
Project management under uncertainty: using flexible resource management to exploit schedule flexibility

João Faria* and Madalena Araújo

ALGORITMI Research Centre,
School of Engineering,
University of Minho, Portugal
Email: joao.faria@algoritmi.uminho.pt
Email: mmaraujo@dps.uminho.pt
*Corresponding author

Erik Demeulemeester

Research Center for Operations Management,
Faculty of Economics and Business,
KU Leuven, Belgium
Email: erik.demeulemeester@kuleuven.be

Anabela Tereso

ALGORITMI Research Centre,
School of Engineering,
University of Minho, Portugal
Email: anabelat@dps.uminho.pt

Abstract: Project management still faces a wide gap separating theory from practice, especially regarding the robustness of the generated project schedules facing the omnipresence of uncertainty. A new approach to deal with uncertainty is presented to explore slack that might exist in a given project schedule. We propose that renewable resources' capacity to perform work can be increased so that they can perform additional work in a time unit or can be decreased with the consequent reduction on the performed work. This possibility combined with the slack that some activities have in a specific schedule can be used to absorb deviations that might occur during a project's execution. When a critical activity is about to have its duration increased, slowing down other non-critical activities by putting their resources in a decreased work mode enables the activity to still be executed within time by using resources in an increased working mode. [Received: 14 February 2018; Revised: 2 January 2019; Revised: 8 June 2019; Accepted: 17 November 2019]

Keywords: project management; scheduling; resource allocation; RCPSP; uncertainty.

Reference to this paper should be made as follows: Faria, J., Araújo, M., Demeulemeester, E. and Tereso, A. (2020) 'Project management under uncertainty: using flexible resource management to exploit schedule flexibility', *European J. Industrial Engineering*, Vol. 14, No. 5, pp.599–631.

Biographical notes: João Faria received a degree in Electronics and Telecommunications Engineering in 1989 at the University of Aveiro, Portugal and a post-graduation in Informatics in 1994 at the University of Minho, Portugal. He worked for 25 years as an engineer and manager for several companies while teaching electronics and informatics. More recently, he received his PhD in Systems and Industrial Engineering in 2016 at the University of Minho, Portugal where he is a researcher of the ALGORITMI Research Centre. Currently, he is a senior project manager at Altran, leading several large scale software projects for the automotive industry.

Madalena Araújo is a chemical engineer at Coimbra University (Portugal) and holds a MSc in Industrial Management and PhD in Production Engineering, both from Birmingham University (UK). She is a Full Professor on Industrial and Technology Management in University of Minho (Portugal). She is a researcher and leader of the Industrial Engineering and Management Group – ALGORITMI Research Centre. Her research interests are on the economics and management of engineering systems – decision and utility modelling, project evaluation and management, risk analysis, multi-criteria modelling, and their applications. She authored and co-authored over hundred papers and supervised dozens of MSc dissertations and PhD theses.

Erik Demeulemeester received his Business Engineering degree in Management Informatics, Master in Business Administration, and PhD in Applied Economics from KU Leuven, Leuven, Belgium, in 1987, 1988 and 1992, respectively. Since 1992, he has been a Full-time Professor with the Research Center for Operations Management, KU Leuven and currently teaches a course on project management and scheduling, a doctoral course on combinatorial optimisation and local search techniques, as well as a seminar on operations management. His main research interests are situated in the field of project scheduling and healthcare planning.

Anabela Tereso obtained a degree in Computer and Systems Engineering, MSc in Informatics and PhD in Production Engineering. She is an Assistant Professor at the Department of Production and Systems Engineering and Researcher in ALGORITMI Research Centre at University of Minho (Portugal). She was responsible for the creation of the Master in Engineering Project Management, having been director and/or member of the board from 2014 till the present. Her research interests are on project management, risk management, decision analysis and their applications. She authored and co-authored over 100 papers and supervised dozens of MSc dissertations and PhD theses.

1 Introduction

As our society becomes increasingly more complex, the need to establish well-defined processes and to define their associated entities and rules arises as mandatory so that complexity can be dealt with. Repetitive procedures coming from the industrial revolution were the first to emerge as such well-defined processes, being a major milestone in this direction the introduction of mass production at Ford with its model T. Several years passed until in the late 1950's, the first generally accepted methodologies that established processes to deal with non-repetitive tasks were developed. Since then much research effort has been made to better model and manage these non-repetitive

tasks known as projects. The definitions of a Project, as for example the definitions given in Kerzner (2013), Meredith and Mantel (2011) or PMBOK (PMI, 2013), commonly agree that it is a onetime endeavour aiming to reach a predefined set of goals. Often this implies a well-defined and committed a priori cost and delivery date. It is therefore imperative that the project team, and especially the project manager, have not only the necessary skills, but also the best tools to help them getting it right the first time. On the other hand, project managers and their teams face increasing challenges as projects become more complex along with increasing competitiveness. A typical scenario for the project execution is that of assigning a set of resources that are available for the duration of the project which leaves little space for coping with uncertainties, especially when the project plan is established as an optimal or near optimal schedule. So, uncertainty resulting from several origins, along with resource unforeseen unavailability (Elmaghraby, 2005), often collides with the demand to deliver on time and with no additional costs. Many times, the method at hand is to use the available resources to work more within the same time unit either by considering this extra work as overtime (with additional costs) or not (Jia et al., 2007; Olsen and Swenson, 2011).

These are the issues that are further presented and a research line is identified to cope with these increasing demands.

In the remaining of this document, the following notation is assumed:

<i>Symbol</i>	<i>Description</i>
G	Project network $G = (V, A)$
V	Set of all project activities
A	Set of precedence relations
i	Activity $i = \{1, \dots, n\} \in \mathbb{N}$
$i = 1$	Dummy start
$i = n$	Dummy end
(i, j)	Precedence: $(i, j) \in A$, i is an immediate predecessor of j
$Pred_i$	Set of (immediate) predecessors of i
$Succ_i$	Set of (immediate) successors of i
s_i	Start time of i
d_i	Duration of i
d_i^{nom}	Nominal (starting or deterministic) duration of i
d_i^{min}	Minimal duration of i
d_i^{max}	Maximal duration of i
f_i	Finish time of i
K	Set of resource types
k	Resource type $k = \{1, \dots, m\}$ $k \in K$
r_{ik}	Resource requirement of i of type k
a_k	Resource availability of type k
a_k^{nom}	Nominal resource availability of type k
α_k^-	Negative resource flexibility (maximum decrease of a_k from nominal)

α_k^+	Positive resource flexibility (maximum increase of a_k from nominal)
u_{kt}	Resource unused capacity of type k at time t
w_{ik}	Work content of i for resource k
e	A project example
P_t	Set of activities in progress (active) at time t
S	A schedule $S = \{s_1, s_2, \dots, s_n\}$
S^b	A baseline schedule
S^w	A working schedule
T	Project duration (completion time)
δ	Project deadline

This document is organised in five sections. Section 1, this chapter, describes the document's research area, notation, and organisation. Section 2 describes the existing context regarding this research field starting with the basic concepts for the scheduling problems and solutions, relevant to this work. The chapter ends with the state of the art in this field whose limitations contributed to motivating this research. Section 3 deals with the proposed model, its theoretical definition and analysis. It identifies the problem, proposing a solution based on the concept of intrinsic schedule flexibility combined with the concept of resource flexibility. These concepts are defined and described in detail throughout the chapter by developing the necessary theoretical foundations while applying them to a project example. Section 4 encloses a computational study, whose primary goal is to evaluate the method's potential. For this purpose, a test environment is defined comprising a software developing environment, a set of typical scheduling algorithms, and a set of project examples. Then, its potential is assessed regarding the flexibility of the generated schedules and the impact that resource flexibility have in limiting its exploitation considering distinct flexibility parameters. Section 5 compiles the main conclusions that can be drawn from this work and identifies future work related to this research.

2 Literature review

The available literature concerning project management, its evolution, and its main concepts, is vast. Project management as a scientific discipline started in the mid of the twentieth century. CPM and PERT were presented in the 1950's, being the first ones to be developed, but did not take resources into account considering that their availability was unlimited, which greatly simplifies the model's complexity and its computational hardness. When resources started being considered, classical models assumed activities with deterministic duration and known resource requirements, and attempted to 'optimally' schedule them. This gave rise to the well-known resource-constrained project scheduling problem (RCPS) that can be stated as the problem of finding a schedule that is simultaneously precedence and resource feasible with the objective of minimising the project's durations. Projects are assumed to have deterministic activity durations which, once started, cannot be interrupted (non-pre-emptive), and precedence relations are of type finish to start with no time lag. This problem formulation plays an important role in project scheduling, mainly because this is a combinatorial NP-hard in the strong sense

problem, as proven by Blazewicz et al. (1983). However, even if the underlying model does integrate resource constraints, it still is a simplified model to most real project scheduling problems since it does not incorporate non-deterministic activity durations or more complex precedence relations, for instance. Even so, this approach was an important step in researching for better ways to establish schedules that are increasingly helpful in project management. Besides some optimal solution approaches, several sub-optimal ones have been developed using new and adapted algorithms from other research fields. Optimal solution methods have a major drawback that only small project instances are assured to be solved to optimality within an acceptable timeframe. As an example of this is the fact that, up to now, the J60 test set, described in Kolisch and Sprecher (1997) has not been fully solved to optimality. Actually, no publication was found to contradict this statement while all references to J60 test set solutions refers that there are still some instances open, e.g., in Vanhoucke and Coelho (2018). Optimal solution methods include mixed integer programming (MIP) formulations which can be found in the work of Pritsker et al. (1969), Kaplan (1988), Alvarez-Valdés and Tamarit (1993) and Mingozzi et al. (1998) or a more recent one proposed by Bianco and Caramia (2013) reformulated by Naber et al. (2014). RCPSP can also be solved to optimality using branch-and-bound based algorithms. This tree solution enumeration technique varies mainly in the search/branching strategy, and on the pruning techniques that are used to narrow the search steps needed to reach an optimal solution, which include proper upper and/or lower bounds. These algorithms can be classified according to their branching schemes being the most relevant ones the *precedence tree*: Patterson et al. (1989) and Talbot (1982); *minimal delaying alternatives*: Christofides et al. (1987) and Demeulemeester and Herroelen (1992); *extension alternatives*: Patterson et al. (1989); *minimal forbidden sets*: Igelmund and Radermacher (1983).

Another approach uses heuristic and meta-heuristic based scheduling algorithms. Heuristic methods can be divided in two major groups: *constructive heuristics*, that build or construct a schedule from scratch, whose description and computational analysis is available in the work of Kolisch (1996) and Sprecher et al. (1995); *improvement heuristics*, that take any existing feasible schedule and try to create a better one, for which an overview and a computational analysis of some of these methods can be found in Hartmann and Kolisch (2000) and Kolisch and Hartmann (2006).

Models like stochastic RCPSP (SRCPSP) assume that possible values and probabilities of activity duration are known, which enables a predefinition of the adequate scheduling policy so that the project's cost is minimised (Ballestín and Leus, 2009; Stork, 2001). If the project parameters (activities and resources) are unknown in advance, then the only possible scheduling approach is to start activities as they are defined, i.e., a purely on-line scheduling model. In between, lies the most realistic cases where probabilistic distribution on problem parameters including activity durations is not fully known, but

- 1 a set of scenarios can be established
- 2 an initial estimation is possible but unexpected changes may arise with impact in activity duration.

