neq \jmath^2}} r_{i\jmath^1} \gamma_{\jmath^1\jmath^2} \leq C_i,$$

$$r_{i\jmath^2} + \sum_{\substack{\jmath^1 \in A \\ \jmath^1 \neq \jmath^2}} r_{i\jmath^1} (z_{\jmath^1\jmath^2} - x_{\jmath^2\jmath^1}) \leq C_i,$$

$$r_{i\jmath^2} + \sum_{\substack{\jmath^1 \in A \\ \jmath^1 < \jmath^2}} r_{i\jmath^1} (z_{\jmath^1\jmath^2} + x_{\jmath^1\jmath^2} - 1) + \sum_{\substack{\jmath^1 \in A \\ \jmath^1 > \jmath^2}} r_{i\jmath^1} (z_{\jmath^1\jmath^2} - x_{\jmath^2\jmath^1}) \leq C_i.$$

Observing that the set of unordered pairs $A^2 = \{\{\jmath^2, \jmath^1\} : \jmath^2 \in A, \jmath^1 \in A, \jmath^2 \neq \jmath^1\}$ can be converted into the set of ordered pairs

$$\bar{A}^2 = \{(\jmath^1, \jmath^2) \in (A \times A) : \jmath^1 < \jmath^2\}$$

without modifying the structure of formulation F1, the above inequalities suggest, for each $(\jmath^1, \jmath^2) \in \bar{A}^2$, the equivalences (that will be proved to be wrong)

$$\gamma_{\jmath^1\jmath^2} = z_{\jmath^1\jmath^2} - x_{\jmath^2\jmath^1}, \tag{3.101}$$

$$\gamma_{\jmath^2\jmath^1} = z_{\jmath^2\jmath^1} - x_{\jmath^1\jmath^2}, \tag{3.102}$$

among F1 and Kop-CT1-m, and

$$\gamma_{\jmath^1\jmath^2} = z_{\jmath^1\jmath^2} + x_{\jmath^1\jmath^2} - 1, \tag{3.103}$$

$$\gamma_{\jmath^2\jmath^1} = z_{\jmath^2\jmath^1} - x_{\jmath^1\jmath^2}, \tag{3.104}$$

among formulations F1 and Kop-CT2-m.

Furthermore, if we write constraints (3.7) and (3.80) for the generic pair $(\jmath^1, \jmath^2) \in \bar{A}^2$, we obtain:

$$t_{\jmath^2} - t_{\jmath^1} \leq H(1 - \theta_{\jmath^2\jmath^1}) - \varepsilon,$$
$$t_{\jmath^2} + \lambda \leq t_{\jmath^1} + H x_{\jmath^1\jmath^2}.$$

Since in model Kop-CT1-m $x_{\jmath^1\jmath^2} = 1 - x_{\jmath^2\jmath^1}$ due to equation (3.78), the previous become:

$$t_{\jmath^2} - t_{\jmath^1} \leq H(1 - \theta_{\jmath^2\jmath^1}) - \varepsilon,$$
$$t_{\jmath^2} - t_{\jmath^1} \leq H(1 - x_{\jmath^2\jmath^1}) - \lambda.$$

Feasibly assuming $\lambda = \varepsilon$ (e.g., 0.1), this suggests the equivalences

$$\theta_{j^2 j^1} = x_{j^2 j^1}, \tag{3.105}$$

$$\theta_{j^2 j^1} = 1 - x_{j^1 j^2}, \tag{3.106}$$

among formulations F1 and Kop-CT1-m, and

$$\theta_{j^2 j^1} = 1 - x_{j^1 j^2}, \tag{3.107}$$

among models F1 and Kop-CT2-m.

Using equations (3.101)-(3.102), (3.105)-(3.106) into Kop-CT1-m or (3.103)-(3.104), (3.107) into Kop-CT2-m, allows to evidence important details. Indeed, if the above substitutions brought to analogous constraints in the three formulations, we would actually be able to prove their theoretical equivalence. Otherwise, looking at the models after these linear transformations, we can easily notice some significant similarities and differences.

In the following, we will only make this analysis for comparing formulation F1 with Kop-CT2-m. Indeed, a similar study including Kop-CT1-m would simply lead to equivalent results, due to the correspondence of the two models. Combining (3.103), (3.104), and (3.107) we will practically consider the following conversions, for each pair $(j^1, j^2) \in \bar{A}^2$:

$$x_{j^1 j^2} = 1 - \theta_{j^2 j^1}, \tag{3.108}$$

$$z_{j^1 j^2} = \gamma_{j^1 j^2} + \theta_{j^2 j^1}, \tag{3.109}$$

$$z_{j^2 j^1} = \gamma_{j^2 j^1} - \theta_{j^2 j^1} + 1. \tag{3.110}$$

Connections between constraints

Before showing the common aspects of F1 and Kop-CT2-m, we write the mathematical models in a more convenient form, using set $\bar{A}^2$ and updated indexes:

(F1) :

$$\min \; w \tag{3.111}$$

$$w \geq t_{j^1} + \tau_{j^1} \qquad \forall j^1 \in A, \tag{3.112}$$

$$t_{j^2} \geq t_{j^1} + \tau_{j^1} \qquad \forall (j^1, j^2) \in Q, \tag{3.113}$$

$$t_{j^2} + \tau_{j^2} - t_{j^1} \leq H(1 - \theta_{j^2 j^1}) + (\tau_{j^2} - \varepsilon)\gamma_{j^2 j^1} \qquad \forall \; (j^1, j^2) \in \bar{A}^2, \tag{3.114}$$

$$t_{j^1} - t_{j^2} \leq H\theta_{j^2 j^1} + \varepsilon\gamma_{j^2 j^1} - \varepsilon \qquad \forall \; (j^1, j^2) \in \bar{A}^2, \tag{3.115}$$

$$t_{j^1} + \tau_{j^1} - t_{j^2} \leq H\theta_{j^2 j^1} + \tau_{j^1}\gamma_{j^1 j^2} \qquad \forall \; (j^1, j^2) \in \bar{A}^2, \tag{3.116}$$

$$t_{j^2} - t_{j^1} \leq H(1 - \theta_{j^2 j^1}) - \varepsilon \qquad \forall \; (j^1, j^2) \in \bar{A}^2, \tag{3.117}$$

$$r_{ij^2} + \sum_{\substack{j^1 \in A \\ j^1 \neq j^2}} r_{ij^1}\gamma_{j^1 j^2} \leq C_i \qquad \forall j^2 \in A, \forall i \in R, \tag{3.118}$$

$$\theta_{j^2 j^1} \in \{0, 1\} \qquad \forall \; (j^1, j^2) \in \bar{A}^2, \tag{3.119}$$

$$\gamma_{\jmath^2\jmath^1} \in \{0,1\} \qquad \forall\ (\jmath^1,\jmath^2) \in \bar{A}^2, \tag{3.120}$$

$$\gamma_{\jmath^1\jmath^2} \in \{0,1\} \qquad \forall\ (\jmath^1,\jmath^2) \in \bar{A}^2, \tag{3.121}$$

$$t_{\jmath^1} \geq 0 \qquad \forall \jmath^1 \in A, \tag{3.122}$$

$$w \geq 0, \tag{3.123}$$

(Kop-CT2-m) :

$$\min\ f_{m+1} \tag{3.124}$$

$$f_{\jmath^1} = t_{\jmath^1} + \tau_{\jmath^1} \qquad \forall \jmath^1 \in (A \cup \{m+1\}), \tag{3.125}$$

$$f_{\jmath^1} \leq t_{\jmath^2} \qquad \forall (\jmath^1,\jmath^2) \in Q \cup (A \times \{m+1\}), \tag{3.126}$$

$$t_{\jmath^1} - t_{\jmath^2} \leq H(1 - x_{\jmath^1\jmath^2}) \qquad \forall\ (\jmath^1,\jmath^2) \in \bar{A}^2, \tag{3.127}$$

$$t_{\jmath^2} - t_{\jmath^1} \leq H x_{\jmath^1\jmath^2} - \lambda \qquad \forall\ (\jmath^1,\jmath^2) \in \bar{A}^2, \tag{3.128}$$

$$f_{\jmath^1} - t_{\jmath^2} \leq H z_{\jmath^1\jmath^2} \qquad \forall\ (\jmath^1,\jmath^2) \in \bar{A}^2, \tag{3.129}$$

$$f_{\jmath^2} - t_{\jmath^1} \leq H z_{\jmath^2\jmath^1} \qquad \forall\ (\jmath^1,\jmath^2) \in \bar{A}^2, \tag{3.130}$$

$$r_{i\jmath^2} + \sum_{\substack{\jmath^1 \in A \\ \jmath^1 < \jmath^2}} r_{i\jmath^1}(z_{\jmath^1\jmath^2} + x_{\jmath^1\jmath^2} - 1) + \sum_{\substack{\jmath^1 \in A \\ \jmath^1 > \jmath^2}} r_{i\jmath^1}(z_{\jmath^1\jmath^2} - x_{\jmath^2\jmath^1}) \leq C_i \qquad \forall \jmath^2 \in A, \forall i \in R, \tag{3.131}$$

$$x_{\jmath^1\jmath^2} \leq z_{\jmath^2\jmath^1} \qquad \forall\ (\jmath^1,\jmath^2) \in \bar{A}^2, \tag{3.132}$$

$$1 - x_{\jmath^1\jmath^2} \leq z_{\jmath^1\jmath^2} \qquad \forall\ (\jmath^1,\jmath^2) \in \bar{A}^2, \tag{3.133}$$

$$t_{\jmath^1} \geq 0 \qquad \forall \jmath^1 \in (A \cup \{m+1\}), \tag{3.134}$$

$$f_{\jmath^1} \geq 0 \qquad \forall \jmath^1 \in (A \cup \{m+1\}), \tag{3.135}$$

$$x_{\jmath^1\jmath^2} \in \{0,1\} \qquad \forall\ (\jmath^1,\jmath^2) \in \bar{A}^2, \tag{3.136}$$

$$z_{\jmath^1\jmath^2} \in \{0,1\} \qquad \forall\ (\jmath^1,\jmath^2) \in \bar{A}^2, \tag{3.137}$$

$$z_{\jmath^2\jmath^1} \in \{0,1\} \qquad \forall\ (\jmath^1,\jmath^2) \in \bar{A}^2. \tag{3.138}$$

It is easy to see that (3.111)-(3.113) are equivalent to (3.124)-(3.126), where the second employ $m+1$ continuous variables $m+1$ equality constraints more than the first ones. Indeed we have

$$f_{m+1} = t_{m+1} + \tau_{m+1} = t_{m+1} = w,$$

and considering the transformation (3.125) into (3.126), inequalities (3.112)-(3.113) are evidently the same of (3.126).

Moreover, substituting $f_{\jmath^1} = t_{\jmath^1} + \tau_{\jmath^1}$ for each $\jmath^1 \in A$ (from equation (3.125)), constraints (3.127)-(3.130) become:

$$t_{\jmath^1} - t_{\jmath^2} \leq H(1 - x_{\jmath^1\jmath^2}) \qquad \forall\ (\jmath^1,\jmath^2) \in \bar{A}^2,$$

$$t_{\jmath^2} - t_{\jmath^1} \leq H x_{\jmath^1\jmath^2} - \lambda \qquad \forall\ (\jmath^1,\jmath^2) \in \bar{A}^2,$$

$$t_{\jmath^1} + \tau_{\jmath^1} - t_{\jmath^2} \leq H z_{\jmath^1\jmath^2} \qquad \forall\ (\jmath^1,\jmath^2) \in \bar{A}^2,$$

$$t_{\jmath^2} + \tau_{\jmath^2} - t_{\jmath^1} \leq H z_{\jmath^2\jmath^1} \qquad \forall\, (\jmath^1, \jmath^2) \in \bar{A}^2,$$

and considering (3.108)-(3.110):

$$t_{\jmath^1} - t_{\jmath^2} \leq H\theta_{\jmath^2\jmath^1} \qquad \forall\, (\jmath^1, \jmath^2) \in \bar{A}^2, \tag{3.139}$$

$$t_{\jmath^2} - t_{\jmath^1} \leq H(1 - \theta_{\jmath^2\jmath^1}) - \lambda \qquad \forall\, (\jmath^1, \jmath^2) \in \bar{A}^2, \tag{3.140}$$

$$t_{\jmath^1} + \tau_{\jmath^1} - t_{\jmath^2} \leq H\theta_{\jmath^2\jmath^1} + H\gamma_{\jmath^1\jmath^2} \qquad \forall\, (\jmath^1, \jmath^2) \in \bar{A}^2, \tag{3.141}$$

$$t_{\jmath^2} + \tau_{\jmath^2} - t_{\jmath^1} \leq H(1 - \theta_{\jmath^2\jmath^1}) + H\gamma_{\jmath^2\jmath^1} \qquad \forall\, (\jmath^1, \jmath^2) \in \bar{A}^2. \tag{3.142}$$

Inequalities (3.140) are equivalent to (3.117) assuming $\lambda = \varepsilon$, while (3.139), (3.141) and (3.142) are respectively similar to (3.115), (3.116) and (3.114), where for each $(\jmath^1, \jmath^2) \in \bar{A}^2$:

- the right-hand side of (3.139) lacks the term $\varepsilon\gamma_{\jmath^2\jmath^1} - \varepsilon$;
- the variable $\gamma_{\jmath^1\jmath^2}$ on the right-hand side of (3.141) is multiplied by H and not by the reduced parameter $\tau_{\jmath^1}$;
- the variable $\gamma_{\jmath^2\jmath^1}$ on the right-hand side of (3.142) is multiplied by H and not by the reduced parameter $\tau_{\jmath^2} - \varepsilon$;

Resource capacity constraints (3.118) and (3.131) are expressed exactly in the same way, considering transformations (3.108)-(3.110). Indeed:

$$\begin{aligned}
&r_{i\jmath^2} + \sum_{\substack{\jmath^1 \in A \\ \jmath^1 < \jmath^2}} r_{i\jmath^1}(z_{\jmath^1\jmath^2} + x_{\jmath^1\jmath^2} - 1) + \sum_{\substack{\jmath^1 \in A \\ \jmath^1 > \jmath^2}} r_{i\jmath^1}(z_{\jmath^1\jmath^2} - x_{\jmath^2\jmath^1}) \\
&= r_{i\jmath^2} + \sum_{\substack{\jmath^1 \in A \\ \jmath^1 < \jmath^2}} r_{i\jmath^1}\gamma_{\jmath^1\jmath^2} + \sum_{\substack{\jmath^1 \in A \\ \jmath^1 > \jmath^2}} r_{i\jmath^1}\gamma_{\jmath^1\jmath^2} \\
&= r_{i\jmath^2} + \sum_{\substack{\jmath^1 \in A \\ \jmath^1 \neq \jmath^2}} r_{i\jmath^1}\gamma_{\jmath^1\jmath^2}
\end{aligned} \tag{3.143}$$

Finally, tightening inequalities (3.132)-(3.133) are only present in formulation Kop-CT2-m. Their equivalent in terms of variables $\theta_{\jmath^2\jmath^1}$, $\gamma_{\jmath^2\jmath^1}$, $\gamma_{\jmath^1\jmath^2}$ would simply give, for each $(\jmath^1, \jmath^2) \in \bar{A}^2$:

$$\begin{aligned}
1 - \theta_{\jmath^2\jmath^1} &\leq \gamma_{\jmath^2\jmath^1} - \theta_{\jmath^2\jmath^1} + 1, \\
1 - 1 + \theta_{\jmath^2\jmath^1} &\leq \gamma_{\jmath^1\jmath^2} + \theta_{\jmath^2\jmath^1},
\end{aligned}$$

that is, $\gamma_{\jmath^2\jmath^1} \geq 0$ and $\gamma_{\jmath^1\jmath^2} \geq 0$, already considered in (3.120)-(3.121).

Main differences between the formulations

Thanks to the above analysis, we can now gather our previous observations and list the main differences among formulations F1, Kop-CT1-m and Kop-CT2-m.

Formulation	Number of constraints	Continuous variables	Binary variables
F1	$2m^2 + mn + \|Q\| + 1$	$m + 1$	$\frac{3}{2}m(m-1)$
Kop-CT1-m	$\frac{7}{2}m^2 + mn + \frac{1}{2}m + \|Q\| + 3$	$2m + 2$	$2m(m-1)$
Kop-CT2-m	$3m^2 + mn + m + \|Q\| + 3$	$2m + 2$	$\frac{3}{2}m(m-1)$

Table 3.8. Number of variables and constraints for formulations F1, Kop-CT1-m and Kop-CT2-m, considering m activities, n resource types, and $|Q|$ precedence relations.

Number of variables and constraints: the number of variables and constraints for each model are reported in Table 3.8. Overall, formulation F1 is that with the smaller dimensions, having, as expected, the same amount of 0-1 variables than Kop-CT2-m and $\frac{1}{2}m(m-1)$ less than Kop-CT1-m.

Role of binary variables in resource capacity constraints: Table 3.9 considers a single pair $(\jmath^1, \jmath^2) \in \bar{A}^2$, and summarizes, for each possible disposition of activities starting time, the implications brought by the three sets of constraints:

- (3.114)-(3.117): included in F1;
- (3.127)-(3.130): included in Kop-CT2-m;
- (3.139)-(3.142): from Kop-CT2-m, using substitutions (3.108)-(3.110);

Case	(3.114)-(3.117)	(3.139)-(3.142)	(3.127)-(3.130)
$t_{\jmath^1} + \tau_{\jmath^1} < t_{\jmath^2}$	$\Rightarrow \{\theta_{\jmath^2\jmath^1} = 0$	$\Rightarrow \{\theta_{\jmath^2\jmath^1} = 0$	$\Rightarrow \begin{cases} x_{\jmath^1\jmath^2} = 1 \\ z_{\jmath^2\jmath^1} = 1 \end{cases}$
$t_{\jmath^2} + \tau_{\jmath^2} < t_{\jmath^1}$	$\Rightarrow \{\theta_{\jmath^2\jmath^1} = 1$	$\Rightarrow \{\theta_{\jmath^2\jmath^1} = 1$	$\Rightarrow \begin{cases} x_{\jmath^1\jmath^2} = 0 \\ z_{\jmath^1\jmath^2} = 1 \end{cases}$
$t_{\jmath^1} < t_{\jmath^2} < t_{\jmath^1} + \tau_{\jmath^1}$	$\Rightarrow \begin{cases} \gamma_{\jmath^1\jmath^2} = 1 \\ \theta_{\jmath^2\jmath^1} = 0 \end{cases}$	$\Rightarrow \begin{cases} \gamma_{\jmath^1\jmath^2} = 1 \\ \theta_{\jmath^2\jmath^1} = 0 \end{cases}$	$\Rightarrow \begin{cases} x_{\jmath^1\jmath^2} = 1 \\ z_{\jmath^1\jmath^2} = 1 \\ z_{\jmath^2\jmath^1} = 1 \end{cases}$
$t_{\jmath^2} < t_{\jmath^1} < t_{\jmath^2} + \tau_{\jmath^2}$	$\Rightarrow \begin{cases} \gamma_{\jmath^2\jmath^1} = 1 \\ \theta_{\jmath^2\jmath^1} = 1 \end{cases}$	$\Rightarrow \begin{cases} \gamma_{\jmath^2\jmath^1} = 1 \\ \theta_{\jmath^2\jmath^1} = 1 \end{cases}$	$\Rightarrow \begin{cases} x_{\jmath^1\jmath^2} = 0 \\ z_{\jmath^1\jmath^2} = 1 \\ z_{\jmath^2\jmath^1} = 1 \end{cases}$
$t_{\jmath^1} = t_{\jmath^2}$	$\Rightarrow \begin{cases} \gamma_{\jmath^1\jmath^2} = 1 \\ \gamma_{\jmath^2\jmath^1} = 1 \\ \theta_{\jmath^2\jmath^1} = 0 \end{cases}$	$\Rightarrow \begin{cases} \gamma_{\jmath^1\jmath^2} = 1 \\ \theta_{\jmath^2\jmath^1} = 0 \end{cases}$	$\Rightarrow \begin{cases} x_{\jmath^1\jmath^2} = 1 \\ z_{\jmath^1\jmath^2} = 1 \\ z_{\jmath^2\jmath^1} = 1 \end{cases}$

Table 3.9. Implications brought by the three different sets of constraints, considering a single pair $(\jmath^1, \jmath^2) \in \bar{A}^2$ and all possible cases for the activities starting times.

The first four rows in the table confirm the similarities between F1 and Kop-CT2-m. Indeed, (3.114)-(3.117) assure the same implications of (3.139)-(3.142), which are obviously compatible with (3.127)-(3.130). However, when $t_{\jmath^1} = t_{\jmath^2}$ (see the fifth case), the implications are not equivalent, due to the difference between constraint (3.115) and (3.139) (indeed the additional term $\varepsilon\gamma_{\jmath^2\jmath^1} - \varepsilon$ in (3.115) guarantees also $\gamma_{\jmath^2\jmath^1} = 1$).

This generates two important considerations: the first is that, also ignoring for a moment the coefficients of binary variables, equations (3.108)-(3.110) do not permit a direct transformation of F1 in Kop-CT2-m (and Kop-CT1-m) and vice versa. Secondly we notice that, since in the last case ($t_{\jmath^1} = t_{\jmath^2}$) constraints (3.114)-(3.117) and (3.127)-(3.130) give

$$\gamma_{\jmath^2\jmath^1} \neq z_{\jmath^2\jmath^1} - x_{\jmath^1\jmath^2},$$

also resource capacity constraints have a different interpretation.

To clarify this fact, we assume there is a pair of activities $(\jmath^1, \jmath^2) \in \bar{A}^2$ which start at the same time (i.e., $t_{\jmath^1} = t_{\jmath^2}$). If we write constraint (3.118) for a certain resource $i \in R$ and for both the activities $\jmath^1$ and $\jmath^2$, we get:

$$r_{i\jmath^1} + \sum_{\jmath \in A,\ \jmath \neq \jmath^1} r_{i\jmath}\gamma_{\jmath\jmath^1} \leq C_i,$$

$$r_{i\jmath^2} + \sum_{\jmath \in A,\ \jmath \neq \jmath^2} r_{i\jmath}\gamma_{\jmath\jmath^2} \leq C_i.$$

Since $\gamma_{\jmath\jmath^1} = \gamma_{\jmath\jmath^2}$ for each $\jmath \in A \backslash \{\jmath^1, \jmath^2\}$, and $\gamma_{\jmath^2\jmath^1} = \gamma_{\jmath^1\jmath^2} = 1$, these inequalities are completely equivalent.

If we now do the same step for constraint (3.131), we obtain:

$$r_{i\jmath^1} + \sum_{\jmath \in A,\ \jmath < \jmath^1} r_{i\jmath}(z_{\jmath\jmath^1} + x_{\jmath\jmath^1} - 1) + \sum_{\jmath \in A,\ \jmath > \jmath^1} r_{i\jmath}(z_{\jmath\jmath^1} - x_{\jmath^1\jmath}) \leq C_i,$$

$$r_{i\jmath^2} + \sum_{\jmath \in A,\ \jmath < \jmath^2} r_{i\jmath}(z_{\jmath\jmath^2} + x_{\jmath\jmath^2} - 1) + \sum_{\jmath \in A,\ \jmath > \jmath^2} r_{i\jmath}(z_{\jmath\jmath^2} - x_{\jmath^2\jmath}) \leq C_i.$$

Here we have

$$\begin{cases} z_{\jmath\jmath^1} + x_{\jmath\jmath^1} - 1 = z_{\jmath\jmath^2} + x_{\jmath\jmath^2} - 1 & \jmath < \jmath^1 < \jmath^2 \\ z_{\jmath\jmath^1} - x_{\jmath^1\jmath} = z_{\jmath\jmath^2} + x_{\jmath\jmath^2} - 1 & \jmath^1 < \jmath < \jmath^2 \\ z_{\jmath\jmath^1} - x_{\jmath^1\jmath} = z_{\jmath\jmath^2} - x_{\jmath^2\jmath} & \jmath^1 < \jmath^2 < \jmath \end{cases}$$

but

$$0 = z_{\jmath^2\jmath^1} - x_{\jmath^1\jmath^2} \neq z_{\jmath^1\jmath^2} + x_{\jmath^1\jmath^2} - 1 = 1.$$

It follows that the two capacity constraints are not equivalent. In particular, the first does not consider (wrongly) the resources of type i used by activity $\jmath^2$, since $z_{\jmath^2\jmath^1} - x_{\jmath^1\jmath^2} = 0$. However, everything is taken into account by the second inequality, which is tighter, having $z_{\jmath^1\jmath^2} + x_{\jmath^1\jmath^2} - 1 = 1$.

The same argument is also valid when the activities that start at the same time are three or more. Among these, only the activity with the highest

index effectively tights the problem, taking into account the capacity of resources occupied by the all the other operations, and thus modeling availability constraints.

This also explains why for formulations Kop-CT1-m and Kop-CT2-m having ordered pairs of indexes is essential (while formulation F1 is independent from the ordering).

Tightening constraints and integer feasible solutions: Formulations Kop-CT1-m and Kop-CT2-m, differently form F1, include tightening constraints (respectively (3.132)-(3.133) and (3.83)) which directly link the binary variables of the problem. Thanks to these inequalities, some non-integer solutions are cut off from the linear relaxation of the problem, assuring the same 0-1 feasible points for variables x and z. Table 3.10 considers a single pair of activities $(j, c) \in \bar{A}^2$ and summarizes the integer feasible values for models Kop-CT1-m and Kop-CT2-m for each disposition of the starting times, ignoring resource capacity limitations. Comparing this table with Table 3.1, it can be noticed that F1 in general admits more 0-1 solutions than Kop-CT1-m/Kop-CT2-m, although possibly reduced by resource availability constraints.

	$f_j \leq t_c$	$t_j < t_c \leq f_j$	$t_j = t_c$	$t_c < t_j \leq f_c$	$f_c \leq t_j$
Kop-CT1-m	(0,1,1,0)	(1,1,1,0)	(1,1,1,0)	(1,1,0,1)	(1,0,0,1)
	(1,1,1,0)				(1,1,0,1)
Kop-CT2-m	(0,1,1)	(1,1,1)	(1,1,1)	(1,1,0)	(1,0,0)
	(1,1,1)				(1,1,0)

Table 3.10. Feasible values of $(z_{jc}, z_{cj}, x_{jc}, x_{cj})$ for Kop-CT1-m and (z_{jc}, z_{cj}, x_{jc}) for Kop-CT2-m, for all starting times alternatives, ignoring resource constraints.

Coefficients of binary variables: As already observed analyzing inequalities (3.139)-(3.142), it is not possible to directly derive the reduced coefficients of formulation F1 (which multiply $\gamma_{\jmath^1\jmath^2}$ and $\gamma_{\jmath^2\jmath^1}$ in constraints (3.114)-(3.117)) by simply applying the substitutions (3.108)-(3.110) and (3.125) into (3.127)-(3.130). This brings out an important difference able to produce significant computational discrepancies, as we will show in the following.

