Universidade do Minho
Escola de Engenharia

João Manuel Peixoto Faria

Project Management Under Uncertainty: A mixed approach using flexible resource management to exploit schedule flexibility

PhD Thesis

Doctoral Program in Industrial and Systems Engineering

Work done under the guidance of

Professor Doctor Maria Madalena Teixeira Araújo

Professor Doctor Erik Demeulemeester

March, 2016

STATEMENT

Name: João Manuel Peixoto Faria

Email: jfaria01@gmail.com

Thesis title:

Project Management Under Uncertainty: A mixed approach using flexible resource management to exploit schedule flexibility

Supervisors:

Professor Doctor Madalena Araújo, University of Minho - Portugal

Professor Doctor Erik Demeulemeester, KU Leuven - Belgium

Conclusion year: 2016

Doctorate name: Doctoral Program in Industrial and Systems Engineering

Universidade do Minho, 4 / 3 / 2016

João Manuel Peixoto Faria

STATEMENT OF INTEGRITY

I hereby declare having conducted my thesis with integrity. I confirm that I have not used plagiarism or any form of falsification of results in the process of the thesis elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

University of Minho, 4th of March, 2016

Full name: João Manuel Peixoto Faria

Signature: João Manuel Peixoto Faria

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisors, Professor Madalena Araújo and Professor Erik Demeulemeester, for their encouragements, guidance and their availability during my research and writing of my thesis.

I must also thank Dr. Anabela Tereso for her support and great help during all the steps of the preparation of my thesis.

I would like to express my sincere thanks to Professor Salah Elmaghraby and Dr. José Coelho for their advices and scientific contribution.

Last but not least, my thanks go to my wife for all the support and patience and to my daughters for all the time I should been there and I could not.

In memory of my father

ABSTRACT

Project management involves onetime endeavours which demand for getting it right the first time. Also, project scheduling, being one of the most modelled project management process stages, still faces a wide gap separating theory from practice. Demanding computational models, and their consequent call for simplification, divert the implementation of such models in project management tools from the actual day to day project management process. Special focus is being made to the robustness of the generated project schedules facing the omnipresence of uncertainty. An "easy" way out is to add, more or less cleverly calculated, time buffers that always result in project duration increase and correspondingly, in increased costs. A better approach to deal with uncertainty can be to explore slack that might exist in a given project schedule, even more when a non-optimal schedule is used, which is what usually happens in practice. The combination of such approach with recent advances in modelling resource allocation and scheduling techniques, to cope with the increasing flexibility in resources, is a promising line of research, in order to generate more adequate project management tools. "Flexible resource profiles Resource Constraint Project Scheduling Problem" (FRCPSP) formulations are a step in this direction but a distinct approach can be followed, considering that, apart from being flexibly allocated or not, resources are themselves flexible by nature. More specifically, when renewable resources are considered, their capacity to perform work in a time unit (e.g. a day), which is considered to be one unit for each resource in traditional models, can be increased so that they can perform additional work in a time unit, or it can be decreased with the consequent reduction on the performed work. In reality, this flexibility is frequently considered by project managers. The objective here is to use this possibility so that, when combined with the slack that some activities have in a specific schedule, deviations that might occur during a project's execution can be absorbed. In the most critical case, in which a critical activity (activity without slack) will have its duration increased, the strategy is basically to slow down non-critical activities (activities with slack), by putting their resources in a decreased work mode, so that the critical activity, which is about to have an increase in its duration delaying the whole project, can still be executed within time by using resources in an increased working mode.

This thesis analyses this combination, its consequences and limitations and proposes models to generate and enhance starting (baseline) schedules that can take a greater advantage of this approach.

KEYWORDS: Project management, Scheduling, Resource allocation, RCPSP, Uncertainty.

RESUMO

A Gestão de Projetos consiste num esforço único que exige que se acerte logo à primeira tentativa. Além disso, o escalonamento de projetos, apesar de ser uma das etapas com mais modelação no processo de gestão de projetos, ainda apresenta uma grande divergência entre a teoria e a prática. Os modelos computacionais exigentes, e o consequente apelo para a sua simplificação, desviam a sua implementação dos processos de gestão de projetos reais. Especial atenção tem sido dada para a robustez dos cronogramas produzidos que enfrentam a omnipresença da incerteza. Uma forma "fácil" de resolver o problema é adicionar *buffers* temporais, mais ou menos habilmente calculados, que resultam sempre num aumento da duração do projeto e, por consequência, do seu custo. Uma abordagem melhor para lidar com a incerteza pode passar por explorar a folga que possa existir num determinado cronograma, especialmente quando é usado um cronograma não-ótimo, o que geralmente acontece na prática. A combinação de tal abordagem com os recentes avanços na modelação da alocação de recursos e das técnicas de escalonamento, para lidar com o aumento da flexibilidade dos recursos, são uma linha de investigação promissora para se obterem ferramentas de escalonamento de projetos mais adequadas. As formulações FRCPSP (*Flexible resource profile Resource Constraint Project Scheduling Problem*) são um passo nessa direção, mas uma abordagem distinta pode ser seguida considerando que, apesar de poderem ou não ser alocados de forma flexível, os recursos são, eles próprios, flexíveis. Mais especificamente, considerando recursos renováveis, a sua capacidade para executar trabalho numa unidade de tempo, considerada unitária nos modelos tradicionais, pode ser aumentada, de forma a executarem trabalho adicional numa unidade de tempo, ou diminuída, com a consequente redução do trabalho realizado. Na realidade, esta flexibilidade é muitas vezes considerada pelos gestores de projeto. O objetivo aqui é usar esta possibilidade para que, quando combinada com a folga existente em algumas atividades num cronograma, os desvios que ocorram durante a execução do projeto possam ser absorvidos. No caso mais crítico, em que uma atividade crítica (atividade sem folga) tenha a sua duração aumentada, a estratégia consiste em desacelerar atividades não críticas (atividades com folga), colocando os seus recursos num modo de trabalho diminuído, para que a atividade crítica, que está prestes a aumentar a sua duração, ainda possa ser executada dentro do tempo, usando os seus recursos num modo de trabalho aumentado.

Esta tese analisa esta combinação, as suas consequências e limitações e propõe alguns modelos para gerar e melhorar o cronograma inicial (*baseline*) de modo a melhor aproveitar esta abordagem.

Palavras-Chave: Gestão de Projetos, Escalonamento, Alocação de recursos, RCPSP, Incerteza.

INDEX

Figure index.......... xv
Table index.......... xvii
Symbols and Acronyms.......... xix
1. Introduction.......... 1
  1.1 A Project.......... 2
  1.2 Project Management.......... 2
    1.2.2 Project Scheduling.......... 5
    1.2.3 Project Control.......... 6
  1.3 Motivation.......... 6
  1.4 Objectives.......... 8
  1.5 Methodology.......... 8
  1.6 Thesis structure.......... 13
2. Background and Related Work.......... 17
  2.1 Overview.......... 17
  2.2 Project Scheduling Concepts and Models.......... 19
    2.2.1 CPM.......... 19
    2.2.2 PERT.......... 22
    2.2.3 Resource types.......... 24
    2.2.4 RCPSP.......... 26
    2.2.5 SRCPSP - Stochastic Project Scheduling.......... 39
  2.3 State of the art.......... 42
    2.3.1 Proactive/Reactive Project Scheduling.......... 44
    2.3.2 FRCPSP.......... 49
    2.3.3 Stochastic work content.......... 50
    2.3.4 Other related work.......... 51
  2.4 Conclusion.......... 52
3. Problem description.......... 53
  3.1 Problem definition.......... 53
  3.2 The model.......... 57
    3.2.1 Capacity.......... 58