The first case can be handled with 'proactive scheduling', also known as 'robust scheduling', while the second can be dealt with 'reactive scheduling'. When assuming deterministic activity duration, a minimum duration schedule is considered optimal, but it

may not be ideal to deal with uncertainty. A compact schedule will be vulnerable to uncertainties because it will have insufficient flexibility to deal with unforeseen events, that is to say, it is not robust. What is then the relation between schedule robustness and schedule flexibility? Robustness of a schedule has to do with its insensitivity to changes in the project's parameters regarding

- 1 the activity start time
- 2 the objective function value while flexibility has to do with its capacity to be repaired.

2.1 *Proactive/reactive project scheduling*

Proactive scheduling is about generating the best possible robust baseline schedules, therefore they have to be optimised according to some measures that quantify their robustness. Several robustness measures exist of which the most typical ones (Herroelen and Leus, 2004) can be classified as *quality robustness*: a schedule's insensitivity to disruptions regarding its solution value (its performance) or *solution robustness*: a schedule's insensitivity to disruptions regarding its solution (the verified difference from the baseline schedule and the realised one). Quality robustness can be defined for the objective function considering, e.g., their expected value ($E[s_n]$) or the probability that a certain goal will be reached ($E[s_n \leq \delta]$) while solution robustness boils down to measuring the distance from the baseline schedule to the realised schedule, which can be expressed in several ways, e.g., as $\Delta(S^B, S^R) = \sum_{i \in V} \omega_i |s_i^B - s_i^R|$ (Van de Vonder, 2006), where s_i^B is the planned and s_i^R is the actual start of activity i and ω_i is a weight factor of such deviation.

Reactive scheduling is required whenever disruptions occur that have an impact on the current schedule. Contrary to proactive scheduling, reactive scheduling is a multi-stage process parallel to project execution. Reactive scheduling procedures can range from simple schedule repairs to full rescheduling. While the first approach has very limited capacity to cope with uncertainty, the latter tends to generate instability or nervousness. To solve this problem regarding time uncertainty, sampling, and weighted earliness-tardiness (WET) procedures were proposed by Van de Vonder et al. (2007) and Lambrechts (2007). Resource uncertainty reactive procedures were proposed by Lambrechts et al. (2007).

2.2 *FRCPS*

The RCPS model is based on the assumption that resource consumption is constant for the duration of each activity which can be fully defined by a deterministic resource and activity indexed variable r_{ik} , with $i \in V$ and $k \in K$. Cases exist, where there is no uniform resource consumption, and where the awareness of its distribution per time period can improve the scheduling process. This is the main goal of FRCSPS, the RCPS with Flexible resource profiles problem. This problem was initially addressed by Kolisch et al. (2003) and Kolisch and Meyer (2006) to model the selection and scheduling of pharmaceutical research projects. The term RCPS FWP (RCPS with flexible work profiles) is used by Ranjbar and Kianfar (2010) with the same meaning as FRCPCP in their proposed solution while Fündeling and Trautmann (2010) use the term 'work

content constraints'. Baumann and Trautmann (2013) used the same terminology to present a MIP formulation while heuristic solutions are proposed in Tritschler et al. (2014) and Rokou et al. (2014). Naber and Kolisch (2014) presented several MIP formulations for the problem and compare them by solving the models using a commercial MIP solver.

2.3 Other related work

Other studies dwell also around similar objectives as this research, but none followed a similar approach. Al-Fawzan and Haouari (2005) proposed a bi-objective model to solve RCPSP in a more robust way. It combined the traditional objective of minimising the project duration with the maximisation of a new robustness measure which they define as $\Omega = \sum_{i=1}^n s_i$, where s_i is the slack defined as the time that an activity can be delayed without delaying the start time of the next activity and not violating resource constraints. Kobylański and Kuchta (2007) disagreed with the proposed robustness measure and pointed out their deficiencies presenting two alternative ones based on the same approach that

- 1 maximises the minimum of slacks
- 2 maximises the minimum of the ratio slacks/duration.

Hazir et al. (2010) followed the same approach to solve the discrete time-cost trade-off problem (DTCTP) by using seven slack based enhanced robustness measures, a critical path based, and a project buffer size based ones. They concluded that none of the slack based measures is the best robustness measure, being all surpassed by the project buffer size based one. Nevertheless, all these studies focused on increasing the schedule robustness by using slack-based objective functions, but did not foresee the use of slack in conjunction with resource flexibility, which is the primary innovation presented in this work. The proposed model is presented in the next chapter.

3 Flexible resource management problem

The most relevant project scheduling techniques in use that deal with project uncertainty tend to focus on activities on the project's critical path (like PERT and critical chain). More elaborated RCPSP and proactive/reactive scheduling techniques did not find their way into commercial use due to their computational hardness and their complexity in modelling day-to-day projects (Demeulemeester and Herroelen, 2009). On the other hand, more elaborated resource allocation techniques, like FRCSP, that can model the increasing need to have flexibility in resource usage, are being studied (Naber and Kolisch, 2014; Rokou et al., 2014; Tritschler et al., 2014). These techniques usually establish an a priori flexible resource allocation which means that resources need not be allocated in constant amounts for the whole duration of an activity but, once assigned, are kept fixed during project execution. Following the principles of proactive/reactive scheduling, the proposal made here is to assume that projects should start with a deterministic baseline schedule, which is stable and constant, but the plan should cope, as much as possible, with uncertainties. However, in this research, the idea to explore is to

use the ‘schedule flexibility’, existing in non-critical activities, to cope with uncertainty. This flexibility can be expressed as the time that an activity’s finish time can be delayed without affecting the remaining schedule (activity slack) which, in conjunction with resource flexibility, can accommodate eventual increases in activity durations (uncertainty).

3.1 Problem definition

The problem is then how to transform a given schedule into a more robust one that will behave better when unscheduled events occur during project execution. The aim is to provide the project manager with a technique that helps him to determine the best schedule, and to assist him in making the best decisions, in response to changes in the project, which will lead to minimal deviation to the original schedule duration and in this way, to the predefined project costs. The idea is to consider a given feasible schedule S^b (a baseline schedule), obtained by any scheduling technique, and to redistribute resource capacity in order to accelerate critical activities at the expense of slowing down non-critical ones. The goal is that the resulting schedule S^w (a working schedule) will be, as long as deviations are within certain boundaries, equivalent to the baseline one in the sense that each activity start time remains the same, but the finish time can be different for some activities: later for activities that are slowed down and earlier for activities that are processed faster which, in the latter case, creates a time buffer to cope with increases in activities’ work content. The resource redistribution assumes that resources are flexible in the sense that they have a nominal work capacity per time unit, but can vary their work capacity downwards (less work capacity per time unit) or upwards (additional work capacity per time unit) from the nominal value. Traditionally, an increase in the workload of an activity can be accommodated in the defined timeframe by adding resources or by adding overtime to the already allocated ones, which typically increases the cost. The proposed approach aims to use a similar concept of overtime (working more within a time period), but which is then compensated by working less when other activities have slack, so that the project’s end date remains valid within certain boundaries.

3.2 The model

Consider the following definitions:

- a_k^{nom} : nominal resource availability of type k
- α_k^- : negative resource flexibility (maximum percentage decrease of a_k from nominal)
- α_k^+ : positive resource flexibility (maximum percentage increase of a_k from nominal).

Assuming that resource availability is flexible in the sense that the effective work that a resource can produce per unit of time can vary downward or upward from their predefined nominal value, the resource availability can be represented by the following expression:

$$a_k^{nom} (1 - \alpha_k^-) \leq a_k \leq a_k^{nom} (1 + \alpha_k^+) \quad (3.1)$$

In this expression, a_k is the effective resource availability for resource k which can vary between the defined values when allocated to a specific project's activity. Considering unitary resource nominal availability $a_k^{nom} = 1$ and setting, as an example, that the resources to be used in a specific project have their flexibility bounded by $\alpha_k^- = \alpha_k^+ = 25\%$, the effective resource unitary availability is defined by $(1 - 0.25) \leq a_k \leq (1 + 0.25)$. Considering a_k to be continuous ($a_k \in \mathbb{R}$), then $a_k \in [0.75, 1.25]$. To assess the quality of the considered flexibility boundaries one can interpret these values as one resource being available for one working day (say 8h) with the flexibility to decrease its availability from 0.75 of a day (6h) up to 1.25 of a day (10h), which can be regarded as a typical flexibility for human resources.

Consider now the project represented in the AoN network of Figure 1 which is limited to 14 activities (12 real activities and the fictitious project start and end) and to two resources, to avoid overloading the example and still cover all relevant cases of the model.

Figure 1 Project example with 14 activities and two resource types (see online version for colours)

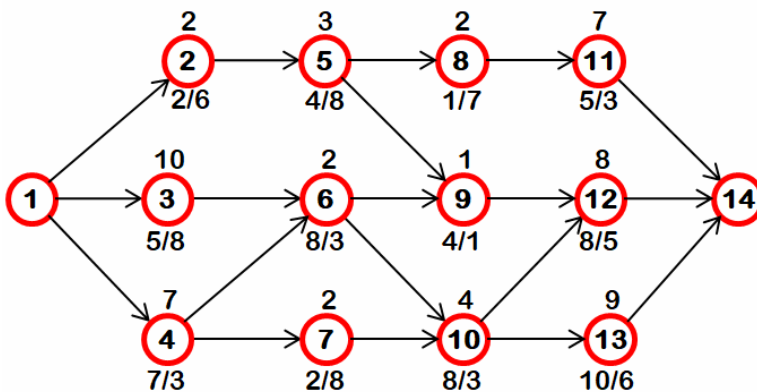
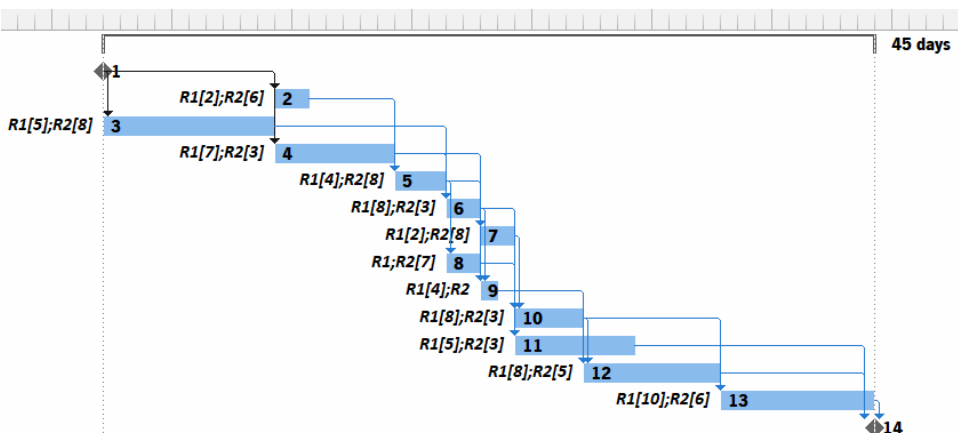


Figure 2 Gantt chart for minimal makespan baseline schedule (see online version for colours)

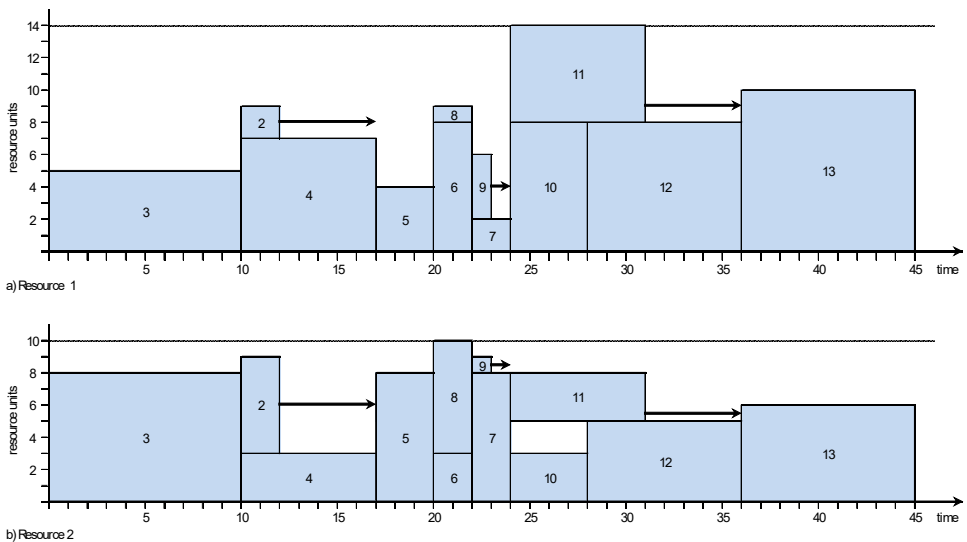


The numbers above the nodes identify activity duration (d_i) and below are resource requirements for each of the two resource types (r_{i1}/r_{i2}). Assuming that resource availability is $a_1 = 14$ and $a_2 = 10$, an optimal solution for the RCPSP problem is obtained with project duration of $T = 45$. Figure 2 shows the schedule represented in Microsoft Project 2013 (MSProject). Bold numbers indicate the activity number while italic characters are resource consumptions per activity in the following format (consumption is 1 if omitted): *resource1*[consumption of resource1]; *resource2*[consumption of resource2].