Notice in addition that this fact is independent from the type of parameters used as big-M in formulations Kop-CT1-m and Kop-CT2-m. Indeed, constraints (3.127)-(3.130) are affected by our choice of substituting the original values computable through preprocessing with the general upper bound H, but a similar result would have been obtained also by using $LS_{\jmath^1} - ES_{\jmath^2}, LS_{\jmath^1} - ES_{\jmath^2} + \lambda, LF_{\jmath^1} - ES_{\jmath^2}$ for all pairs $(\jmath^1, \jmath^2) \in A \times A$.

Nevertheless, in order to analyze the sensitivity of formulations F1, Kop-CT1-m and Kop-CT2-m to these parameters, we introduce the following additional variations to the models:

F1-w: equivalent to F1, where constraints (3.4)-(3.7) are replaced by:

$$
\begin{aligned}
& t_j + \tau_j - t_c \leq (LF_j - ES_c)(1 - \theta_{jc}) + (\tau_j - \varepsilon)\gamma_{jc} && \forall \{j,c\} \in A^2, \\
& t_c - t_j \leq (LS_c - ES_j + \varepsilon)\theta_{jc} + \varepsilon\gamma_{jc} - \varepsilon && \forall \{j,c\} \in A^2, \\
& t_c + \tau_c - t_j \leq (LF_c - ES_j)\theta_{jc} + \tau_c\gamma_{cj} && \forall \{j,c\} \in A^2, \\
& t_j - t_c \leq (LS_j - ES_c)(1 - \theta_{jc}) - \varepsilon && \forall \{j,c\} \in A^2, \\
& \theta_{jc} = 1 && \forall \{j,c\} \in A^2 : LS_j < ES_c.
\end{aligned}
$$

Kop-CT1-w: equivalent to Kop-CT1-m, where constraints (3.79)-(3.81) are replaced by:

$$
\begin{aligned}
& t_j \leq t_c + (LS_j - ES_c)x_{cj} && \forall (c,j) \in (A \times A) : c > j, \\
& t_c + \lambda \leq t_j + (LS_c - ES_j + \lambda)x_{jc} && \forall (c,j) \in (A \times A) : c > j, \\
& f_j - t_c \leq (LF_j - ES_c)z_{jc} && \forall (c,j) \in (A \times A) : c \neq j, \\
& x_{jc} = 1 && \forall (c,j) \in (A \times A), c \neq j : LS_j < ES_c.
\end{aligned}
$$

Kop-CT2-w: equivalent to Kop-CT2-m, where constraints (3.91)-(3.93) are replaced by:

$$
\begin{aligned}
& t_j \leq t_c + (LS_j - ES_c)(1 - x_{jc}) && \forall (c,j) \in (A \times A) : c > j, \\
& t_c + \lambda \leq t_j + (LS_c - ES_j + \lambda)x_{jc} && \forall (c,j) \in (A \times A) : c > j, \\
& f_j - t_c \leq (LF_j - ES_c)z_{jc} && \forall (c,j) \in (A \times A) : c \neq j, \\
& x_{jc} = 1 && \forall (c,j) \in (A \times A), c > j : LS_j < ES_c.
\end{aligned}
$$

To compute the parameters ES_j, LS_j and LF_j for each activity $j \in A$ we use for all formulations the critical path method combined with the serial schedule generation scheme heuristic from Kelley (1963).

3.4.2 Preliminary computational analysis

In previous sections, we have considered different mathematical formulations for the RCPSP, including some relevant modifications. Here we take into account six of them in order to conduct some preliminary experiments on the test set j30 of PSPLib, validate the results, and analyze in detail some important aspects of their experimental behavior. The final aim of this investigation is indeed to provide a comprehensive overview of the performances reachable by the different modeling possibilities encountered so far, motivating also our choice of the formulations to further compare in Section 3.5, where we extend our study to additional test instances.

We thus start examining the following models: F1, Kop-CT1-m, Kop-CT2-m, F1-w, Kop-CT1-w, and Kop-CT2-w. The results obtained on set j30, containing 480 instances, are reported in Table 3.11, where the same KPIs of Table 3.2 are used. As before, the experiments are performed on an Intel Core i7 CPU 5600U 2.60

Formulation	Feas.%	Opt.%	Out.Gap%	Act.Gap%	Time(sec.)
F1	100.00	96.88	14.27	0.96	9.64
F1-w	100.00	96.25	13.38	1.07	9.16
Kop-CT1-m	100.00	91.25	15.18	2.03	9.94
Kop-CT1-w	99.79	91.88	16.30	2.19	11.54
Kop-CT2-m	100.00	91.25	15.23	1.97	12.16
Kop-CT2-w	99.79	91.25	15.99	2.02	11.21

Table 3.11. Computational results for F1, F1-w, Kop-CT1-m, Kop-CT1-w, Kop-CT2-m, and Kop-CT2-w on j30 instances.

GHz with 16GB RAM, running ILOG CPLEX Concert Technology 12.8.0. under standard configurations, with a time limit of 300 CPU seconds.

As evident, the performances of F1 and F1-w are comparable, and both produce better solutions than Kop-CT1-m, Kop-CT1-w, Kop-CT2-m, and Kop-CT2-w, which have similar scores. The use of preprocessing to calculate activities time-windows partially increases the number of optimal solutions found by Kop-CT1-w, although makes it impossible to find a feasible point for a particular instance running Kop-CT1-w or Kop-CT2-w (notice that in theory a feasible solution is available, being the one calculated by the upper bounding heuristic). Differently, F1-w provides few optima less than F1, and this is probably due to the different branching behavior induced by the use of lower coefficients for variables θ. Indeed as θ_{jc} "loses some power" in satisfying constraints (3.4)-(3.7), variables γ_{jc} and γ_{cj} will be more encouraged to take non-zero values in the relaxation, assuming a higher importance in the branching and thus modifying the performances of the algorithm (see observations in the last part of Section 3.2.2). This is however a tricky issue, that naturally brings to the interrogative on which are the best values for big-M parameters, being both sufficiently high to stimulate particular algorithmic actions and reasonably small to avoid numerical problems. In any case, such discussions are merely technical, relatively significant, and not in line with the scope of our work. For this reason, we prefer to continue our analysis, assuming $H = \sum_{j \in A} \tau_j$ (as in formulations F1, Kop-CT1-m, and Kop-CT2-m) is a good enough value, and to only investigate models that do not involve preprocessing procedures.

Thus, focusing on the direct comparison of formulation F1 with Kop-CT1-m and Kop-CT2-m, we can observe the outputs in Tables 3.12 and 3.13, which in general reveal similar results (for instance detailed outputs, see Appendix A). Notice that both models Kop-CT1-m and Kop-CT2-m find the optimum for 438 test instances, but not exactly the same. All of these are easily solved by F1, spending on average respectively less than one half and less than one third of the time employed by the others. Also the 15 problems for which no one method could prove the optimality before the time limit confirm the good performances of F1, being the one with the lowest gaps (output by CPLEX and actual, with respect to the known optima). Moreover, the instances optimally solved by F1 but not by Kop-CT1-m/Kop-CT2-m in both cases return an average computation time of about 100 second for F1 and an

		Kop-CT1-m	
		Optimal	**Sub/Non-proven optimal**
F1	**Optimal**	Instances: 438 (91.25%) F1: Out.Gap%: 0.00; Act.Gap%: 0.00; Time(sec.): 4.12; Avg.Nodes: 4533 Kop-CT1-m: Out.Gap%: 0.00; Act.Gap%: 0.00; Time(sec.): 9.94; Avg.Nodes: 5996	Instances: 27 (5.62%) F1: Out.Gap%: 0.00; Act.Gap%: 0.00; Time(sec.): 99.24; Avg.Nodes: 115456 Kop-CT1-m: Out.Gap%: 9.87; Act.Gap%: 1.57; Time(sec.): 300.00; Avg.Nodes: 130100
	Sub/Non-proven optimal	Instances: 0 (0.00%) – –	Instances: 15 (3.13%) F1: Out.Gap%: 14.27; Act.Gap%: 0.96; Time(sec.): 300.00; Avg.Nodes: 261134 Kop-CT1-m: Out.Gap%: 24.73; Act.Gap%: 2.87; Time(sec.): 300.00; Avg.Nodes: 112056

Table 3.12. Results comparison between formulations F1 and Kop-CT1-m

output gap of approximately 10% for the other formulations (with Act.Gap%=1.57 for Kop-CT1-m, and Act.Gap%=1.23 for Kop-CT2-m).

Looking at the number of nodes visited by the algorithms in each situation, we may notice that F1 in general needs to explore fewer directions than Kop-CT1-m and Kop-CT2-m to prove the optimality of the solutions. In accordance with this, when instances can not be solved, F1 visits more than twice the number of nodes examined by Kop-CT1-m and almost three times that of Kop-CT2-m in the same amount of time of 300 seconds, finally reaching an actual gap under the 1%. This evidences the influence of three interrelated aspects:

1. Probably due to the lower number of variables and constraints, formulation F1 in general finds the optimal solution of nodes relaxation faster. This is also confirmed by Table 3.14, which reports the average amounts of milliseconds required by each method to solve the relaxed problems at the root for the instances in set j30.

2. The particular structure of formulation F1 enables a more effective branching than those of formulations Kop-CT1-m and Kop-CT2-m. We have already

		Kop-CT2-m	
		Optimal	**Sub/Non-proven optimal**
F1	**Optimal**	Instances: 438 (91.25%) F1: {Out.Gap%: 0.00, Act.Gap%: 0.00, Time(sec.): 4.01, Avg.Nodes: 4288} Kop-CT2-m: {Out.Gap%: 0.00, Act.Gap%: 0.00, Time(sec.): 12.16, Avg.Nodes: 6468}	Instances: 27 (5.62%) F1: {Out.Gap%: 0.00, Act.Gap%: 0.00, Time(sec.): 100.99, Avg.Nodes: 119429} Kop-CT2-m: {Out.Gap%: 10.14, Act.Gap%: 1.23, Time(sec.): 300.00, Avg.Nodes: 109000}
	Sub/Non-proven optimal	Instances: 0 (0.00%) – –	Instances: 15 (3.13%) F1: {Out.Gap%: 14.27, Act.Gap%: 0.96, Time(sec.): 300.00, Avg.Nodes: 261134} Kop-CT2-m: {Out.Gap%: 24.39, Act.Gap%: 3.30, Time(sec.): 300.00, Avg.Nodes: 97335}

Table 3.13. Results comparison between formulations F1 and Kop-CT2-m

observed in previous sections how F1 allows to obtain good algorithmic performances, encouraging the search in specific directions and quickly tightening the feasible region through the use of reduced parameters. In the following, we will try to further investigate this fact also providing an insight into a direct comparison among different methods.

3. As a result of the preceding point, F1 normally reaches deep layers of the search tree sooner than other formulations. Nodes in these domains have a high number of binary variables fixed to 0 or 1, and for this reason permit an even faster solution of linear relaxation problems. This way the positive effect is propagated through the branches, consistently improving the overall exploration.

Before continuing our investigation of performances, we provide the last data point to support our analysis and assess the validity of the results. Table 3.15 reports the outputs obtained by running the same type of experiments on set j30 for Kop-CT1-m and Kop-CT2-m, disabling – as done for formulations F1, F2 and F3 – all the advanced procedures automatically used by CPLEX to speed up the solution process. The results for F1 are reported again for a direct comparison.

Formulation	Time (milliseconds)
F1	8.09
Kop-CT1-m	26.73
Kop-CT2-m	9.90

Table 3.14. Average solving times of root node relaxation for F1, Kop-CT1-m, and Kop-CT2-m on j30 instances.

Formulation	Feas.%	Opt.%	Out.Gap%	Act.Gap%	Time(sec.)
F1	100.00	91.67	12.83	2.04	15.87
Kop-CT1-m	100.00	81.25	15.79	0.95	17.21
Kop-CT2-m	100.00	78.54	19.59	0.80	23.25

Table 3.15. Computational results for F1, Kop-CT1-m and Kop-CT2-m on j30 instances with CPLEX advanced methods disabled.

Comparison between branching properties

As said, formulation F1 has the most efficient algorithmic behavior among methods compared. To provide an intuitive idea of the reason, we consider again the simple example involving five activities proposed in Section 3.2.2. The branching trees explored by F1, Kop-CT1-m and Kop-CT2-m are shown in Figure 3.4, where black nodes and white nodes are respectively associated to decisions regarding the sequencing variables (θ for F1 and x for Kop-CT1-m and Kop-CT2-m) or the other binaries (γ for F1 and z for Kop-CT1-m and Kop-CT2-m). The original node logs used to build the figure are reported in Appendix B for completeness.

Although this representation can only statically illustrate the decisions taken by the algorithms in a single and extremely simple case, it is able to provide an insight into the logical steps followed by each approach. Starting from the root and proceeding toward the leaves, it is clear that, while F1 automatically prioritizes fixing variables θ, Kop-CT1-m and Kop-CT2-m alternate between x and z (remember that Kop-CT1-m has twice as many x variables as Kop-CT2-m). This happens regardless of the variable selection strategy employed, being a consequence of the type of solutions found to optimize the linear relaxation of nodes. For example, the optimal $(\gamma, \theta), (z^1, x^1), (z^2, x^2)$ (respectively for F1, Kop-CT1-m and Kop-CT2-m) at the root nodes are given by the following:

$$\gamma = \begin{bmatrix} - & 0 & 0 & 0 & 0 \\ 0 & - & 0 & 0 & 0 \\ 0 & 0 & - & 0 & 0 \\ 0 & 0 & 0 & - & 0 \\ 0 & 0 & 0 & 0 & - \end{bmatrix} \qquad \theta = \begin{bmatrix} - & - & - & - & - \\ 0.2 & - & - & - & - \\ 0.2 & 0.13 & - & - & - \\ 0.27 & 0.2 & 0.27 & - & - \\ 0.33 & 0.27 & 0.33 & 0.33 & - \end{bmatrix}$$

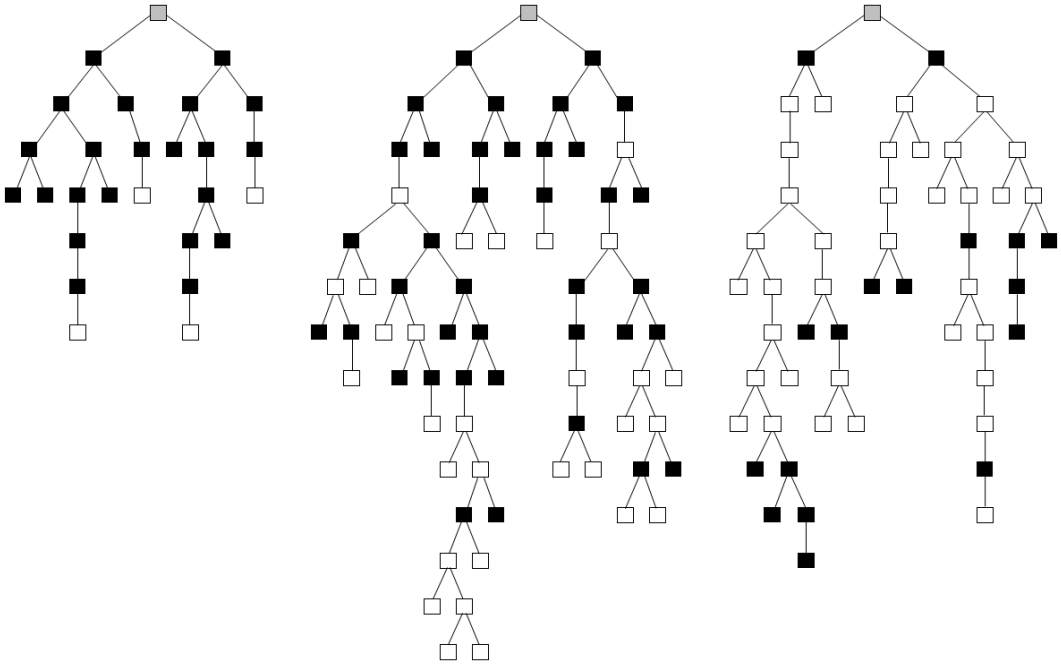

Figure 3.4. Branching trees generated (from left to right) by F1, Kop-CT1-m and Kop-CT2-m while solving a simple instance with five activities. Black and white nodes indicate the type of variable selected for branching.

$$z^1 = \begin{bmatrix} - & 0 & 0.07 & 0.07 & 0.07 \\ 1 & - & 0.2 & 0.2 & 0.2 \\ 1 & 0.93 & - & 0.2 & 0.2 \\ 0.93 & 0.8 & 0.8 & - & 0.27 \\ 0.93 & 0.8 & 0.8 & 0.73 & - \end{bmatrix} \qquad x^1 = \begin{bmatrix} - & 1 & 1 & 0.93 & 0.93 \\ 0 & - & 0.93 & 0.8 & 0.8 \\ 0 & 0.07 & - & 0.8 & 0.8 \\ 0.07 & 0.2 & 0.2 & - & 0.73 \\ 0.07 & 0.2 & 0.2 & 0.27 & - \end{bmatrix}$$

$$z^2 = \begin{bmatrix} - & 0.91 & 0.99 & 0.75 & 0.99 \\ 0.21 & - & 1 & 1 & 1 \\ 0.2 & 0.12 & - & 0.99 & 0.91 \\ 0.27 & 0.19 & 0.27 & - & 0.67 \\ 0.33 & 0.25 & 0.33 & 0.33 & - \end{bmatrix} \qquad x^2 = \begin{bmatrix} - & 0.09 & 0.01 & 0.25 & 0.01 \\ - & - & 0 & 0 & 0 \\ - & - & - & 0.01 & 0.09 \\ - & - & - & - & 0.33 \\ - & - & - & - & - \end{bmatrix}$$

where values are rounded at the second decimal place for simplicity.

In Section 3.2.2 we have already observed that formulation F1 can easily find a solution for the relaxations, possibly setting the θ variables to fractional values and many γ to 0. This indeed allows to satisfy all inequalities linking t, θ and γ, without generating conflicts on the resources limited capacities. The preceding solution seems to confirm this idea.

Differently from F1, Kop-CT1-m and Kop-CT2-m admit many fractional values among both variables z and x. This of course depends on the characteristics of the multidimensional feasible region, which however is really hard to visualize. Thus, in order to give an idea, we can just make some intuitive observations. Let us momentarily consider only one pair of activities $(c, j) \in \bar{A}^2$, and write inequalities

(3.76),(3.78)-(3.81),(3.83) of formulation Kop-CT1-m without integrality constraints. We get:

$$f_j = t_j + \tau_j, \tag{3.144}$$

$$f_c = t_c + \tau_c, \tag{3.145}$$

$$x_{cj} = 1 - x_{jc}, \tag{3.146}$$

$$t_j - t_c \leq H x_{cj}, \tag{3.147}$$

$$t_c - t_j + \lambda \leq H x_{jc}, \tag{3.148}$$

$$f_j - t_c \leq H z_{jc}, \tag{3.149}$$

$$f_c - t_j \leq H z_{cj}, \tag{3.150}$$

$$x_{jc} \leq z_{cj}, \tag{3.151}$$

$$x_{cj} \leq z_{jc}. \tag{3.152}$$

Independently from the starting and finish time of the activities, exactly one of the left-hand sides of (3.147) and (3.148), and at least one among those of (3.149) and (3.150) are strictly positive. Consequently, the values of at least two or three of the binary variables on the right-hand sides need to be greater than zero. In particular, constraints (3.146), (3.151)-(3.152) impose that if $x_{cj} < 1$ then $x_{jc} > 0$ and $z_{cj} > 0$, and if $x_{jc} < 1$ then $x_{cj} > 0$ and $z_{jc} > 0$. We can distinguish two cases:

1. If activities c and j have starting and ending times such that they are sequenced (i.e., $f_c \leq t_j$ or $f_j \leq t_c$), either $0 < x_{jc} \leq z_{cj}$ or $0 < x_{cj} \leq z_{jc}$. In this situation a feasible integer solution obviously exists. For example, if c precedes j, it is possible to set $x_{jc} = 0, x_{cj} = 1, z_{jc} = 1, z_{cj} = 0$, and if j precedes c, $x_{jc} = 1, x_{cj} = 0, z_{jc} = 0, z_{cj} = 1$. This happens for instance to activities 1 and 2 in the example, since they are connected by the precedence relation $(1, 2) \in Q$.

2. If activities c and j overlap, then the situation is different because both (3.149) and (3.150) have positive left-hand sides. Consequently, $z_{jc} > 0, z_{cj} > 0$, and at the same time $x_{jc} + x_{cj} = 1$ due to (3.146). So one could imagine two integer solutions, compatibly with the starting times of the activities: $(x_{jc} = 0, x_{cj} = 1, z_{jc} = 1, z_{cj} = 1)$ or $(x_{jc} = 1, x_{cj} = 0, z_{jc} = 1, z_{cj} = 1)$. However, as long as the resource constraints have some tightening effect, these solutions are difficultly feasible. Indeed if we write inequalities (3.82) for activities c and j, and every resource $i \in R$, we obtain:

$$r_{ic} + \sum_{\substack{h \in A \\ h \neq c}} r_{ih}(z_{hc} - x_{ch}) \leq C_i \qquad \forall i \in R,$$

$$r_{ij} + \sum_{\substack{h \in A \\ h \neq j}} r_{ih}(z_{hj} - x_{jh}) \leq C_i \qquad \forall i \in R.$$

In the first summation we have the terms $r_{ij}(z_{jc} - x_{cj})$, in the second the terms $r_{ic}(z_{cj} - x_{jc})$. In order not to overcome the maximum capacities, they

have to be someway balanced with all the corresponding nonnegative terms of all the pairs of activities for each type of resource. It follows that setting $z_{jc} = 1, x_{cj} = 0$, so $r_{ij}(z_{jc} - x_{cj}) = r_{ij}$ (or $z_{cj} = 1, x_{jc} = 0$, with $r_{ic}(z_{cj} - x_{jc}) = r_{ic}$), may in general be a strategy for some overlapping activity pairs, but would force some others to give fractional values to many (or all) of their associated binary variables. To make an example, supposing that solution $(x_{jc} = 0.5, x_{cj} = 0.5, z_{jc} = 0.5, z_{cj} = 0.5)$ satisfies all (3.147)-(3.150), it involves $r_{ij}(z_{jc} - x_{cj}) = 0$ and $r_{ic}(z_{cj} - x_{jc}) = 0$, which allows to completely avoid resource conflicts for the considered pair.

These two cases summarily describe the type of solution which one could expect from the linear relaxation at the root or, considering the associated effects, at every node of the branching tree where some binary variables still need to be fixed. However, an important aspect has to be taken into account before completing this analysis, which is connected to the quality of the linear relaxation. Model F1 and Kop-CT1-m, as all the Continuous-Time approaches in the literature, provide poor relaxation bounds (possibly equal to the length of the critical path, ignoring resource constraints). This is basically a consequence of the previous observations, which indeed bring to values of binary variables rather distant from 0 or 1, being anyway able to satisfy indicator constraints. Looking at this fact exactly from the opposite point of view, we can notice the following. When the linear relaxation is solved at the root node – if the problem is not trivial – it provides an optimal non-integer solution with value LB_r. This is the smallest lower bound achievable at every node, since no binary variables are fixed. In other terms $LB_r \leq LB_{n_b}$ for each node n_b in the branching tree. Similarly, if a node n_b^c derives from a parent n_b^p, of course $LB_{n_b^p} \leq LB_{n_b^c}$. So we can generally assert that, going from the root toward the leaves of the tree, the optimized project makespan resulting from problem relaxations can only increase. Imagining the representation of this idea on a Gantt chart, we may visualize two diverging situations:

At the root, the makespan is very low. The sequence relations are respected, but resource limited capacities do not effectively constraint the disposition of activities over time. Consequently, many pairs (c, j) overlap, falling within the second of the cases previously considered, and thus generating different fractional values.

At deeper nodes, the makespan is higher. This is a consequence of the increasing power of resource constraints, which are able to avoid some unfeasible activities parallelizations, having some binary variables fixed. Many pairs (c, j) must be necessarily sequenced, thus increasing the number of possibly integer solutions to the linear relaxation.

The aim of these observations is to try a generalization of some tricky aspects that however significantly depend on the type of instance considered. In the example with five activities provided above, only one precedence relation linked the variable starting time, thus permitting a very cumulative solution to the initial relaxation (indeed $LB_r = 5$, with $t_1 = t_3 = t_4 = t_5 = 0$ and $t_2 = 1$). If more sequence constraints were involved, we could probably expect to get less fractional values

for the binary variables, but in any case distributed among both the z and x. For instance, consider the following solutions found for the relaxation of the root nodes, including into Q the additional pair $(2, 3)$:

$$\gamma = \begin{bmatrix} - & 0 & 0 & 0 & 0 \\ 0 & - & 0 & 0 & 0 \\ 0 & 0 & - & 0 & 0 \\ 0 & 0 & 0 & - & 0 \\ 0 & 0 & 0 & 0 & - \end{bmatrix} \qquad \theta = \begin{bmatrix} - & - & - & - & - \\ 0.2 & - & - & - & - \\ 0.4 & 0.33 & - & - & - \\ 0.27 & 0.2 & 0.07 & - & - \\ 0.33 & 0.27 & 0.13 & 0.33 & - \end{bmatrix}$$

$$z^1 = \begin{bmatrix} - & 0 & 0 & 0.07 & 0.07 \\ 1 & - & 0 & 0.2 & 0.2 \\ 1 & 1 & - & 0.4 & 0.4 \\ 0.93 & 0.8 & 0.6 & - & 0.27 \\ 0.93 & 0.8 & 0.6 & 0.73 & - \end{bmatrix} \qquad x^1 = \begin{bmatrix} - & 1 & 1 & 0.93 & 0.93 \\ 0 & - & 1 & 0.8 & 0.8 \\ 0 & 0 & - & 0.6 & 0.6 \\ 0.07 & 0.2 & 0.4 & - & 0.73 \\ 0.07 & 0.2 & 0.4 & 0.27 & - \end{bmatrix}$$

$$z^2 = \begin{bmatrix} - & 0.93 & 0.6 & 0.84 & 0.99 \\ 0.2 & - & 0.73 & 1 & 1 \\ 0.4 & 0.33 & - & 1 & 1 \\ 0.27 & 0.2 & 0.07 & - & 0.67 \\ 0.33 & 0.27 & 0.13 & 0.33 & - \end{bmatrix} \qquad x^2 = \begin{bmatrix} - & 0.07 & 0.4 & 0.16 & 0.01 \\ - & - & 0.27 & 0 & 0 \\ - & - & - & 0 & 0 \\ - & - & - & - & 0.33 \\ - & - & - & - & - \end{bmatrix}$$

This again confirms the difference between F1 and Kop-CT1-m/Kop-CT2-m. Notice indeed that – although not addressed here – arguments similar to those considered for Kop-CT1-m could be tentatively sketched also for formulation Kop-CT2-m, taking into account the modified expression of resource constraints, where variable x_{jc} (with $j < c$) appears not once but twice with different weights and opposite signs.

Observe that the fact that the relaxed version of a formulation admits many integers, by itself, does not constitute an advantage for the branching, or at least not in the first phases, when some fractional values are anyway returned and the search keeps going on. This brings us to the main important difference among our approach and the other models considered.