3.2.2 Slack ... 59
3.2.3 Duration ... 61
3.2.4 Score ... 65
3.2.5 Balance ... 68
3.2.6 More on durations ... 69
3.2.7 Alternative schedules ... 71
3.2.8 Extreme cases ... 73
3.2.9 Additional remarks ... 75
3.3 MIP formulations ... 77
3.3.1 Resource flexibility ... 77
3.3.2 Slack management ... 78
3.3.3 Deadline ... 78
3.3.4 Bi-objective ... 79
3.3.5 Target flexibility ... 79
3.3.6 Notes on formulations ... 80
3.4 Scheduling process ... 80
3.4.1 General procedure ... 81
3.4.2 Multi-project management ... 82
3.4.3 Modes ... 83
3.4.4 Scheduling procedure ... 84
3.5 Comparative notes ... 88
4. Computational Study ... 91
4.1 Test environment ... 91
4.1.1 Development environment ... 92
4.1.2 Test algorithms/tools ... 92
4.1.3 Test set ... 94
4.2 Preliminary study ... 94
4.2.1 Results ... 94
4.2.2 Conclusions ... 97
4.3 Flexibility Data: schedule flexibility ... 98
4.3.1 Slack ... 99
4.3.2 SIF - Schedule Intrinsic Flexibility ... 109

4.3.3 Analysis ... 115
4.4 Flexibility data: resource flexibility ... 115
4.4.1 Optimal $\alpha-, \alpha+$ ... 116
4.4.2 Impact of varying $\alpha-, \alpha+$ ... 117
4.5 Enhance schedule flexibility ... 121
4.5.1 Heuristic procedure ... 121
4.5.2 Modified DH B&B ... 124
4.5.3 Results overview ... 127
4.6 Study summary ... 128
5. Conclusions ... 131
5.1 Concluding remarks ... 132
5.2 Future/Open Work ... 136
Bibliography ... 141
Appendix I (Proofs) ... 151
1. Proof for expression (3.12) ... 151
2. Proof for expression (3.13) ... 152
3. Proof for expression (3.17) ... 153
4. Proof for expression (3.26) ... 154
Appendix II (Additional examples) ... 155
1. Alternative schedule's slack and score ... 155
Appendix III (Charts) ... 157
Appendix IV (Tables) ... 161
Annex I (Algorithms) ... 189
1. The B&B DH algorithm ... 189
2. Minimal Delaying Alternatives algorithm ... 194
3. Serial SGS and priority rules ... 195

FIGURE INDEX

Figure 1: Elementary project management process ... 4

Figure 2: Project example with 14 activities in AoN format ... 21

Figure 3: CPM schedules for project example ... 22

Figure 4: Example a) for conditions (3.4), b) for condition (3.5) and c) for condition (3.6) ... 56

Figure 5: Project example with 14 activities and 2 resource types in AoN format ... 57

Figure 6: Gantt chart for minimal makespan baseline schedule (representation in MSProject) ... 57

Figure 7: Resource profile for optimal baseline schedule: a) for resource 1 b) for resource 2 ... 58

Figure 8: Fully stacked resource profile example ... 59

Figure 9: Resource profile for buffered schedule: a) for resource 1 b) for resource 2 ... 64

Figure 10: Resource profile for end buffered schedule: a) for resource 1 b) for resource 2 ... 65

Figure 11: Resource profile for alternative optimal schedule: a) for resource 1 b) for resource 2 ... 72

Figure 12: Resource profile for non-optimal schedule: a) for resource 1 b) for resource 2 ... 73

Figure 13: Project duration for all 480 J30 instances ... 95

Figure 14: Amplified detail of project duration chart ... 95

Figure 15: #NC for each scheduling method for all J30 instances (radar chart) ... 100

Figure 16: #NC for each scheduling method for all J30 instances (x-y chart) ... 100

Figure 17: Normalized frequency of #NC for each scheduling method ... 101

Figure 18: Average #NC for J30 instances with same <NC,RF,RS> ... 103

Figure 19: #NC for instances with same <NC,RF,RS> having avg(#NC) a) lowest and b) highest ... 104

Figure 20: Σslack for each scheduling method for all J30 instances ... 105

Figure 21: Average Σslack for J30 instances with same <NC,RF,RS> ... 106

Figure 22: Σslack for instances with same <NC,RF,RS> and avg(Σslack) a) Lowest b) Highest ... 107

Figure 23: Average SIF for instances with same <NC,RF,RS> for each resource type ... 110

Figure 24: SIF for instances with same <NC,RF,RS> for each resource type with lowest avg(SIF) ... 111

Figure 25: SIF for instances with same <NC,RF,RS> for each resource type with highest avg(SIF) ... 112

Figure 26: Average SIF/Σ(dr) for instances with same <NC,RF,RS> for each resource type ... 114

Figure 27: Number of activities with slack according to α^- for each scheduling technique ... 118

Figure 28: Number of activities that can benefit from resource flexibility according to α^+ ... 120

Figure 29: SGSS based enhanced schedule flexibility ... 122

Figure 30: Priority rules' relation with enhanced schedule flexibility 123
Figure 31: DH-B&B based enhanced schedule flexibility 126
Figure 32: Enhanced schedule flexibility results 128
Figure 33: Project duration for all 480 J30 instances (large chart) 158
Figure 34: Σslack of each scheduling method for all J30 instances (larger chart) 159

TABLE INDEX

Table 1: The Nine Schools of Project Management Thought 10

Table 2: Fundamental Project Scheduling Methods 18

Table 3: CPM calculations for project example 21

Table 4: Project scheduling methods under uncertainty 47

Table 5: Example slack values 60

Table 6: Possible duration span 61

Table 7: Allowed duration span 63

Table 8: Example score extreme values 66

Table 9: Example balance values 68

Table 10: Project execution modes 84

Table 11: Methodology comparison 88

Table 12: J30 project duration summary 96

Table 13: J30 average resources 97

Table 14: Slack aggregated values for J30 instances 108

Table 15: Relative slack aggregated values for J30 instances 108

Table 16: SIF aggregated vales for each resource type and scheduling technique 113

Table 17: SIF/Σ(dr) aggregated values for each resource type and scheduling technique 115

Table 18: Resource flexibility parameters 117

Table 19: Activity minimal durations regarding α^- 119

Table 20: Average number of activities with slack according to α^- for each scheduling technique 119

Table 21: Activity minimal durations regarding α^+ 120

Table 22: Slack and score for distinct schedules (S, S', S'') 155

Table 23: Average #NC and Σslack for J30 instances with same <NC,RF,RS> 162

Table 24: T, #NC and Σslack for all J30 instances 163

Table 25: Average SIF for J30 instances with same <NC,RF,RS> 175

Table 26: Average relative SIF for J30 instances with same <NC,RF,RS> 176

Table 27: SIF for each resource type for all J30 instances 177

Table 28: Priority rules 196

SYMBOLS AND ACRONYMS

Symbol	Description
G	Project network (digraph or DAG - Directed acyclic graph) $G = (V, A)$
V	Set of all project activities (vertices or nodes)
A	Set of precedence relations (arcs or edges)
i	Activity (job or task) $i = \{1, \dots, n\} \in \mathbb{N}$
$i = 1$	Dummy start activity
$i = n$	Dummy end activity
(i, j)	Precedence: $(i, j) \in A$, i is an immediate predecessor of activity j
$Pred_i$	Set of (immediate) predecessors of activity i
$Succ_i$	Set of (immediate) successors of activity i
$Pred_i^*$	Set of all (immediate and transitive) predecessors of activity i
$Succ_i^*$	Set of all (immediate and transitive) successors of activity i
s_i	Start time of activity i
d_i	Duration (processing time) of activity i
d_i^{nom}	Nominal (starting or deterministic) duration of activity i
d_i^{min}	Minimal duration of activity i
d_i^{max}	Maximal duration of activity i
f_i	Finish (end or completion) time of activity i
K	Set of resource types
k	Resource type $k = \{1, \dots, m\}, k \in K$
r_k	Resource requirement of type k
r_{ik}	Resource requirement of activity i of type k
a_k	Resource availability of type k
a_k^{nom}	Nominal resource availability of type k
a_k^-	Minimal resource availability of type k
a_k^+	Maximal 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)

u_{kt}	Resource unused capacity of type k at instant t
w_{ik}	Work content of activity i of resource k
δ	Project deadline
δ_i	Activity i deadline
e	A project example
P_t	Set of activities in progress at time t (active activities)
E	Set of eligible activities
D	Set of finished activities (done activities)
F	Forbidden set ($F \subseteq V$)
$\underline{F}$	Minimal forbidden set
B	Delaying alternative ($B \subseteq F$)
$\underline{B}$	Minimal delaying alternative
S	A schedule $S = \{s_1, s_1, \dots, s_n\}$
PS	A partial schedule (already scheduled activities)
S^b	A baseline schedule
S^w	A working schedule
C	Set of all cutset activities
T	Project duration (completion time)
p	Level of the search tree (B&B)
m	Decision point (time)
$LB(x)$	Lower Bound of x
$UB(x)$	Upper Bound of x
$O(g(n))$	Order of complexity of $g(n)$)
$P[x]$	Probability of x
$E[x]$	Expected value of x
$avg(x)$	Average or arithmetical mean of x
$\lceil x \rceil$	Smallest integer not less than x
$\lfloor x \rfloor$	Largest integer not greater than x
$==$	Test for equality in pseudo-code
$//$	Start of a comment in pseudo-code