The Gantt chart shows both the basic project information (activities, durations, precedence relations and resource requirements) and the schedule information (the sequence of activities, and their start and finish times). To better identify the resource capacity transferral process between activities, the resource profile chart will be necessary. Figure 3 shows the corresponding resource profiles for

- a resource 1
- b resource 2.

Figure 3 Resource profile for optimal baseline schedule (see online version for colours)



3.2.1 Capacity

With this type of chart it is easier to identify the unused capacity per resource (available resource capacity that is not used by any activity) and the slack per activity which are two key elements that will be explored to cope with uncertainty. The unused resource capacity of resource k at time t can be defined as:

$$u_{kt} = a_k - \sum_{i \in P_t} r_{ik} \quad \text{for each } k \in K, t \in [0, T] \tag{3.2}$$

where $P_t \subseteq V$ is the active set (the set of activities in progress at time instant t).

3.2.2 Slack

The activity slack will be defined as the time each activity can finish later without affecting the remainder of the schedule, i.e., without affecting any other activity. Because the project is an RCPSP instance, both precedence and resource constraints must be considered.

$$\begin{aligned} \text{slack}_{ik} &= \max \left(\tau \in \left\{ \left[\left[0, \min_{j \in \text{Succ}_i} (s_j) - s_i - d_i \right] \cap \mathbb{N} \right] \right\} \right) \\ r_{ik} &\leq a_k - \sum_{t \in \Pi} r_{hk}, t = s_i + d_i, \dots, s_i + d_i + \tau \quad \text{for each } k \in K, i \in V \end{aligned} \quad (3.3)$$

where Succ_i is the set of successor activities of i .

Regardless of resources being dependent or independent from each other, they must comply with the activity duration. This means that they must be executed within each activity at the same rate which is the rate of the most demanding one, i.e., the pace imposed by the resource that has less slack for the given activity. Accordingly, the activity slack can be defined as:

$$\text{slack}_i = \min_{k \in K} (\text{slack}_{ik}) \quad \text{for each } i \in V \quad (3.4)$$

The activity slack is illustrated in Figure 3 as a straight arrow beginning in the activity finish time and with a length corresponding to the activity's slack. According to the definition (3.4), their length is equal on both resource charts. Table 1 presents the values for slack for the given example, along with the start time s_i , the duration d_i and the finish time f_i , for each activity. As might be expected, the slack can be quite different regarding each resource type.

If resources are flexible as described in equation (3.1) and according to these slack values, activities 2, 9 and 11 are candidates to be slowed down in their execution in order to, if necessary, liberate resources to compensate any other activities that, for any reason, might require more resources than initially estimated.

3.2.3 Duration

Assuming resource flexibility is again bounded by $\alpha_k^- = \alpha_k^+ = 25\%$, the effective resource unitary ($a_k^{\text{nom}} = 1$) availability is $0.75 \leq a_k \leq 1.25$. These values can now be applied to activity duration to determine the duration span for each activity. The results are presented in Table 2 where:

- $d_i^{\text{nom}} = d_i$: the nominal duration equals the initial (deterministic) duration d_i .
- $\bar{d}_i = d_i^{\text{nom}} / (1 - \alpha_k^-)$: the maximal activity duration is its duration when executed at its slowest rate (minimal resource unitary availability). In this case the minimal rate is 75%.
- $\underline{d}_i = d_i^{\text{nom}} / (1 - \alpha_k^+)$: the minimal activity duration is its duration when executed at its fastest rate (maximal resource unitary availability). In this case the maximal rate is 125%.

\overline{d}_i and \underline{d}_i can assume non-integer values which are not commonly used values for activity duration. Therefore, their corresponding limit integer values are considered (also presented in Table 2):

- $d_i^{\max} = \lfloor \overline{d}_i \rfloor$: maximal integer activity duration
- $d_i^{\min} = \lceil \underline{d}_i \rceil$: minimal integer activity duration.

Table 1 Example slack values

<i>i</i>	<i>s_i</i>	<i>d_i</i>	<i>f_i</i>	<i>slack_{i1}</i>	<i>slack_{i2}</i>	<i>slack_i</i>
1	0	0	0	0	0	0
2	10	2	12	5	5	5
3	0	10	10	10	0	0
4	10	7	17	3	0	0
5	17	3	20	0	0	0
6	20	2	22	0	0	0
7	22	2	24	0	0	0
8	20	2	22	2	0	0
9	22	1	23	1	5	1
10	24	4	28	0	0	0
11	24	7	31	5	14	5
12	28	8	36	0	0	0
13	36	9	45	0	0	0
14	45	0	45	0	0	0

An immediate conclusion can be drawn that if only integer activity durations are considered, smaller activities like activity 2 do not take advantage of resource flexibility.

In fact, to take advantage of this methodology, the activity duration must meet the following criteria:

$$d_i^{nom} \geq \frac{1}{\alpha_k^-} - 1 \quad \alpha_k^- \in \mathbb{R}_0^+ \tag{3.5}$$

$$d_i^{nom} \geq \frac{1}{\alpha_k^+} + 1 \quad \alpha_k^+ \in \mathbb{R}_0^+ \tag{3.6}$$

These criteria only mean that extra care should be taken with activities that have ‘small’ durations, as defined by expressions (3.5) and (3.6), because they will not participate and therefore will not take advantage of this methodology. The above expressions quantify the term ‘small’ duration as a function of each of the resource flexibility parameters α_k^- and α_k^+ .

The possible activity duration interval is not schedule dependent, depending only on the deterministic (initial) activity duration and their resources’ flexibility. On the other hand, slack is schedule dependent. The next step is to combine these concepts. Executing an activity in a smaller duration will never have an impact on the remaining activities

which means that the minimal duration of an activity is not affected by the schedule used to execute the project. On the other hand, allowing activity duration to be larger than d_i^{nom} , the duration used to establish the schedule, affects the schedule when the increase in duration is greater than the activity's slack. Therefore, an additional variable d_{iS}^{max} can be considered to express the maximal activity duration taking into account the schedule's limitation which is defined by expression (3.7). Expression (3.8) defines the corresponding minimal duration variable d_{iS}^{min} that emphasises the fact that minimal durations are not schedule dependent.

$$d_{iS}^{max} = \min(d_i^{max}, d_i^{nom} + slack_i) \tag{3.7}$$

$$d_{iS}^{min} = d_i^{min} \tag{3.8}$$

Table 3 shows the values resulting from these expressions when applied to the project example. Activities 3, 4, 12 and 13 (critical activities identified in red) can be executed in their nominal duration or less while activity 11 (non-critical activity identified in light green) can also be executed in its nominal duration or less but, additionally, can be executed with a higher duration than its nominal value. Even though it is possible that activities like this one could be executed at a faster rate, w.l.o.g., only slower rates (longer durations) will be considered for not hampering the explanation. All other activities in this example will not participate in this methodology due to their small duration.

Table 2 Possible duration span

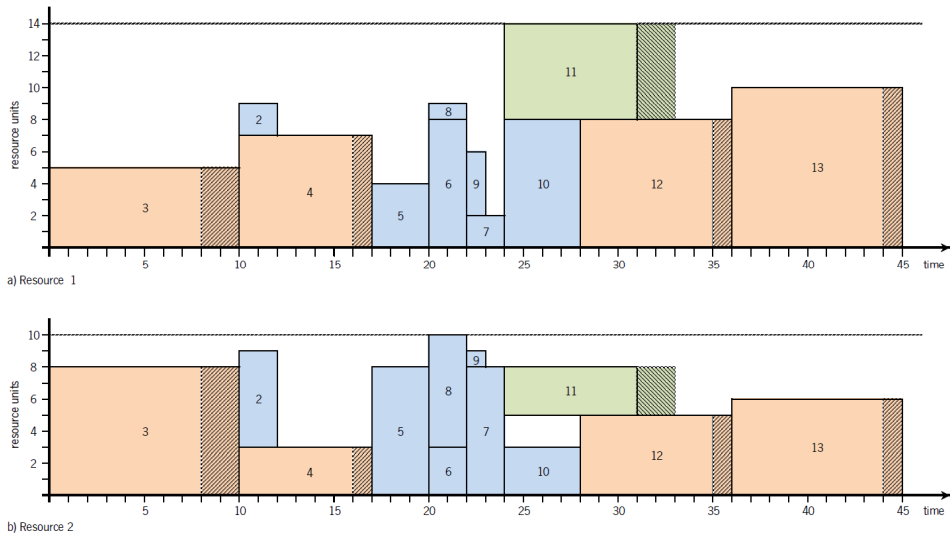
<i>i</i>	100%	75%		125%	
	d_i^{nom}	\bar{d}_i	d_i^{max}	\underline{d}_i	d_i^{min}
1	0	0.00	0	0.00	0
2	2	2.67	2	1.60	2
3	10	13.33	13	8.00	8
4	7	9.33	9	5.60	6
5	3	4.00	4	2.40	3
6	2	2.67	2	1.60	2
7	2	2.67	2	1.60	2
8	2	2.67	2	1.60	2
9	1	1.33	1	0.80	1
10	4	5.33	5	3.20	4
11	7	9.33	9	5.60	6
12	8	10.67	10	6.40	7
13	9	12.00	12	7.20	8
14	0	0.00	0	0.00	0

In Figure 4 activities are displayed with its nominal duration and their possible alternative durations' are identified by diagonal filling lines (positive slope for decreased, negative slope for increased durations).