As already observed, formulation F1 generally works as follows: given a pair (c, j), the variable which is most likely selected for the first branching among those associated to such pair is θ_{cj}, being probably fractional. Whether θ_{cj} is fixed to 0 or to 1, two constraints (among the four (3.4)-(3.7)) are immediately tightened, leaving to γ_{cj} and γ_{jc} the final possibility to further influence the feasible space for the activities starting times. In particular, having a typical relaxation solution a considerable number of γ set to 0, if c and j have not to be parallelized, it is in general simple for F1 to return a solution with integer γ_{cj} and γ_{jc}, and specifically $\gamma_{cj} = 0, \gamma_{jc} = 0$. Moreover, γ_{cj} and γ_{jc} will really need to be moved from 0 only if $t_c = t_j$. All these aspects positively affect the branching procedure, which can thus be completed relatively fast.

The same considerations do not apply to Kop-CT1-m/Kop-CT2-m. Indeed, due to the characteristics of the optimal solutions provided by the continuous relaxation of nodes, the branching decisions can generally be taken both on variables of type z or x. If for example, considering formulation Kop-CT1-m and a given pair (c, j), x_{cj}

is fixed to 0 or 1, then also the x_{jc} and one among z_{cj} and z_{jc} are necessarily forced to take an integer value (due to (3.78) and (3.83)), which is the best case. Since two of these three binaries equal 1 and multiply a big number, only one constraint among (3.79) and (3.80) is effectively tightened (i.e., the one involving the x which is set to 0). The remaining variable z could eventually generate another branching in case of overlapping between c and j, otherwise can be feasibly set to 0 by the relaxation, which is convenient. Differently, when z_{cj} or z_{jc} are fixed as first, there are two possibilities. If for example $z_{cj} = 0$ is set, the situation is similar to the previous case because also the x_{cj} and x_{jc} automatically become integer. However a successive branching on variable z_{jc} is expected because $z_{cj} = 0$ and $z_{jc} = 0$ are not possible together, and so z_{jc} needs to move closer to 1, while resource constraints in general try to lower its value, potentially generating fractional quantities. The worst case anyway happens when a variable z is set to 1. In this situation no other binaries are necessarily forced to 0 or 1, and constraints (3.81) do not tight the feasible region. Thus the search continues fixing a different variable without any additional information.

Although these observations clearly simplify the real complexity of the branching procedure, considering only one pair of activities and no interactions with the others, they anyway provide an insight into some possible reasons why F1 produces better performances than Kop-CT1-m. In this respect, notice that similar arguments could be also derived for formulation Kop-CT2-m, taking of course into account the reduced number of variables x. This may lead to different selections of the variables used for branching at the first steps, potentially influencing the performance of the method. Indeed, in general there is not an evident reason to justify an experimental difference between Kop-CT1-m and Kop-CT2-m. Kopanos et al. (2014) in their work affirmed Kop-CT1 obtained better results than Kop-CT2. Actually, these formulations seem to perform similarly, with a slight advantage for the first one. We think that the different behavior can probably derive from the choice of the first variables used for branching. Having Kop-CT2-m twice as many z variables as x, the probability to prioritize fixing z increases, which generally leads to less efficient steps.

3.5 Experimental results

Starting from the results obtained in the previous section, we further investigated the performances of our proposed formulation, considering 2280 additional test problems: 1800 of set RanGen2 (split into Set 1, Set 2, Set 3, Set 4 and Set 5) with 30 activities, and 480 of set j60 of PSPLib with 60 activities. Model Kop-CT1-m is also included for comparison. Everything was implemented in JAVA using ILOG CPLEX Concert Technology 12.8.0 under standard configurations. The experiments were performed on an Intel Core i7 CPU 5600U 2.60 GHz with 16GB RAM, limiting the time for each instance to 300 CPU seconds.

Table 3.16 lists our results, where all the measures are calculated as in preceding testings with the exception of Act.Gap%, that indicates the average gap of the integer suboptimal or non-proven optimal solutions from the best known solution available.

Figure 3.5 directly compares the amounts of instances optimally solved by

Set	Formulation	Feas.%	Opt.%	Out.Gap%	Act.Gap%	Time(sec.)
j30	F1	100.00	96.88	14.27	0.96	9.64
	Kop-CT1-m	100.00	91.25	15.18	2.03	9.94
j60	F1	100.00	74.17	24.02	9.53	21.29
	Kop-CT1-m	100.00	69.17	28.06	17.84	30.77
Set 1	F1	100.00	75.44	25.70	3.82	13.65
	Kop-CT1-m	100.00	66.78	33.57	10.33	14.19
Set 2	F1	100.00	60.56	23.57	3.98	12.38
	Kop-CT1-m	100.00	53.33	33.74	9.35	27.43
Set 3	F1	100.00	91.67	11.79	1.16	16.73
	Kop-CT1-m	100.00	86.25	18.15	2.98	19.52
Set 4	F1	100.00	69.58	26.92	4.38	7.06
	Kop-CT1-m	100.00	62.50	34.07	10.24	15.63
Set 5	F1	100.00	72.50	19.82	2.89	7.41
	Kop-CT1-m	100.00	67.08	30.38	8.45	19.49

Table 3.16. Computational results for F1 and Kop-CT1-m on sets j30, j60, and sets 1-5 of RanGen2.

formulations F1 and Kop-CT1-m for each set, showing the dominance of the first model. In order to understand how the optimal solutions are distributed among the different approaches, Figure 3.6 must be considered. It basically divides the test problems of every set into four groups on the basis of the exit status provided by the algorithms. The most numerous groups are those composed of instances that were solved to optimality by both F1 and Kop-CT1-m, while only in two cases (in Set 3) Kop-CT1-m could return optimal solutions that were not found (or not certified as optimal) by F1. This means that in general F1 is able to solve the same problems as Kop-CT1-m plus some additional ones.

Figures 3.7 and 3.8 respectively compare the performances of the formulations in case of proved optimality or not. In particular, Figure 3.7 considers for each model the instances in all test sets for which the optimal solution was found, and shows the distribution of computation times over different ranges. More than the 35% of problems solved by F1 required less than 0.3 seconds, while Kop-CT1-m generally needs more time. For example, 19.85% of the optimal instances for Kop-CT1-m employ more than 20 seconds, while the same value for F1 is only 12.53%. Similar observations come out from Figure 3.8 where only the instances that reached the time limit without proving the optimality are considered, and the ranges refer to the actual gaps from the best known values. More than 50% of problems returned a solution with a gap lower than 4% for F1, while only 30% of the instances processed by Kop-CT1-m produced a gap under this threshold. To reach the 50% it is necessary to also include the ranges of (4%-6%] and (6%-8%]. Moreover, a significant difference

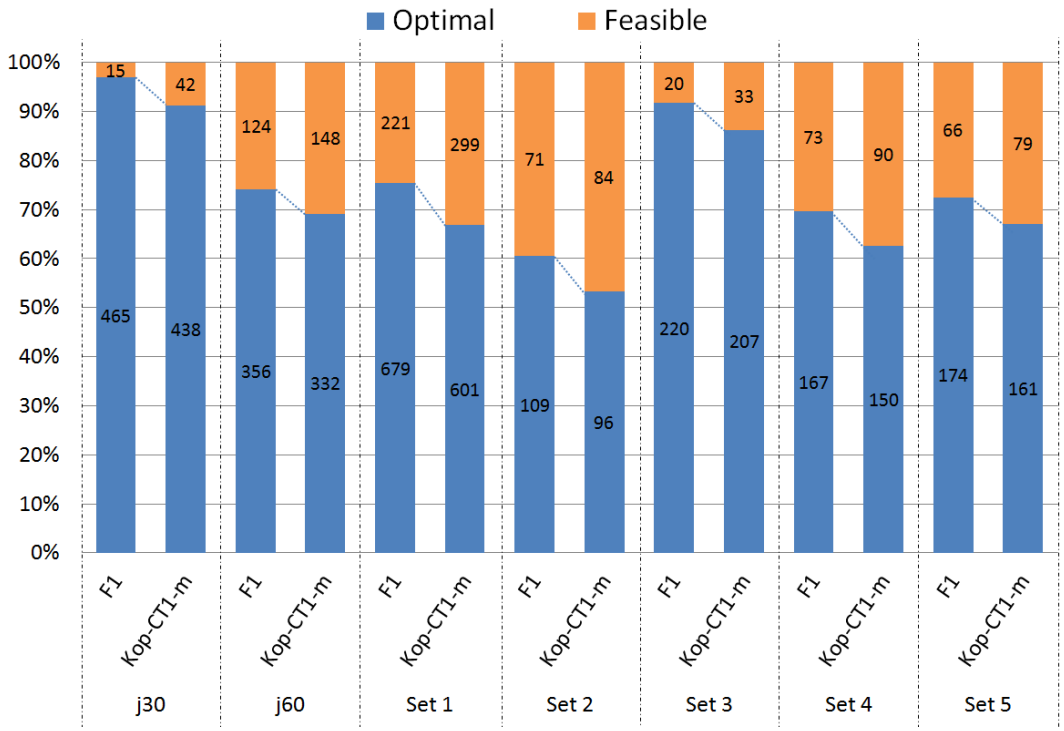

Figure 3.5. Comparison between the amounts of test problems in each set optimally or non-optimally solved by F1 and Kop-CT1-m.

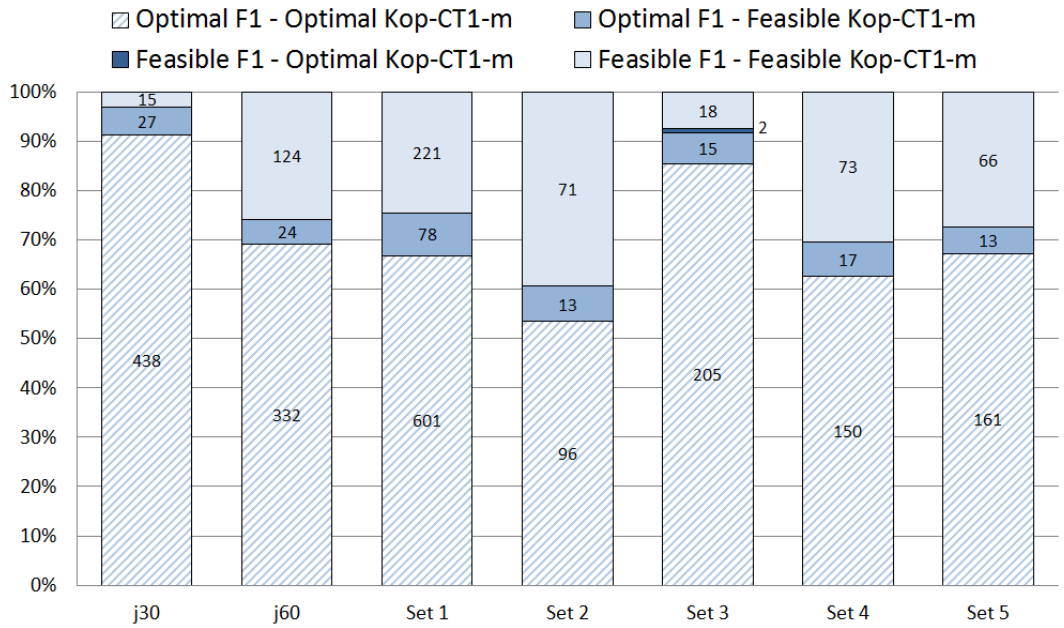

Figure 3.6. Amounts of instances in each set optimally solved by both models, by none of the two, or by one formulation and not by the other (optimal for F1 and feasible for Kop-CT1-m or optimal for Kop-CT1-m and feasible for F1).

appears on the right side of the graph, in particular for gaps over 20% (1.69% of incidence for F1 vs. 14.71% for Kop-CT1-m). Separated charts of results like 3.7 and 3.8 for each test set are provided in Appendix C, bringing to analogous conclusions.

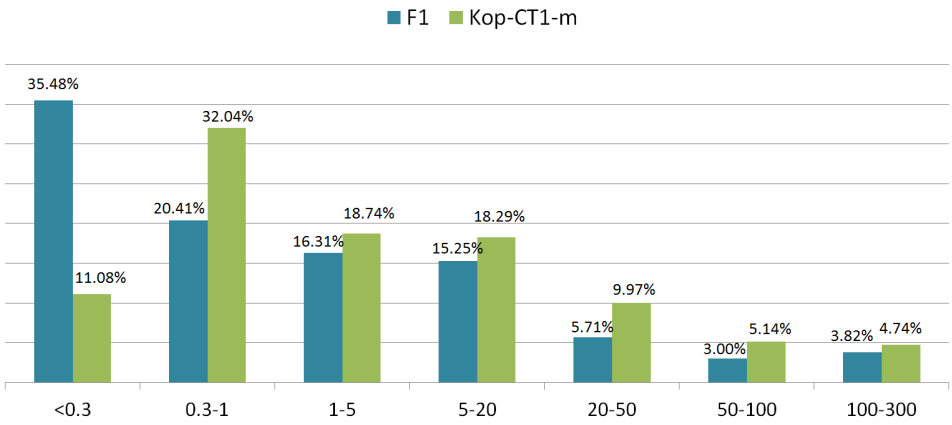

Figure 3.7. Distribution of computation times over different ranges for F1 and Kop-CT1-m, considering only the instances optimally solved among those of all sets.

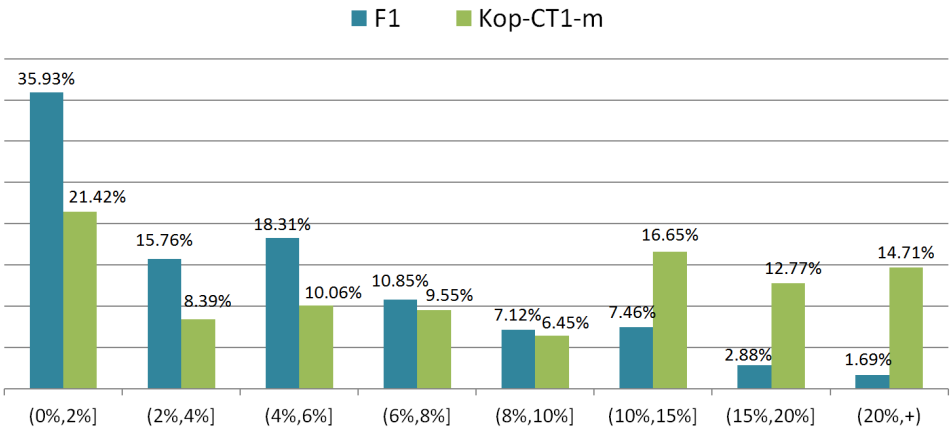

Figure 3.8. Distribution of actual gaps over different ranges for F1 and Kop-CT1-m, considering only the suboptimal or non-proven optimal instances among those of all sets.

Figure 3.9 constitutes a summary for the previous considerations, showing for each set and each formulation the percentage of instances:

- solved to optimality in less than 1 second;
- solved to optimality in more than 1 second and less than 50 seconds;
- solved to optimality in more than 50 seconds and less than 300 seconds;
- not optimally solved providing a solution equivalent to the best known;
- not optimally solved with an actual gap from the best known solution lower than 3%;

- not optimally solved with an actual gap from the best known solution higher than 3%.

Finally, Figure 3.10 collects the average values of the KPIs over all the test sets.

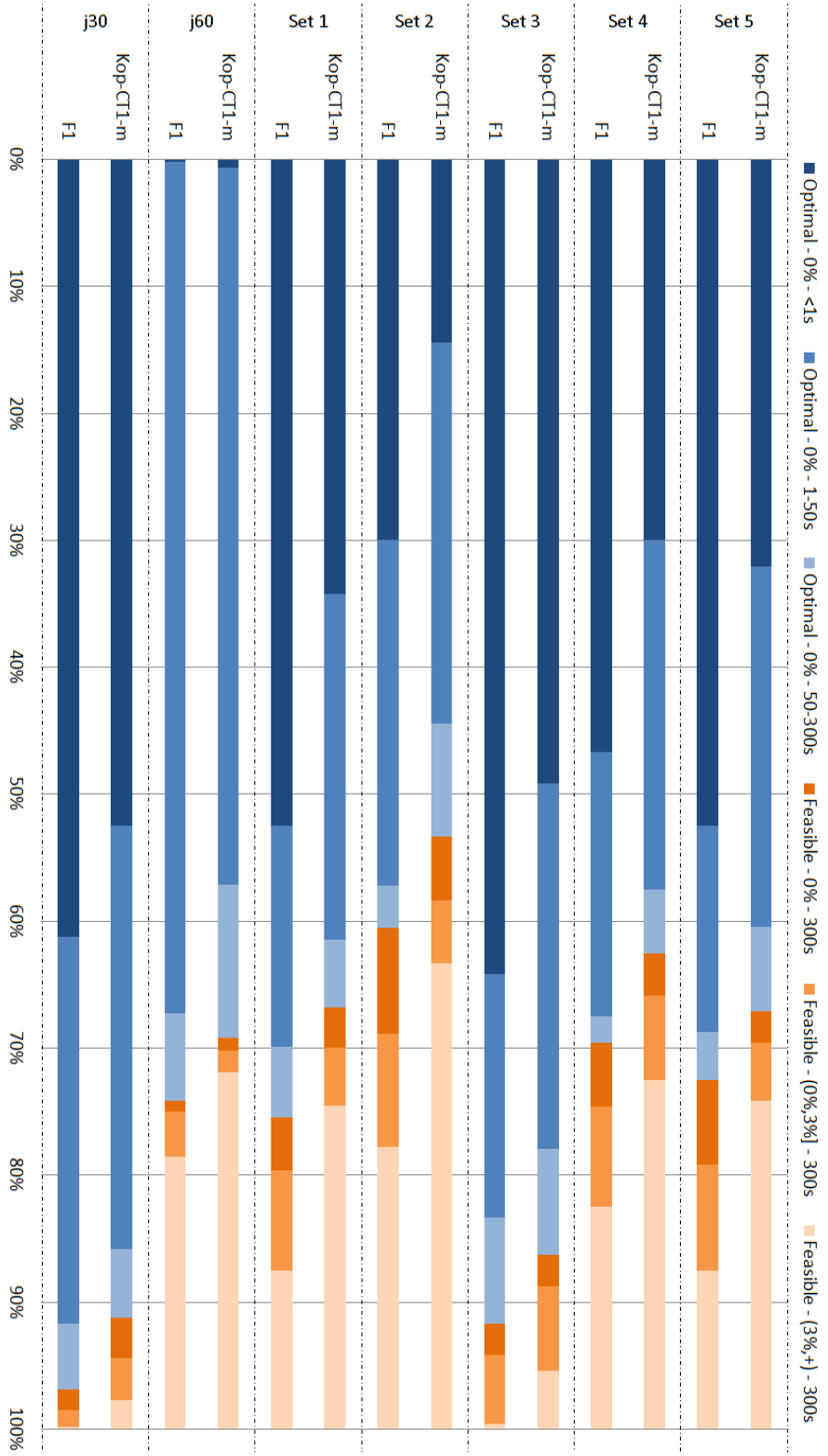

Figure 3.9. Percentages of instances solved or not solved to optimality by each method for each test set, detailed for different ranges of computation times and actual gaps from the best known solutions.

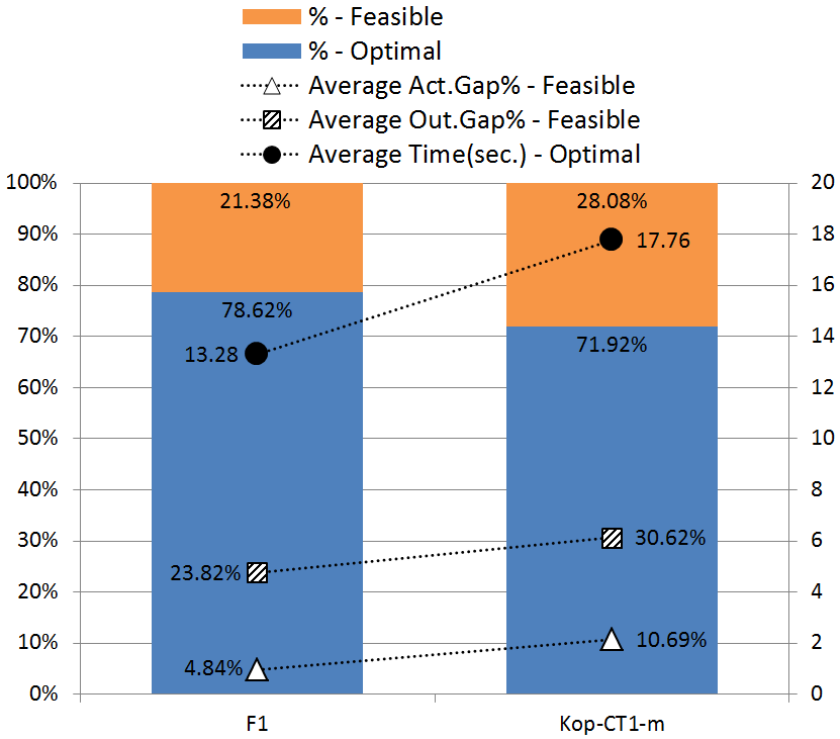

Figure 3.10. Average indicator values for all the instances considered.

Chapter 4

A new framework for workforce planning and activities scheduling

In this chapter, we extend the arguments previously addressed to provide an example of application of the MILP formulation proposed for the Resource Constrained Project Scheduling Problem.

Sections 4.1 and 4.2 are devoted to the introduction of the framework and its main components. Section 4.3 presents the main simulation-optimization aspects. The formal statement of the problem is provided in Section 4.4, and a bilevel programming formulation is proposed in Section 4.5. The framework structure is outlined in Section 4.6, while Section 4.7 concludes the chapter with three example applications.

4.1 Introduction and motivations

Nowadays, a key success factor for many large enterprises is the ability to properly manage labor cost and timetables. This is the reason why workforce planning and scheduling tools are now getting more and more developed.

Two are the typical issues arising in such applications: the first is related to the medium and long-term goal of estimating the amount of workers that the company will require in future periods. The second, mostly linked to short-term operations, involves the assignment of human resources to activities in order to meet deadlines and industrial plans.

In practice, to conduct a complete analysis and evaluate the effectiveness of a solution both time and financial objectives must be taken into account, considering not only the need of minimizing durations and delays but also the ability to limit the required budgets. The result is a trade-off problem which looks at the same time at avoiding resource underutilization and incapacity to comply with due dates.

In the following, we present a new approach to solve the workforce scheduling problem in complex applicative contexts such as manufacturing and logistics, characterized by the simultaneous processing of several activities, the occupation of wide areas, the coexistence of independent workloads, the use of advanced machineries

and, above all, the employment of different types of operators, having various abilities and experience levels.

Standard approaches usually address this issue by defining distinct planning, scheduling and allocation problems. However, within the considered context, the problem of providing the right number of workers with the right skills at the right time is inherently linked to the schedule of the activities. For this reason, we rather propose a strategy to tackle all these aspects together, taking into account a reasonable time horizon. As a result, we obtain a large problem requiring not only a suitable representation of processes complexity, but also a feasible assignment of operators to tasks and an optimized activities scheduling.

In what follows, the structure of the problem is formalized and a specialized simulation-based optimization framework is proposed.

4.2 Main components of the framework

In general, our framework applies to systems where one or more processes are executed. Roughly, a process can be described in terms of three basic definitions: the resources employed, the skills required and the component activities.

4.2.1 Resources

By *resource* we mean any operator, means of transport, machine, equipment, etc. that is scarce in nature and necessary for some operations. In particular, we refer to the concept of *renewable resources*, used in the literature to indicate that a specified number of units or quantity is available in every period. In other terms, resources as manpower and machines have a limited initial capacity which is continuously "renewed" during the project without the possibility to be exceeded at any time. In contrast, *nonrenewable resources* have a given capacity whose consumption is limited within the entire planning horizon. Typical examples are the raw materials for a production lot or the budget for a project.

In the following, we will not directly take into account nonrenewable resources; the motivation for that is not only justified by a speculative initial choice, since the analysis and solution of problems with renewable or non-renewable resources have different theoretical bases, but it derives also from practical reasons. Indeed, the applications considered for the development of our work mainly focus on workforce planning decisions, thus identifying the resources as humans, that are by definition classifiable as renewable.

Anyway, it is also worth to notice that the results provided by our approach could be easily extended to any kind of renewable resource, irrespective of considering people or things. For this reason in what follows without loss of generality we will refer to either operators, machines or other resources indistinguishably.

4.2.2 Skills

Especially when considering humans - but by extension even when dealing with an advanced machinery, for example - it is possible to list the tasks that each type of operator/resource is able to perform. In other words, in order to enable a feasible

allocation, it is important to express a form of compatibility between resources and operations. We enclose this idea in the concept of *skill*: if an operator/machine/object has different skills and, among them, the skill k, this means that he/it can be assigned to every activity requiring (also) that particular ability.

Skills can be therefore used to characterize resource types. For example, given two different resources having exactly the same skills, two possibilities are available in the decision process: the first is two consider them separately for the assignment, the second is to group them in a unique set (or type) and to admit a partial allocation that does not exceed the overall capacity of 2.

4.2.3 Activities

We are now able to introduce the last basic notion: the *activity*, representing any non-interruptible operation requiring time and resources to be completed. Activities have a duration (or processing time) and may be subject to release date and due date limits, representing organizational and strategical restrictions to be satisfied by the variable starting times. Furthermore they may be linked by some precedence constraints that are due to technological requirements and that specify which are the sequence relations among different activities.

From a scheduling point of view, activities may be considered mutually exclusive or not; this depends on their way of requiring resources. In particular, we say that a resource is *exclusive* when its overall capacity is exclusively occupied by an activity at a time for its entire duration. Examples are a specific engineer signing a project, a dedicated space for shipbuilding or a particular unit in a single machine configuration. Conversely, the electrical power in a factory, the operators in a call center or the parallel machines in a manufacturing environment are *nonexclusive* resources, since they have a capacity that can be shared between different actors at the same time.

The same argument applies also when taking into account the concept of skill. Generally speaking, we will assume that an activity can employ a certain amount of a resource capacity to cover a given skill request. Then, we will say that the use of resources is exclusive if it necessarily implies the entire capacity occupation, irrespective of involved skills.

4.3 Simulation-based optimization

From what just said, processes are sequences of activities requiring particular skills to be performed. Operators may have one or more skills and can be split in different sets representing resource types. Each type contains resources that are equally able in executing tasks and that singularly contribute in defining a common capacity for the set. Starting from this, operators of a certain type can be selected and allocated to activities to take on a role that is consistent with one of their skills.

About allocation quantities, a basic assumption of our approach is that the number of resources and skills required is not fixed and therefore there exist many feasible combinations of operators guaranteeing the completion of an activity. In particular, allowing to vary the workforce assignments between a lower and an upper limit, we evidently admit variability to operations processing times. Such aspect

heavily characterizes our procedure. Assuming it is not possible to derive analytic functions expressing the link between allocated skills and time to complete the activities, we have based our solution method on the use of a set of ad hoc simulators, having as input a vector of worker availabilities and as output a duration estimate. Figure 4.1 shows an example of an activity simulator: in correspondence of different resources and skills assignments the system returns different processing times.