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 goal or, more generally, a 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 (due to, for example, increasing technological evolution, multidisciplinarity and globalization), along with increasing competitiveness. In this scenario, project managers face, right from the start, the challenge to balance the scope-time-cost triangle, where time and cost "cannot" deviate from the agreed upon values, but the scope embraces/encompasses a whole set of uncertainties. A typical scenario for the project execution is that of assigning a set of resources that are available for the duration of the project. While this approach seems quite comfortable for the project manager, it leaves little space for coping with uncertainties especially when the project plan is established as an optimal or near optimal schedule, which is the correct option if one wants to be at its best competitive form. This is one of the reasons that lead to budget overruns and delays that occur in the majority of large projects (Couto & Teixeira, 2007; Flyvbjerg, Bruzelius, & Rothengatter, 2003).

So, uncertainty resulting from several origins, like not fully understood technical challenges and/or requirements, leading to incorrect estimations of the necessary work to be done, along with resource unforeseen unavailability (Elmaghraby, 2005), collides many times 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 (typically a day) either by considering this extra work as overtime (in which case there will be additional costs) or not (Jia, Fan, & Lu, 2007; Olsen & Swenson, 2011).

These are the issues that will be further studied in this thesis and a research line will be identified that enables the development of a prototype for further supporting project managers to cope with these increasing demands.

1.1 A Project

A project is then the basic concept underlying this research. It always involves the notion of a temporary event in the sense that it is limited in time, having a start time and a finish time, with a set of (implicitly or explicitly assigned) resources, to achieve a set of goals.

Projects, as defined in this simplistic way, always existed but, with the increasing demand to deliver them on time, on budget and within a context of increasing resource scarcity, the requirement to adequately manage them became mandatory.

According to its basic definition, a project can be seen as a three dimensional entity known as the project management triangle of scope-time-cost, which will be evaluated to provide measures of the project's management success, that is to say, the ability to reach the predefined goals (scope) within the given timeframe (time) and with the allocated resources (cost).

In order to better contextualize what is the concern of this research, the following section will further address Project management.

1.2 Project Management

In the most abstract formulation, project management is a process that deals with activities and resources and how to organize them in achieving a set of goals. This definition gives little information about the actual understanding of what managing a project is, but already unveils some subjects to be decided upon regarding:

- How activities are defined and can they change while the project is being executed;
- The type of resources (human, machine, money, others) and their availability;

- The nature of the goals (artefacts to be built, changes to be done, deadlines);
- Supra project goals (quality, risks, ethics);
- Outside factors (to the project) and other organizational issues.

It is not of particular interest, in this context, to go into detail in answering these questions, but the point in mentioning them is that this subject deals with different aspects of our society. It is a huge task to study all of them as a whole so, as any engineer learns from theory and experiences in practice, one should split huge problems into manageable ones in order to succeed.

- But how can this problem be divided?
- Into which pieces?
- Where to start?

Since its early days, project management aimed to answer these questions and many others that arise from these ones. The search to answer some of these questions led to the development of what is commonly accepted as the first modern project management techniques that emerged in the late 1950's: the "Critical Path Method" (CPM) and the "Program Evaluation and Review Technique" (PERT). Henceforth, a vast number of techniques were developed and enormous research effort was put on understanding their ability to deal with projects and how to manage them. This evolution enabled projects, not only to be defined systematically as such, but to be increasingly managed in a specific way, i.e., project management was increasingly performed within a common or standardized process.

1.2.1 Project Management Process

The "Project Management Process" (PMP) purpose is to define the main stages that the project management should go through in order to assure the successful delivery of the project's goals.

Figure 1 presents an elementary project management process, simply to highlight the distinct possible research areas and to identify the focus of the present research.

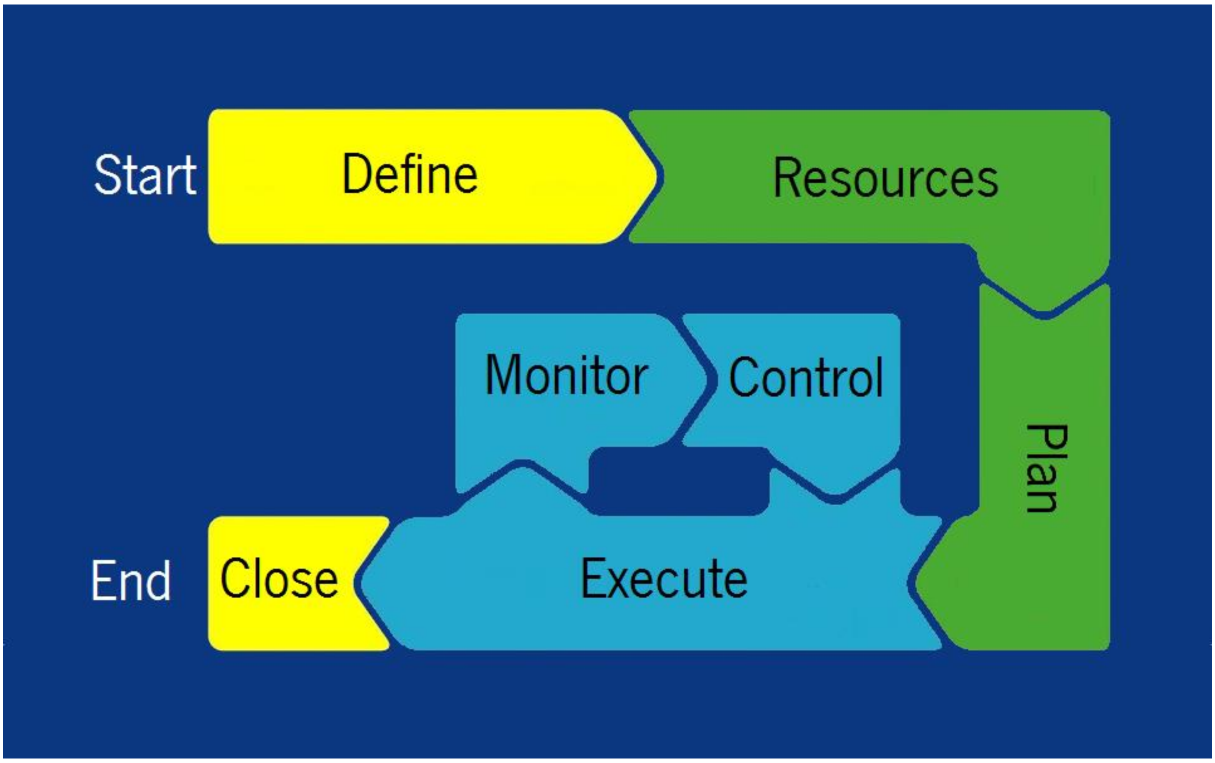


Figure 1: Elementary project management process

This project management process emphasises the temporal nature of a project with a clear "Start" and "End". In its most basic form, the output of the first stage (Define) should be the project's scope definition while the last (Close) deals with the goals fulfilment evaluation. In between, resources are assigned to the project (Resources), allowing estimating its cost, a plan is established for its execution (Plan), allowing estimating its time and the actual project's execution is performed (Execution). At the execution stage, the role of the project manager is to adequately perceive how the execution is being done (Monitor) and to define and implement corrective actions whenever necessary to assure the project's goals will be achieved (Control).