Table 3 Allowed duration span (see online version for colours)

i	d_i^{\min}	d_{iS}^{\min}	d_i^{nom}	slack_i	d_i^{\max}	d_{iS}^{\max}
1	0	0	0	0	0	0
2	2	2	2	5	2	2
3	8	8	10	0	13	10
4	6	6	7	0	9	7
5	3	3	3	0	4	3
6	2	2	2	0	2	2
7	2	2	2	0	2	2
8	2	2	2	0	2	2
9	1	1	1	1	1	1
10	4	4	4	0	5	4
11	6	6	7	5	9	9
12	7	7	8	0	10	8
13	8	8	9	0	12	9
14	0	0	0	0	0	0

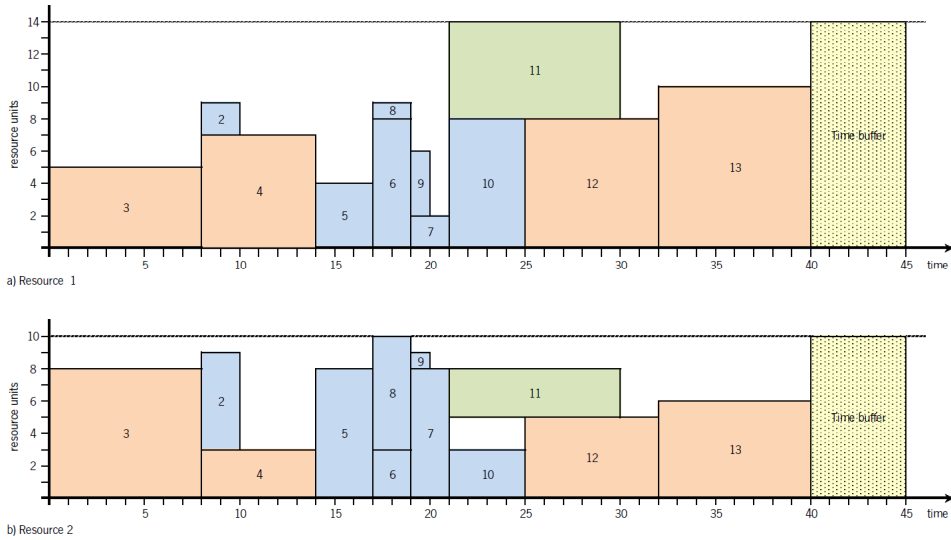
Figure 4 Resource profile for buffered schedule (see online version for colours)



This can be interpreted as having time buffers on some activities, not in the usual sense of adding some extra time to an activity increasing its planned duration, but rather in a way that, by increasing the complexity of dealing with flexible resources, does not increase the project planned makespan. Buffers are added to non-critical activities by allowing an increase in their duration, not violating their constraints (precedence, slack, and resources flexibility) and allowing a decrease in duration for critical activities. To emphasise this view, and because the ‘critical sequences’ of this schedule always involve the activities that have time buffers (if this were not the case, this view would be more difficult), the

schedule can be displayed with the time buffers joined together at the end of the project (Figure 5). The purpose of this method is not to alter the start time of activities, but rather the opposite. The intent here is only to emphasise the effect on dealing with uncertainty, enabling an easier comparison with other time buffer insertion mechanisms.

Figure 5 Resource profile for end buffered schedule (see online version for colours)



3.2.4 Score

The definition of the possible duration that each activity might have assumes that the estimation made regarding the work content (or workload), remains valid throughout the project execution. If this is true, there is no uncertainty on the project and it can resume according to the baseline schedule. Additional actions are needed otherwise. To proceed, the concept of work content is necessary. It is defined as the product of the activity duration and its resource requirements or, when more than one resource type is considered, as given by expression (3.9), where w_{ik} is the work content of activity i for resource type k .

$$w_{ik} = d_i r_{ik} \quad \text{for each } i \in V, k \in K \tag{3.9}$$

The proposed method is based on slowing down non-critical activities in order to respond to critical activities that might be in danger of being delayed due to having somehow increased their work content. The contribution of an activity has not only to do with its duration, but also with its resource requirements, that is to say, its work content. Variable $score_{ik}$ measures the weighted working time deviation per activity i and per resource type k and is formally defined by the expression:

$$score_{ik} =_{ik} (d_i - d_i^{nom}) \quad \text{for each } i \in V, k \in K \tag{3.10}$$

where d_i is the effective activity duration ($d_i^{\min} \leq d_i \leq d_i^{\max}$).

$score_{ik}$ can be seen as a measure of the deviation of the work content which becomes evident by rearranging it as: $score_{ik} = r_{ik}d_i - w_{ik}^{nom}$.

The extreme values of $score$ for the project example are presented in Table 4. As with the assumptions made to derive Figure 4, extreme values for $score$ consider the maximal contribution of each activity using:

- For non-critical activities: maximal duration, which sets the maximal positive score.
- For critical activities: minimal duration, which sets the maximal negative score.

Table 4 Example score extreme values (see online version for colours)

i	r_{i1}	r_{i2}	d_i^{nom}	d_i	$score_{i1}$	$score_{i2}$
1	0	0	0	0	0	0
2	2	6	2	2	0	0
3	5	8	10	8	-10	-16
4	7	3	7	6	-7	-3
5	4	8	3	3	0	0
6	8	3	2	2	0	0
7	2	8	2	2	0	0
8	1	7	2	2	0	0
9	4	1	1	1	0	0
10	8	3	4	4	0	0
11	6	3	7	9	12	6
12	8	5	8	7	-8	-5
13	10	6	9	8	-10	-6
14	0	0	0	0	0	0
$\sum_i score_{ik}, \text{ for each } k \in K :$					-23	-24

As expected, activities 3, 4, 12 and 13 will ‘consume’ score (negative values) if executed at higher rates, and therefore with less duration, enabling the possibility that they can use additional time and resources (i.e., work content) if uncertainty becomes relevant. This can be compensated with score ‘produced’ (positive values) by activity 11 if executed at lower rates and, therefore, taking a longer time to be completed. But in general, as in this particular case, $\sum_i score_{ik}$ will not be 0 for all k . The following cases might occur for resource k :

- $\sum_i score_{ik} > 0$: it is not necessary to slow down all possible activities to enable a faster rate for all possible critical activities.
- $\sum_i score_{ik} = 0$: slowing down all possible activities enables a faster rate for all possible critical activities (the easy case).
- $\sum_i score_{ik} < 0$: it is not possible to enable a faster rate for all possible critical activities even if all possible non-critical ones are slowed down.

Cases a) and b) fully solves the problem within the defined parameters, but in case c) there are additional constraints. If $\sum_i score_{ik}$ is restricted to be not less than 0 then its ‘consumption’ must be done until it reaches 0.

In the most general case, a project deals with more than one resource type. This can be dealt with by defining an overall score for the schedule by considering the most demanding case, that is, the score of the resource type with the lowest score. This can be defined by the following expression:

$$score = \min_k \left(\sum_i score_{ik} \right) \quad \text{for all } i \in V, k \in K \tag{3.11}$$

To develop this issue further an additional variable will be used to keep track of the cumulated score.

3.2.5 Balance

The *balance* variable will quantify the cumulated score and should be defined such that it takes into account the order in which activities can use it. To apply this concept to the project example, the activity index will be used which leads to the following definition:

$$balance_{ik} = \sum_{i=1}^i score_{ik} \quad \text{for each } i, j \in V, k \in K \tag{3.12}$$

The resulting values for the example are presented in Table 5. As can be seen, $balance_{ik}$ is always less than or equal to zero and, more critically, it ends with a negative value. This means that the resource flexibility is violated within the scope of the project.

Table 5 Example balance values

<i>i</i>	<i>Resource 1</i>		<i>Resource 2</i>	
	<i>score</i> ₁₁	<i>balance</i> ₁₁	<i>score</i> ₁₂	<i>balance</i> ₁₂
1	0	0	0	0
2	0	0	0	0
3	-10	-10	-16	-16
4	-7	-17	-3	-19
5	0	-17	0	-19
6	0	-17	0	-19
7	0	-17	0	-19
8	0	-17	0	-19
9	0	-17	0	-19
10	0	-17	0	-19
11	12	-5	6	-13
12	-8	-13	-5	-18
13	-10	-23	-6	-24
14	0	-23	0	-24

As long as the final *balance* is greater than or equal to 0, i.e., $balance_{V|k} \geq 0$ for all $k \in K$, the project's flexibility resulting from its slack, and the resource flexibility should be able to absorb its uncertainties. Therefore, it is interesting to define the following balance:

$$balance_k^+ = \sum_{i=1}^n score_{ik} \mid score_{ik} > 0, i \in V, k \in K \tag{3.13}$$

This can be set to the schedule's initial balance that can be defined as $balance_{0k}$ (which is being implicitly set to 0 up to now), i.e., $balance_{0k} = balance_k^+$.

Setting the balance initial value in this way, means that the positive score of each activity is 'moved' to the project start. Therefore, it must be withdrawn from each of the individual activities such that:

$$score_{ik}^+ = 0 \mid score_{ik} > 0 \quad \text{for each } i \in V, k \in K \tag{3.14}$$

In the example, the resulting values are $balance_{01} = 12$ and $balance_{02} = 6$. As the project is executed, it can 'consume' additional resources as long as $balance_{nk} \geq 0$.

3.2.6 More on durations

Expression (3.7) can be written as (3.15). The maximal activity duration is schedule dependent besides being dependent on the activity's nominal duration and the resource flexibility α_k^- .

$$d_{iS}^{\max} = \min \left(\min_k \left[\frac{d_i^{nom}}{(1 - \alpha_k^-)} \right], d_i^{nom} + slack_i \right) \quad \text{for each } k \in K, i \in V \tag{3.15}$$

Similarly, expression (3.8) can be written as (3.16) to emphasise that the minimal activity duration is not schedule dependent, depending only on the activity's nominal duration d_i^{nom} and the resources flexibility α_k^+ .

$$d_{iS}^{\min} = \max_k \left[\frac{d_i^{nom}}{(1 + \alpha_k^+)} \right] \quad k \in K, i \in V \tag{3.16}$$

This is to say that the capacity to absorb uncertainties, concerning the release of resources to critical activities, is both schedule and resource flexibility dependent, and is limited to the most demanding (minimal) one:

- Resource flexibility dependent term $= \min_k \left[\frac{d_i^{nom}}{(1 - \alpha_k^-)} \right]$: the *resource flexibility* (capacity to take advantage of the schedule's flexibility).
- Schedule dependent term $d_i^{nom} + slack_i$: the *schedule flexibility* (capacity to take advantage of the resources' capacity).

The schedule flexibility can also be identified in score as $score_{iV} = r_{ik} (d_i - d_i^{nom})$ and, for activities with slack, its maximal value is obtained when $d_i = d_i^{nom} + slack_i$. Therefore, if

the resource flexibility is not taken into account, it can be expressed as $score_{ir} = r_{ik}slack_i$. With this result the ‘schedule intrinsic flexibility’ (SIF) can be defined as:

$$SIF_k = \sum_{i=1}^n r_{ik}slack_i \quad \text{for each } k \in K \quad (3.17)$$

This definition is made per resource type which is useful if resources can be managed independently. If not, the following definition should be used:

$$SIF = \min_k \left(\sum_{i=1}^n r_{ik}slack_i \right) \quad \text{for all } k \in K \quad (3.18)$$

Note that the SIF_k can be regarded as the $balance_k^+$ considering the schedule’s flexibility only, i.e., when resource flexibility is not taken into account.

3.3 Scheduling process

This methodology can be regarded as a Proactive-Reactive scheduling approach. Generically, in the proactive phase (before execution starts) a schedule is determined, according to project and organisational parameters, and the schedule’s flexibility is computed. Then, in the reactive phase, the project execution is monitored, reacting with the increase or decrease in the activities’ execution rate within the computed intervals (in the proactive phase). If the schedule’s flexibility limits are not sufficient to cope with the deviations, additional actions are needed that will typically lead to rescheduling.