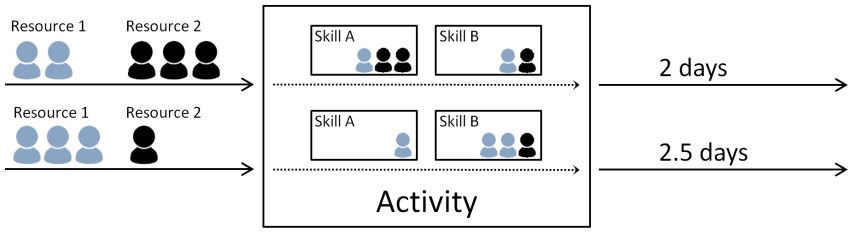

Figure 4.1. Activity simulator example model.

Notice that this methodology has several advantages in terms of adaptability to represent real processes. In fact, the use of simulation on one hand makes it possible to model even extremely complex activities, on the other allows to develop a really flexible approach, able to use exactly the same paradigm in very different cases.

Therefore, whether it is very simple calculations or intricate time-consuming simulations, the optimization engine can neglect the details and the peculiar complexity of specific cases, only providing a feasible assignment of resources and waiting for a feedback reporting the time required in that configuration.

Nevertheless, it is worth to consider the significant effects that the outlined approach has on the problem to be solved. In particular, some observations can be made:

- *The use of simulations makes the problem analytically unresolvable and not treatable by the usual optimization techniques.* In reality this comment hides two assumptions that are true in the vast majority of real applications: firstly that a complete enumeration of all operator-skill-activity assignments would be practically impossible; secondly that, even in those cases where the number of combinations is limited, it would not be practical to execute all the simulations in a preprocessing phase, due to their expensive computational cost.

- *The decision process can be defined as a black-box optimization process* where simulators are the "black boxes", whose internal structure is theoretically unknown to the optimizer. It follows from this that solving algorithms must be able to work with an almost null or only partial knowledge of the problem structure, facing in practice the challenge of finding an optimal solution without taking advantage of analytical search methods and optimality test criteria.

- *In general, few hypotheses can be made.* This is a further consequence of the use of simulators, due to the undetermined form of functions they mimic. However, while maintaining a generic definition of the goals to be optimized, it is still possible to draw some non-unrealistic assumptions about the mathematical

structure of the problem under consideration. For example, we can generally suppose that:

(a) all other inputs being fixed, increasing the number of operators of a certain type assigned to a given activity contributes to reduce the required processing time;

(b) too high allocations of resources or too long process durations imply excessive costs or unfeasible solutions. Thus, two opposite effects always appear in the objective function or in the constraints. The result is a simulation-optimization problem facing a typical trade-off between different goals. On the one hand it aims at reducing the employment cost, minimizing the number of necessary skilled operators, on the other, it encourages an optimal activities scheduling, trying to parallelize the tasks and decrease the overall completion time.

4.4 Problem statement

In order to introduce the general mathematical formulation we have developed for this problem, we first need to list some basic definitions.

We will consider m non-preemptive activities indexed by the set

$$A = \{1, ..., j, ..., m\}.$$

Each activity is characterized by a variable continuous-valued processing time. Furthermore a release date r_j and a due date d_j are defined for the j-th activity. Both r_j and d_j are real nonnegative parameters, eventually set to zero and infinity to relax the restrictions.

The set of indexes for the n resources considered is given by

$$R = \{1, ..., i, ..., n\}.$$

Then we consider p different skills. Let

$$S = \{1, ..., k, ..., p\}$$

be the associated set of indexes.

We define V as the set of non-parallelizzable activity index pairs, that is all the combinations of two activities that for some reason cannot be executed at the same time, e.g. because they have a logical conflict or because they use the same machine, space, or other exclusive resource that is not worth to explicitly consider in the model.

Precedence relations are given by the set Q of ordered index pairs, such that $(\jmath^1, \jmath^2) \in Q$ means that the execution of activity $\jmath^2$ must start after the end of activity $\jmath^1$. These predefined sequences are supposed consistent with each other.

Our problem formulation involves three main types of decision variables. First, the total number of operators made available for each resource type is represented by a vector $y \in \mathbb{N}^n$, such that y_i denotes the availability of resource i. Second, integer variables x_{ijk} are required to indicate the number of workers of type i assigned

to activity j to cover the request of skill k. Finally, the starting-time continuous variables t_j are introduced for each activity j, thus making the scheduling possible.

We assume all variables to be nonnegative. Furthermore, for every triple (i, j, k) we bound x_{ijk} between a lower (l_{ijk}) and an upper (u_{ijk}) limit, indicating the minimum and maximum requirements of resource i with skill k for activity j. Similarly, we denote by l_i and u_i (with $0 \leq l_i \leq u_i$) the limits for variable y_i, and by l_j, u_j, l_{jk}, u_{jk}, l_{ij} and u_{ij} respectively the minimum and maximum requirements for each activity, activity-skill or for each resource-activity.In addition, l_{ik} and u_{ik} allow to control the amount of resources of type i assigned to tasks involving skill k.

All the previous restrictions can be used to specify logical or physical constraints, and respectively have zero and infinity as default values.

Ultimately, minding our assumption on the dependence among operators assignments and time to complete the activities, we can identify the output of the j-th simulator with the symbol $\tau_j(x_{\cdot j\cdot}) = \phi_j(x_{1j1}, .., x_{ijk}, .., x_{njp})$, so expressing the processing time of activity j as an unknown function of the variable resource allocations.

Starting from these definitions and formally indicating with $A(t)$ the set of activities in process at time t, a conceptual formulation of the problem can be written as follows:

$$\min_{x,y,t} f(x, y, t) \tag{4.1}$$

$$s.t.\ \ l_{ijk} \leq x_{ijk} \leq u_{ijk} \qquad i \in R,\ j \in A,\ k \in S, \tag{4.2}$$

$$l_i \leq y_i \leq u_i \qquad i \in R, \tag{4.3}$$

$$l_j \leq \sum_{i \in R} \sum_{k \in S} x_{ijk} \leq u_j \qquad j \in A, \tag{4.4}$$

$$l_{ij} \leq \sum_{k \in S} x_{ijk} \leq u_{ij} \qquad i \in R,\ j \in A, \tag{4.5}$$

$$l_{jk} \leq \sum_{i \in R} x_{ijk} \leq u_{jk} \qquad j \in A,\ k \in S, \tag{4.6}$$

$$l_{ik} \leq \sum_{j \in A} x_{ijk} \leq u_{ik} \qquad i \in R,\ k \in S, \tag{4.7}$$

$$\max_{j \in A} \sum_{k \in S} x_{ijk} \leq y_i \leq \sum_{j \in A} \sum_{k \in S} x_{ijk} \qquad i \in R, \tag{4.8}$$

$$\tau_j(x_{\cdot j\cdot}) = \phi_j(x_{1j1}, .., x_{ijk}, .., x_{njp}) \qquad j \in A, \tag{4.9}$$

$$r_j \leq t_j \leq d_j - \tau_j(x_{\cdot j\cdot}) \qquad j \in A, \tag{4.10}$$

$$t_{j^2} \geq t_{j^1} + \tau_{j^1}(x_{\cdot j^1\cdot}) \qquad (j^1, j^2) \in Q, \tag{4.11}$$

$$t_{j^1} \geq t_{j^2} + \tau_{j^2}(x_{\cdot j^2\cdot}) \vee t_{j^2} \geq t_{j^1} + \tau_{j^1}(x_{\cdot j^1\cdot}) \qquad \{j^1, j^2\} \in V, \tag{4.12}$$

$$\sum_{j \in A(t)} \sum_{k \in S} x_{ijk} \leq y_i \qquad i \in R, \tag{4.13}$$

$$y_i \in \mathbb{N} \qquad i \in R, \tag{4.14}$$

$$x_{ijk} \in \mathbb{N} \qquad i \in R,\ j \in A,\ k \in S, \tag{4.15}$$

$$t_j \in \mathbb{R}^+ \qquad j \in A. \qquad (4.16)$$

The objective function is expressed in a generic form by (4.1), involving all the variables x, y, t. Constraints (4.2) and (4.3) are the bounds for variables x_{ijk} and y_i. Expressions (4.4), (4.5) and (4.6) represent limits on the overall amount of resources respectively for each activity, for each resource type and activity, and for each activity and skill. Relations (4.7) allow to govern the type of tasks assigned to the different resources, restricting the amounts of activity assignments requiring particular skills. Constraints (4.8) express two concepts: the availability of operators of type i must be *(i)* enough to guarantee that each activity can be independently executed (e.g., if scheduled in sequence with the others), and *(ii)* not more than the total amount of resources that would be needed if all the activities were parallelized. Equations (4.9) bring processing time simulations into the formulation. Constraints (4.10) give release date and deadlines limits, while inequalities (4.11) describe the precedence relations between activities. The disjunctive expressions in (4.12) ensure that pairs of activities belonging to V are not executed at the same time, in particular specifying that either activity $\jmath^1$ starts after the end of activity $\jmath^2$ or the opposite. Constraints (4.13) indicate only the conceptual relation between available and allocated operators, i.e. the sum of resources which are busy at a certain generic time t cannot exceed the total number of workers, for each type $i \in R$. This purely descriptive form can be converted in a MILP structure, paying the cost of introducing a (finite) large number of constraints depending on m and n. Finally, (4.14)-(4.16) define problem variables.

4.5 A bilevel programming formulation

The presence of simulators naturally imposes to split the solving procedure into different parts. This is the reason why we formalize the problem in a bilevel programming formulation, which has as upper-level and lower-level objectives two generic functions. Their global effect can be thought of as the combination of two conflicting components: the first accounting for the workforce cost, the second expressing a time objective. As an example of this trade-off, we can consider a situation where variables y_i are, at the same time, pushed down to lower salaries expenses, and pushed up to relax resource constraints and obtain better results in activities scheduling, improving, for example, the overall *makespan*, the sum of projects completion times or the average finish time of activities.

The following formulation is proposed:

$$\min_{x,y,\tau,t,\delta,\gamma,\theta} \; f_1(x, y, \tau, t, \delta, \gamma, \theta) \qquad (4.17)$$

$$s.t. \quad l_{ijk} \leq x_{ijk} \leq u_{ijk} \qquad i \in R,\ j \in A,\ k \in S, \quad (4.18)$$

$$l_i \leq y_i \leq u_i \qquad i \in R, \quad (4.19)$$

$$l_j \leq \sum_{i \in R} \sum_{k \in S} x_{ijk} \leq u_j \qquad j \in A, \quad (4.20)$$

$$l_{ij} \leq \sum_{k \in S} x_{ijk} \leq u_{ij} \qquad i \in R,\ j \in A, \quad (4.21)$$

$$l_{jk} \leq \sum_{i \in R} x_{ijk} \leq u_{jk} \qquad j \in A,\ k \in S, \quad (4.22)$$

$$l_{ik} \leq \sum_{j \in A} x_{ijk} \leq u_{ik} \qquad i \in R,\ k \in S, \quad (4.23)$$

$$\sum_{k \in S} x_{ijk} \leq y_i \qquad i \in R,\ j \in A, \quad (4.24)$$

$$y_i \leq \sum_{j \in A} \sum_{k \in S} x_{ijk} \qquad i \in R, \quad (4.25)$$

$$\tau_j = \phi_j(x_{1j1}, .., x_{ijk}, .., x_{njp}) \qquad j \in A, \quad (4.26)$$

$$y_i \in \mathbb{N} \qquad i \in R, \quad (4.27)$$

$$x_{ijk} \in \mathbb{N} \qquad i \in R,\ j \in A,\ k \in S, \quad (4.28)$$

$$\tau_j \in \mathbb{R}^+ \qquad j \in A, \quad (4.29)$$

$$(t, \delta, \gamma, \theta) \in \arg\min_{t,\delta,\gamma,\theta} f_2(t, \delta, \gamma, \theta) \quad (4.30)$$

$$s.t. \quad r_j \leq t_j \leq d_j - \tau_j \qquad j \in A, \quad (4.31)$$

$$t_{\jmath^2} \geq t_{\jmath^1} + \tau_{\jmath^1} \qquad (\jmath^1, \jmath^2) \in Q, \quad (4.32)$$

$$t_{\jmath^1} + \tau_{\jmath^1} - t_{\jmath^2} \leq M(1 - \delta_{\jmath^1\jmath^2}) \qquad \{\jmath^1, \jmath^2\} \in V, \quad (4.33)$$

$$t_{\jmath^2} + \tau_{\jmath^2} - t_{\jmath^1} \leq M\delta_{\jmath^1\jmath^2} \qquad \{\jmath^1, \jmath^2\} \in V, \quad (4.34)$$

$$\sum_{k \in S} x_{i\jmath^1 k} + \sum_{\substack{\jmath^2 \in A \\ \jmath^2 \neq \jmath^1}} \sum_{k \in S} x_{i\jmath^2 k}\gamma_{\jmath^1\jmath^2} \leq y_i \qquad i \in R,\ \jmath^1 \in A, \quad (4.35)$$

$$t_{\jmath^1} + \tau_{\jmath^1} - t_{\jmath^2} \leq M(1 - \theta_{\jmath^1\jmath^2}) + (\tau_{\jmath^1} - \varepsilon)\gamma_{\jmath^1\jmath^2} \qquad \{\jmath^1, \jmath^2\} \in A^2, \quad (4.36)$$

$$t_{\jmath^2} - t_{\jmath^1} \leq M\theta_{\jmath^1\jmath^2} + \varepsilon\gamma_{\jmath^1\jmath^2} - \varepsilon \qquad \{\jmath^1, \jmath^2\} \in A^2, \quad (4.37)$$

$$t_{\jmath^2} + \tau_{\jmath^2} - t_{\jmath^1} \leq M\theta_{\jmath^1\jmath^2} + \tau_{\jmath^2}\gamma_{\jmath^2\jmath^1} \qquad \{\jmath^1, \jmath^2\} \in A^2, \quad (4.38)$$

$$t_{\jmath^1} - t_{\jmath^2} \leq M(1 - \theta_{\jmath^1\jmath^2}) - \varepsilon \qquad \{\jmath^1, \jmath^2\} \in A^2, \quad (4.39)$$

$$\theta_{\jmath^1\jmath^2} \in \{0, 1\} \qquad \{\jmath^1, \jmath^2\} \in A^2, \quad (4.40)$$

$$\gamma_{\jmath^1\jmath^2} \in \{0, 1\} \qquad \{\jmath^1, \jmath^2\} \in A^2, \quad (4.41)$$

$$\gamma_{\jmath^2\jmath^1} \in \{0, 1\} \qquad \{\jmath^1, \jmath^2\} \in A^2, \quad (4.42)$$

$$\delta_{\jmath^1\jmath^2} \in \{0, 1\} \qquad \{\jmath^1, \jmath^2\} \in V, \quad (4.43)$$

$$t_j \in \mathbb{R}^+ \qquad j \in A, \quad (4.44)$$

where:

- M is a large constant (e.g., $M = \sum_{j \in A} \tau_j$);
- ε is a small parameter (e.g., $\varepsilon = 10^{-(\bar{\nu}+1)}$, with $\bar{\nu} = \min\{\nu \in \mathbb{N} : 10^{\nu}\tau_j \in \mathbb{N}, \forall\ j \in A\}$);
- $A^2 = \{\{\jmath^1, \jmath^2\} : \jmath^1 \in A,\ \jmath^2 \in A,\ \jmath^1 \neq \jmath^2\}$.

In this formulation, constraints (4.24)-(4.25) coincide with (4.8), inequalities (4.33)-(4.34) express (4.12), and (4.35)-(4.39) assure (4.13).

Notice that if $f_2(t, \delta, \gamma, \theta)$ is a linear function – that is the case of many typical scheduling objectives – the inner formulation results to be a MILP model. Thus, depending on the size of the instance considered, it can be exactly (or approximately, if a gap greater than 0 is admitted) solved with standard branch and bound techniques, as well as separately tackled with ad-hoc heuristic procedures.

4.6 The structure of the framework

Our solution framework is composed of three main nested blocks, as shown in Figure 4.2. The most external one is a *black-box* optimization formulation working on variables y_i and x_{ijk}, subject to constraints (4.18)-(4.25) and (4.27)-(4.28). Its objective function, denoted by $\tilde{f}$, has the structure of (4.17) and is calculated every time from the results of inner blocks.

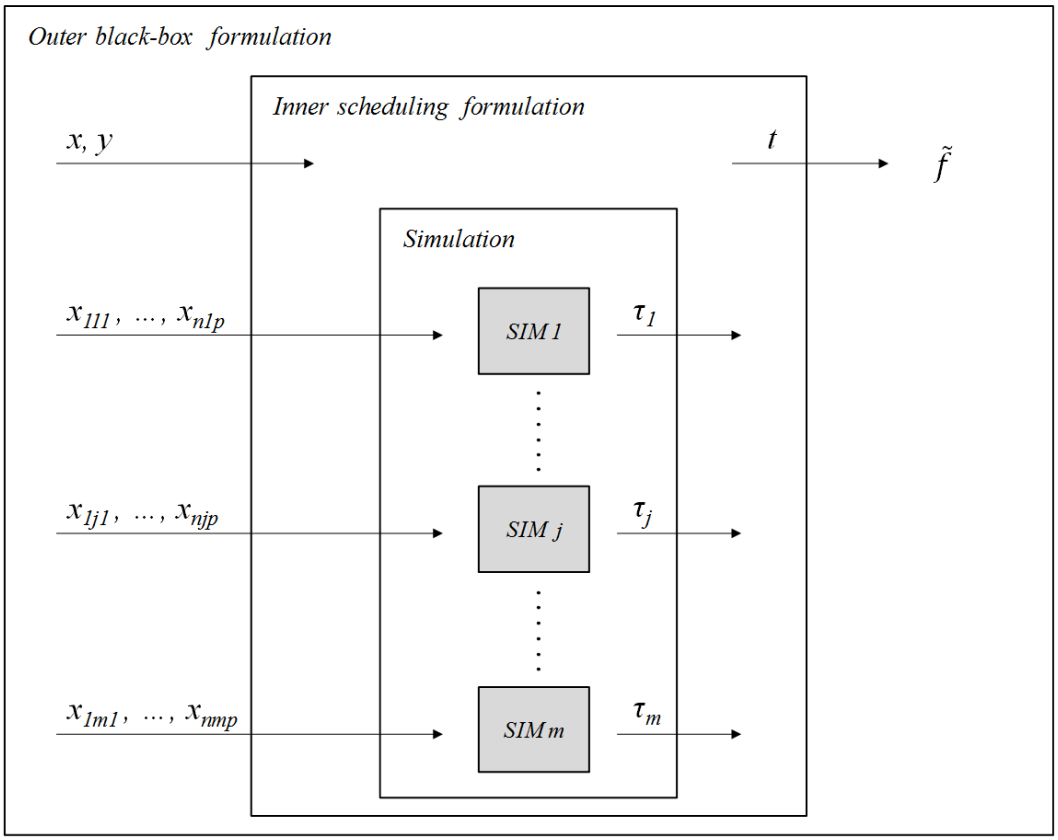

Figure 4.2. Framework structure.

In turn, the second module, represented by the resource constrained scheduling formulation given by (4.30)-(4.44), is (approximately) solved at every iteration, immediately after the execution of the third block, that takes the x_{ijk} as inputs, runs a parallel simulation for each activity j, and returns the processing times τ_j. If a

feasible solution of the inner scheduling model does not exist for certain assignments of resources (or if it is not found within a predefined time limit), a message must be sent to the outer block, which automatically generates a new proposal to be tested.

Figure 4.3 provides a schematic representation of a typical iteration of the solving procedure.

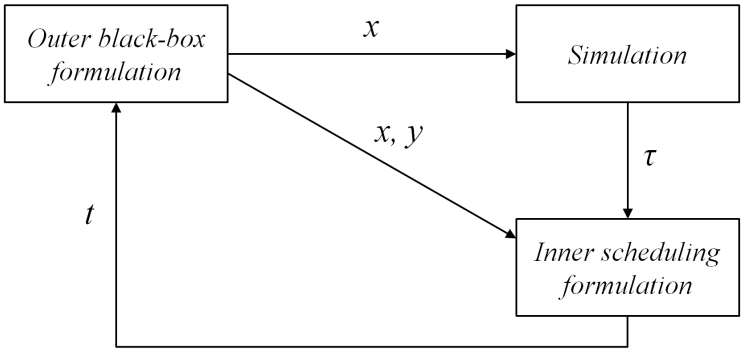

Figure 4.3. Schematic representation of a typical iteration of the framework.

4.7 Examples of applications

Some examples of applications of the proposed framework are provided in this section. They present simplified versions of the real restrictions influencing each case, but allow to clarify the modeling flexibility offered by the combination of simulation and optimization tools in a single black-box framework.

4.7.1 Application 1: Big plant construction

Big plant construction projects may be influenced by many elements of uncertainty and variability. A possible option to try managing the associated risks is to use simulators able to take into account such disturbing effects and provide as output high-fidelity estimates and/or worst-case results for each activity. In this way, the tool can be employed to establish the right number of operators of each type that would be necessary for the entire period, determining also their tasks assignments. In general, the skills are not relevant in these decisions, which only depend on the available resource types. Many precedence relations and some conflicts are defined between the activities, due to the complexity of the processes considered. A simple prototype of data for illustration is given by the following:

- $|A| = m$;
- $|R| = n$;
- $|S| = 1$;
- $|Q| = q > 0$;
- $|V| = v > 0$;

- $l_{ik} = l_i = 0, u_{ik} = u_i = \infty, \forall i \in R$;
- $l_{ijk} = l_{ij} \geq 0, u_{ijk} = u_{ij} \geq l_{ij}, \forall i \in R, \forall j \in A$;
- $l_{jk} = l_j \geq 0, u_{jk} = u_j \geq l_j, \forall j \in A$;
- $r_j \geq 0, d_j = \infty, \forall j \in A$;
- $f_2 = makespan$;
- $f_1 = makespan \sum_{i \in R} c_i y_i$, with $c_i > 0$ representing the cost per period of each resource of type $i \in R$.

4.7.2 Application 2: Software development projects

There are some contexts in which the skills involved are fundamental for the execution of tasks, that may be performed in several ways. An example is the development of complex pieces of software, subject to tightening due dates. In such a case, resource types generally coincide with particular individuals which are assumed to be known at the start of the planning horizon. Different activities requiring many skills are involved and some sequence constraints are defined. Simulators can be used to model the particular abilities of workers as well as their actual availability during each day. The objectives may be many. For instance, to keep as low as possible the (weighted) quantity of personnel allocated to the project while meeting all the due dates, one could consider:

- $|A| = m$;
- $|R| = n$;
- $|S| = p$;
- $|Q| = q > 0$;
- $|V| = 0$;
- $l_i = 0, u_i = 1, \forall i \in R$ (since resource types are single individuals, $y_i \in \{0, 1\}$);
- $l_{ijk} = 0, u_{ijk} = 1, \forall i \in R, \forall j \in A, \forall k \in S$ (since resource types are single individuals, $x_{ijk} \in \{0, 1\}$);
- $l_{ij} = 0, u_{ij} = 1, \forall i \in R, \forall j \in A$;
- $l_j \geq 0, u_j \geq l_j, \forall j \in A$;
- $l_{jk} \geq 0, u_{jk} \geq l_{jk}, \forall j \in A, \forall k \in S$;
- $l_{ik} \geq 0, u_{ik} \geq l_{ik}, \forall i \in R, \forall k \in S$ (for example, $l_{ik} = 1$ to assign at least a task requiring skill k to resource i);
- $r_j = 0, \forall j \in A$;
- $d_j < \infty$, for some $j \in A$;

- $f_2 = \sum_{j \in A} t_j$;
- $f_1 = \sum_{i \in R} w_i y_i$, with w_i representing the weight of resource $i \in R$.

4.7.3 Application 3: Operators allocation in manufacturing and logistic processes

Manufacturing and logistic processes are naturally characterized by a typical complexity which is due to many coexisting aspects. In such fields the estimations of workloads are often really difficult to be addressed, and in several cases practically impossible through the use of standard tools only. The development of suitable simulation models able to reproduce the reality and predict the required processing times frequently offers a valid answer in these contexts. Thus, assuming to have the opportunity to simulate many different scenarios for each activity, corresponding to distinct allocations of operators to tasks, it is possible to perform an extensive "what-if" analysis which does not only evaluate the best configuration for a particular process, but considers the entire system and balances the needs in order to find a solution which is acceptable in general. For example, if several activities are subject to deadlines (referred as $\bar{d}_j, \forall j \in A$, to be distinguished by the mandatory due dates d_j), the objective could realistically be to minimize the overall tardiness, taking into account every process and properly distributing the available operators of each type, supposed known a priori and denoted by $n_i, \forall i \in R$. The input data in this case would involve:

- $|A| = m$;
- $|R| = n$;
- $|S| = p$;
- $|Q| = q \geq 0$;
- $|V| = v \geq 0$;
- $l_i = u_i = n_i > 0, \forall i \in R$;
- $l_{ijk} \geq 0, u_{ijk} \geq l_{ijk}, \forall i \in R, \forall j \in A, \forall k \in S$;
- $l_{ij} \geq 0, u_{ij} \geq l_{ij}, \forall i \in R, \forall j \in A$;
- $l_j \geq 0, u_j \geq l_j, \forall j \in A$;
- $l_{jk} \geq 0, u_{jk} \geq l_{jk}, \forall j \in A, \forall k \in S$;
- $l_{jk} \geq 0, u_{jk} \geq l_{jk}, \forall j \in A, \forall k \in S$;
- $r_j \geq 0, \forall j \in A$;
- $d_j < \infty$, for some $j \in A$;
- $f_2 = \sum_{j \in A} \max\{0, t_j + \tau_j - \bar{d}_j\}$;

- $f_1 = \sum_{i \in R} \sum_{j \in A} \sum_{k \in S} c_{ijk} x_{ijk} \tau_j + \rho \sum_{j \in A} \max\{0, t_j + \tau_j - \bar{d}_j\}$, where the first term represents an estimation of the allocation cost and the second is the overall tardiness, multiplied by a general scaling factor ρ.

Conclusions

In this dissertation we presented a new mixed-integer linear formulation for the RCPSP. The proposed model was compared to one of the best state-of-the-art methods from the literature, outperforming it on all the experiments and on all the KPIs considered.

The new formulation avoids the discretization of time, which is treated as a continuous variable. This on the one hand permits to write the model in a compact form to be directly passed to standard solvers, on the other allows to deal with problem instances where activity durations and resource requirements are possibly non-integer valued, without incurring into approximation errors or excessive modeling sizes.

Binary indicator variables are defined to identify the overlapping of activities and their sequence. The interconnection between these variables and the associated big-M constraints is deeply analyzed and studied in order to get insights on the reasons why such good performance is achieved on standard instances.

Furthermore, a potential application to workforce planning and scheduling problems is presented as a black-box simulation-optimization approach, which employs suitable simulators to flexibly model complex activities, and involves the definition of a bilevel programming formulation to address conflicting time-cost objectives at the same time.