Project managers as the major authority within the project context are involved in all project stages. But while project definition and resource assignment are stages that typically have to be negotiated with other project stakeholders and agreed upon, planning activities and the subsequent monitoring and control are mostly of the project manager's responsibility. Typically, the closing stage will not affect the project's goals but serves mainly the purpose of defining its end, thereby fulfilling its temporary nature. It should also serve to evaluate what has been achieved and how it was achieved so that these conclusions can be gathered as lessons learned to be used in the context of future projects. Accordingly, this research will disregard the entire project’s workflow and associated processes, but will focus on the stages most closely related and dependent on the project manager - planning, monitoring and controlling. Specifically, it will concentrate on the actions that build the plan to be executed and, after perceiving deviations, taking re-planning actions that, while coping with those deviations, can still enable the goals to be reached and therefore to still keep the project in a successful trajectory.

In this context, the project scheduling process that is studied in this thesis is the task of building the project's execution plan, assuming that activities are identified, enough resources are allocated to the project which are already assigned to activities (not necessarily in a unique mode) and the project's goals are defined.

1.2.2 Project Scheduling

Project scheduling is then the problem of sequencing activities in time such that the project's goals are met without violating any of the project's constraints.

The project's goals here are not the ones referred to in the "Define" stage. These have to do with the goals that the project manager needs to set in the definition of the plan, which from now on will be referred to as a schedule, such that the project's goals (defined at the "Define" stage) are reached. The goals set for generating the sequencing of activities can be more precisely named as project scheduling objectives. An example can help clarifying these concepts: the project's goal can be to develop a new aircraft, and a project scheduling objective can be to minimize the project's duration, which in turn can be associated with an additional project goal, to build the new aircraft as fast as possible.

The project's constraints play an important role in project scheduling and can also be closely related to the project's scheduling objectives. They define conditions that must be met in sequencing activities and assume two basic forms that can model most of existing constraints: precedence constraints and resource constraints. Precedence constraints model the fact that some activities must precede (or succeed, which is the same if activities are inverted in the relation) some others, for whatever reason, for some amount of time. Resource constraints model the fact that resources are finite and therefore some activities might not be executed in parallel with some others, even if they do not have any impeding precedence relation, due to the non-existence of enough available resources to execute them all.

The project can now start its execution stage according to the defined schedule which, in a simplistic and deterministic scenario (activities have deterministic durations and resource requirements), is a set containing activity start times (equivalently, finish times can be used, since they differ only by the activities duration). The initially established schedule can be referred to as the "initial baseline schedule" or simply "baseline schedule".

Then, while the project is being "executed", the project manager initiates monitor and control stages. In this context, the focus will be on controlling the project when deviations occur, not being concerned about how they are perceived within the monitoring stage.

1.2.3 Project Control

Project control can be performed in several forms in order to safeguard the project's goals. Some relevant ones are concerned with the economic project evolution, namely the Earned Value Management (EVM), which can be further explored for example in Anbari (2003) or more broadly in Fleming and Koppelman (2010), or with dealing with the impact of foreseen deviations like Risk Management (RM), which can be further explored in Elmaghraby (2005) or more broadly in Chapman and Ward (2003). The focus here will be limited to the specific consequences on the current schedule, not being concerned with other consequences beyond the current schedule. Therefore, project control will be limited to project re-planning or re-scheduling in a broad sense, because activity start times will not necessarily change. This process will generate new schedules that can be referred to as working schedules which, once determined, will be the new current schedule. This process of planning and control, involving the project's scheduling and re-scheduling, can use distinct approaches, depending on the selected model to represent the project and its schedule. These models can range from the most simplistic ones that assume that all variables are deterministic and infinite resources are available (no resource constraints), or can be more complex, assuming that there are resource constraints or that some variables are stochastic, namely activity durations. In the latter case, the aim is to produce robust schedules, in the sense that they can deal with the uncertainty that projects face, represented in the stochastic variables, without endangering the project's defined timeframe and thus, the project's success. Robust scheduling can be achieved within an integrated strategy of project planning and control or, as mentioned before, a scheduling/re-scheduling approach. This model is commonly named as proactive/reactive project scheduling.

These topics will be further detailed in the next chapter.

1.3 Motivation

This project started with the will of the researcher to further understand the project management process and particularly to delve into the project scheduling problematic. His previous extensive experience as a project manager professional gave him a broad perspective of the issues and difficulties involved in this subject and motivation to contribute to the field’s improvement. Organizations tend to have standard project management procedures that help. These procedures, either general or organization specific, cover the whole range of the project's management process but, more often than not, fail in providing tools and techniques to fully explore optimal resource allocation and dealing with unforeseen events, leaving too much space for empiric decisions that are hard to control and can led to far from optimal results.

On the other hand, it is the belief of the researcher, resulting mainly from his experience as project manager, that activity duration is not the best (independent) variable for a project manager to gather information for a defined activity but rather the activity's work content. Having this information, resources can be allocated and durations are determined. This approach is followed by Tereso et al. since 2002 (Tereso, Araújo, & Elmaghraby, 2004; Tereso, Mota, & Lameiro, 2006; Tereso, Novais, Araújo, & Elmaghraby, 2009). This approach, combined with new proactive/reactive scheduling models, which enable to develop plans taking into account the uncertainties involved, can be used to develop new methods, hopefully producing more interesting solutions for effective project management under uncertainty, resulting in better control of overruns and delays.

The researcher also faced the challenge of coping with a changing environment where the project management scope-time-cost triangle does not remain perfectly defined after a project schedule is settled. During project execution, change requests (change in projects' features) are submitted (impacting in the set of projects' activities), due dates can be crashed (or in fewer cases relaxed), resource availability changes (rotation, illness, concurrent projects) and, more than often, costs have to be cut (it is not worth mentioning the case where costs can be increased).

More generally, in a fast changing environment like the one faced nowadays, there are many factors that contribute to budget overruns and delays that occur in the majority of large projects (Couto and Teixeira 2007, Flyvbjerg et al. 2003). Despite the abundance of commercial tools for project management (Kolisch, 1999), the assistance provided to managers has not allowed their proper planning, according to project's progress, especially when projects subject to uncertainty face deviations. Such tools fail in effectively managing the risk involved and simultaneously keep projects in time and on budget.

The question is then how can a project manager develop and control a plan that is time effective and is simultaneously able to cope with uncertainties?

1.4 Objectives

The previous section defined the reason ("why?") this research project is being made. In the next one, the approach used ("how?") will be presented in a thorough way, enabling its positioning in the scientific research process. The solution ("what?") will be stated as an objective to be reached together with the corresponding research question to be answered.

As stated before, the mix of two lines of research will be the driver of this project. They are:

- the resource allocation problem considering stochastic work contents;
- the proactive/reactive scheduling techniques.

Accordingly, this project's ultimate goal is to improve project management practices through the development of new models and methods for project scheduling and resource allocation that can be reflected in a tool to assist in solving some of the decision problems faced by projects managers.

Or, translating this statement into a research question, for which an answer will be the focus of this research:

> *"Can Project Management in Resource Constrained Projects be improved by using a combination of the assumption that activities have stochastic work contents with the use of proactive/reactive scheduling techniques?"*

1.5 Methodology

The research that will be undertaken is not intended to develop any new ground breaking theory. It will rather focus on a particular aspect of project management and, as it is expected, will develop some "substantive theory" that will fill a gap in existing theories. This option will not result in any live changing theory but it is of the researcher's concern that the theory produced will contribute to reduce the typical gap between research and practice.

It is worth mentioning that this has been the usual way to identify research questions in many scientific areas, and also in project management research, as demonstrated in recent studies (Hällgren, 2012). This way to spot research opportunities is sometimes called "gap-spotting". It is claimed that researchers should not only fill-in these gaps in theory, but to become bolder in challenging the long-held theories. Believing that one should have a critical view about existing assumptions, it also cannot be underestimated the validity of filling-in these gaps in theory or, most commonly, to create increments in existing theory.

Accordingly, it is not intended to break the problem into smaller pieces, solving them all and reaching an overall conclusion (theory). The contribution can be centred on a selected piece, solve it and achieve a conclusion (theory) for this particular piece.

Going back to the questions raised, one way to consistently undertake the research is to analyse the problem under a philosophical view and select a perspective most suited to the cross examination of the specific research question and the researcher's affinity.

The project management process is, inherently, a social activity. It always involves some kind of human intervention and interaction, from the very beginning of the project idea, passing through the definition of activities, selecting resources or defining constraints, until deciding to consider it concluded and reaching some conclusions.

On another level, when it concerns its applicability, it may range from a project to produce some kind of artefact[1] to a massive change in an organization.