3.3.1 General procedure

The procedure to apply this methodology to project management might be:

- *PREPROCESSING* (before the project starts): define resource flexibility parameters (α_k^-, α_k^+) .
- *STEP 0* (before scheduling): define project data and determine possible durations (d_i^{\min}, d_i^{\max}) .
- *STEP 1* (within the project scheduling stage): establish the baseline schedule S^B using an adequate scheduling technique and determine slack ($slack_i$), the schedule specific possible maximal durations (d_{iS}^{\max}) and the schedule flexibility ($balance_k^+$). Set the initial project balance to its $balance_k^+$ and establish a working schedule S^w , having the same start times as S^B , but with finish times defined in the intervals resulting from the possible activity durations.
- *STEP 2* (during project execution): If deviations occur, check if they can be absorbed by the schedule’s flexibility and that there is enough balance left. If this is possible, update the working schedule accordingly. If not, rescheduling is necessary.

Next chapter presents a computational study to show this method’s potential using well known test set projects and scheduling techniques.

4 Computational study

Till now the model was presented using a simple project example. Here, a deeper evaluation of the model is made using a set of project examples and scheduling techniques under predefined conditions.

Once a baseline schedule is available, the necessary computations to determine the flexibility data are not too demanding. The easiest way to determine a baseline schedule is to use a professional tool. This is also an interesting case to be analysed because the proposed methodology can be integrated in such a context. However, there are several such tools, each having their own way of determining a schedule. As most tools base their scheduling method on heuristic approaches, it is useful to consider such a time sub-optimal scheduling technique. On the other hand, the extreme case of a time optimal (minimum) schedule is also of interest as it poses additional challenges to the proposed method due to their tightness. It seems (empirically) reasonable that, as the project's duration decreases schedules are less tight and tend to have less slack. Remember that the schedule for the project example used so far is an optimal time duration one.

That being said, the decision was to consider the most common heuristic scheduling method and also a time optimal one which is a computationally demanding task.

The implemented scheduling algorithms (coded in Microsoft Visual Studio 2012[®] using C++) to obtain the baseline schedule are:

- a For optimal solutions (Opt): Demeulemeester and Herroelen branch-and-bound (DH-B&B) algorithm (1992, 1997).
- b To represent a heuristic scheduling method (SSS): serial scheduling generation scheme (SSGS), selecting the best solution from using the following priority rules (Kolisch, 1996):
 - lowest job number (LJN)
 - random (RND)
 - shortest processing time (SPT)
 - longest processing time (LPT)
 - most immediate successors (MIS)
 - most total successors (MTS)
 - least number of related jobs (LNRJ)
 - greatest rank positional weight (GRPW)
 - earliest start time (ST)
 - earliest finish time (EFT)
 - latest start time (LST)
 - latest finish time (LFT)
 - minimum slack (MSLK)
 - greatest resource work content (GRWC)
 - greatest cumulative resource work content (GCRWC).
- c Microsoft Project 2013[®] (MSP) was used to include a project management software tool.

The test set used is the PSPLIB J30 (Kolisch and Sprecher, 1997) which consists of 480 project examples with 32 activities, including dummy start and end ones. Ten project examples, called instances, are generated having the same triplet $\langle NC, RF, RS \rangle$ where NC, RF, RS are variable parameters which can be generically described as:

- NC : network complexity – defined as the average (arithmetic mean) number of immediate predecessors each activity has in a project;
- RF : resource factor – defined as the average fraction of (renewable) resources used by an activity.
- RS : resource strength – defined as the level of resource scarcity.

For generating the J30 instances, the following values are assumed:

- $NC \in \{1.5, 1.8, 2.1\}$: increasing number of precedence constraints.
- $RF \in \{0.25, 0.5, 0.75, 1\}$: $RF = 0$ (not considered in the set) denotes that activities require no resources while $RF = 1$ denotes that each activity requires every resource (at least one unit of each). Considering that one of the base parameters to generate J30 instances is that $|K| = 4$, the average required resources for each activity belongs to the set $\{1, 2, 3, 4\}$. From now on, this alternative way of expressing RF will be used so that the presentation is clearer.
- $RS \in \{0.2, 0.5, 0.7, 1\}$: $RS = 0$ (not considered in the set) denotes that for at least one activity and one resource type all available resources are used ($r_{ik} = a_k, i \in V, k \in K$) while $RS = 1$ denote that there is no scarcity of resource, and therefore no explicit resource allocation is necessary to generate a schedule.

A full description of the project generator and their parameters, including these ones, can be found in Sprecher et al. (1995).

4.1 Flexibility data: schedule flexibility

As concluded before, there is schedule specific flexibility data that can be calculated for a given project schedule which is independent from the resource flexibility.

4.1.1 Slack

Considering V_e as the set of all activities of the project example e , slack related values are considered in the following two distinct ways:

- $\#NC$: the number of activities that have positive slack (non-critical) of the project example according to the selected schedule, given by the expression $\#NC = |\{i \in V_e, slack_i > 0\}|$.
- $\Sigma slack$: the sum of the slack for all activities of the project example, given by the expression $\sum slack = \sum_{i \in V_e} slack_i$.

Figure 6 presents $\#NC$ calculated for each instance regarding each scheduling method.

Figure 6 #NC for each scheduling method (see online version for colours)

instances for RS=1	10	7	11	9	9	12	10	11	8	13	9	10	10	7	11	9	9	12	10	11	8	13	9	10	10	8	11	8	9	12	11	11	7	13	7	10	12				
	9	7	12	8	9	12	8	10	10	12	12	12	10	7	12	8	9	12	8	10	10	12	12	12	10	8	12	5	8	12	8	9	10	12	13	13	10				
	8	10	11	11	10	9	8	11	12	15	14	5	8	10	11	11	10	9	8	11	12	15	14	5	8	11	13	10	11	7	6	11	13	14	14	5	7				
	7	6	8	10	11	8	10	8	11	8	10	9	10	8	10	11	8	10	8	10	11	8	10	9	10	8	8	10	10	7	10	9	9	8	10	7	11				
	6	9	8	7	7	8	10	12	10	11	8	11	11	9	8	7	7	8	10	12	10	11	8	11	11	10	10	8	8	9	8	10	9	9	10	7	8	10			
	5	11	11	9	8	9	8	10	6	11	10	11	12	11	11	9	8	9	8	10	6	11	10	11	12	11	11	7	8	8	8	10	7	10	11	10	10				
	4	9	11	9	11	11	13	13	11	12	10	8	12	9	11	9	11	11	11	13	13	11	12	10	8	12	11	11	10	11	8	13	13	10	12	8	8	12			
	3	10	10	11	9	11	9	8	10	12	14	10	10	10	10	10	11	9	8	10	12	12	14	10	10	10	10	12	12	14	10	7	10	10	9	11	9	10	13	10	
	2	13	9	6	10	9	11	10	10	10	12	10	11	13	9	6	10	10	9	11	10	10	12	10	11	13	9	6	10	8	9	11	10	9	11	6	13	10	11		
	1	0	7	8	10	12	11	10	14	11	13	0	10	11	7	8	10	12	11	10	14	11	13	0	10	11	11	13	0	11	0	11	11	12	10	14	0	13	8	8	
instances for RS=0.8	10	10	10	7	10	11	12	7	4	11	10	7	3	11	8	6	9	11	12	10	5	11	11	7	4	11	7	7	10	12	12	7	3	11	7	8	6				
	9	10	10	6	10	8	10	4	12	10	5	7	11	10	10	6	10	8	8	4	12	13	9	8	10	12	9	8	10	8	8	5	10	11	10	11	7	8			
	8	10	10	9	7	8	9	10	4	13	11	6	8	9	6	10	9	9	8	5	13	9	4	7	9	9	10	5	9	10	7	6	12	10	5	7	10	7			
	7	12	7	9	8	11	9	8	7	9	13	9	7	12	7	9	8	13	9	9	7	9	12	9	7	9	12	9	6	14	11	10	5	12	9	7	10				
	6	11	9	8	8	10	11	6	5	11	10	5	3	11	10	7	5	9	11	6	6	10	10	6	5	10	10	6	5	10	11	7	6	10	9	4	6	11	12	7	5
	5	10	9	8	5	8	9	10	4	11	5	5	5	12	8	9	6	9	10	8	5	12	5	5	7	13	9	9	5	7	11	13	5	11	11	7	4	11	11	7	4
	4	10	9	8	9	12	6	9	10	11	10	10	6	10	9	9	10	6	9	9	11	6	10	5	9	11	6	10	5	9	10	7	9	11	9	8	12	13	6	8	
	3	8	7	9	6	11	11	7	7	12	10	7	8	9	8	10	5	12	11	5	9	12	8	8	7	9	11	8	6	11	9	11	8	11	7	6	3	11	7	6	3
	2	11	9	7	8	13	11	8	7	10	12	9	8	9	8	7	10	12	11	8	8	10	12	9	8	10	12	9	8	14	10	6	6	10	12	11	9	7	10		
	1	11	7	4	3	9	12	7	6	9	9	7	9	9	9	5	9	12	11	4	7	11	9	8	9	12	10	6	6	11	10	8	8	9	9	6	7	11	7	8	
instances for RS=0.5	10	7	7	7	9	11	7	8	5	9	10	7	6	7	8	5	6	11	9	7	6	7	10	6	8	11	8	6	13	10	9	5	10	8	9	6	10	9	6		
	9	8	8	10	3	10	13	3	3	12	8	6	6	7	8	8	7	11	12	0	4	12	8	12	6	9	9	9	8	10	6	5	12	11	9	9	11	9	9		
	8	0	0	4	7	13	4	9	1	9	7	5	4	9	8	7	8	11	0	6	4	0	10	5	5	11	5	0	4	13	10	9	8	7	7	7	7	7			
	7	12	7	7	3	8	8	4	6	11	10	7	4	11	11	5	2	5	7	4	7	11	10	9	3	10	9	11	4	8	7	8	7	9	10	8	3	10	8	3	
	6	12	9	5	4	7	8	8	5	9	10	7	4	11	8	8	7	8	7	8	4	9	11	10	5	12	7	11	9	7	12	9	6	8	9	5	8	11	9	8	
	5	7	11	3	2	9	10	10	3	8	11	7	3	8	11	7	4	9	9	8	5	11	9	9	5	6	10	9	9	9	9	8	5	11	9	11	8	11	8		
	4	11	7	8	3	7	9	7	4	11	10	7	6	10	9	6	6	8	7	5	6	11	9	6	10	9	9	7	9	3	9	6	13	11	8	8	13	11	8	8	
	3	9	7	3	3	8	10	8	9	10	9	8	4	9	8	7	4	8	11	8	6	10	10	5	10	9	7	5	10	10	6	6	9	10	6	6	9	10	6	6	
	2	11	8	7	4	10	9	8	3	10	9	8	7	4	11	10	7	4	11	9	7	5	12	9	6	6	8	8	7	6	9	12	10	6	10	9	4	6	6	7	
	1	11	8	8	4	14	9	6	3	11	10	7	4	11	10	8	8	12	8	6	5	9	6	7	7	11	6	6	8	15	8	9	4	10	12	10	7	10	7		
instances for RS=0.2	10	10	7	7	4	8	7	7	2	11	12	10	6	7	11	8	4	6	9	5	5	11	13	10	5	7	7	8	8	7	12	8	6	7	8	8	3	10	8	3	
	9	9	10	2	3	8	8	7	4	11	7	6	6	9	8	8	5	11	7	7	8	9	6	7	6	7	10	8	6	11	8	7	6	9	5	9	8	9	8		
	8	9	7	6	8	11	8	7	3	12	10	7	3	8	9	5	8	10	11	7	6	13	7	7	7	9	9	9	7	11	10	6	4	9	10	6	6	9			
	7	7	10	5	1	11	7	5	3	10	7	7	3	8	8	7	4	11	8	8	2	10	10	7	6	11	8	7	6	11	7	6	6	6	3	8	8	5	10	6	5
	6	10	10	8	2	11	9	3	3	11	10	6	4	9	11	2	4	11	12	6	4	10	7	7	6	10	8	7	9	11	10	6	4	10	7	7	6	10	7	6	
	5	10	4	9	1	8	9	3	4	10	6	8	7	9	9	10	3	8	5	8	5	10	10	6	8	7	10	8	6	10	9	5	7	8	7	8	7	8	7		
	4	10	7	7	3	8	6	3	3	10	8	7	6	12	10	5	4	8	9	8	7	10	8	9	8	11	9	6	7	8	10	5	7	5	10	7	6	10	7	6	
	3	10	10	5	5	11	9	5	2	10	10	8	7	11	10	5	7	11	11	7	5	11	8	9	7	11	8	8	6	8	9	6	6	10	8	7	6	10	8	7	
	2	12	9	8	2	10	5	4	6	11	10	9	8	12	8	6	6	10	11	6	5	10	10	8	9	12	6	6	6	11	11	5	9	9	10	7	8	10	7	8	
	1	7	7	9	5	9	7	5	0	8	8	7	5	9	10	6	5	11	7	6	3	6	7	8	3	8	8	7	5	6	6	2	7	8	11	7	7	8			
RF:	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
	NC=1.5				1.8				2.1				1.5				1.8				2.1				1.5																