Three main future research directions are envisaged for this work. The first is intended to further investigate the performance of the proposed mathematical model, with a particular focus to more difficult and larger instances of the RCPSP. It is true, indeed, that MILP approaches are by nature subject to computational limitations. However, different improving steps can be conceived, as the introduction of suitable preprocessing techniques to tighten constraints and reduce the number of variables.

A second opportunity is offered by the willing to fully leverage the new modeling approach presented in the second chapter of this dissertation, with the aim of applying it to extensions of the RCPSP and other important problems characterized by availability constraints.

Last but not least, the simulation-optimization framework proposed in Chapther 4 will be further developed and tested. The aim is to extend it to fully represent more complex situations and to employ it as a decision support tool in real applications.

Appendix A

Detailed results for set j30 of PSPLib

The results obtained by formulations F1, Kop-CT1-m, and Kop-CT2-m on each instance of set j30 of PSPLib are reported in the table. For every method, there are listed:

- the output status;
- the objective value;
- the computation time in seconds;
- the output gap provided by CPLEX;
- the actual gap from the optimum;
- the number of nodes visited.

instance		F1						Kop-CT1-m						Kop-CT2-m					
1	1	Opt	43	0.14	0	0	0	Opt	43	1.36	0	0	1076	Opt	43	1.72	0	0	1103
1	2	Opt	47	0.84	0	0	555	Opt	47	1.34	0	0	1120	Opt	47	3.23	0	0	3066
1	3	Opt	47	0.56	0	0	232	Opt	47	0.9	0	0	742	Opt	47	1.19	0	0	540
1	4	Opt	62	0.76	0	0	303	Opt	62	3.95	0	0	4668	Opt	62	5.37	0	0	2595
1	5	Opt	39	0.97	0	0	655	Opt	39	2.79	0	0	2893	Opt	39	1.76	0	0	1023
1	6	Opt	48	0.62	0	0	465	Opt	48	1.11	0	0	789	Opt	48	1.58	0	0	750
1	7	Opt	60	0.48	0	0	103	Opt	60	0.7	0	0	18	Opt	60	0.72	0	0	92
1	8	Opt	53	0.13	0	0	0	Opt	53	0.25	0	0	0	Opt	53	0.23	0	0	0
1	9	Opt	49	1.43	0	0	1840	Opt	49	2.03	0	0	2322	Opt	49	3.6	0	0	3891
1	10	Opt	45	0.78	0	0	658	Opt	45	1.23	0	0	922	Opt	45	1.78	0	0	1312
2	1	Opt	38	0.59	0	0	122	Opt	38	0.51	0	0	0	Opt	38	1.34	0	0	561
2	2	Opt	51	0.8	0	0	480	Opt	51	0.53	0	0	119	Opt	51	1.73	0	0	1118
2	3	Opt	43	0.12	0	0	0	Opt	43	0.23	0	0	0	Opt	43	0.23	0	0	0
2	4	Opt	43	0.59	0	0	73	Opt	43	0.53	0	0	12	Opt	43	0.75	0	0	0
2	5	Opt	51	0.36	0	0	0	Opt	51	0.65	0	0	148	Opt	51	0.84	0	0	153
2	6	Opt	47	0.39	0	0	0	Opt	47	0.47	0	0	0	Opt	47	1	0	0	154
2	7	Opt	47	0.7	0	0	63	Opt	47	0.39	0	0	0	Opt	47	1.48	0	0	747
2	8	Opt	54	0.41	0	0	0	Opt	54	0.78	0	0	24	Opt	54	1.55	0	0	1148
2	9	Opt	54	0.7	0	0	215	Opt	54	1.06	0	0	750	Opt	54	1.09	0	0	399
2	10	Opt	43	0.89	0	0	458	Opt	43	0.73	0	0	254	Opt	43	1.23	0	0	805
3	1	Opt	72	0.34	0	0	0	Opt	72	0.3	0	0	0	Opt	72	0.37	0	0	0
3	2	Opt	40	0.53	0	0	70	Opt	40	1.36	0	0	0	Opt	40	0.61	0	0	155
3	3	Opt	57	0.55	0	0	0	Opt	57	0.37	0	0	0	Opt	57	0.61	0	0	6
3	4	Opt	98	0.28	0	0	0	Opt	98	0.3	0	0	0	Opt	98	0.33	0	0	0
3	5	Opt	53	0.55	0	0	206	Opt	53	0.64	0	0	42	Opt	53	0.97	0	0	289
3	6	Opt	54	0.48	0	0	123	Opt	54	0.39	0	0	0	Opt	54	0.61	0	0	17
3	7	Opt	48	0.56	0	0	178	Opt	48	0.44	0	0	9	Opt	48	0.53	0	0	0
3	8	Opt	54	0.47	0	0	34	Opt	54	0.34	0	0	0	Opt	54	0.67	0	0	56
3	9	Opt	65	0.55	0	0	440	Opt	65	0.66	0	0	289	Opt	65	1	0	0	536
3	10	Opt	59	0.55	0	0	411	Opt	59	0.36	0	0	8	Opt	59	0.58	0	0	165
4	1	Opt	49	0.17	0	0	0	Opt	49	0.2	0	0	0	Opt	49	0.2	0	0	0
4	2	Opt	60	0.23	0	0	0	Opt	60	0.2	0	0	0	Opt	60	0.67	0	0	190
4	3	Opt	47	0.52	0	0	213	Opt	47	0.27	0	0	0	Opt	47	0.48	0	0	0

4	4	Opt	57	0.14	0	0	0	Opt	57	0.19	0	0	0	Opt	57	0.19	0	0	0
4	5	Opt	59	0.5	0	0	0	Opt	59	1.19	0	0	0	Opt	59	0.38	0	0	0
4	6	Opt	45	0.17	0	0	0	Opt	45	0.23	0	0	0	Opt	45	0.2	0	0	0
4	7	Opt	56	0.45	0	0	88	Opt	56	0.25	0	0	0	Opt	56	0.47	0	0	0
4	8	Opt	55	0.34	0	0	0	Opt	55	0.22	0	0	0	Opt	55	0.17	0	0	0
4	9	Opt	38	0.14	0	0	0	Opt	38	0.23	0	0	0	Opt	38	0.22	0	0	0
4	10	Opt	48	0.36	0	0	0	Opt	48	0.38	0	0	0	Opt	48	0.55	0	0	0
5	1	Opt	53	2.37	0	0	3696	Opt	53	8.7	0	0	4173	Opt	53	9.36	0	0	3750
5	2	Opt	82	2.76	0	0	4447	Opt	82	15.44	0	0	6135	Opt	82	13.9	0	0	5213
5	3	Opt	76	5.4	0	0	3954	Opt	76	36.32	0.01	0	12745	Opt	76	32.4	0	0	9715
5	4	Opt	63	16.68	0	0	17514	Opt	63	30.78	0	0	10892	Opt	63	45.3	0	0	13500
5	5	Opt	76	2.36	0	0	3871	Opt	76	11.75	0	0	5533	Opt	76	9.75	0	0	3623
5	6	Opt	64	4.2	0	0	3651	Opt	64	24.29	0	0	11859	Opt	64	13.39	0	0	6236
5	7	Opt	76	8.71	0	0	7097	Opt	76	79.31	0	0	19323	Opt	76	60.04	0	0	15285
5	8	Opt	67	9.27	0	0	9209	Opt	67	28.14	0	0	10236	Opt	67	40.19	0	0	15752
5	9	Opt	49	3.6	0	0	2548	Opt	49	10.12	0	0	5052	Opt	49	16.16	0	0	6179
5	10	Opt	70	2.04	0	0	2133	Opt	70	7.63	0	0	3569	Opt	70	10.33	0	0	3839
6	1	Opt	59	4.66	0	0	3897	Opt	59	9.19	0	0	4996	Opt	59	14.82	0	0	7975
6	2	Opt	51	2.61	0	0	2617	Opt	51	4.06	0	0	2983	Opt	51	8.53	0	0	4996
6	3	Opt	48	3.31	0	0	2902	Opt	48	2.65	0	0	2612	Opt	48	2.09	0	0	1514
6	4	Opt	42	3.95	0	0	3680	Opt	42	5.02	0	0	3365	Opt	42	9.17	0	0	4767
6	5	Opt	67	8.49	0	0	19224	Opt	67	3.92	0	0	4593	Opt	67	3.49	0	0	3454
6	6	Opt	37	1.51	0	0	1257	Opt	37	2.96	0	0	3210	Opt	37	2.28	0	0	1546
6	7	Opt	46	1.26	0	0	1732	Opt	46	1.48	0	0	1619	Opt	46	2.51	-2.17	0	2525
6	8	Opt	39	4.13	0	0	3625	Opt	39	7.05	0	0	4561	Opt	39	4.99	0	0	2789
6	9	Opt	51	0.72	0	0	392	Opt	51	1.28	0	0	1014	Opt	51	2.64	0	0	2421
6	10	Opt	61	3.82	0	0	3804	Opt	61	6.02	0	0	3783	Opt	61	6.18	0	0	3584
7	1	Opt	55	0.52	0	0	234	Opt	55	2.03	0	0	0	Opt	55	1.03	0	0	170
7	2	Opt	42	0.95	0	0	783	Opt	42	0.36	0	0	0	Opt	42	0.41	0	0	0
7	3	Opt	42	1.12	0	0	1201	Opt	42	1	0	0	141	Opt	42	1.39	0	0	606
7	4	Opt	44	0.92	0	0	746	Opt	44	1.34	0	0	12	Opt	44	1.31	0	0	288
7	5	Opt	44	1.42	0	0	1757	Opt	44	2.46	0	0	2308	Opt	44	1.44	0	0	191
7	6	Opt	35	0.69	0	0	156	Opt	35	1.76	0	0	0	Opt	35	1.12	0	0	0
7	7	Opt	50	0.17	0	0	0	Opt	50	0.39	0	0	0	Opt	50	0.34	0	0	0
7	8	Opt	44	0.55	0	0	147	Opt	44	1.76	0	0	607	Opt	44	1.79	0	0	1458
7	9	Opt	60	0.69	0	0	284	Opt	60	1.12	0	0	775	Opt	60	1.44	0	0	1025
7	10	Opt	49	0.8	0	0	724	Opt	49	1.81	0	0	1600	Opt	49	3.01	0	0	2184
8	1	Opt	44	0.51	0	0	72	Opt	44	0.8	0	0	0	Opt	44	0.66	0	0	0
8	2	Opt	51	0.42	0	0	20	Opt	51	0.59	0	0	0	Opt	51	0.59	0	0	0
8	3	Opt	53	0.61	0	0	193	Opt	53	0.52	0	0	0	Opt	53	0.69	0	0	0
8	4	Opt	48	0.45	0	0	0	Opt	48	0.56	0	0	0	Opt	48	0.94	0	0	0
8	5	Opt	58	0.53	0	0	0	Opt	58	0.86	0	0	0	Opt	58	0.7	0	0	9
8	6	Opt	47	0.53	0	0	178	Opt	47	0.59	0	0	28	Opt	47	0.81	0	0	0
8	7	Opt	41	0.56	0	0	81	Opt	41	0.31	0	0	0	Opt	41	0.31	0	0	0
8	8	Opt	51	0.45	0	0	0	Opt	51	0.64	0	0	7	Opt	51	0.92	0	0	113
8	9	Opt	39	0.58	0	0	149	Opt	39	0.44	0	0	0	Opt	39	1.08	0	0	43
8	10	Opt	67	0.2	0	0	0	Opt	67	0.3	0	0	0	Opt	67	0.67	0	0	8
9	1	Opt	83	13.38	0	0	14503	Feas	83	300	6.02	0	79809	Feas	83	300	9.64	0	98365
9	2	Feas	92	300	11.96	0	301317	Feas	94	300	29.79	2.17	110220	Feas	94	300	29.86	2.17	103901
9	3	Opt	68	18.16	0	0	18083	Feas	72	300	16.67	5.88	61590	Opt	68	183.15	0	0	47670
9	4	Opt	71	34.09	0	0	40451	Feas	72	300	2.78	1.41	82714	Feas	72	300	4.17	1.41	67375
9	5	Opt	70	4.2	0	0	4404	Opt	70	11.95	0	0	5233	Opt	70	21.03	0	0	7745
9	6	Opt	59	55.44	0	0	52944	Feas	59	300	3.39	0	106375	Opt	59	283.72	0	0	91219
9	7	Opt	63	20.09	0	0	10852	Feas	63	300	9.52	0	75679	Feas	63	300	7.1	0	60976
9	8	Opt	91	43.18	0	0	58787	Opt	91	197.82	0	0	118930	Opt	91	251.74	0	0	94380
9	9	Opt	63	28.14	0	0	35090	Opt	63	234.16	0	0	76645	Feas	64	300	9.37	1.59	80022
9	10	Opt	88	31.21	0	0	31002	Feas	88	300	10.23	0	77904	Opt	88	234.42	0	0	67395
10	1	Opt	42	3.34	0	0	3818	Opt	42	2.06	0	0	838	Opt	42	2.68	0	0	2522
10	2	Opt	56	6.36	0	0	6808	Opt	56	13.51	0	0	9336	Opt	56	11.51	0	0	8592
10	3	Opt	62	10.53	0	0	7588	Opt	62	42.71	0	0	22942	Opt	62	17.3	0	0	8742
10	4	Opt	58	5.46	0	0	6109	Opt	58	7.96	0	0	5470	Opt	58	7.75	0	0	3750
10	5	Opt	41	5.32	0	0	4701	Opt	41	22.23	0	0	12409	Opt	41	8.91	0	0	5412
10	6	Opt	44	7.16	0	0	6408	Opt	44	95.94	0	0	32922	Opt	44	192.44	0	0	68998
10	7	Opt	49	2.2	0	0	2531	Opt	49	6.33	0	0	3461	Opt	49	4.38	0	0	2598
10	8	Opt	54	22.7	0	0	22646	Opt	54	15.05	0	0	9085	Opt	54	45.94	0	0	11514
10	9	Opt	49	0.64	0	0	240	Opt	49	1.26	0	0	618	Opt	49	2.78	0	0	1691
10	10	Opt	41	19.55	0	0	10513	Opt	41	44.87	0	0	20745	Opt	41	117.36	0	0	41172
11	1	Opt	54	3.92	0	0	5446	Opt	54	4.35	0	0	3879	Opt	54	5.02	0	0	4228
11	2	Opt	56	3.57	0	0	4086	Opt	56	4.31	0	0	2886	Opt	56	7.36	0	0	6136
11	3	Opt	81	0.34	0	0	0	Opt	81	1.5	0	0	0	Opt	81	1.56	0	0	720
11	4	Opt	63	1.54	0	0	2877	Opt	63	0.81	0	0	346	Opt	63	3.48	0	0	2437
11	5	Opt	49	5.2	0	0	4671	Opt	49	8.05	0	0	5989	Opt	49	5.88	0	0	5067
11	6	Opt	44	1.7	0	0	1159	Opt	44	2.31	0	0	928	Opt	44	3.82	0	0	3149
11	7	Opt	36	4.02	0	0	3681	Opt	36	6.27	0	0	5189	Opt	36	4.7	0	0	2805
11	8	Opt	62	0.76	0	0	623	Opt	62	6.63	0	0	4841	Opt	62	6.76	0	0	4196
11	9	Opt	67	0.42	0	0	124	Opt	67	0.98	0	0	605	Opt	67	1.31	0	0	431
11	10	Opt	38	0.16	0	0	0	Opt	38	0.41	0	0	0	Opt	38	1.23	0	0	0
12	1	Opt	47	0.53	0	0	160	Opt	47	0.42	0	0	0	Opt	47	0.47	0	0	0
12	2	Opt	46	0.56	0	0	219	Opt	46	0.44	0	0	0	Opt	46	0.42	0	0	0
12	3	Opt	37	0.47	0	0	11	Opt	37	1.65	0	0	0	Opt	37	1.05	0	0	0
12	4	Opt	63	0.51	0	0	247	Opt	63	0.37	0	0	0	Opt	63	0.64	0	0	0
12	5	Opt	47	0.59	0	0	284	Opt	47	0.72	0	0	0	Opt	47	0.69	0	0	0
12	6	Opt	53	0.22	0	0	0	Opt	53	0.98	0	0	200	Opt	53	1.15	0	0	327
12	7	Opt	55	0.69	0	0	296	Opt	55	0.69	0	0	49	Opt	55	0.89	0	0	0
12	8	Opt	35	0.77	0	0	128	Opt	35	0.39	0	0	0	Opt	35	0.37	0	0	0
12	9	Opt	52	0.64	0	0	27	Opt	52	1.15	0	0	40	Opt	52	0.94	0	0	0
12	10	Opt	57	0.42	0	0	0	Opt	57	0.83	0	0	0	Opt	57	0.94	0	0	0
13	1	Feas	60	300	16.8	3.45	131359	Feas	61	300	27.7	5.17	87892	Feas	63	300	28.41	8.62	48444
13	2	Feas	63	300	20.09	1.61	283764	Feas	66	300	31.67	6.45	74370	Feas	67	300	36.14	8.06	55780
13	3	Feas	76	300	8.42	0	291628	Feas	76	300	23.86	0	105987	Feas	79	300	26.58	3.95	72695
13	4	Opt	72	286.04	0.01	0	468467	Feas	73	300	9.59	1.39	118846	Feas	72	300	8.33	0	132398

13	5	Feas	69	300	23.04	2.99	102413	Feas	72	300	32.28	7.46	56222	Feas	72	300	32.23	7.46	64014
13	6	Feas	65	300	23.92	1.56	207386	Feas	66	300	26.62	3.13	79153	Feas	67	300	30.35	4.69	60907
13	7	Feas	77	300	6.17	0	240037	Feas	80	300	22.3	3.9	129702	Feas	79	300	20.25	2.6	92418
13	8	Opt	106	263.45	0	0	398594	Feas	114	300	32.12	7.55	99515	Feas	113	300	36.95	6.6	87669
13	9	Opt	71	110.43	0.01	0	101833	Feas	73	300	11.56	2.82	141997	Feas	72	300	11.11	1.41	131065
13	10	Feas	64	300	4.37	0	238526	Feas	65	300	16.92	1.56	146584	Feas	64	300	9.37	0	81324
14	1	Opt	50	35.88	0	0	34180	Opt	50	47.66	0	0	24427	Opt	50	60.18	0	0	49384
14	2	Opt	53	96.71	0	0	40467	Feas	54	300	7.31	1.89	82161	Feas	54	300	7.41	1.89	57752
14	3	Opt	58	10.65	0	0	9811	Opt	58	65.24	0	0	44560	Opt	58	50.12	0	0	34236
14	4	Opt	50	193.08	0	0	138270	Feas	50	300	2	0	246400	Feas	50	300	2	0	129594
14	5	Opt	52	8.21	0	0	8323	Opt	52	14.51	0	0	11330	Opt	52	19.86	0	0	16553
14	6	Opt	35	16.1	0	0	10307	Opt	35	15.54	0	0	8523	Opt	35	71.21	0	0	34400
14	7	Opt	50	231.19	0	0	53507	Feas	51	300	5.88	2	104867	Feas	50	300	4	0	90495
14	8	Opt	54	1.2	0	0	1945	Opt	54	2.48	0	0	3289	Opt	54	1.93	0	0	1767
14	9	Opt	46	79.97	0	0	70244	Opt	46	129.57	0	0	82207	Feas	47	300	6.38	2.17	162295
14	10	Opt	61	9.92	0	0	26185	Opt	61	4.91	0	0	7713	Opt	61	4.95	0	0	5513
15	1	Opt	46	0.8	0	0	380	Opt	46	0.79	0	0	242	Opt	46	1.03	0	0	0
15	2	Opt	47	0.7	0	0	416	Opt	47	0.78	0	0	0	Opt	47	0.87	0	0	403
15	3	Opt	48	0.97	0	0	1427	Opt	48	0.72	0	0	141	Opt	48	2.15	0	0	1898
15	4	Opt	48	0.53	0	0	133	Opt	48	0.41	0	0	0	Opt	48	0.87	0	0	0
15	5	Opt	58	75.71	0	0	37040	Opt	58	73.24	0	0	56086	Opt	58	123.71	0	0	46071
15	6	Opt	67	0.61	0	0	411	Opt	67	1.08	0	0	645	Opt	67	1.83	0	0	1271
15	7	Opt	47	0.69	0	0	654	Opt	47	1.4	0	0	1035	Opt	47	1.67	0	0	892
15	8	Opt	50	1.53	0	0	1796	Opt	50	1.31	0	0	788	Opt	50	2.56	0	0	1735
15	9	Opt	54	0.64	0	0	308	Opt	54	1.45	0	0	1047	Opt	54	2.47	0	0	2278
15	10	Opt	65	0.37	0	0	248	Opt	65	4.91	0	0	5214	Opt	65	1.54	0	0	885
16	1	Opt	51	0.34	0	0	0	Opt	51	0.62	0	0	0	Opt	51	1.17	0	0	256
16	2	Opt	48	0.59	0	0	351	Opt	48	0.36	0	0	0	Opt	48	1.06	0	0	452
16	3	Opt	36	0.13	0	0	0	Opt	36	0.22	0	0	0	Opt	36	0.36	0	0	0
16	4	Opt	47	0.4	0	0	0	Opt	47	0.34	0	0	0	Opt	47	0.62	0	0	4
16	5	Opt	51	0.36	0	0	0	Opt	51	0.34	0	0	0	Opt	51	0.69	0	0	0
16	6	Opt	51	0.39	0	0	0	Opt	51	0.47	0	0	0	Opt	51	0.67	0	0	48
16	7	Opt	34	0.14	0	0	0	Opt	34	0.33	0	0	0	Opt	34	0.34	0	0	0
16	8	Opt	44	0.37	0	0	0	Opt	44	0.9	0	0	0	Opt	44	0.98	0	0	248
16	9	Opt	44	0.45	0	0	0	Opt	44	0.45	0	0	0	Opt	44	0.94	0	0	465
16	10	Opt	51	0.65	0	0	309	Opt	51	0.5	0	0	0	Opt	51	1.29	0	0	612
17	1	Opt	64	1.87	0	0	3301	Opt	64	3.4	0	0	3426	Opt	64	4.85	0	0	5440
17	2	Opt	68	0.34	0	0	0	Opt	68	0.36	0	0	0	Opt	68	0.47	0	0	179
17	3	Opt	60	0.16	0	0	0	Opt	60	0.22	0	0	0	Opt	60	0.2	0	0	0
17	4	Opt	49	0.75	0	0	1226	Opt	49	0.87	0	0	636	Opt	49	0.81	0	0	488
17	5	Opt	47	1.18	0	0	1803	Opt	47	0.81	0	0	458	Opt	47	1.75	0	0	1971
17	6	Opt	63	0.14	0	0	0	Opt	63	0.23	0	0	0	Opt	63	0.2	0	0	0
17	7	Opt	57	1.08	0	0	1506	Opt	57	0.83	0	0	693	Opt	57	1.5	0	0	1208
17	8	Opt	61	0.14	0	0	0	Opt	61	0.58	0	0	130	Opt	61	0.92	0	0	396
17	9	Opt	48	0.61	0	0	188	Opt	48	0.81	0	0	408	Opt	48	0.98	0	0	859
17	10	Opt	66	0.13	0	0	0	Opt	66	0.2	0	0	0	Opt	66	0.2	0	0	0
18	1	Opt	53	0.14	0	0	0	Opt	53	0.39	0	0	0	Opt	53	0.73	0	0	36
18	2	Opt	55	0.14	0	0	0	Opt	55	0.22	0	0	0	Opt	55	0.2	0	0	0
18	3	Opt	56	0.47	0	0	62	Opt	56	0.36	0	0	0	Opt	56	0.42	0	0	0
18	4	Opt	70	0.12	0	0	0	Opt	70	0.3	0	0	0	Opt	70	0.3	0	0	0
18	5	Opt	52	0.42	0	0	167	Opt	52	0.41	0	0	0	Opt	52	0.81	0	0	0
18	6	Opt	62	0.98	0	0	1251	Opt	62	0.59	0	0	291	Opt	62	0.86	0	0	240
18	7	Opt	48	0.16	0	0	0	Opt	48	0.25	0	0	0	Opt	48	0.22	0	0	0
18	8	Opt	52	0.16	0	0	0	Opt	52	0.23	0	0	0	Opt	52	0.23	0	0	0
18	9	Opt	47	0.14	0	0	0	Opt	47	0.42	0	0	0	Opt	47	0.23	0	0	0
18	10	Opt	49	0.83	0	0	925	Opt	49	0.53	0	0	0	Opt	49	0.8	0	0	332
19	1	Opt	40	0.48	0	0	154	Opt	40	1.76	0	0	0	Opt	40	0.81	0	0	187
19	2	Opt	58	0.36	0	0	0	Opt	58	0.27	0	0	0	Opt	58	0.37	0	0	0
19	3	Opt	83	0.28	0	0	8	Opt	83	0.33	0	0	0	Opt	83	0.39	0	0	0
19	4	Opt	39	0.13	0	0	0	Opt	39	0.19	0	0	0	Opt	39	0.22	0	0	0
19	5	Opt	48	0.17	0	0	0	Opt	48	0.53	0	0	109	Opt	48	0.22	0	0	0
19	6	Opt	49	0.16	0	0	0	Opt	49	0.2	0	0	0	Opt	49	0.22	0	0	0
19	7	Opt	57	0.38	0	0	1	Opt	57	0.39	0	0	0	Opt	57	0.61	0	0	0
19	8	Opt	55	0.55	0	0	259	Opt	55	1.9	0	0	0	Opt	55	0.56	0	0	0
19	9	Opt	38	0.13	0	0	0	Opt	38	0.25	0	0	0	Opt	38	0.22	0	0	0
19	10	Opt	47	0.14	0	0	0	Opt	47	0.27	0	0	0	Opt	47	0.22	0	0	0
20	1	Opt	57	0.3	0	0	0	Opt	57	0.25	0	0	0	Opt	57	0.3	0	0	0
20	2	Opt	70	0.33	0	0	12	Opt	70	0.23	0	0	0	Opt	70	0.41	0	0	0
20	3	Opt	49	0.16	0	0	0	Opt	49	0.22	0	0	0	Opt	49	0.16	0	0	0
20	4	Opt	43	0.11	0	0	0	Opt	43	0.22	0	0	0	Opt	43	0.2	0	0	0
20	5	Opt	61	0.13	0	0	0	Opt	61	0.23	0	0	0	Opt	61	0.2	0	0	0
20	6	Opt	51	0.16	0	0	0	Opt	51	0.22	0	0	0	Opt	51	0.19	0	0	0
20	7	Opt	42	0.33	0	0	0	Opt	42	0.22	0	0	0	Opt	42	0.2	0	0	0
20	8	Opt	51	0.16	0	0	0	Opt	51	0.16	0	0	0	Opt	51	0.19	0	0	0
20	9	Opt	41	0.16	0	0	0	Opt	41	0.3	0	0	0	Opt	41	0.19	0	0	0
20	10	Opt	37	0.14	0	0	0	Opt	37	0.25	0	0	0	Opt	37	0.2	0	0	0
21	1	Opt	84	1.39	0	0	2420	Opt	84	6.99	0	0	3979	Opt	84	4.49	0	0	2905
21	2	Opt	59	2.65	0	0	4614	Opt	59	6.65	0	0	4332	Opt	59	7.64	0	0	2853
21	3	Opt	76	1.01	0	0	1057	Opt	76	7.04	0	0	4253	Opt	76	5.71	0	0	4235
21	4	Opt	70	3.23	0	0	3410	Opt	70	11.7	0	0	6050	Opt	70	13.77	0.01	0	8396
21	5	Opt	55	1.64	0	0	2643	Opt	55	5.83	0	0	3829	Opt	55	11.23	0	0	6286
21	6	Opt	76	2.21	0	0	4430	Opt	76	8.35	0	0	5056	Opt	76	10.98	0	0	4061
21	7	Opt	65	5.62	0	0	9436	Opt	65	11.34	0	0	6497	Opt	65	12.23	0	0	5793
21	8	Opt	62	1.23	0.01	0	2491	Opt	62	11.23	0	0	6847	Opt	62	10.28	0	0	4931
21	9	Opt	69	1.89	0	0	3018	Opt	69	14.54	0	0	7202	Opt	69	19.02	0	0	7064
21	10	Opt	69	1.37	0	0	2463	Opt	69	5.16	0	0	5654	Opt	69	5.52	0	0	2929
22	1	Opt	42	2.18	0	0	4374	Opt	42	1.78	0	0	1918	Opt	42	1.04	0	0	578
22	2	Opt	45	0.13	0	0	0	Opt	45	0.67	0	0	0	Opt	45	1.31	0	0	1002
22	3	Opt	63	0.5	0	0	119	Opt	63	2.17	0	0	2050	Opt	63	1.39	0	0	1054
22	4	Opt	42	2	0	0	2975	Opt	42	2.28	0	0	2801	Opt	42	2.04	0	0	1791
22	5	Opt	52	1.62	0	0	2493	Opt	52	2.53	0	0	1731	Opt	52	4.45	0	0	3804