At this point, a major decision has to be made. Will project management be regarded as a “social entity” or rather as a “machine” or will it be something in between?

There is no consensus on how many different ways there are to categorize these views. As an example, it can be mentioned the work published by Anbari, Bredillet and Turner (2008) in which they organize some different views in nine distinct categories that they called "schools of thought". This study is mentioned, not because such a partitioning is of any particular interest to this research, but because it identifies and describes, in a very brief way, several aspects that are useful to be mentioned within this argumentation.

---

[1] Producing an artefact should not be underestimated. It might range from something very simple to a very complex system. It can also mean producing a prototype that might be intended to be mass produced.

Table 1: The Nine Schools of Project Management Thought

Reproduced from Anbari et al. (2008, TABLE 1)

School	**Metaphor**	**Key idea**	**Came to prominence**	**Influence**
Optimization	The project as a machine	Optimize outcome of the project using mathematical tools	Late 1940s	Operations Research
Modelling	The project as a mirror	Use of hard and soft-systems theory to model the project	Hard-systems: mid 1950s; Soft-systems: 1990s	Systems theory, Soft systems methodology
Governance	The project as a legal entity	Govern the project and the relationship between project participants	Contracts: early 1970s; Temporary organization and governance: 1990s	Contracts and law, Governance, Transaction costs, Agency theory
Behaviour	The project as a social system	Manage the relationships between people on the project	OB: mid 1970s HRM: early 2000s	OB (Organizational Behaviour) HRM (Human Resources Man.)
Success	The project as a business objective	Define success and failure Identify causes	Mid 1980s	Internal to Project Management
Decision	The project as a computer	Information processing through the project life cycle	Late 1980s	Decision sciences, Transaction costs
Process	The project as an algorithm	Find an appropriate path to the desired outcome	Late 1980s	Information systems, Strategy
Contingency	The project as a chameleon	Categorize the project type to select appropriate systems	Early 1990s	Contingency theory, Leadership theory
Marketing	The project as a billboard	Communicate with all stakeholders to obtain their support	Stakeholders: mid 1990s Board: early 2000s	Stakeholder management, Governance, Strategy

Table 1 of Anbari et al. (2008) is reproduced here in order to enlighten that the terminology that is being used does not have the exact same meaning as the one used in the referred study. The term that refers a project as a "social entity" used in this text has a broader meaning than the term "social system" used in the study. The same is valid for the term "machine". The envisioned model for this analysis assumes a much more continuous spectrum from "social" to "machine" but, to make an effort to compare with the division proposed by Anbari et al. (2008), one can say that there are far fewer steps between them.

If the research is more focused on understanding how human behaviour influences one or more aspects in the project management process, it is most likely that the researcher's philosophical view of the problem is an interpretivist one. As anyone that managed a project knows, in almost all processes involving humans (like stakeholders or resources), social aspects[2] must be taken into account. They may not be explicitly expressed in the defined activities, resources or constraints and consequently in

---

[2] Social aspects concerns human behaviour and interaction with others in a social environment.

the project's plan, but they have to be taken into account. Choosing, primarily, to understand these aspects falls into this vision. Examples of questions (not to be confused with research questions) concerning this are:

- What is the influence of cultural aspects in defining a project?
- How do organizations influence the project management process?
- Which additional aspects have to be considered when using a multi-disciplinary team?

Or, eventually, some harder questions, like the ones an individual can face in managing projects related to creative areas, like how to integrate inspiration or motivation in project management, can be posed.

In this interpretivist view of project management, the researcher will probably be following a deductive approach and will be mostly dealing with qualitative data (either solely or in combination with some quantitative data), requiring most certainly the use of some kind of human interaction based data collection.

On the other hand, if the researcher assumes that social aspects are negligible, or more realistically, they are incorporated somehow in the defined project parameters (to be materialized later on), the research can be focused on a more self-contained environment. The questions raised in this simplified view of project management can be dealt with a more classical positivistic view. Within this view, the most suitable approach to be used is a deductive one in which the proposed theory will be validated with quantitative data. These data might be collected in real life ongoing[3] projects, as in action research or using simulated or archival data (real life or simulated), as is usually done if one uses an experiment strategy. This is the traditional option within the engineering community in spite of some increase in alternative ones (Borrego, Douglas, & Amelink, 2009).

These options represent two extreme visions of the problem. There are additional visions in between and its classification might be a bit more difficult. As an example of such a distinct vision, the study performed by Gul (2011) is referred. In it the "critical realism" paradigm is advocated to be used in project management leading to a mixed methodological approach. The case study is the preferred research strategy and the use of qualitative and quantitative data becomes natural within this paradigm.

This discussing is intentionally kept simplistic because its only purpose is to substantiate the option taken, presenting in a generic and non-exhaustive way, the options the researcher faced.

---

[3] Ongoing projects means projects that are been executed while the research is being performed.

Given the researcher's background in engineering, the option toward positivism is the obvious choice. However, his professional experience in managing projects, whose main resources are human resources, in a transnational environment, enables him to be aware of how social aspects are crucial and have to be taken into account in successfully managing a project, i.e., in reaching the project's goals. Recalling what was said previously about incorporating somehow, in the defined project parameters, these social aspects, is the core concern of this research project. It follows a positivist approach to project management, integrating in the project parameters some kind of "buffer"[4] to accommodate these social aspects that usually result in project delays and budget overruns. In real life projects, an experienced project manager, that has a good knowledge of the project's team, will also incorporate some of these variables, on an empirical basis, without any explicit supporting methodological framework. But this is hard to reach in large projects because it has not been made explicit. This way, organizations may be tied up to these project managers, since they may not share the knowledge they acquired with the organization. This may be good for the individual but it is not good for the organization.

The "buffers" mentioned above will be materialized using a combination of a statically defined parameter (defined before the project is executed) and a dynamically defined parameter (may change during the project's execution). The first is to consider that work content is defined by a stochastic function and the second is proactive/reactive scheduling. These techniques can accommodate uncertainties in projects from any sources including those due to social factors enabling to neglect human interactions effects.

As such, ontologically this research project interprets reality in an objectivistic view which means that human interpretation will not be of great concern and, as such, human interaction will not be either.

Epistemologically, a classical positivistic philosophical view of knowledge will be reflected in the methodology to be followed.

The approach will be deductive because it is stated that by merging two existing theories we can enhance the project management process (the premise), given that it has to be proven when applied to one or more sets of data.

On the research design level, an experimental strategy and (mono method) quantitative data will be used. The experiences and associated data to be used as input will be performed in a cross-sectional time horizon. After developing the combination of project management theories (the stochastic work

---

[4] Here, the term "buffer" is not used in the Project Management terminology sense

contents' approach and the proactive/reactive scheduling techniques), embodied in a set of algorithms, models and methods that will be automated in a software tool, a set of predefined experiences will be conducted using this and other techniques, in order to obtain comparable results (the dependent variables) to be analyzed. Input data (independent variables) will be gathered from benchmark databanks, like Kolisch and Sprecher (1997)'s PSPLIB, that are available for this purpose. Whenever it becomes necessary, data adaptations and extensions might be made in order to create specific test conditions. In any case, data integrity and consistency must be assured.

Once results are collected, they must be analysed. Typically, these techniques will result in some kind of formulation that involves an optimization formula like minimizing the overall project cost, the use of a particular resource or the projects' duration. This means that no complex data analyses like complex statistics will be necessary but rather simple values (and their basic statistics) like distance from optimal solutions and processing time.

1.6 Thesis structure

This thesis is organized in five main chapters, four appendices and one annex.

Chapter 1, this one, serves as a general introduction to the thesis, describing its main domain and research area, what motivated the author to engage in such an endeavour and specifically on why focusing in project planning, monitoring and control. Having established this focus, the research question and the problem statement further refines the scope of this project and the chapter ends with the definitions and justification of how is the research performed in methodological research terms.

Chapter 2 describes the existing context regarding this research field. It starts by a general approach in identifying the most relevant authors in the field and their respective published work. This is followed by a description of the basic concepts presenting in a greater detail the most typical scheduling problems and solutions that are relevant to the present work. The chapter ends with the state of the art in this field whose limitations contributed to motivating this research and refine the research question.