From the chart, it can be seen that only two instances do not have any activities without slack which happens for two of the scheduling methods. On the other end, three instances have half their real activities with slack which occurs for all scheduling methods. All remaining cases lie between $0 < \#NC < 15$. Figure 7 presents the normalised frequency distribution for $\#NC$. For more than 90% of the cases, the number of activities with slack lies in the interval [4, 13]. This means that, in most cases, the number of activities with slack is between 13.3% and 43.3% of the total number of real activities in a project, whatever the scheduling technique used.

Aggregated values for $\#NC$ and $\Sigma slack$, for all instances, regarding their minimum (*min*), their average (*avg*), and their maximum (*max*), are presented in Table 6. Relative aggregated percentage values are also presented for $\#NC/30$ (ratio to the number of real activities in each project) and $\Sigma slack/T$ (ratio to the project duration for the respective scheduling method).

Table 6 Slack aggregated values

	<i>min</i>	<i>avg</i>	<i>max</i>	<i>min</i>	<i>avg</i>	<i>max</i>
<i>Absolute values</i>		$\#NC$			$\Sigma slack$	
Opt	0	8.25	15	0	49.19	187
SSS	0	8.57	15	0	52.91	192
MSP	2	8.66	15	3	59.50	165
<i>Relative values (%)</i>		$\#NC/30$			$\Sigma slack/T$	
Opt	0.00%	27.49%	50.00%	0.00%	87.49%	268.63%
SSS	0.00%	28.56%	50.00%	0.00%	91.61%	268.63%
MSP	6.67%	28.86%	50.00%	4.76%	98.67%	268.63%

While some instances have no activities with slack, on average more than 25% of activities do have slack, reaching an upper limit of 50%, whatever the scheduling technique.

4.1.2 Schedule intrinsic flexibility

The SIF, as defined in (3.17) and (3.18), denotes a better measure for the schedule’s flexibility than slack as was explained. Aggregated values that result from the first expression $(SIF_k = \sum_{i=1}^n r_{ik} slack_i)$ are presented in Table 7 for minimum, average, and maximum values, for each resource type and scheduling technique. Also presented in the table are relative measures (SIF%) concerning the ratio of SIF and the sum of the required resources, calculated using the expression $SIF_k / \sum_i (d_i r_{ik})$. The purpose of this analysis is to evaluate the SIF’s magnitude when compared to the overall resource consumption expressed in its global work content $w_k / \sum_i (d_i r_{ik})$. These values present a significant variation that can, in some cases, exceed 100%. It is significant that the global average values are between 26.6% and 33.3% with extreme cases near to 300%.

Table 7 SIF and SIF% aggregated vales

		<i>r</i> ₁			<i>r</i> ₂		
		<i>min</i>	<i>avg</i>	<i>max</i>	<i>min</i>	<i>avg</i>	<i>max</i>
SIF	Opt	0	134.6	667	0	132.0	689
	SSS	0	147.7	755	0	140.5	735
	MSP	0	165.5	673	0	157.1	846
SIF%	Opt	0.0%	27.5%	225.7%	0.0%	27.1%	191.7%
	SSS	0.0%	29.4%	202.7%	0.0%	28.1%	191.7%
	MSP	0.0%	33.3%	298.6%	0.0%	30.5%	240.4%
		<i>r</i> ₃			<i>r</i> ₄		
		<i>min</i>	<i>avg</i>	<i>max</i>	<i>min</i>	<i>avg</i>	<i>max</i>
SIF	Opt	0	130.6	655	0	130.8	670
	SSS	0	142.8	655	0	139.9	670
	MSP	0	160.3	818	0	157.0	670
SIF%	Opt	0.0%	26.9%	176.4%	0.0%	26.6%	209.3%
	SSS	0.0%	28.7%	176.4%	0.0%	27.4%	209.3%
	MSP	0.0%	32.0%	238.5%	0.0%	30.4%	153.7%

4.2 Flexibility data: resource flexibility

To be able to explore the flexibility resulting from the existence of slack, while not changing activity start times, resources have to be flexible. The impact of setting some resource flexibility parameters on constraining schedule flexibility is then analysed.

4.2.1 Optimal α^- , α^+

Resource flexibility parameters α^- and α^+ have optimal minimal values regarding activity duration so that all activities can benefit from resource flexibility. From expression (3.8),

α^- has to comply with the condition $\alpha^- \geq \frac{1}{d_i^{nom} + 1}$ to enable all activities to contribute

with their slack. Similarly, to enable all activities to benefit from slack, from expression

(3.9), α^+ has to comply with $\alpha^+ \geq \frac{1}{d_i^{nom} - 1}$.

One of the parameters used to generate the test set (Kolisch and Sprecher, 1997) is the possible duration of any non-dummy activity, which deserves special mention because of its great impact in this analysis. It is defined as $d_j \in [1, 10] \cap \mathbb{N}$, which is an integer and must lie in the interval [1, 10]. Considering that durations are expressed in days, the following conclusions are drawn:

- If $\alpha \geq 50\%$ all activities can contribute with their slack.
- Activities with $d_j = 1$ never benefit from slack.

On the other hand, considering for example that activity durations are expressed in five working days per week, a less demanding scenario regarding resource flexibility is required. Activity duration will lie in the interval $d_j \in [5, 50]$ and the optimal resource flexibility parameters can be lowered to $\alpha^- \geq 16.7\%$ and $\alpha^+ \geq 25\%$. A more detailed analysis follows considering the first more demanding scenario.

4.2.2 Impact of varying α^- , α^+

Being a day the time unit, Table 8 presents possible flexible resources parameter values that lead to integer working hours per day. While α^- cannot assume values greater than 100%, α^+ can assume values up to 200%. However, regarding this analysis, α values are limited to more (human resources) reasonable ones which are highlighted in Table 8.

Table 8 Resource flexibility parameters (see online version for colours)

Values for α^- and α^+	Unitary resource availability	Working hours per day
0.0%	$1 \leq a_k \leq 1$	$r = 8$
12.5%	$0.875 \leq a_k \leq 1.125$	$7 \leq r \leq 9$
25.0%	$0.75 \leq a_k \leq 1.25$	$6 \leq r \leq 10$
37.5%	$0.625 \leq a_k \leq 1.375$	$5 \leq r \leq 11$
50.0%	$0.5 \leq a_k \leq 1.5$	$4 \leq r \leq 12$
62.5%	$0.375 \leq a_k \leq 1.625$	$3 \leq r \leq 13$
75.0%	$0.25 \leq a_k \leq 1.75$	$2 \leq r \leq 14$
87.5%	$0.125 \leq a_k \leq 1.875$	$1 \leq r \leq 15$
100.0%	$0 \leq a_k \leq 2$	$0 \leq r \leq 16$

First, the impact of resource parameter α^- on the number of non-critical activities that can be slowed down is studied.

To simplify the readability of the charts presented in the next two figures, individual values are not presented for each instance, but rather an average value for the ten instances having the same triplet $\langle NC, RF, RS \rangle$.

Figure 8 presents the values based on the number of activities that have slack and whose duration satisfies expression (3.5), that is $u = \left| i \in V, slack_i > 0, d_i^{nom} \geq \left\lceil \frac{1}{\alpha^-} - 1 \right\rceil \right|$.

Table 9 Activity minimal durations regarding α^-

α^-	Activity duration
12.5%	$d_i^{nom} \geq 7$
25.0%	$d_i^{nom} \geq 3$
37.5%	$d_i^{nom} \geq 2$
50.0%	$d_i^{nom} \geq 1$

From the charts, it can be seen that the effect of α^- is quite straightforward except for $\alpha^- = 12.5\%$ which has greater impact. This has to do with the activity durations (1 to 10)

and their random distribution on the test set which combined with the values of minimal durations according to α , have that effect. The values of such minimal durations are presented in Table 9.

The same conclusion can be drawn when analysing averages as can be seen in the following table:

Figure 8 Number of activities with slack according to α (see online version for colours)