22	6	Opt	52	3.01	0	0	2500	Opt	52	4.54	0	0	3036	Opt	52	4.09	0	0	2969
22	7	Opt	60	3.68	0	0	4584	Opt	60	5.63	0	0	4109	Opt	60	4.91	0	0	2934
22	8	Opt	55	9.16	0	0	6773	Opt	55	55.22	0	0	21884	Opt	55	21.06	0	0	10778
22	9	Opt	76	0.31	0	0	0	Opt	76	0.56	0	0	10	Opt	76	0.58	0	0	11
22	10	Opt	55	2.92	0	0	2577	Opt	55	4.79	0	0	3570	Opt	55	4.1	0	0	2740
23	1	Opt	63	0.59	0	0	361	Opt	63	0.62	0	0	0	Opt	63	0.53	0	0	0
23	2	Opt	53	0.37	0	0	0	Opt	53	0.5	0	0	0	Opt	53	0.55	0	0	0
23	3	Opt	46	0.58	0	0	461	Opt	46	0.56	0	0	96	Opt	46	1.39	0	0	652
23	4	Opt	65	0.64	0	0	551	Opt	65	1.29	0	0	1455	Opt	65	1.33	0	0	806
23	5	Opt	52	0.51	0	0	156	Opt	52	0.72	0	0	9	Opt	52	0.97	0	0	43
23	6	Opt	48	0.19	0	0	0	Opt	48	1.19	0	0	828	Opt	48	0.69	0	0	354
23	7	Opt	60	0.16	0	0	0	Opt	60	1.14	0	0	1083	Opt	60	1.64	0	0	1274
23	8	Opt	48	0.76	0	0	732	Opt	48	0.41	0	0	0	Opt	48	0.55	0	0	0
23	9	Opt	63	0.64	0	0	479	Opt	63	1.25	0	0	1215	Opt	63	1.2	0	0	327
23	10	Opt	61	0.13	0	0	0	Opt	61	0.25	0	0	0	Opt	61	0.22	0	0	0
24	1	Opt	53	0.34	0	0	0	Opt	53	0.31	0	0	0	Opt	53	0.22	0	0	0
24	2	Opt	58	0.36	0	0	19	Opt	58	0.38	0	0	0	Opt	58	0.59	0	0	129
24	3	Opt	69	0.28	0	0	0	Opt	69	0.34	0	0	0	Opt	69	0.23	0	0	0
24	4	Opt	53	0.38	0	0	0	Opt	53	0.42	0	0	0	Opt	53	0.62	0	0	0
24	5	Opt	51	0.45	0	0	14	Opt	51	0.36	0	0	0	Opt	51	0.84	0	0	340
24	6	Opt	56	0.39	0	0	0	Opt	56	0.5	0	0	0	Opt	56	0.55	0	0	0
24	7	Opt	44	0.17	0	0	0	Opt	44	0.31	0	0	0	Opt	44	0.25	0	0	0
24	8	Opt	38	0.17	0	0	0	Opt	38	0.33	0	0	0	Opt	38	0.25	0	0	0
24	9	Opt	43	0.14	0	0	0	Opt	43	0.36	0	0	0	Opt	43	0.25	0	0	0
24	10	Opt	53	0.34	0	0	0	Opt	53	0.36	0	0	0	Opt	53	0.7	0	0	0
25	1	Opt	93	67.47	0	0	105564	Feas	95	300	8.42	2.15	115709	Feas	94	300	10.64	1.08	129588
25	2	Opt	75	39.44	0	0	51258	Feas	75	300	7.97	0	123007	Feas	75	300	7.9	0	90576
25	3	Opt	76	98.06	0.01	0	92031	Feas	76	300	15.9	0	87113	Feas	77	300	15.02	1.32	87895
25	4	Opt	81	55.66	0	0	78383	Feas	83	300	13.25	2.47	145502	Feas	83	300	16.53	2.47	155675
25	5	Opt	72	81.65	0.01	0	125001	Feas	72	300	5.42	0	172341	Feas	72	300	9.72	0	147274
25	6	Opt	58	39.9	0	0	38512	Feas	60	300	15	3.45	84266	Feas	58	300	10.34	0	71554
25	7	Opt	95	12.56	0	0	13997	Opt	95	121.12	0	0	40320	Opt	95	193.5	0	0	49065
25	8	Opt	69	13.48	0	0	12870	Opt	69	146.89	0	0	58002	Opt	69	148.86	0	0	26461
25	9	Opt	84	6.8	0	0	6425	Opt	84	39.7	0	0	13220	Opt	84	38.33	0.01	0	15941
25	10	Opt	58	5.35	0	0	5877	Opt	58	35.43	0	0	19635	Opt	58	16.19	0	0	9942
26	1	Opt	59	0.16	0	0	0	Opt	59	2.65	0	0	2425	Opt	59	0.76	0	0	0
26	2	Opt	40	0.53	0	0	113	Opt	40	1.09	0	0	266	Opt	40	0.94	0	0	19
26	3	Opt	58	0.83	0	0	987	Opt	58	1.26	0	0	926	Opt	58	3.1	0	0	2994
26	4	Opt	62	2.76	0	0	3378	Opt	62	2.43	0	0	3537	Opt	62	4.2	0	0	3257
26	5	Opt	74	0.89	0	0	1475	Opt	74	4.4	0	0	3591	Opt	74	5.4	0	0	2951
26	6	Opt	53	23.43	0	0	35525	Opt	53	19.72	0	0	17728	Opt	53	12.7	0	0	9500
26	7	Opt	56	1.12	0	0	1331	Opt	56	0.5	0	0	0	Opt	56	0.89	0	0	0
26	8	Opt	66	0.58	0	0	300	Opt	66	2.7	0	0	2604	Opt	66	3.15	0	0	2488
26	9	Opt	43	10.81	0	0	10781	Opt	43	14.13	0	0	11052	Opt	43	52.74	0	0	25475
26	10	Opt	49	3.1	0	0	3372	Opt	49	4.95	0	0	3921	Opt	49	6.07	0	0	3868
27	1	Opt	43	0.16	0	0	0	Opt	43	0.3	0	0	0	Opt	43	0.55	0	0	0
27	2	Opt	58	0.56	0	0	263	Opt	58	0.67	0	0	186	Opt	58	0.65	0	0	121
27	3	Opt	60	0.59	0	0	220	Opt	60	0.61	0	0	47	Opt	60	0.77	0	0	114
27	4	Opt	64	0.36	0	0	102	Opt	64	0.34	0	0	0	Opt	64	0.25	0	0	0
27	5	Opt	49	0.14	0	0	0	Opt	49	0.34	0	0	0	Opt	49	0.31	0	0	0
27	6	Opt	59	0.75	0	0	1086	Opt	59	0.44	0	0	0	Opt	59	0.26	0	0	0
27	7	Opt	49	1.29	0	0	2267	Opt	49	2.56	0	0	3353	Opt	49	2.98	0	0	3526
27	8	Opt	66	0.48	0	0	210	Opt	66	1.03	0	0	83	Opt	66	0.84	0	0	0
27	9	Opt	55	1.14	0	0	2035	Opt	55	1.03	0	0	186	Opt	55	1.53	0	0	950
27	10	Opt	62	0.47	0	0	295	Opt	62	0.61	0	0	0	Opt	62	0.7	0	0	31
28	1	Opt	69	0.28	0	0	0	Opt	69	0.66	0	0	50	Opt	69	0.67	0	0	0
28	2	Opt	57	0.45	0	0	15	Opt	57	0.44	0	0	0	Opt	57	0.73	0	0	180
28	3	Opt	40	0.44	0	0	109	Opt	40	0.58	0	0	150	Opt	40	1.34	0	0	378
28	4	Opt	49	0.52	0	0	253	Opt	49	0.33	0	0	0	Opt	49	0.31	0	0	0
28	5	Opt	73	0.33	0	0	44	Opt	73	0.52	0	0	0	Opt	73	0.59	0	0	0
28	6	Opt	55	0.31	0	0	0	Opt	55	0.34	0	0	0	Opt	55	0.25	0	0	0
28	7	Opt	48	0.19	0	0	0	Opt	48	0.38	0	0	0	Opt	48	0.33	0	0	0
28	8	Opt	53	0.14	0	0	0	Opt	53	0.34	0	0	0	Opt	53	0.28	0	0	0
28	9	Opt	62	0.39	0	0	6	Opt	62	0.36	0	0	0	Opt	62	0.27	0	0	0
28	10	Opt	59	0.47	0	0	233	Opt	59	0.47	0	0	0	Opt	59	0.45	0	0	0
29	1	Opt	85	161.15	0	0	262904	Feas	87	300	4.6	2.35	155395	Feas	87	300	10.1	2.35	84165
29	2	Opt	90	183.83	0.01	0	209664	Feas	91	300	14.07	1.11	113937	Feas	93	300	17.63	3.33	88031
29	3	Feas	78	300	21.18	0	200546	Feas	80	300	29.59	2.56	99375	Feas	80	300	29.69	2.56	79619
29	4	Feas	104	300	10.58	0.97	397608	Feas	104	300	21.25	0.97	106501	Feas	104	300	18.08	0.97	91885
29	5	Opt	98	33.4	0	0	45682	Feas	99	300	6.06	1.02	154964	Feas	99	300	11.11	1.02	114169
29	6	Feas	92	300	15.37	0	227086	Feas	95	300	28.03	3.26	83160	Feas	94	300	25.43	2.17	131206
29	7	Feas	74	300	5	1.37	399564	Feas	74	300	9.46	1.37	157701	Feas	73	300	10.96	0	135246
29	8	Feas	82	300	19.13	2.5	149568	Feas	84	300	27.18	5	108715	Feas	85	300	27.06	6.25	82597
29	9	Opt	97	115.61	0	0	140559	Feas	102	300	23.53	5.15	112154	Feas	101	300	19.8	4.12	81860
29	10	Opt	76	3.28	0	0	7270	Opt	76	82.73	0	0	60822	Opt	76	56.74	0.01	0	27714
30	1	Opt	47	11.67	0	0	8238	Opt	47	29.83	0	0	23277	Opt	47	39.91	0	0	19317
30	2	Opt	68	43.9	0	0	34106	Opt	68	104.36	0	0	52815	Opt	68	149.98	0	0	49391
30	3	Opt	55	32.4	0	0	27812	Opt	55	36.72	0	0	27190	Opt	55	20.11	0	0	24263
30	4	Opt	53	17.64	0	0	16917	Opt	53	22.57	0	0	12662	Opt	53	12.92	0	0	9662
30	5	Opt	54	5.85	0	0	6698	Opt	54	6.72	0	0	6277	Opt	54	13.93	0	0	6939
30	6	Opt	62	56.68	0	0	18515	Opt	62	38.73	0	0	21911	Opt	62	39.02	0	0	29691
30	7	Opt	68	5.85	0	0	5385	Opt	68	10.37	0	0	7515	Opt	68	21.22	0	0	12164
30	8	Opt	46	39.69	0	0	41674	Feas	46	300	2.17	0	251738	Opt	46	183.71	0	0	147695
30	9	Opt	46	69.73	0	0	31681	Opt	46	60.37	0	0	45024	Opt	46	122.02	0	0	62514
30	10	Opt	53	66.91	0	0	44540	Opt	53	68.23	0	0	55199	Opt	53	93.65	0	0	54733
31	1	Opt	43	0.48	0	0	113	Opt	43	0.53	0	0	0	Opt	43	1.01	0	0	0
31	2	Opt	63	0.56	0	0	331	Opt	63	0.58	0	0	0	Opt	63	0.92	0	0	193
31	3	Opt	58	0.59	0	0	317	Opt	58	0.48	0	0	0	Opt	58	1.29	0	0	0
31	4	Opt	50	0.53	0	0	232	Opt	50	0.48	0	0	0	Opt	50	1.08	0	0	591
31	5	Opt	52	3.07	0	0	2878	Opt	52	3.62	0	0	3133	Opt	52	6.05	0	0	4501
31	6	Opt	53	0.78	0	0	500	Opt	53	0.8	0	0	241	Opt	53	0.95	0	0	28

31	7	Opt	61	0.19	0	0	0	Opt	61	0.42	0	0	0	Opt	61	0.5	0	0	0
31	8	Opt	58	0.81	0	0	747	Opt	58	2	0	0	2182	Opt	58	2.25	0	0	1635
31	9	Opt	50	15.13	0	0	16524	Opt	50	37.67	0	0	49512	Opt	50	51.14	0	0	52558
31	10	Opt	55	31.33	0	0	21852	Opt	55	68.48	0	0	53516	Opt	55	183.88	0	0	181255
32	1	Opt	61	0.31	0	0	0	Opt	61	0.37	0	0	0	Opt	61	0.78	0	0	253
32	2	Opt	60	0.37	0	0	0	Opt	60	0.25	0	0	0	Opt	60	0.31	0	0	0
32	3	Opt	57	0.53	0	0	172	Opt	57	0.23	0	0	0	Opt	57	0.36	0	0	0
32	4	Opt	68	0.25	0	0	0	Opt	68	0.36	0	0	0	Opt	68	0.3	0	0	0
32	5	Opt	54	0.39	0	0	0	Opt	54	0.31	0	0	0	Opt	54	0.3	0	0	0
32	6	Opt	44	0.16	0	0	0	Opt	44	0.19	0	0	0	Opt	44	0.31	0	0	0
32	7	Opt	35	0.17	0	0	0	Opt	35	0.33	0	0	0	Opt	35	0.28	0	0	0
32	8	Opt	54	0.34	0	0	0	Opt	54	0.33	0	0	0	Opt	54	0.69	0	0	0
32	9	Opt	65	0.36	0	0	0	Opt	65	0.33	0	0	0	Opt	65	0.52	0	0	0
32	10	Opt	51	0.48	0	0	129	Opt	51	0.38	0	0	0	Opt	51	0.3	0	0	0
33	1	Opt	65	0.19	0	0	0	Opt	65	0.75	0	0	432	Opt	65	0.72	0	0	393
33	2	Opt	60	0.5	0	0	459	Opt	60	0.48	0	0	22	Opt	60	0.64	0	0	309
33	3	Opt	55	0.16	0	0	0	Opt	55	0.44	0	0	0	Opt	55	0.59	0	0	87
33	4	Opt	77	0.38	0	0	112	Opt	77	0.39	0	0	11	Opt	77	0.52	0	0	4
33	5	Opt	53	0.14	0	0	0	Opt	53	0.75	0	0	140	Opt	53	0.59	0	0	253
33	6	Opt	59	0.36	0	0	111	Opt	59	0.39	0	0	0	Opt	59	0.62	0	0	189
33	7	Opt	58	0.13	0	0	0	Opt	58	0.2	0	0	0	Opt	58	0.23	0	0	0
33	8	Opt	61	0.52	0	0	0	Opt	61	0.47	0	0	77	Opt	61	0.62	0	0	178
33	9	Opt	65	0.61	0	0	643	Opt	65	0.59	0	0	121	Opt	65	1	0	0	695
33	10	Opt	53	0.16	0	0	0	Opt	53	0.23	0	0	0	Opt	53	0.2	0	0	0
34	1	Opt	68	0.48	0	0	130	Opt	68	0.34	0	0	12	Opt	68	0.47	0	0	0
34	2	Opt	44	0.13	0	0	0	Opt	44	0.28	0	0	0	Opt	44	0.19	0	0	0
34	3	Opt	69	0.11	0	0	0	Opt	69	0.3	0	0	0	Opt	69	0.38	0	0	0
34	4	Opt	67	0.11	0	0	0	Opt	67	0.19	0	0	0	Opt	67	0.2	0	0	0
34	5	Opt	63	0.36	0	0	16	Opt	63	0.38	0	0	0	Opt	63	0.45	0	0	1
34	6	Opt	52	0.31	0	0	0	Opt	52	0.34	0	0	0	Opt	52	0.41	0	0	47
34	7	Opt	58	0.48	0	0	425	Opt	58	0.37	0	0	0	Opt	58	0.61	0	0	35
34	8	Opt	58	0.41	0	0	0	Opt	58	0.44	0	0	247	Opt	58	0.67	0	0	343
34	9	Opt	60	0.16	0	0	0	Opt	60	0.3	0	0	0	Opt	60	0.23	0	0	0
34	10	Opt	47	0.14	0	0	0	Opt	47	0.22	0	0	0	Opt	47	0.2	0	0	0
35	1	Opt	57	0.14	0	0	0	Opt	57	0.22	0	0	0	Opt	57	0.19	0	0	0
35	2	Opt	53	0.11	0	0	0	Opt	53	0.22	0	0	0	Opt	53	0.19	0	0	0
35	3	Opt	60	0.11	0	0	0	Opt	60	0.2	0	0	0	Opt	60	0.19	0	0	0
35	4	Opt	50	0.14	0	0	0	Opt	50	0.36	0	0	0	Opt	50	0.61	0	0	0
35	5	Opt	60	0.13	0	0	0	Opt	60	0.23	0	0	0	Opt	60	0.16	0	0	0
35	6	Opt	58	0.14	0	0	0	Opt	58	0.31	0	0	0	Opt	58	0.41	0	0	0
35	7	Opt	61	0.14	0	0	0	Opt	61	0.33	0	0	0	Opt	61	0.31	0	0	0
35	8	Opt	63	0.3	0	0	66	Opt	63	0.2	0	0	0	Opt	63	0.19	0	0	0
35	9	Opt	59	0.26	0	0	0	Opt	59	0.27	0	0	0	Opt	59	0.31	0	0	9
35	10	Opt	59	0.17	0	0	0	Opt	59	0.23	0	0	0	Opt	59	0.42	0	0	0
36	1	Opt	66	0.13	0	0	0	Opt	66	0.22	0	0	0	Opt	66	0.2	0	0	0
36	2	Opt	44	0.16	0	0	0	Opt	44	0.23	0	0	0	Opt	44	0.19	0	0	0
36	3	Opt	61	0.14	0	0	0	Opt	61	0.2	0	0	0	Opt	61	0.19	0	0	0
36	4	Opt	59	0.13	0	0	0	Opt	59	0.16	0	0	0	Opt	59	0.17	0	0	0
36	5	Opt	64	0.14	0	0	0	Opt	64	0.23	0	0	0	Opt	64	0.2	0	0	0
36	6	Opt	46	0.11	0	0	0	Opt	46	0.23	0	0	0	Opt	46	0.19	0	0	0
36	7	Opt	56	0.14	0	0	0	Opt	56	0.16	0	0	0	Opt	56	0.16	0	0	0
36	8	Opt	63	0.14	0	0	0	Opt	63	0.16	0	0	0	Opt	63	0.16	0	0	0
36	9	Opt	59	0.17	0	0	0	Opt	59	0.27	0	0	0	Opt	59	0.22	0	0	0
36	10	Opt	59	0.12	0	0	0	Opt	59	0.17	0	0	0	Opt	59	0.19	0	0	0
37	1	Opt	79	1.81	0	0	3611	Opt	79	7.5	0	0	5404	Opt	79	5.62	0	0	3187
37	2	Opt	69	1.39	0	0	2638	Opt	69	6.35	0	0	5344	Opt	69	4.7	0	0	6665
37	3	Opt	81	1.72	0	0	2641	Opt	81	7.47	0	0	4716	Opt	81	9.39	0	0	6205
37	4	Opt	83	3.2	0	0	7374	Opt	83	7.13	0	0	4402	Opt	83	8.94	0	0	4968
37	5	Opt	80	2.39	0	0	5392	Opt	80	2.5	0	0	3592	Opt	80	1.95	0	0	2352
37	6	Opt	73	1.01	0	0	1605	Opt	73	1.95	0	0	2328	Opt	73	1.9	0	0	2378
37	7	Opt	92	4.9	0	0	6152	Opt	92	26.85	0	0	14020	Opt	92	28.28	0	0	16729
37	8	Opt	72	0.83	0	0	627	Opt	72	3.95	0	0	2690	Opt	72	2.9	0	0	3671
37	9	Opt	57	1.08	0	0	1518	Opt	57	4.38	0	0	6400	Opt	57	4.07	0	0	5296
37	10	Opt	81	2.12	0	0	5366	Opt	81	6.05	0	0	5360	Opt	81	7.27	0	0	5909
38	1	Opt	48	1.34	0	0	2728	Opt	48	1.92	0	0	2998	Opt	48	2.6	0	0	4036
38	2	Opt	54	0.5	0	0	0	Opt	54	0.73	0	0	662	Opt	54	1.11	0	0	740
38	3	Opt	59	0.64	0	0	668	Opt	59	1.65	0	0	1882	Opt	59	1.64	0	0	1687
38	4	Opt	59	0.19	0	0	0	Opt	59	1.15	0	0	1187	Opt	59	0.92	0	0	835
38	5	Opt	71	0.75	0	0	1129	Opt	71	1.36	0	0	1776	Opt	71	2.2	0	0	2799
38	6	Opt	63	0.16	0	0	0	Opt	63	0.34	0	0	0	Opt	63	0.23	0	0	0
38	7	Opt	65	0.72	0	0	954	Opt	65	0.89	0	0	919	Opt	65	1.67	0	0	845
38	8	Opt	61	0.51	0	0	344	Opt	61	0.94	0	0	792	Opt	61	1.09	0	0	638
38	9	Opt	63	0.78	0	0	932	Opt	63	2.43	0	0	2942	Opt	63	1.79	0	0	2078
38	10	Opt	60	0.67	0	0	749	Opt	60	1.33	0	0	1600	Opt	60	1.3	0	0	1618
39	1	Opt	55	0.44	0	0	48	Opt	55	0.44	0	0	0	Opt	55	0.34	0	0	0
39	2	Opt	54	0.14	0	0	0	Opt	54	0.33	0	0	0	Opt	54	0.22	0	0	0
39	3	Opt	54	0.53	0	0	453	Opt	54	0.59	0	0	0	Opt	54	0.62	0	0	472
39	4	Opt	53	0.16	0	0	0	Opt	53	0.28	0	0	0	Opt	53	0.27	0	0	0
39	5	Opt	55	0.14	0	0	0	Opt	55	0.25	0	0	0	Opt	55	0.23	0	0	0
39	6	Opt	69	0.37	0	0	216	Opt	69	0.28	0	0	0	Opt	69	0.81	0	0	540
39	7	Opt	56	0.44	0	0	17	Opt	56	0.5	0	0	3	Opt	56	1.05	0	0	723
39	8	Opt	67	0.5	0	0	175	Opt	67	0.33	0	0	0	Opt	67	0.27	0	0	0
39	9	Opt	64	0.51	0	0	280	Opt	64	0.65	0	0	534	Opt	64	0.66	0	0	106
39	10	Opt	60	0.41	0	0	88	Opt	60	0.53	0	0	3	Opt	60	1.08	0	0	622
40	1	Opt	51	0.22	0	0	0	Opt	51	0.36	0	0	0	Opt	51	0.25	0	0	0
40	2	Opt	56	0.17	0	0	0	Opt	56	0.25	0	0	0	Opt	56	0.23	0	0	0
40	3	Opt	57	0.19	0	0	0	Opt	57	0.33	0	0	0	Opt	57	0.25	0	0	0
40	4	Opt	57	0.14	0	0	0	Opt	57	0.23	0	0	0	Opt	57	0.33	0	0	0
40	5	Opt	65	0.12	0	0	0	Opt	65	0.34	0	0	0	Opt	65	0.2	0	0	0
40	6	Opt	60	0.16	0	0	0	Opt	60	0.27	0	0	0	Opt	60	0.23	0	0	0
40	7	Opt	46	0.16	0	0	0	Opt	46	0.27	0	0	0	Opt	46	0.27	0	0	0