Chapter 3 deals with the problem description and its theoretical definition and analysis. It starts by identifying the problem and proposing a solution model, based on the concept of intrinsic schedule flexibility, which is the flexibility existing in schedules resulting from their activity slack, combined with the concept of resource flexibility, that is the capacity a resource can have to work below or above its nominal work capacity. These concepts are defined and described in detail throughout the chapter by developing the necessary theoretical foundations and applying them in a project example. Some conceptual integer programming formulations are presented that, based on the Resource Constrained Project Scheduling Problem (RCPSP) model, extend its application to include schedule flexibility optimization that leads to an increase in the model capability to deal with uncertainty. Then, the model is used within a typical scheduling process, starting with a general procedure which is followed by a description of its advantages, when used in a multi-project environment, and by identifying some possible execution modes or execution constraints, finally leading to a more general scheduling procedure example. The chapter ends with a brief comparison of the proposed methodology with other related ones.

Chapter 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 or tools and a set of project examples. Within this environment, a preliminary study is performed in order to put them into perspective regarding their ability to generate good schedules in the sense of minimal duration and considering nominal (or deterministic) activity durations. This enables an increased perception of the potential use of the proposed method in combination with the tested scheduling algorithms. Then, using the same test environment, the proposed method's potential is assessed, regarding the generated schedules' flexibility and the impact that resource flexibility, with distinct flexibility parameters, have in limiting its exploitation. The chapter ends with some tests to evaluate the possibility to use the scheduling algorithms in order to select the best resulting schedules regarding its highest flexibility values.

Chapter 5 compiles the main conclusions that can be drawn from this work. It starts with a summary of the main features of the proposed methodology and to which extent the research question is answered. It finishes with the enumeration of future work that can be done related to this research, some of which has been already referred to in the previous chapters, whenever improvements or limitations of the present work are assumed to exist. Some hints that can be followed to proceed with the research of the identified topics are also given.

The main text is followed by the list of references, a set of appendices that complete the main text, and an annex that presents all algorithms used in the implementation of the computational study.

Appendix I presents proofs for some expressions defined in the main text and Appendix II presents some additional project examples. Then Appendix III reproduces some charts presented in the main text with increased size and consequently with better readability. Appendix IV includes several tables presenting detailed results that are mentioned in the main text, serving as basic data to generate charts and aggregated data.

Finally, in Annex I the existing algorithms used in the implementation are presented: first the Demeulemeester and Herroelen Branch-and-Bound algorithm (Demeulemeester & Herroelen, 1992, 1997), including the Minimal Delay Alternatives algorithm, and then the Serial Scheduling Generation Scheme (Kolisch, 1996b) with the used priority rules.

2. BACKGROUND AND RELATED WORK

In this chapter the previously addressed topics related to this research, either because they are the basis over which the new developments are made or because they focus on solving similar problems, will be further described and relevant literature will be identified. It will be made according to an Overview > Concepts > State-of-the-art approach, which is to say that it begins with a general literature review to give an overview of some of the related fields, detailing then some basic concepts where the current research relies on, focusing later on more recent advances or the state of the art of this research area.

2.1 Overview

The available literature concerning project management, its evolution and its main concepts is vast. As mentioned earlier, project management as a scientific discipline, started in the mid of the twentieth century. Before this, Henry Gantt created the Gantt charts, which are still in use today (Wilson, 2003). Later, several concepts were formulated and some associated project management tools were developed. Most of these concepts or models became scientific theories that led to further scientific research.

Table 2 identifies some fundamental project scheduling methods whose underlying concepts are the basis for most of the existing scheduling methodologies and tools, which will be further described in the next topic. The table classifies such methods according to their ability to incorporate two important project variables, specifically project activity variables:

- The resource availability limitations (Resources) which if considered (Limited) require the definition of the amount of resources that each activity requires to be processed, and;
- The nature of activity durations (Durations) which can be considered as being defined by a fixed value (Deterministic) or that cannot, in which case they must be represented by a more complex (Stochastic) variable.

Early models simply did not take resources into account, considering that their availability was unlimited which greatly simplifies the model's complexity and its computational hardness.

Table 2: Fundamental Project Scheduling Methods

Resources / **Durations**	**Unlimited**	**Limited**
Deterministic	CPM (Critical Path Method)	RCPSP (Resource Constrained Project Scheduling Problem)
Stochastic	PERT (Program Evaluation and Review Technique)	SRCPSP (Stochastic RCPSP)

When resources started being considered limited, classical models assumed that each activity has a deterministic duration and known resource requirements, and attempted to “optimally” schedule the activities, in whichever sense optimality was defined. This gave rise to the well-known RCPSP (Resource-Constrained Project Scheduling Problem). The majority of these studies suffer from the serious flaw of ignoring the uncertainty present in real life projects. Unfortunately, the inclusion of uncertainty in these models seemed to meet with insurmountable obstacles. Initial attempts to overcome these obstacles used more or less complex probability distributions to model time uncertainties, assuming averages (or other single value probability representation) to be the values to use in traditional models (PERT falls into this category). This approach proved to be insufficient to model real world projects (Elmaghraby, 2005).

Therefore, researchers had to deal with random variables and had to increase the estimate of the time of realization of certain “key events” by an allowance (or “buffer”) that would absorb delays in case some activities took longer than estimated. It would thus enable to achieve a higher degree of robustness of the resulting schedules, in what is sometimes referred to as the stability/makespan trade-off. It could be achieved simply by right shifting non-started activities where makespan is sacrificed on behalf of the project schedule stability.

Nevertheless, the use of such resource limits and non-deterministic activity durations can be quite relevant in modelling real life projects. As will be presented in the next section, resource constrained models are not too complex, but the resulting problem is computationally hard to solve. On the other hand, modelling uncertain activity durations is much harder to model because of the countless possibilities. To cope with this complexity, most existing methods assume that some knowledge exists about the possible deviations, expressing durations as stochastic variables.

2.2 Project Scheduling Concepts and Models

The basic concepts presentation of project scheduling will start with the models that do not consider resource constraints: CPM and PERT. Such models were the first ones, chronologically, to be developed and were presented in the 1950 decade. At that time, both relied on the Activity-on-Arc (AoA) project network representation, but now are more commonly used with the Activity-on-Node project network diagrams which will be used from now on. A comparison of these network diagramming methods can be found in Yang and Wang (2010).

2.2.1 CPM

CPM stands for "Critical Path Method" and applies to projects with fixed duration activities in which resources are not considered, i.e., are available in virtually infinite amounts. It gets its name from focusing the project scheduling process in identifying and controlling the sequence of activities that determines the project's duration which is named as the project's "critical path". This simplistic approach to project scheduling enables fast schedule computation[5] from which two main scheduling approaches can be used:

- Early Start Scheduling: start all activities as soon as possible;
- Late Start Scheduling: start all activities as late as possible.

The "Early Start Schedule" is obtained by the "forward pass" algorithm:

```
$ES_1 = EF_1 = 0$
for $i = 2$ to $n$ do
    $ES_i = \max \{EF_j \mid j \in Pred_i\}$
    $EF_i = ES_i + d_i$
$T = EF_n$
```

where ES_i is the Early Start time, EF_i is the Early Finish time and $Pred_i$ is the set of all immediate predecessors of activity i. The project duration is then T.

After performing the forward pass, the "Late Start Schedule" can be obtained by the "backward pass" algorithm that starts creating the schedule from project finish to start, using the project duration T:

---

[5] CPM computation is of complexity $O(n^2)$ where n is the number of the activities in the project.

```
$LF_n = LS_n = T$
for $i = n - 1$ to $1$ do
    $LF_i = \min\{LS_j \mid j \in Succ_i\}$
    $LS_i = LF_i - d_i$
```

where LS_i is the Late Start time, LF_i is the Late Finish time and $Succ_i$ is the set of all immediate successors of activity i.

Note that the project duration is again T, but the algorithm could start by scheduling the dummy end activity n in any time instant and shifting the resulting schedule by the time instant of the dummy start activity 1, i.e.:

```
for $i = 1$ to $n$ do
    $LF_i^0 = LF_i - LF_1$
    $LS_i^0 = LS_i - LS_1$
```

where LF_i^0 and LS_i^0 denote the latest finish and the latest start of activity i for a project starting at time instant $t = 0$.