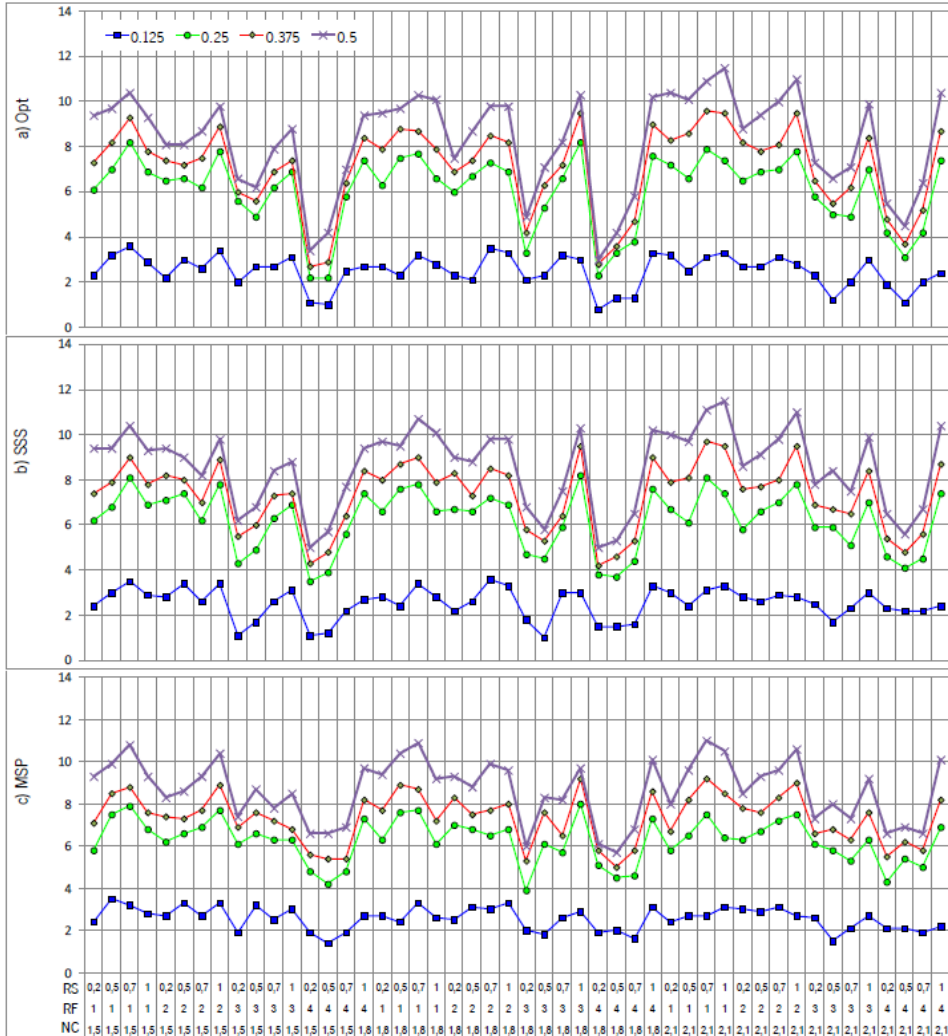
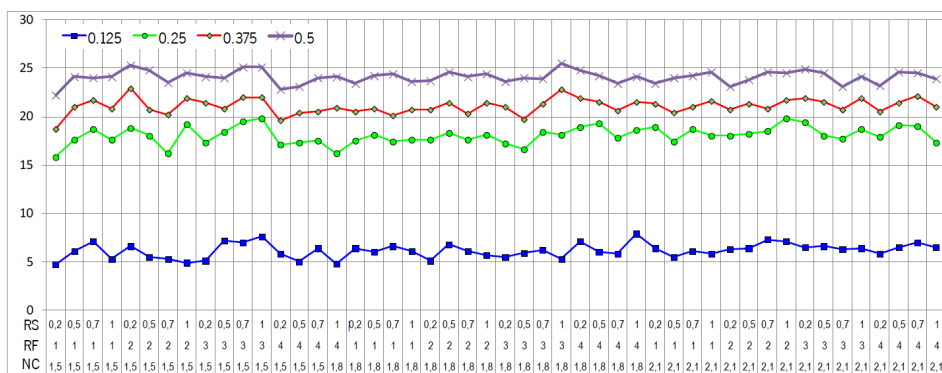


Table 10 Average number of activities with slack according to α

$\alpha =$	12.5%	25.0%	37.5%	50.0%
Opt	2.50	6.06	7.09	8.25
SSS	2.52	6.21	7.32	8.57
MSP	2.56	6.27	7.34	8.66

Contrary to the impact of α , which concerned only non-critical activities, the analysis of α^+ involves all non-dummy activities since, while its major impact regards critical activities, it can also be applied to non-critical ones if they become critical. The goal is then to determine which activities can be executed at a faster rate, if necessary, when resources have a certain flexibility to work faster which is expressed in the α^+ parameter. Therefore, the schedule being used is neutral in determining the impact of α^+ on the number of activities that can benefit from resource flexibility. Accordingly, Figure 9 has only one chart presenting the number of activities that can benefit from resource flexibility, expressed as their average values for each set of instances with same $\langle NC, RF, RS \rangle$, plotted for each selected value of α^+ . These values are computed, for each instance, according to expression (3.6).

Figure 9 Number of activities that can benefit from resource flexibility according to α^+ (see online version for colours)



The upper limit for any of the plots is 30, but it is only reached when $\alpha^+ = \infty$. According to the given expression, and for similar reasons as the ones presented for α , values are quite straightforward except for $\alpha^+ = 12.5\%$. Again, it has to do with minimal durations with respect to α^+ which results in a ‘big jump’ from $\alpha^+ = 12.5\%$ to $\alpha^+ = 25\%$ as can be seen in Table 11.

Table 11 Activity minimal durations regarding α^+

α^+	Activity duration
12.5%	$d_i^{nom} \geq 9$
25.0%	$d_i^{nom} \geq 5$
37.5%	$d_i^{nom} \geq 4$
50.0%	$d_i^{nom} \geq 3$

A similar analysis can be performed for $\Sigma slack$ and SIF but, given the statistical nature of the generated instances regarding durations and their required resources, results are expected to lead to the same conclusion which is that the resource flexibility parameters and minimal activity duration are closely related. Therefore, it is crucial to set them correctly in order to cope with uncertainties, whenever this methodology is used.

4.3 Study summary

Using a benchmark test set and typical scheduling algorithms, the following conclusions can be drawn:

- Regarding the flexibility of the generated schedules:
 - a The number of activities with slack ($\#NC$) are, on average, 27% to 29% of the total number of non-dummy activities in the projects with a maximum of 50%.
 - b The schedule's total slack ($\Sigma slack$) is, on average, 87% to 99% of the schedule's duration, ranging up to more than 200%.
 - c The *SIF* is, on average, 27% to 33% of the total project's resource consumption, ranging up to about 300%.
- Regarding the impact of resource flexibility in limiting the use of the schedule flexibility:
 - a The optimal value of α should be at least 50% and α^* should be set as high as possible when activity durations of the test set are expressed in days. If durations are expressed in weeks, the α parameter can be set as low as 16.7% and α^* can be limited to 25%.
 - b Varying α and α^* has a significant impact in limiting the number of activities that may be included in the procedure, especially for the lowest considered value (12.5%).

The purpose of this analysis is to demonstrate that, even for small projects as are the J30 ones, the proposed model is applicable as for typical resource flexibility factors the slack existing in some schedules can be used to cope with deviations that may occur during project execution. Nevertheless, additional research should be made to demonstrate that project schedules resist better to deviations when using this model.

Next chapter presents the main conclusions that can be drawn from this analysis.

5 Conclusions

This research presents a new approach to deal with uncertainties that might arise when a project is being executed, and its applicability is evaluated when applied to a benchmark test set. None of the analysed literature uses such approach nor proposes similar procedures. On the other hand, with a few exceptions, existing scheduling techniques generate schedules for the used test set that can use this procedure to cope with uncertainties because they have activities with slack. Therefore, if resources are flexible, projects that might fail the objective of being finished in time due to deviations in work contents, have additional chances to still be finished in time.

5.1 Concluding remarks

If the fundamental requirement that resources are flexible is fulfilled, then this methodology can be applied to any feasible baseline schedule, assuring that its flexibility (the slack existing in their activities) can be used to cope with eventual deviations in the project's estimations such that the schedule is as stable as possible regarding the start

time of their activities and therefore of the project's duration. Several benefits of such schedule stability were already mentioned, but can be complemented with additional considerations. If the existing slack is not managed (e.g., as with this methodology), resources might be:

- Idle, meaning that no work will be done.
- Set to work in other projects or tasks, in which case setup events may be necessary resulting in inefficiency and disturbance on the project.
- When resources are persons, they may be tempted to add non agreed upon features, changing the project's scope with unplanned functionalities with the consequences of informal approaches.
- Or, if they are not so helpful, 'consume' slack by delaying the work using the 'student syndrome' or other forms of the 'Parkinson's law'.

Activities without slack, which are schedule critical, face additional challenges when deviations to plan occur. If deviations tend to decrease their duration they gain slack becoming non-critical. Some might be tempted to anticipate the start of dependent activities, following a 'Roadrunner mentality' as is the case of 'Prochain' (Herroelen and Leus, 2001). This will collide with the premise of protecting the start time of activities which follows the 'railways scheduling' approach. The stability of the plan is preferred over reducing the project's duration either at the planning phase or at execution. But, more often than not, deviations tend to increase activity duration. With this methodology, the flexibility existing in the schedule can be used to compensate such deviations. This means that, as far as schedule flexibility exists, there is a mechanism to prevent such activities to propagate their deviations. Such deviations propagate by activities starting later than planned, jeopardising the schedule thereafter. Alternatives like 'activity crashing' and 'activity overlapping' or 'fast tracking' (Gerk and Qassim, 2008), being an immediate one to allow to overload the resources or 'overtime'. This last approach, to be comparable, must be applied within each time period, i.e., allow resources to work more within each time period so that an activity with increased work content still does not increase its duration. This approach, besides the cost factor, which is not dealt within the scope of this research, places another issue: why does one consider 'overtime' and not the opposite. In fact, taking into account both effects can lead to a greater liability to planners, project managers, and estimators in general because:

- Planning by excess will not result in work capacity being thrown away.
- Planning by default will result in unplanned costs.

The proposed methodology tends to promote responsibility over estimations and provides mechanisms to deal with the consequences of 'bad' planning by having flexibility in the resources used.

5.2 Future/open work

This methodology can be further validated by simulating its operation to demonstrate that the working schedule (S^w) resists better to deviations than the baseline schedule (S^b). This validation can be extended to include, in conjunction, other scheduling techniques that

cope with project uncertainties. The other important subject that was not explicitly considered is cost analysis. Implicitly project costs are regarded as proportional to time, namely activity duration and resources. No consideration is made for additional costs regarding schedule deviations or distinct costs for each resource type. Flexibility costs are also assumed to be merely proportional to durations, but other models are possible like defining them as $cost_i = f(exec.rate_i)$, $exec.rate_i = d_i^{nom} / d_i$ (a function of execution rate) or $cost_k = f(\alpha_k^-, \alpha_k^+)$ (a function of resource flexibility).

5.3 Epilogue

While some limitations are identified and put into perspective, results show the potential of the presented approach, which can be named as *RCPSP-FRM* standing for ‘RCPSP with flexible resource management’. Additional analysis and further testing and benchmarking will certainly provide a broader perception of its capabilities and applicability. Nevertheless, the achieved results are promising which is encouraging for further research.