40	8	Opt	57	0.12	0	0	0	Opt	57	0.3	0	0	0	Opt	57	0.23	0	0	0
40	9	Opt	64	0.16	0	0	0	Opt	64	0.25	0	0	0	Opt	64	0.22	0	0	0
40	10	Opt	51	0.16	0	0	0	Opt	51	0.25	0	0	0	Opt	51	0.25	0	0	0
41	1	Opt	86	2.59	0	0	5753	Opt	86	7.54	0	0	4868	Opt	86	15.09	0	0	8899
41	2	Opt	89	12.98	0	0	21490	Opt	89	100.93	0	0	45668	Opt	89	198.01	0	0	148514
41	3	Opt	85	4.07	0	0	5350	Opt	85	21.62	0	0	13195	Opt	85	20.4	0	0	11333
41	4	Opt	78	9.69	0	0	11031	Opt	78	83.98	0	0	41710	Opt	78	60.51	0	0	28168
41	5	Opt	99	2.98	0	0	7460	Opt	99	36.57	0	0	21838	Opt	99	18.84	0	0	11629
41	6	Opt	103	89.39	0.01	0	164689	Feas	103	300	4.66	0	205483	Feas	103	300	2.91	0	224370
41	7	Opt	92	6.79	0	0	9855	Opt	92	68.03	0	0	41782	Opt	92	76.16	0.01	0	63075
41	8	Opt	88	22.92	0	0	32887	Opt	88	148.82	0	0	60980	Feas	88	300	7.17	0	77652
41	9	Opt	92	121.15	0	0	342706	Opt	92	61.78	0	0	33147	Opt	92	39.28	0	0	24113
41	10	Opt	99	123.04	0	0	222311	Feas	99	300	7.98	0	112674	Feas	99	300	5.96	0	164666
42	1	Opt	58	0.94	0	0	1595	Opt	58	0.53	0	0	0	Opt	58	0.23	0	0	0
42	2	Opt	50	7.22	0	0	7986	Opt	50	5.26	0	0	6859	Opt	50	5.76	0	0	4742
42	3	Opt	60	4.1	0	0	5103	Opt	60	8.38	0	0	5733	Opt	60	6.12	0	0	4084
42	4	Opt	49	8.07	0	0	7524	Opt	49	11.37	0	0	7536	Opt	49	17.27	0	0	8833
42	5	Opt	52	0.86	0	0	1236	Opt	52	1.81	0	0	1242	Opt	52	3.21	0	0	3847
42	6	Opt	66	2.53	0	0	4112	Opt	66	2.74	0	0	3645	Opt	66	3.68	0	0	2746
42	7	Opt	66	1.09	0	0	1455	Opt	66	0.89	0	0	695	Opt	66	1.81	0	0	2190
42	8	Opt	82	0.69	0	0	578	Opt	82	2.76	0	0	3817	Opt	82	4.13	0	0	2716
42	9	Opt	60	4.32	0	0	5697	Opt	60	5.66	0	0	6323	Opt	60	9.3	0	0	7611
42	10	Opt	75	0.44	0	0	185	Opt	75	0.58	0	0	0	Opt	75	0.95	0	0	658
43	1	Opt	55	4.06	0	0	5525	Opt	55	4.62	0	0	3917	Opt	55	6.66	0	0	5877
43	2	Opt	43	0.12	0	0	0	Opt	43	0.37	0	0	0	Opt	43	0.25	0	0	0
43	3	Opt	57	1.43	0	0	2266	Opt	57	2.73	0	0	2873	Opt	57	4.63	0	0	3805
43	4	Opt	67	0.5	0	0	283	Opt	67	0.66	0	0	110	Opt	67	1.97	0	0	2401
43	5	Opt	64	2.96	0	0	3587	Opt	64	3.67	0	0	3930	Opt	64	5.37	0	0	3794
43	6	Opt	58	1.18	0	0	2374	Opt	58	0.84	0	0	804	Opt	58	1.34	0	0	1137
43	7	Opt	52	0.76	0	0	820	Opt	52	0.47	0	0	0	Opt	52	0.45	0	0	0
43	8	Opt	62	0.25	0	0	0	Opt	62	1.72	0	0	2141	Opt	62	3.39	0	0	4986
43	9	Opt	57	3.43	0	0	7517	Opt	57	4.35	0	0	6007	Opt	57	4.57	0	0	5735
43	10	Opt	60	1.25	0	0	2703	Opt	60	2	0	0	2869	Opt	60	1.98	0	0	2824
44	1	Opt	50	0.34	0	0	0	Opt	50	0.55	0	0	90	Opt	50	0.58	0	0	0
44	2	Opt	54	0.16	0	0	0	Opt	54	0.36	0	0	0	Opt	54	0.25	0	0	0
44	3	Opt	51	0.38	0	0	0	Opt	51	0.25	0	0	0	Opt	51	0.25	0	0	0
44	4	Opt	57	0.14	0	0	0	Opt	57	0.31	0	0	0	Opt	57	0.33	0	0	0
44	5	Opt	55	0.19	0	0	0	Opt	55	0.37	0	0	0	Opt	55	0.27	0	0	0
44	6	Opt	56	0.28	0	0	0	Opt	56	0.22	0	0	0	Opt	56	0.23	0	0	0
44	7	Opt	42	0.17	0	0	0	Opt	42	0.38	0	0	0	Opt	42	0.27	0	0	0
44	8	Opt	49	0.28	0	0	0	Opt	49	0.33	0	0	0	Opt	49	0.27	0	0	0
44	9	Opt	64	0.14	0	0	0	Opt	64	0.22	0	0	0	Opt	64	0.2	0	0	0
44	10	Opt	63	0.16	0	0	0	Opt	63	0.33	0	0	0	Opt	63	0.28	0	0	0
45	1	Opt	82	49.2	0	0	65741	Feas	82	300	12.6	0	157415	Feas	84	300	12.27	2.44	98795
45	2	Feas	125	300	16.43	0	390950	Feas	125	300	22.15	0	182892	Feas	125	300	19.83	0	242620
45	3	Opt	92	3.54	0	0	4604	Opt	92	19.95	0	0	11129	Opt	92	34.63	0.01	0	27033
45	4	Opt	84	17.3	0	0	14784	Opt	84	31.12	0	0	20381	Opt	84	76.78	0	0	53144
45	5	Opt	86	4.81	0	0	6484	Opt	86	23.63	0	0	21130	Opt	86	52.51	0	0	33865
45	6	Feas	129	300	11.63	0	355254	Feas	129	300	22.17	0	152365	Feas	129	300	21.66	0	117374
45	7	Opt	101	11.36	0	0	15238	Opt	101	102.55	0.01	0	71971	Opt	101	127.95	0.01	0	102852
45	8	Opt	94	11.71	0.01	0	19287	Opt	94	129.15	0	0	98088	Opt	94	66.55	0	0	51932
45	9	Opt	82	13.96	0	0	16366	Opt	82	156.22	0	0	104629	Opt	82	146.72	0.01	0	78581
45	10	Opt	90	60.81	0	0	112745	Opt	90	245.39	0	0	191695	Feas	90	300	6.67	0	136543
46	1	Opt	59	16.75	0	0	20035	Opt	59	13.68	0	0	14175	Opt	59	28.45	0	0	27417
46	2	Opt	67	41.15	0	0	31878	Opt	67	35.51	0	0	30140	Opt	67	27.47	0	0	22355
46	3	Opt	65	29.31	0	0	29512	Opt	65	67.28	0	0	54660	Opt	65	79.97	0	0	61494
46	4	Opt	64	5.3	0	0	7143	Opt	64	3.87	0	0	5003	Opt	64	6.79	0	0	6891
46	5	Opt	57	2.2	0	0	3529	Opt	57	3.98	0	0	3835	Opt	57	4.23	0	0	3092
46	6	Opt	59	33.09	0	0	40179	Opt	59	42.89	0	0	45038	Opt	59	49.22	0	0	42178
46	7	Opt	59	148.62	0.01	0	104372	Feas	60	300	7.9	1.69	243132	Feas	59	300	3.39	0	92184
46	8	Opt	58	45.05	0.01	0	25827	Opt	58	7.58	0	0	6542	Opt	58	20.08	0	0	13358
46	9	Opt	49	5.57	0	0	5711	Opt	49	4.32	0	0	4209	Opt	49	9.69	0	0	7585
46	10	Opt	55	63.65	0	0	27374	Opt	55	59.3	0	0	32868	Opt	55	73.04	0	0	31920
47	1	Opt	58	0.16	0	0	0	Opt	58	0.3	0	0	0	Opt	58	0.23	0	0	0
47	2	Opt	59	0.47	0	0	0	Opt	59	0.45	0	0	0	Opt	59	1.05	0	0	1097
47	3	Opt	55	0.9	0	0	956	Opt	55	1.14	0	0	1422	Opt	55	1.33	0	0	746
47	4	Opt	49	1.56	0	0	2510	Opt	49	2.43	0	0	4647	Opt	49	3.15	0	0	4919
47	5	Opt	47	3.28	0	0	3543	Opt	47	3.12	0	0	2948	Opt	47	5.88	0	0	3643
47	6	Opt	53	5.43	0	0	6718	Opt	53	3.74	0	0	5374	Opt	53	3.59	0	0	3293
47	7	Opt	66	13.93	0	0	47803	Opt	66	3.14	0	0	5170	Opt	66	4.73	0	0	5165
47	8	Opt	48	0.19	0	0	0	Opt	48	0.25	0	0	0	Opt	48	0.28	0	0	0
47	9	Opt	65	0.62	0	0	793	Opt	65	0.31	0	0	0	Opt	65	0.39	0	0	0
47	10	Opt	60	42.77	0	0	58890	Opt	60	42.64	0	0	55999	Opt	60	24.27	0	0	35358
48	1	Opt	63	0.25	0	0	0	Opt	63	0.22	0	0	0	Opt	63	0.36	0	0	0
48	2	Opt	54	0.28	0	0	0	Opt	54	0.36	0	0	0	Opt	54	0.56	0	0	37
48	3	Opt	50	0.34	0	0	0	Opt	50	0.25	0	0	0	Opt	50	0.28	0	0	0
48	4	Opt	57	0.39	0	0	109	Opt	57	0.27	0	0	0	Opt	57	0.23	0	0	0
48	5	Opt	58	0.44	0	0	37	Opt	58	0.48	0	0	44	Opt	58	1.05	0	0	1010
48	6	Opt	58	0.34	0	0	0	Opt	58	0.27	0	0	0	Opt	58	0.23	0	0	0
48	7	Opt	55	0.4	0	0	133	Opt	55	0.27	0	0	0	Opt	55	0.22	0	0	0
48	8	Opt	44	0.16	0	0	0	Opt	44	0.2	0	0	0	Opt	44	0.22	0	0	0
48	9	Opt	59	0.33	0	0	0	Opt	59	0.28	0	0	0	Opt	59	0.27	0	0	0
48	10	Opt	54	0.33	0	0	0	Opt	54	0.33	0	0	0	Opt	54	0.27	0	0	0

Table A.1. Results for F1, Kop-CT1-m, and Kop-CT2-m on each instance of set j30.

Appendix B

Node logs for an example with 5 activities

Here we report the node logs of the formulations F1, Kop-CT1-m and Kop-CT2-m for the example introduced in Section 3.2.2. These logs were used to build the branching trees in Figure 3.4.

	Node	Nodes Left	Objective	IInf	Best Integer	Cuts/ Best Bound	ItCnt	Gap	Variable	B	NodeID	Parent	Depth
	0	0	5.0000	10		5.0000	12						
	0	2	5.0000	8		5.0000	12				0		0
	1	3	9.0000	4		5.0000	17		theta4.5	D	4	0	1
	2	3	9.0000	4		5.0000	17		theta4.5	U	3	0	1
	3	4	12.0000	1		5.0000	20		theta3.4	D	8	4	2
	4	4	9.0000	3		5.0000	22		theta3.4	U	2	4	2
	5	5	12.0000	1		9.0000	25		theta3.5	D	7	3	2
	6	7	12.0000	2		9.0000	31		theta3.5	D	6	2	3
	7	5	9.0000	3		9.0000	27		theta3.5	U	1	3	2
	8	9	14.0000	1		9.0000	35		theta2.4	D	10	6	4
	9	10	12.0000	1		9.0000	41		theta1.5	D	5	1	3
	10	6	12.0000	1		9.0000	26		theta2.3	D	12	8	3
*	11	8	integral	0	15.0000	9.0000	36	40.00%					
	11	8	15.0000	0	15.0000	9.0000	36	40.00%	theta1.4	D	14	10	5
	12	9	-1.00000e+75	0	15.0000	9.0000	42	40.00%					
	12	9	15.0000	0	15.0000	9.0000	42	40.00%	theta3.4	D	9	5	4
	13	8	12.0000	1	15.0000	9.0000	34	40.00%	theta2.3	D	11	7	3
*	14	8	integral	0	12.0000	9.0000	41	25.00%					
	14	8	12.0000	0	12.0000	9.0000	41	25.00%	gamma3.2	U	16	12	4
	15	10	cutoff		12.0000	9.0000	49	25.00%	theta3.4	U	13	5	4
	16	9	-1.00000e+75	0	12.0000	9.0000	44	25.00%					
	16	9	12.0000	0	12.0000	9.0000	44	25.00%	gamma3.2	U	15	11	4
	17	11	9.0000	3	12.0000	9.0000	53	25.00%	theta1.5	U	17	1	3
	18	11	14.0000	1	12.0000	9.0000	45	25.00%	theta1.4	U	18	10	5
	19	12	12.0000	2	12.0000	9.0000	56	25.00%	theta3.4	D	21	17	4
	20	11	14.0000	1	12.0000	9.0000	46	25.00%	theta1.3	D	22	18	6
	21	13	12.0000	2	12.0000	9.0000	59	25.00%	theta3.4	U	25	17	4
	22	9	-1.00000e+75	0	12.0000	9.0000	47	25.00%					
	22	9	14.0000	0	12.0000	9.0000	47	25.00%	gamma3.1	U	26	22	7
	23	14	14.0000	1	12.0000	9.0000	60	25.00%	theta2.5	D	29	21	5
	24	2	cutoff		12.0000	9.0000	59	25.00%	theta3.5	U	30	2	3
	25	14	14.0000	1	12.0000	9.0000	61	25.00%	theta1.3	D	33	29	6
	26	11	-1.00000e+75	0	12.0000	9.0000	62	25.00%					
	26	11	14.0000	0	12.0000	9.0000	62	25.00%	gamma3.1	U	37	33	7

Figure B.1. Node log of formulation F1.

	Node	Nodes Left	Objective	IInf	Best Integer	Cuts/ Best Bound	ItCnt	Gap	Variable	B	NodeID	Parent	Depth
	0	0	5.0000	33		5.0000	31						
	0	2	5.0000	21		5.0000	31				0		0
	1	3	9.0000	12		5.0000	37		x5.4	U	3	0	1
	2	3	9.0000	13		5.0000	40		x5.4	D	4	0	1
	3	4	9.0000	11		5.0000	46		x5.3	U	7	3	2
	4	4	12.0000	7		5.0000	45		x5.3	D	2	3	2
	5	5	9.0000	6		9.0000	59		x2.5	D	11	7	3
	6	5	infeasible			9.0000	64		x4.3	U	6	2	3
	7	4	9.0000	11		9.0000	53		x4.3	U	8	4	2
	8	5	12.0000	7		9.0000	53		x4.3	D	1	4	2
	9	6	12.0000	4		9.0000	65		x4.3	D	10	2	3
	10	5	9.0000	9		9.0000	59		z4.3	D	12	8	3
	11	8	12.0000	1		9.0000	78		x3.2	U	14	10	4
	12	7	9.0000	11		9.0000	72		z5.2	D	15	11	4
	13	6	infeasible			9.0000	75		x5.3	U	5	1	3
*	14	8	integral	0	13.0000	9.0000	94	30.77%					
	14	8	13.0000	0	13.0000	9.0000	94	30.77%	z1.3	D	18	14	5
	15	8	9.0000	8	13.0000	9.0000	75	30.77%	x4.2	U	19	15	5
	16	7	12.0000	4	13.0000	9.0000	76	30.77%	x5.3	D	9	1	3
	17	8	9.0000	7	13.0000	9.0000	91	30.77%	x4.2	U	16	12	4
	18	6	12.0000	1	13.0000	9.0000	98	30.77%	x3.2	U	13	9	4
	19	7	9.0000	9	13.0000	9.0000	102	30.77%	z4.2	D	20	16	5
*	20	6	integral	0	12.0000	9.0000	95	25.00%					
	20	6	12.0000	0	12.0000	9.0000	95	25.00%	z1.3	U	22	14	5
	21	5	-1.00000e+75	0	12.0000	9.0000	99	25.00%					
	21	5	12.0000	0	12.0000	9.0000	99	25.00%	z1.3	U	17	13	5
	22	7	12.0000	4	12.0000	9.0000	106	25.00%	x5.3	D	24	20	6
	23	9	9.0000	6	12.0000	9.0000	78	25.00%	x3.2	U	23	19	6
	24	7	12.0000	2	12.0000	9.0000	110	25.00%	x4.1	U	28	24	7
	25	10	9.0000	10	12.0000	9.0000	80	25.00%	x3.2	D	27	19	6
	26	7	12.0000	4	12.0000	9.0000	116	25.00%	z4.1	D	32	28	8
	27	11	9.0000	7	12.0000	9.0000	82	25.00%	x4.3	U	31	27	7
	28	8	12.0000	1	12.0000	9.0000	119	25.00%	x3.2	U	36	32	9
	29	7	cutoff		12.0000	9.0000	121	25.00%	z1.3	D	40	36	10
	30	11	9.0000	5	12.0000	9.0000	87	25.00%	x5.1	U	35	31	8
	31	6	-1.00000e+75	0	12.0000	9.0000	122	25.00%					
	31	6	12.0000	0	12.0000	9.0000	122	25.00%	z1.3	U	44	36	10
	32	5	9.0000	7	12.0000	9.0000	128	25.00%	x5.3	U	26	20	6
	33	3	cutoff		12.0000	9.0000	130	25.00%	x4.2	D	48	12	4
	34	4	infeasible		12.0000	9.0000	128	25.00%	x4.1	D	21	26	7
	35	6	9.0000	5	12.0000	9.0000	132	25.00%	x4.1	U	30	26	7
	36	3	infeasible		12.0000	9.0000	136	25.00%	z2.5	D	52	30	8
	37	12	9.0000	8	12.0000	9.0000	91	25.00%	z5.1	D	39	35	9
	38	5	9.0000	4	12.0000	9.0000	140	25.00%	z2.5	U	34	30	8
	39	13	9.0000	7	12.0000	9.0000	92	25.00%	z4.1	U	43	39	10
	40	12	infeasible		12.0000	9.0000	95	25.00%	x4.1	U	47	43	11
	41	5	11.0000	4	12.0000	9.0000	146	25.00%	z5.2	D	38	34	9
	42	3	cutoff		12.0000	9.0000	143	25.00%	z5.2	U	56	34	9
	43	4	infeasible		12.0000	9.0000	146	25.00%	x3.2	D	25	38	10
	44	6	11.0000	2	12.0000	9.0000	147	25.00%	x3.2	U	42	38	10
	45	3	infeasible		12.0000	9.0000	150	25.00%	z5.3	U	46	42	11
	46	13	10.0000	4	12.0000	9.0000	97	25.00%	x4.1	D	51	43	11
	47	12	infeasible		12.0000	9.0000	97	25.00%	z4.3	U	55	51	12
	48	13	13.0000	2	12.0000	9.0000	98	25.00%	z4.3	D	59	51	12
	49	12	infeasible		12.0000	9.0000	98	25.00%	z5.3	U	63	59	13
	50	3	cutoff		12.0000	9.0000	152	25.00%	z5.3	D	60	42	11
	51	13	13.0000	1	12.0000	9.0000	99	25.00%	z5.3	D	67	59	13
	52	14	13.0000	1	12.0000	9.0000	100	25.00%	z2.3	D	71	67	14
	53	13	-1.00000e+75	0	12.0000	9.0000	101	25.00%					
	53	11	13.0000	0	12.0000	9.0000	101	25.00%	z2.3	U	75	67	14
	54	6	infeasible		12.0000	9.0000	186	25.00%	z4.1	D	50	39	10
	55	6	infeasible		12.0000	9.0000	186	25.00%	x4.3	D	29	27	7
	56	6	infeasible		12.0000	9.0000	187	25.00%	z5.3	U	64	23	7
	57	8	9.0000	9	12.0000	9.0000	188	25.00%	z5.3	D	79	23	7
	58	5	infeasible		12.0000	9.0000	187	25.00%	x5.1	D	68	31	8
	59	7	11.0000	7	12.0000	9.0000	189	25.00%	x4.2	D	54	15	5
	60	7	cutoff		12.0000	9.0000	191	25.00%	x4.3	D	83	79	8
	61	8	11.0000	8	12.0000	9.0000	191	25.00%	z5.3	D	58	54	6
	62	7	9.0000	6	12.0000	9.0000	194	25.00%	x4.3	U	87	79	8
	63	7	cutoff		12.0000	9.0000	194	25.00%	x4.3	D	62	58	7
	64	6	infeasible		12.0000	9.0000	194	25.00%	z4.3	U	91	87	9
	65	7	11.0000	4	12.0000	9.0000	199	25.00%	x4.3	U	66	58	7
	66	5	cutoff		12.0000	9.0000	196	25.00%	x2.5	U	95	7	3
	67	6	infeasible		12.0000	9.0000	202	25.00%	z4.3	U	70	66	8
	68	5	infeasible		12.0000	9.0000	204	25.00%	z5.3	U	74	54	6

Figure B.2. Node log of formulation Kop-CT1-m.

	Node	Nodes Left	Objective	IInf	Best Integer	Cuts/ Best Bound	ItCnt	Gap	Variable	B	NodeID	Parent	Depth
	0	0	5.0000	24		5.0000	30						
	0	2	5.0000	15		5.0000	30				0		0
	1	3	9.0000	9		5.0000	37		x4.5	D	4	0	1
	2	4	9.0000	8		5.0000	39		z4.3	U	8	4	2
	3	3	9.0000	9		5.0000	39		x4.5	U	3	0	1
	4	5	9.0000	7		5.0000	41		z5.3	U	12	8	3
	5	6	9.0000	8		9.0000	49		z5.3	U	7	3	2
	6	5	infeasible			9.0000	56		z3.5	U	16	12	4
	7	5	12.0000	5		9.0000	59		z3.5	D	20	12	4
	8	6	12.0000	3		9.0000	60		x3.4	U	24	20	5
	9	5	9.0000	7		9.0000	42		z5.3	D	1	8	3
	10	4	12.0000	5		9.0000	47		z4.3	D	2	4	2
	11	8	14.0000	1		9.0000	62		z4.2	D	28	24	6
	12	7	9.0000	9		9.0000	54		z5.1	U	11	7	3
	13	8	infeasible			9.0000	81		z5.3	U	6	2	3
*	14	7	integral	0	15.0000	9.0000	63	40.00%					
	14	7	15.0000	0	15.0000	9.0000	63	40.00%	z4.1	D	32	28	7
	15	9	12.0000	4	15.0000	9.0000	82	40.00%	z5.3	D	10	2	3
	16	10	12.0000	3	15.0000	9.0000	83	40.00%	z3.1	U	14	10	4
	17	11	12.0000	1	15.0000	9.0000	84	40.00%	z3.2	U	18	14	5
	18	9	9.0000	8	15.0000	9.0000	69	40.00%	z3.1	U	5	1	4
	19	10	12.0000	5	15.0000	9.0000	73	40.00%	x3.4	U	9	5	5
	20	8	9.0000	10	15.0000	9.0000	57	40.00%	z1.5	D	15	11	4
*	21	8	integral	0	12.1000	9.0000	86	25.62%	x1.3	D	22	18	6
	22	9	9.0000	7	12.1000	9.0000	59	25.62%	z4.3	U	19	15	5
	23	9	14.0000	2	12.1000	9.0000	78	25.62%	z4.1	U	36	28	7
	24	10	12.0000	4	12.1000	9.0000	74	25.62%	x1.4	D	13	9	6
	25	9	14.0000	1	12.1000	9.0000	81	25.62%	z1.4	D	40	36	8
*	26	6	integral	0	12.0000	9.0000	89	25.00%					
	26	6	12.0000	0	12.0000	9.0000	89	25.00%	x1.3	U	26	18	6
	27	9	-1.00000e+75	0	12.0000	9.0000	77	25.00%					
	27	9	15.0000	0	12.0000	9.0000	77	25.00%	x1.3	D	17	13	7
	28	5	infeasible		12.0000	9.0000	102	25.00%	z3.1	D	30	1	4
	29	10	9.0000	9	12.0000	9.0000	63	25.00%	z4.3	D	23	15	5
	30	5	cutoff		12.0000	9.0000	105	25.00%	x3.4	D	21	5	5
	31	9	14.0000	2	12.0000	9.0000	84	25.00%	z5.1	U	44	40	9
	32	9	14.0000	1	12.0000	9.0000	87	25.00%	x1.5	U	48	44	10
	33	8	-1.00000e+75	0	12.0000	9.0000	89	25.00%					
	33	8	14.0000	0	12.0000	9.0000	89	25.00%	z3.1	U	52	48	11
	34	10	9.0000	7	12.0000	9.0000	66	25.00%	z5.2	U	27	19	6
	35	11	9.0000	8	12.0000	9.0000	73	25.00%	z2.5	D	31	27	7
	36	12	9.0000	7	12.0000	9.0000	74	25.00%	z3.1	U	35	31	8
	37	13	9.0000	8	12.0000	9.0000	76	25.00%	z3.2	U	39	35	9
	38	14	11.0000	5	12.0000	9.0000	77	25.00%	x2.4	D	43	39	10
	39	15	12.0000	6	12.0000	9.0000	82	25.00%	x2.4	U	47	39	10
	40	14	infeasible		12.0000	9.0000	86	25.00%	x3.4	D	51	43	11
	41	15	14.0000	3	12.0000	9.0000	90	25.00%	x3.4	U	55	43	11
	42	14	-1.00000e+75	0	12.0000	9.0000	91	25.00%					
	42	14	15.0000	0	12.0000	9.0000	91	25.00%	x1.4	D	59	55	12
	43	4	infeasible		12.0000	9.0000	158	25.00%	z5.2	D	25	19	6
	44	4	cutoff		12.0000	9.0000	160	25.00%	z5.3	D	67	3	2
	45	4	cutoff		12.0000	9.0000	161	25.00%	z3.2	D	34	35	9
	46	3	infeasible		12.0000	9.0000	160	25.00%	z3.1	D	71	31	8
	47	6	9.0000	9	12.0000	9.0000	163	25.00%	z3.1	U	56	23	6
	48	5	infeasible		12.0000	9.0000	163	25.00%	x1.3	D	60	56	7
	49	5	9.0000	8	12.0000	9.0000	166	25.00%	x1.3	U	64	56	7
	50	6	9.0000	7	12.0000	9.0000	167	25.00%	z1.3	D	68	64	8
	51	5	infeasible		12.0000	9.0000	167	25.00%	z3.5	D	72	68	9
	52	4	cutoff		12.0000	9.0000	169	25.00%	z3.5	U	76	68	9

Figure B.3. Node log of formulation Kop-CT2-m.

Appendix C

Additional graphs of results

The additional graphs of results, which are structured as those in Figures 3.7 and 3.8, and which separately consider the instances in each test set, are reported here.

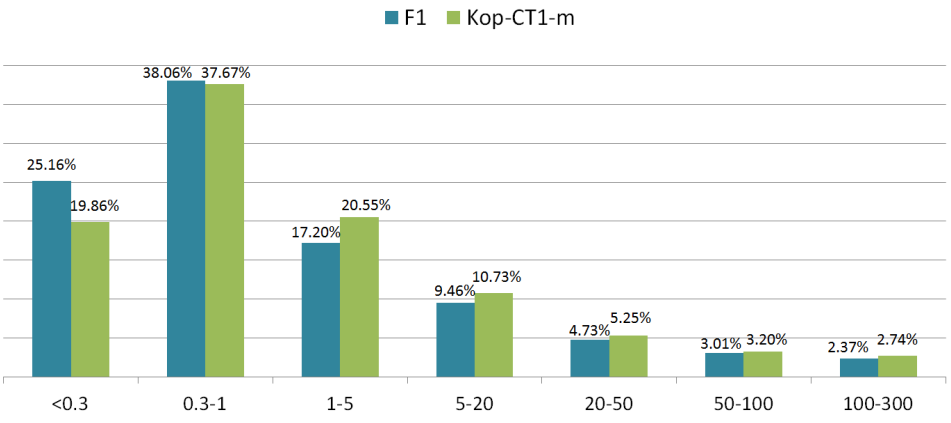

Figure C.1. Distribution of computation times over different ranges for F1 and Kop-CT1-m, considering only the instances optimally solved among those of set j30 of PSPLib.