Having performed the "forward pass" followed by the "backwards pass", each activity has two possible start times and, equivalently, two possible finish times, differing only by the activities' duration d_i. These values define the time interval in which an activity can start (equivalently finish) while not having any impact on the project's duration T. The difference of these values is known as the Float or Slack of the activity and is given by the expression:

$$Float_i = LS_i - ES_i = LF_i - EF_i. \tag{2.1}$$

If $Float_i = 0$ the activity i is named as critical because it must start (equivalently finish) at exactly that time instant otherwise the project will run late. A sequence of critical activities from a single start to a single end is named as "critical path" which is the reason for the CPM naming.

Float as described here is also called "Total Float" (TF) because there are additional types of float including:

- Free Float (FF): The allowable delay in starting activity i while not having any impact in any of its successors, i.e., on the remaining schedule. $FF_i = \min\{ES_j \mid j \in Succ_i\} - EF_i$.
- Safety Float (SF): The allowable delay in starting activity i when all its predecessors finish as late as possible. $SF_i = LS_i - \max\{LF_j \mid j \in Pred_i\}$.

Note that the term slack, as calculated here, is applicable only to the CPM and CPM-like methodologies. When resource constraints are considered, it has the same meaning but is calculated differently in order to consider such additional constraints.

Consider the project example presented in an AoN format in Figure 2. This project has twelve real activities and two dummy activities to univocally define the project's start and end. Activities are topologically numbered and above each respective node their duration is displayed. All precedence relations are of finish-to-start type with no lag (FS=0).

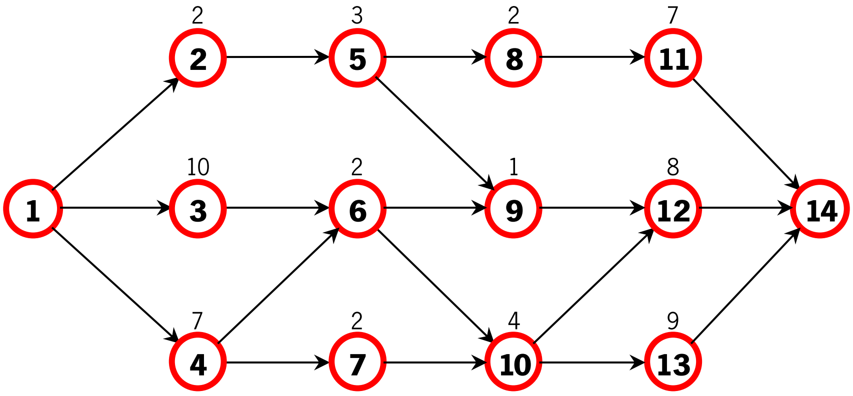


Figure 2: Project example with 14 activities in AoN format

Applying the CPM calculations the critical path of this project can be determined by identifying the path from activity 1 to activity 14 where activities have a Total Float of zero. Table 3 presents such CPM calculation, with the critical activities highlighted, resulting in the critical path 1-3-6-10-13-14. There could be more than one critical path which is not applicable in this case.

Table 3: CPM calculations for project example

i	d_i	ES_i	EF_i	LS_i	LF_i	TF_i
1	0	0	0	0	0	0
2	2	0	2	11	13	11
3	10	0	10	0	10	0
4	7	0	7	3	10	3
5	3	2	5	13	16	11
6	2	10	12	10	12	0
7	2	7	9	10	12	3
8	2	5	7	16	18	11
9	1	12	13	16	17	4
10	4	12	16	12	16	0
11	7	7	14	18	25	11
12	8	16	24	17	25	1
13	9	16	25	16	25	0
14	0	25	25	25	25	0

The resulting CPM schedules: Early Start Schedule and Late Start Schedule, are presented in Figure 3.

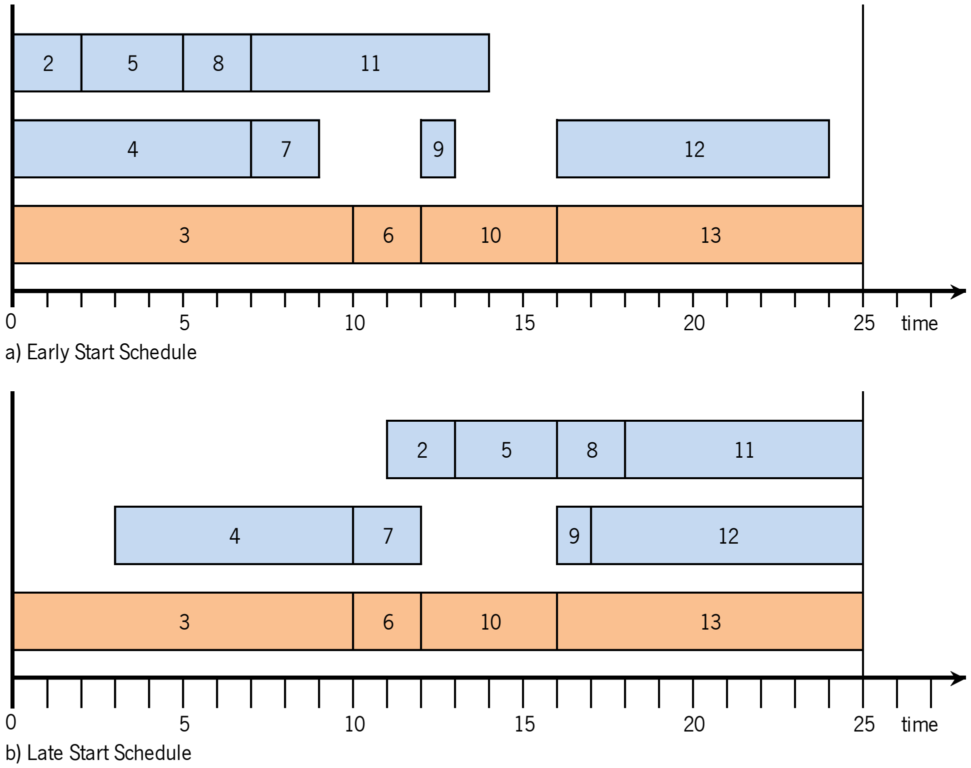


Figure 3: CPM schedules for project example

CPM can be extended to cope with resource limitations by generating a non resource constrained schedule, either the early start schedule or the late start schedule, and respectively right-shifting or left-shifting activities to solve resource over-allocations.

This expeditious scheduling heuristic procedure to cope with resource limitations does not ensure optimal schedule durations, but is frequently used as a basis by project management software to perform scheduling (e.g. Kastor & Sirakoulis, 2009).

2.2.2 PERT

PERT was developed independently from CPM, but shares with it the same basic principles. It differs mainly in the assumption that activities do not, or more strictly may not, have deterministic durations. PERT is the first systematic approach in considering stochastic activity durations in project scheduling. Again the main concern is to identify and control the project's "critical path" but considering that the duration of each activity is defined not as a deterministic value but as a statistical variable that has an average and an associated statistical variance value. When PERT was being developed, the goal was to estimate the conclusion time of a project (developing the ballistic missile "Polaris") that had activities with precedence relations but whose durations where quite uncertain. This approach to project scheduling is more elaborated to cope with the uncertainties, but retains its underlying simplicity that also enables fast schedule computation due to its lack of resource constraints which, for that specific project, Polaris, was good enough since most of its activities were subcontracted.

With PERT, activity durations are considered to follow a beta distribution, which is not necessarily symmetrical with finite non-negative extreme points. It considers three estimations for each activity's duration:

- o - optimistic: the shortest expected duration (when everything goes favourably);
- m - most likely: the expected duration (when everything goes normal);
- p - pessimistic: the longest expected duration (when everything foreseeable goes bad).

Then, the expected activity duration is given by the expression:

$$d_i = \frac{o_i + 4m_i + p_i}{6} \tag{2.2}$$

with a statistical variance of:

$$v_i = \left(\frac{p_i - o_i}{6}\right)^2 \tag{2.3}$$

With these calculated weighted average and variance it is possible to estimate the probability that a project is completed within a determined time period if certain assumptions are made. PERT assumes several ones (MacCrimmon & Ryavec, 1964) including the following ones:

- Activities' durations are statistically independent;
- The critical path is the longest path obtained by summing up their expected activity durations;
- The critical path follows a normal probability distribution, i.e., the critical path contains enough activities such that the central limit theorem applies.