References

- Al-Fawzan, M.A. and Haouari, M. (2005) ‘A bi-objective model for robust resource-constrained project scheduling’, *International Journal of Production Economics*, Vol. 96, No. 2, pp.175–187, <http://doi.org/10.1016/j.ijpe.2004.04.002>
- Alvarez-Valdés, R. and Tamarit, J.M. (1993) ‘The project scheduling polyhedron: dimension, facets and lifting theorems’, *European Journal of Operational Research*, Vol. 67, No. 2, pp.204–220, [http://doi.org/10.1016/0377-2217\(93\)90062-R](http://doi.org/10.1016/0377-2217(93)90062-R).
- Ballestín, F. and Leus, R. (2009) ‘Resource-constrained project scheduling for timely project completion with stochastic activity durations’, *Production and Operations Management*, Vol. 18, No. 4, pp.459–474, <http://doi.org/10.1111/j.1937-5956.2009.01023.x>.
- Baumann, P. and Trautmann, N. (2013) ‘Optimal scheduling of work-content-constrained projects’, in *2013 IEEE International Conference on Industrial Engineering and Engineering Management*, pp.395–399, IEEE, <http://doi.org/10.1109/IEEM.2013.6962441>.
- Bianco, L. and Caramia, M. (2013) ‘A new formulation for the project scheduling problem under limited resources’, *Flexible Services and Manufacturing Journal*, pp.6–24, <http://doi.org/10.1007/s10696-011-9127-y>.
- Blazewicz, J., Lenstra, J.K. and Kan, A.H.G.R. (1983) ‘Scheduling subject to resource constraints: classification and complexity’, *Discrete Applied Mathematics*, Vol. 5, No. 1, pp.11–24, [http://doi.org/10.1016/0166-218X\(83\)90012-4](http://doi.org/10.1016/0166-218X(83)90012-4).
- Christofides, N., Alvarez-Valdés, R. and Tamarit, J.M. (1987) ‘Project scheduling with resource constraints: a branch and bound approach’, *European Journal of Operational Research*, Vol. 29, No. 3, pp.262–273, [http://doi.org/10.1016/0377-2217\(87\)90240-2](http://doi.org/10.1016/0377-2217(87)90240-2).
- Demeulemeester, E. and Herroelen, W. (1992) ‘A branch-and-bound procedure for the multiple resource-constrained project scheduling problem’, *Management Science*, Vol. 38, No. 12, pp.1803–1818, <http://doi.org/doi:10.1287/mnsc.38.12.1803>.
- Demeulemeester, E. and Herroelen, W. (1997) ‘New benchmark results for the resource-constrained project scheduling problem’, *Management Science*, Vol. 43, No. 11, pp.1485–1492.
- Demeulemeester, E. and Herroelen, W. (2009) ‘Robust project scheduling’, *Foundations and Trends in Technology, Information and Operations Management*, Vol. 3, Publishers Inc., <http://doi.org/10.1561/02000000021>

- Elmaghraby, S. (2005) 'On the fallacy of averages in project risk management', *European Journal of Operational Research*, Vol. 165, No. 2, pp.307–313, <http://doi.org/10.1016/j.ejor.2004.04.003>.
- Fündeling, C-U. and Trautmann, N. (2010) 'A priority-rule method for project scheduling with work-content constraints', *European Journal of Operational Research*, Vol. 203, No. 3, pp.568–574, <http://doi.org/10.1016/j.ejor.2009.09.019>.
- Gerk, J.E.V. and Qassim, R.Y. (2008) 'Project acceleration via activity crashing, overlapping, and substitution', *IEEE Transactions on Engineering Management*, Vol. 55, No. 4, pp.590–601, <http://doi.org/10.1109/TEM.2008.927786>.
- Hartmann, S. and Kolisch, R. (2000) 'Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem', *European Journal of Operational Research*, Vol. 127, No. 2, pp.394–407, [http://doi.org/10.1016/S0377-2217\(99\)00485-3](http://doi.org/10.1016/S0377-2217(99)00485-3).
- Hazır, Ö., Haouari, M. and Erel, E. (2010) 'Robust scheduling and robustness measures for the discrete time/cost trade-off problem', *European Journal of Operational Research*, Vol. 207, No. 2, pp.633–643, <http://doi.org/10.1016/j.ejor.2010.05.046>.
- Herroelen, W. and Leus, R. (2001) 'On the merits and pitfalls of critical chain scheduling', *Journal of Operations Management*, Vol. 19, No. 5, pp.559–577, [http://doi.org/10.1016/S0272-6963\(01\)00054-7](http://doi.org/10.1016/S0272-6963(01)00054-7).
- Herroelen, W. and Leus, R. (2004) 'Robust and reactive project scheduling – a review and classification of procedures', *International Journal of Production Research*, Vol. 42, No. 8, pp.1599–1620, <http://doi.org/10.1080/00207540310001638055>.
- Igelmund, G. and Radermacher, F.J. (1983) 'Preselective strategies for the optimization of stochastic project networks under resource constraints', *Networks*, Vol. 13, No. 1, pp.1–28, <http://doi.org/10.1002/net.3230130102>.
- Jia, J., Fan, X. and Lu, Y. (2007) 'System dynamics modeling for overtime management strategy of software project', *International System Dynamics Conference*, Boston, Massachusetts, USA [online] <http://www.systemdynamics.org/conferences/2007/proceed/papers/JIA341.pdf> (accessed 2 December 2015).
- Kaplan, L.A. (1988) *Resource-constrained Project Scheduling with Preemption of Jobs*, University of Michigan [online] https://books.google.pt/books/about/RESOURCE_CONSTRAINED_PROJECT_SCHEDULING.html?id=PPDZNAAACAAJ&pgis=1 (accessed 29 December 2015).
- Kerzner, H. (2013) *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*, John Wiley & Sons [online] <https://books.google.com/books?id=CSHhsh2TpLwC&pgis=1> (accessed 23 January 2016).
- Kobyłański, P. and Kuchta, D. (2007) 'A note on the paper by M. A. Al-Fawzan and M. Haouari about a bi-objective problem for robust resource-constrained project scheduling', *International Journal of Production Economics*, Vol. 107, No. 2, pp.496–501, <http://doi.org/10.1016/j.ijpe.2006.07.012>.
- Kolisch, R. (1996) 'Serial and parallel resource-constrained project scheduling methods revisited: theory and computation', *European Journal of Operational Research*, Vol. 90, No. 2, pp.320–333, [http://doi.org/http://dx.doi.org/10.1016/0377-2217\(95\)00357-6](http://doi.org/http://dx.doi.org/10.1016/0377-2217(95)00357-6).
- Kolisch, R. and Hartmann, S. (2006) 'Experimental investigation of heuristics for resource-constrained project scheduling: an update', *European Journal of Operational Research*, Vol. 174, No. 1, pp.23–37, <http://doi.org/10.1016/j.ejor.2005.01.065>.
- Kolisch, R. and Meyer, K. (2006) 'Selection and scheduling of pharmaceutical research projects', in Józefowska J. and Węglarz, J. (Eds.): *Perspectives in Modern Project Scheduling*, Chapter 13, Springer Science+Business Media, LLC [online] http://link.springer.com/chapter/10.1007/978-0-387-33768-5_13#page-1 (accessed 26 November 2015).
- Kolisch, R. and Sprecher, A. (1997) 'PSPLIB – a project scheduling problem library', *European Journal of Operational Research*, Vol. 96, No. 1, pp.205–216, [http://doi.org/10.1016/S0377-2217\(96\)00170-1](http://doi.org/10.1016/S0377-2217(96)00170-1).

- Kolisch, R., Meyer, K., Mohr, R., Schwindt, C. and Urmann, M. (2003) 'Ablaufplanung für die Leitstrukturoptimierung in der Pharmaforschung', *Zeitschrift Für Betriebswirtschaft*, Vol. 73, No. 8, pp.825–848.
- Lambrechts, O. (2007) *Robust Project Scheduling Subject to Resource Breakdowns*, KU Leuven, Belgium, 27 November [online] <https://lirias.kuleuven.be/handle/1979/1009> (accessed 12 January 2016).
- Lambrechts, O., Demeulemeester, E. and Herroelen, W. (2007) 'Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities', *Journal of Scheduling*, Vol. 11, No. 2, pp.121–136, <http://doi.org/10.1007/s10951-007-0021-0>.
- Meredith, J.R. and Mantel, S.J. (2011) *Project Management: A Managerial Approach*, John Wiley & Sons [online] https://books.google.ca/books/about/Project_Management.html?id=xGRtQetWjNsC&pgis=1 (accessed 23 January 2016).
- Mingozi, A., Maniezzo, V., Ricciardelli, S. and Bianco, L. (1998) *An Exact Algorithm for the Resource-Constrained Project Scheduling Problem Based on a New Mathematical Formulation* [online] <http://pubsonline.informs.org/action/showCitFormats?doi=10.1287%2Fmns.44.5.714> (accessed 29 December 2015).
- Naber, A. and Kolisch, R. (2014) 'MIP models for resource-constrained project scheduling with flexible resource profiles', *European Journal of Operational Research*, Vol. 239, No. 2, pp.335–348, <http://doi.org/http://dx.doi.org/10.1016/j.ejor.2014.05.036>.
- Naber, A., Kolisch, R., Bianco, L. and Caramia, M. (2014) 'The resource-constrained project scheduling model of Bianco and Caramia: clarifications and an alternative model formulation', *Flexible Services and Manufacturing Journal*, Vol. 26, No. 3, pp.454–459, <http://doi.org/10.1007/s10696-014-9197-8>.
- Olsen, B. and Swenson, D. (2011) 'Overtime effects on project team effectiveness', *Midwest Instruction and Computing Symposium*, April, Duluth, MN – USA [online] <http://www2.css.edu/mics/Submissions/submissions/Overtime-Effects-on-Project-Team-Effectiveness.pdf> (accessed 2 December 2015).
- Patterson, J.H., Słowiński, R., Talbot, F.B. and Węglarz, J. (1989) 'An algorithm for a general class of precedence and resource constrained scheduling problems', *Advances in Project Scheduling*, Elsevier, <http://doi.org/10.1016/B978-0-444-87358-3.50005-5>.
- PMI (2013) *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)*, 5th ed., *Project Management Journal*, Vol. 44, <http://doi.org/10.1002/pmj.21345>.
- Pritsker, A.A.B., Waiters, L.J. and Wolfe, P.M. (1969) 'Multiproject scheduling with limited resources: a zero-one programming approach', *Management Science*, Vol. 16, No. 1, pp.93–108, <http://doi.org/10.1287/mnsc.16.1.93>.
- Ranjbar, M. and Kianfar, F. (2010) 'Resource-constrained project scheduling problem with flexible work profiles: a genetic algorithm approach', *Transaction E: Industrial Engineering*, Vol. 17, No. 1, pp.25–35 [online] http://www.sid.ir/en/VEWSSID/J_pdf/95520101E05.pdf (accessed 26 November 2015).
- Rokou, E., Dermizakis, M. and Kirytopoulos, K. (2014) 'Multi-project flexible resource profiles project scheduling with ant colony optimization', in *2014 IEEE International Conference on Industrial Engineering and Engineering Management*, IEEE, pp.642–646, <http://doi.org/10.1109/IEEM.2014.7058717>.
- Sprecher, A., Kolisch, R. and Drexel, A. (1995) 'Semi-active, active, and non-delay schedules for the resource-constrained project scheduling problem', *European Journal of Operational Research*, Vol. 80, No. 1, pp.94–102, [http://doi.org/10.1016/0377-2217\(93\)E0294-8](http://doi.org/10.1016/0377-2217(93)E0294-8).
- Stork, F. (2001) *Stochastic Resource-Constrained Scheduling*, PhD thesis, Technische Universität Berlin [online] <papers2://publication/uuid/0E0D6199-4A7F-4527-832B-D160769539A6> (accessed 30 November 2015).
- Talbot, F.B. (1982) 'Resource-constrained project scheduling with time-resource tradeoffs: the nonpreemptive case', *Management Science*, Vol. 28, No. 10, pp.1197–1210, <http://doi.org/10.1287/mnsc.28.10.1197>.

- Tritschler, M., Naber, A. and Kolisch, R. (2014) 'A genetic algorithm for the resource-constrained project scheduling problem with flexible resource profiles', *Proceedings of the 14th International Conference on Project Management and Scheduling*, pp.230–233.
- Van de Vonder, S. (2006) *Proactive-Reactive Procedures for Robust Project Scheduling*, PhD thesis, Faculty of Economics and Applied Economics, Department of Decision Sciences and Information Management, K.U. Leuven, Belgium.
- Van de Vonder, S., Ballestín, F., Demeulemeester, E. and Herroelen, W. (2007) 'Heuristic procedures for reactive project scheduling', *Computers and Industrial Engineering*, Vol. 52, No. 1, pp.11–28, <http://doi.org/10.1016/j.cie.2006.10.002>.
- Vanhoucke, M. and Coelho, J. (2018) 'A tool to test and validate algorithms for the resource-constrained project scheduling problem', *Computers and Industrial Engineering*, Vol. 118, pp.251–265, <http://doi.org/https://doi.org/10.1016/j.cie.2018.02.001>.