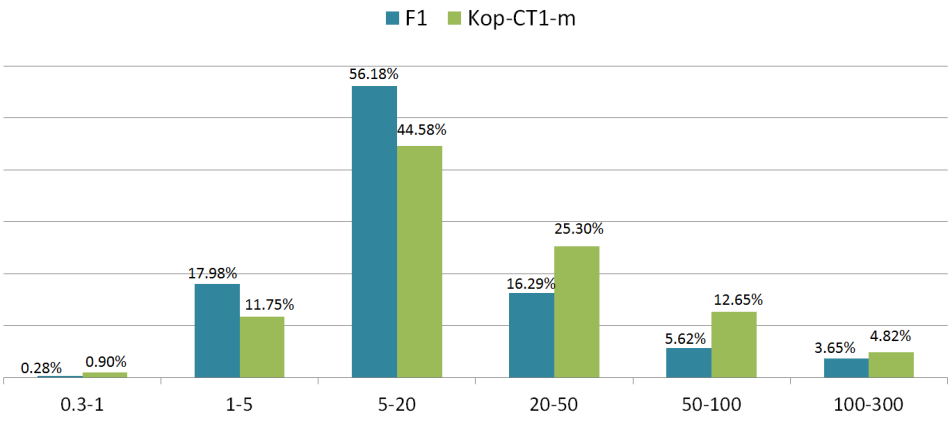

Figure C.2. Distribution of computation times over different ranges for F1 and Kop-CT1-m, considering only the instances optimally solved among those of set j60 of PSPLib.

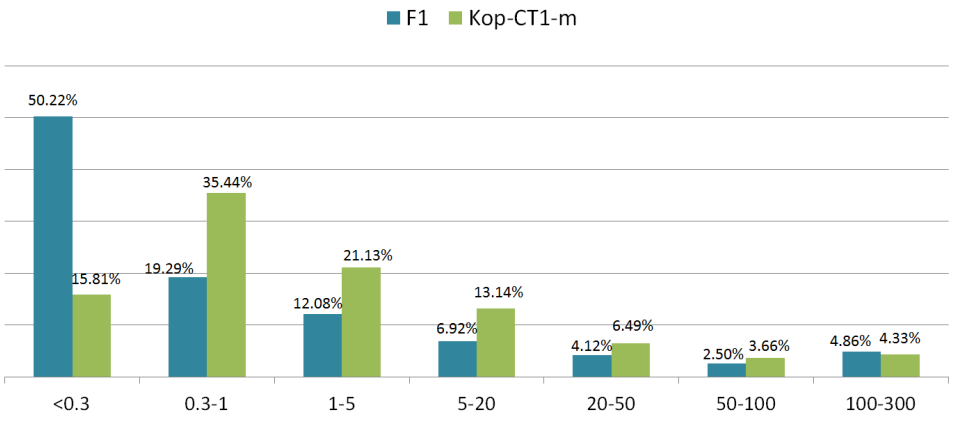

Figure C.3. Distribution of computation times over different ranges for F1 and Kop-CT1-m, considering only the instances optimally solved among those of Set 1 of RanGen2.

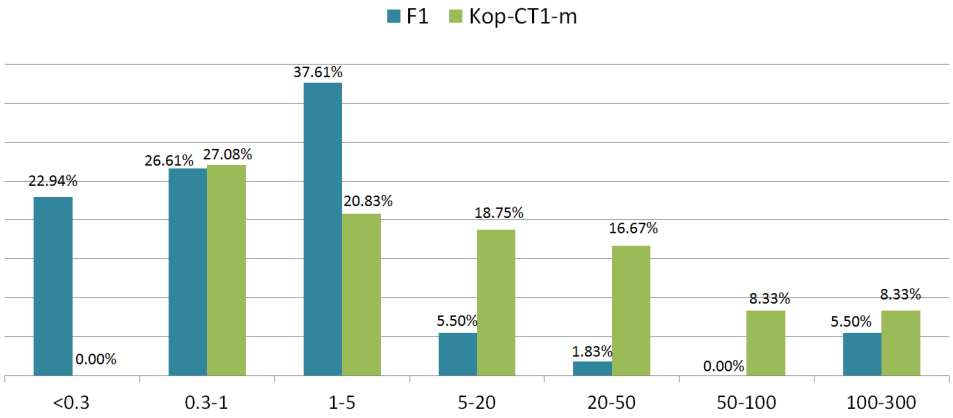

Figure C.4. Distribution of computation times over different ranges for F1 and Kop-CT1-m, considering only the instances optimally solved among those of Set 2 of RanGen2.

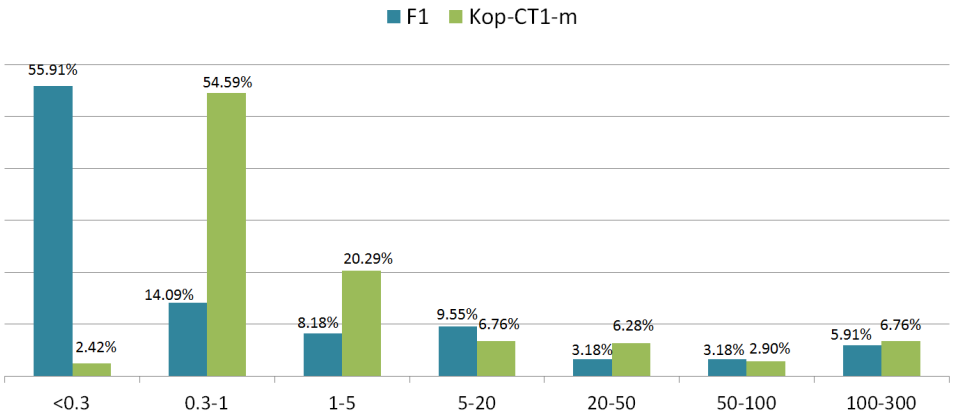

Figure C.5. Distribution of computation times over different ranges for F1 and Kop-CT1-m, considering only the instances optimally solved among those of Set 3 of RanGen2.

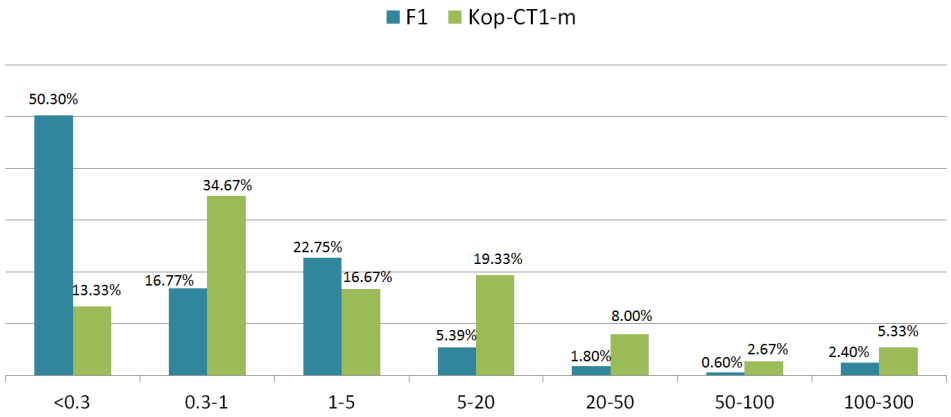

Figure C.6. Distribution of computation times over different ranges for F1 and Kop-CT1-m, considering only the instances optimally solved among those of Set 4 of RanGen2.

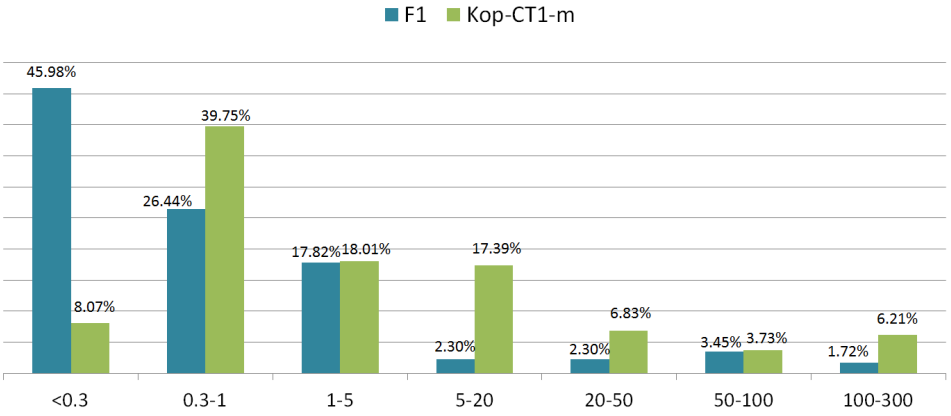

Figure C.7. Distribution of computation times over different ranges for F1 and Kop-CT1-m, considering only the instances optimally solved among those of set Set 5 of RanGen2.

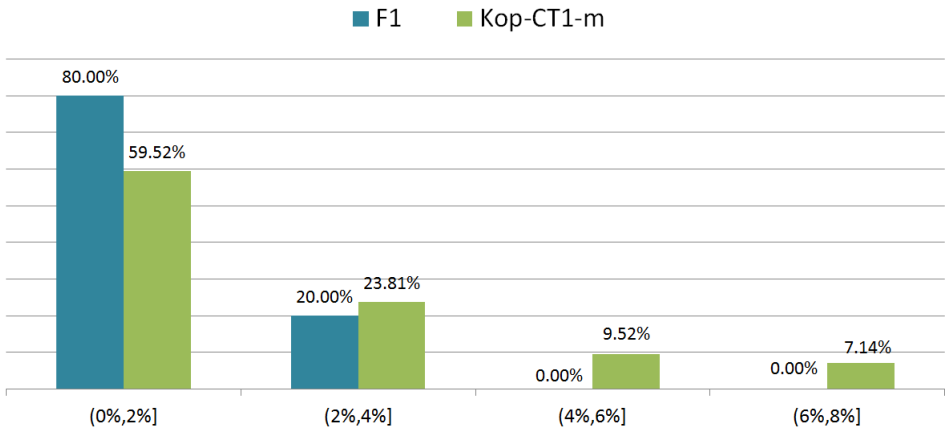

Figure C.8. Distribution of actual gaps over different ranges for F1 and Kop-CT1-m, considering only the sub-/non-proven optimal instances among those of set j30 of PSPLib.

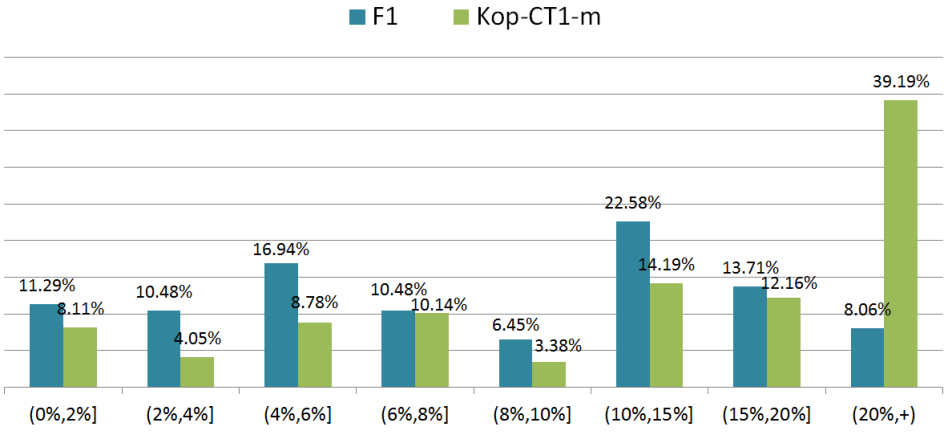

Figure C.9. Distribution of actual gaps over different ranges for F1 and Kop-CT1-m, considering only the suboptimal or non-proven optimal instances among those of set j60 of PSPLib.

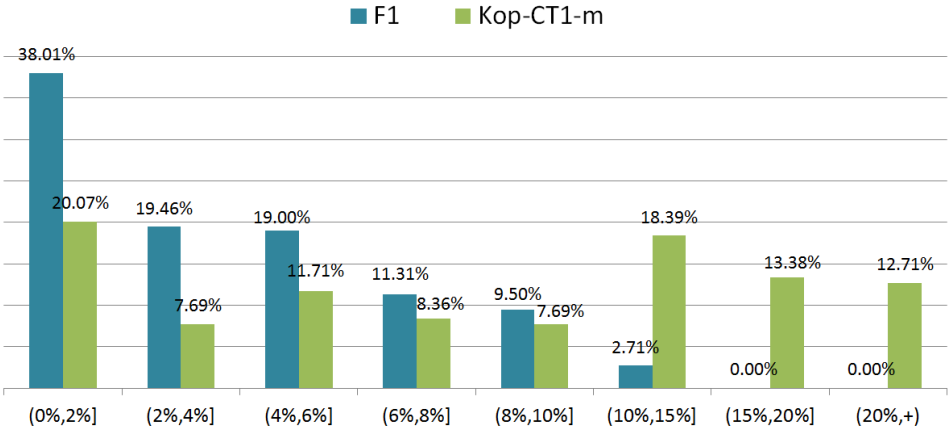

Figure C.10. Distribution of actual gaps over different ranges for F1 and Kop-CT1-m, considering only the sub-/non-proven optimal instances among those of Set 1 of RanGen2.

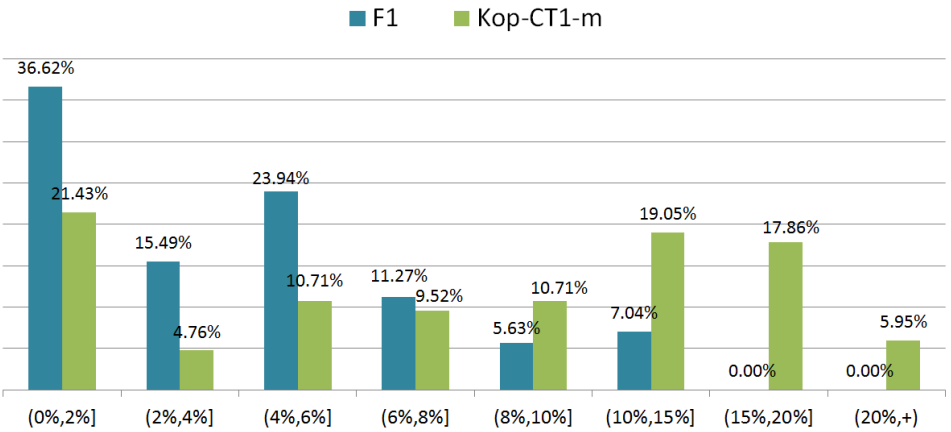

Figure C.11. Distribution of actual gaps over different ranges for F1 and Kop-CT1-m, considering only the sub-/non-proven optimal instances among those of Set 2 of RanGen2.

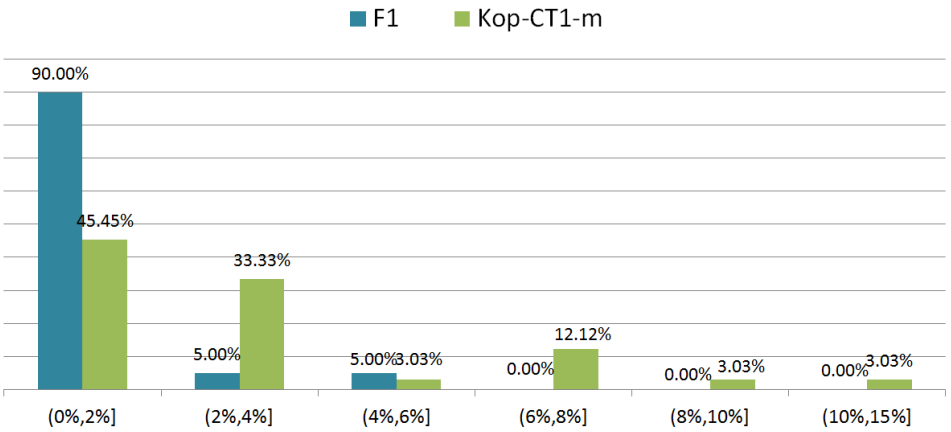

Figure C.12. Distribution of actual gaps over different ranges for F1 and Kop-CT1-m, considering only the sub-/non-proven optimal instances among those of Set 3 of RanGen2.

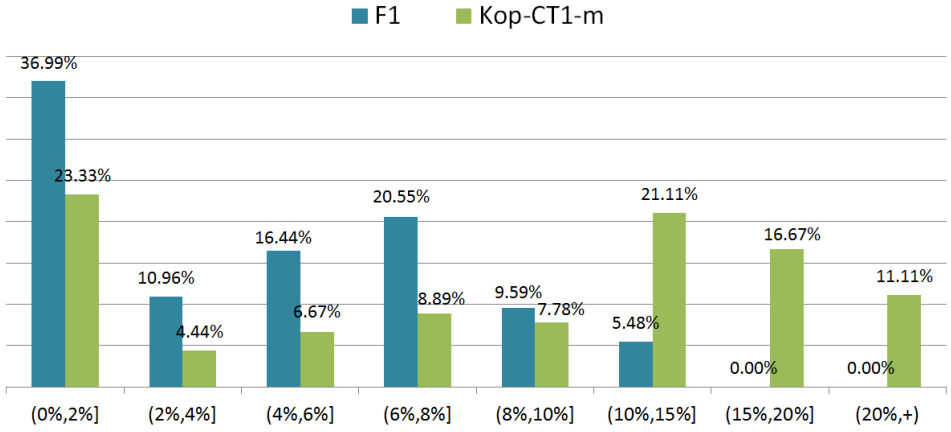

Figure C.13. Distribution of actual gaps over different ranges for F1 and Kop-CT1-m, considering only the sub-/non-proven optimal instances among those of Set 4 of RanGen2.

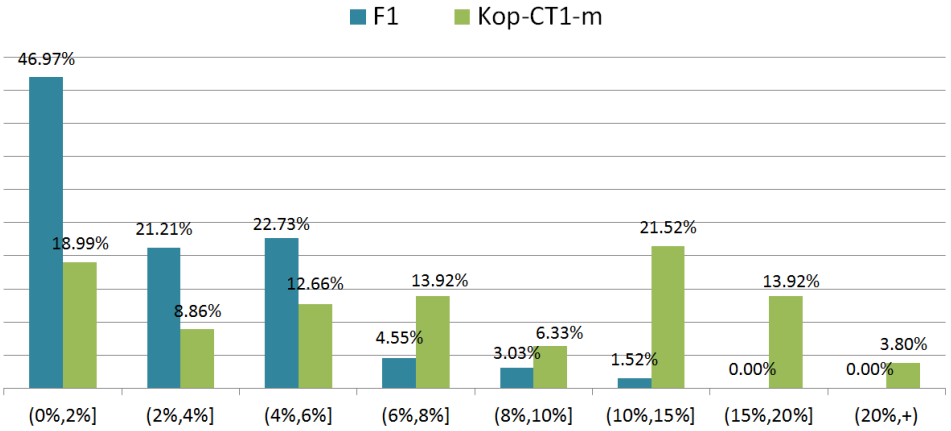

Figure C.14. Distribution of actual gaps over different ranges for F1 and Kop-CT1-m, considering only the sub-/non-proven optimal instances among those of Set 5 of RanGen2.

Bibliography

Alvarez-Valdés, R. and J. M. Tamarit (1993). The project scheduling polyhedron: Dimension, facets and lifting theorems. *European Journal of Operational Research 67*(2), 204 – 220.

Artigues, C., S. Demassey, and E. Neron (2013). *Resource-constrained project scheduling: models, algorithms, extensions and applications*. John Wiley & Sons.

Artigues, C., P. Michelon, and S. Reusser (2003). Insertion techniques for static and dynamic resource-constrained project scheduling. *European Journal of Operational Research 149*(2), 249 – 267.

Ballestín, F. and R. Blanco (2011). Theoretical and practical fundamentals for multi-objective optimisation in resource-constrained project scheduling problems. *Computers & Operations Research 38*(1), 51 – 62.

Baptiste, P. and C. L. Pape (2000). Constraint propagation and decomposition techniques for highly disjunctive and highly cumulative project scheduling problems. *Constraints 5*(1), 119–139.

Bartusch, M., R. H. Möhring, and F. J. Radermacher (1988). Scheduling project networks with resource constraints and time windows. *Annals of Operations Research 16*(1), 199–240.

Bianco, L. and M. Caramia (2013). A new formulation for the project scheduling problem under limited resources. *Flexible Services & Manufacturing Journal 25*(1/2), 6 – 24.

Blazewicz, J., J. Lenstra, and A. Kan (1983). Scheduling subject to resource constraints: classification and complexity. *Discrete Applied Mathematics 5*(1), 11 – 24.

Brucker, P., A. Drexl, R. Möhring, K. Neumann, and E. Pesch (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research 112*(1), 3 – 41.

Brucker, P., S. Knust, A. Schoo, and O. Thiele (1998). A branch and bound algorithm for the resource-constrained project scheduling problem1supported by the deutsche forschungsgemeinschaft, project 'komplexe maschinen-schedulingprobleme'.1. *European Journal of Operational Research 107*(2), 272 – 288.

Chand, S., H. K. Singh, and T. Ray (2017). A heuristic algorithm for solving resource constrained project scheduling problems. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 225–232.

Chen, R.-M. (2011). Particle swarm optimization with justification and designed mechanisms for resource-constrained project scheduling problem. *Expert Systems with Applications 38*(6), 7102 – 7111.

Christofides, N., R. Alvarez-Valdés, and J. Tamarit (1987). Project scheduling with resource constraints: A branch and bound approach. *European Journal of Operational Research 29*(3), 262 – 273.

Demassey, S., C. Artigues, and P. Michelon (2005). Constraint-propagation-based cutting planes: An application to the resource-constrained project scheduling problem. *INFORMS Journal on Computing 17*(1), 52–65.

Demeulemeester, E. and W. Herroelen (1992). A branch-and-bound procedure for the multiple resource-constrained project scheduling problem. *Management Science 38*(12), 1803–1818.

Dorndorf, U., E. Pesch, and T. Phan-Huy (2000). A branch-and-bound algorithm for the resource-constrained project scheduling problem. *Mathematical Methods of Operations Research 52*(3), 413–439.

Hartmann, S. (2000). Packing problems and project scheduling models: an integrating perspective. *Journal of the Operational Research Society 51*(9), 1083–1092.

Hartmann, S. and D. Briskorn (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research 207*(1), 1 – 14.

Hartmann, S. and R. Kolisch (2000). Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research 127*(2), 394 – 407.

Herroelen, W. (2005). Project scheduling - theory and practice. *Production and Operations Management 14*(4), 413–432.

Herroelen, W. and R. Leus (2005). Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research 165*(2), 289 – 306.

Jia, Q. and Y. Seo (2013). Solving resource-constrained project scheduling problems: Conceptual validation of flp formulation and efficient permutation-based abc computation. *Computers & Operations Research 40*(8), 2037 – 2050.

Kaplan, L. A. (1988). *Resource-constrained project scheduling with preemption of jobs.* Ph. D. thesis, University of Michigan.

Kelley, J. (1963). The critical path method: Resources planning and scheduling. *Industrial Scheduling 13*, 347 – 365.

Klein, R. (2000). Project scheduling with time-varying resource constraints. *International Journal of Production Research 38*(16), 3937–3952.

Kolisch, R. and S. Hartmann (2006). Experimental investigation of heuristics for resource-constrained project scheduling: An update. *European Journal of Operational Research 174*(1), 23 – 37.

Kolisch, R. and A. Sprecher (1997). Psplib - a project scheduling problem library: Or software - orsep operations research software exchange program. *European Journal of Operational Research 96*(1), 205 – 216.

Koné, O., C. Artigues, P. Lopez, and M. Mongeau (2011). Event-based milp models for resource-constrained project scheduling problems. *Computers & Operations Research 38*(1), 3 – 13.

Koné, O., C. Artigues, P. Lopez, and M. Mongeau (2013). Comparison of mixed integer linear programming models for the resource-constrained project scheduling problem with consumption and production of resources. *Flexible Services and Manufacturing Journal 25*(1), 25–47.

Kopanos, G. M., T. S. Kyriakidis, and M. C. Georgiadis (2014). New continuous-time and discrete-time mathematical formulations for resource-constrained project scheduling problems. *Computers & Chemical Engineering 68*, 96 – 106.

Laborie, P. (2005). Complete mcs-based search: Application to resource constrained project scheduling. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, San Francisco, CA, USA, pp. 181–186. Morgan Kaufmann Publishers Inc.

Mingozzi, A., V. Maniezzo, S. Ricciardelli, and L. Bianco (1998). An exact algorithm for the resource-constrained project scheduling problem based on a new mathematical formulation. *Management Science 44*(5), 714–729.

Naber, A. and R. Kolisch (2014). Mip models for resource-constrained project scheduling with flexible resource profiles. *European Journal of Operational Research 239*(2), 335 – 348.

Naber, A., R. Kolisch, L. Bianco, and M. Caramia (2014). The resource-constrained project scheduling model of bianco and caramia: Clarifications and an alternative model formulation. *Flexible Services and Manufacturing Journal 26*(3), 454–459.

Pinto, J. and I. Grossmann (1996). A continuous time milp model for short term scheduling of batch plants with pre-ordering constraints. *Computers & Chemical Engineering 20*, S1197 – S1202.

Pritsker, A. A. B., L. J. Waiters, and P. M. Wolfe (1969). Multiproject scheduling with limited resources: A zero-one programming approach. *Management Science 16*(1), 93–108.

Rihm, T. and N. Trautmann (2017). An assignment-based continuous-time milp model for the resource-constrained project scheduling problem. In *Industrial*

Engineering and Engineering Management (IEEM), 2017 IEEE International Conference on, pp. 35–39. IEEE.

Schwindt, C., J. Zimmermann, et al. (2015). *Handbook on Project Management and Scheduling*. Springer.

Vanhoucke, M., J. Coelho, D. Debels, B. Maenhout, and L. V. Tavares (2008). An evaluation of the adequacy of project network generators with systematically sampled networks. *European Journal of Operational Research 187*(2), 511 – 524.

Varakantham, P., N. Fu, and H. C. Lau (2016). A proactive sampling approach to project scheduling under uncertainty. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pp. 3195–3201. AAAI Press.

Weglarz, J., J. Józefowska, M. Mika, and G. Waligóra (2011). Project scheduling with finite or infinite number of activity processing modes - a survey. *European Journal of Operational Research 208*(3), 177 – 205.

Zamani, R. (2013). A competitive magnet-based genetic algorithm for solving the resource-constrained project scheduling problem. *European Journal of Operational Research 229*(2), 552 – 559.

Zapata, J. C., B. M. Hodge, and G. V. Reklaitis (2008). The multimode resource constrained multiproject scheduling problem: Alternative formulations. *AIChE Journal 54*(8), 2101–2119.

Zheng, X.-L. and L. Wang (2015). A multi-agent optimization algorithm for resource constrained project scheduling problem. *Expert Systems with Applications 42*(15), 6039 – 6049.