These assumptions are arguable and several studies analyse such assumptions and their impact (e.g. Golenko-Ginzburg, 1989; Herrerías-Velasco, Herrerías-Pleguezuelo, & Van Dorp, 2011), but they enable the calculation of the project's expected duration using the following expressions:

$$T_e = \sum_{i \in C} d_i \tag{2.4}$$

$$V_p = \sum_{i \in C} v_i \tag{2.5}$$

where C is the set of activities in the critical path.

Then, according to PERT, the probability that a project is concluded within a given time period T_d can be computed by:

$$P(T \leq T_d) = \Phi(z), \qquad with\, z = \frac{T_d - T_e}{\sqrt{V_p}} \tag{2.6}$$

where $\Phi(z)$ is the standard normal cumulative distribution function.

PERT's popularity is due to its simplicity both in its statistical duration modelling and its CPM like scheduling methodology (also referred to as CPM/PERT scheduling) which is at the same time the reason for its major limitations: not considering resource constraints and not focusing also on non-critical activities that are classified as such using assumptions that may become invalid during the project’s execution. This latter limitation can be overcome by the use of additional activity duration estimations such as those obtained by Monte Carlo simulation, which was proposed just after PERT was presented (Van Slyke, 1963).

2.2.3 Resource types

Before describing scheduling techniques that consider resource constraints, it is helpful to understand the concept of resources and how they constrain a project. After precedence constraints, resources are the most frequent type of constraint that restrains an activity from being executed in the best timing regarding the project's objective, e.g., as soon as possible if a minimum project duration is to be achieved.

Resources are most often classified regarding the way in which they are available to be used in projects and also how they interact with each other.

Regarding their availability, three major resource types are usually considered (e.g. Slowinski, 1980):

- Renewable: these resource are available within each time period, i.e., the available amount is renewed from a time period to the next. The resource limitations relate to each time period, which are independent from each other. These project constraints can be defined by the expression:

  $$\sum_{i=1}^{n} r_{ikt}^{R} \leq a_k^R, \qquad k \in K^R, t \in [0,T], i \in V \tag{2.7}$$

  where V is the set of n project activities i consuming r_{ikt}^{R} renewable resources k of the set of all renewable resources K^R, at time instant t, which are available in a_k^R units for each time unit of a project being executed from time instant 0 to T.

  Additionally, resources can also be partially renewable which means that they only behave as renewable resources within certain defined time periods.

  Human resources are a typical renewable resource;

- Non-renewable: these resources have a fixed, non-renewable amount that can be used by a project, i.e., they are no longer available if consumed whatever the time period. These project constraints can be defined by the expression:

  $$\sum_{i=1}^{n} r_{ik}^{N} \leq a_k^N, \qquad k \in K^N, i \in V \tag{2.8}$$

  where V is the set of n project activities i consuming r_{ik}^{N} non-renewable resources k of the set of all non-renewable resources K^N, which are available in a_k^N units.

  A project budget (money) is a typical non-renewable resource;

- Doubly constrained: these are resources that cumulatively have the characteristics of the previous ones, i.e., their usage is both limited per time period as well as project wide.

Money limited to a project budget but which has also time periods' limitations, e.g. cannot be used more than a defined limit within each time period, is an example of a doubly constrained resource;

Some projects require that resources are taken into account regarding their dependency to be correctly modelled. Generically, resources can be independent, which means that they do not depend on other resources, or dependent, requiring that a relation to other resources is assumed and taken into account. Typical examples of such resources are researchers and special labs that they can use which can be found in Naber and Kolisch (2014). These authors make a distinction regarding independent resources as "principal resources", that are independent resources that determine the use of the dependent ones, and (strictly) "independent resources" that are independent and do not interfere with other resources.

Resources can also be classified as continuous or discrete entities. Continuous resources are the ones that require a continuous variable to be modelled which is straightforward for e.g. power (electric, hydraulic, pneumatic or fuel) but can also be used to represent partially allocated discrete resources e.g. 40% of a machine is allocated to work in an activity. Discrete resources are then resources that are adequately represented by a discrete variable like (fully dedicated to an activity) persons or machines.

Regarding their type, if not stated otherwise, resources are discrete and, without loss of generality (w.l.o.g.), it will be assumed that they are renewable and independent. Therefore, K^R, a_k^R and r_{ikt}^R referred in expression (2.7) will be referred to as K, a_k and r_{ikt} respectively.

2.2.4 RCPSP

RCPSP stands for "Resource Constrained Project Scheduling Problem" and is the problem of finding a schedule that is simultaneously precedence and resource feasible with the objective of minimizing the project's durations. Projects are assumed to have deterministic activity durations which once started cannot be interrupted (non-preemptive) and all precedence relations are of type finish-to-start with no time lag (FS=0). This problem formulation plays an important role in project scheduling mainly due to the fact that this combinatorial problem is NP-hard in the strong sense as it was proven by Blazewicz, Lenstra and Kan (1983). This means that solutions to this problem are expected to be exponentially time consuming regarding the project's number of activities. However, even if the underlying model does integrate resource constraints, it still is a simplified model to most of real project scheduling problems, since it does not incorporate non-deterministic activity durations or more complex precedence relations, for instance. Even so, this approach is an important step in researching for better ways to establish schedules that are increasingly helpful in project management as will be detailed when describing RCPSP extensions and generalizations.

RCPSP being a NP-hard problem has been subject of extensive research. Besides some optimal solution approaches, several sub-optimal ones have been developed using new and adapted algorithms from other research fields. The major research lines will be presented starting with the optimal ones.

Optimal solution methods have a major drawback: only small project instances are assured to be solved to optimality within an acceptable timeframe, more explicitly, only project instances up to about 60 activities, which can only represent a small project (large projects typically have thousands of activities), can be solved by these methods. Interestingly, up to now the J60 test set, which is described in Kolisch and Sprecher (1997)[6], have not been fully solved to optimality.

Conceptually, RCPSP can be expressed by the following MIP (Mixed Integer Programming) formulation:

Conceptual Formulation

$$minimize\ (s_n) \tag{2.9}$$

$$subject\ to:$$

$$s_j \geq s_i + d_i, \qquad (i,j) \in A \tag{2.10}$$

$$s_1 = 0 \tag{2.11}$$

$$\sum_{i \in P_t} r_{ik} \leq a_k\ , \qquad t = 1, \dots, s_n, k \in K \tag{2.12}$$

where P_t is the set of the activities in progress at time instance t.

Expression (2.9) sets the objective function to minimize the start time of the dummy end activity which is the same as minimizing the project's duration, (2.10) ensures the precedence relations by constraining the start time of an activity to be greater or equal to the finish time of all its predecessors, (2.11) sets the project to start at time instant 0 and (2.12) ensures resource constraints. i.e., forces resource consumption to be within their availability.

---

[6] Available in http://www.om-db.wi.tum.de/psplib/getdata.cgi?mode=sm

Such conceptual formulation has no easy way to be implemented which led to the development of several new ones starting with the formulation of Pritsker, Waiters and Wolfe (1969), based on binary variables defined as:

$$x_{it} = \begin{cases} 1, & \text{if activity } i \text{ is finished at the end of period } t \\ 0, & \text{otherwise} \end{cases}, \qquad i \in V, t \in [EF_i, LF_i] \tag{2.13}$$

The formulation is then:

$$minimize \sum_{t=EF_n}^{LF_n} tx_{nt} \tag{2.14}$$

$$subject\ to\text{:}$$

$$\sum_{t=EF_i}^{LF_i} x_{it} = 1, \qquad i \in V \tag{2.15}$$

$$\sum_{t=EF_j}^{LF_j} (t - d_j)x_{jt} \geq \sum_{t=EF_i}^{LF_i} t_i x_{it}\,, \qquad (i,j) \in A \tag{2.16}$$

$$\sum_{i=1}^{n} \sum_{q=\max\{t,EF_i\}}^{\min\{t+d_i-1,LF_i\}} r_{ik}x_{it} \leq a_k\,, \qquad t = 1, \dots, T \text{ and } k \in K \tag{2.17}$$

$$x_{it} \in \{0,1\}, \qquad i \in V, t \in [EF_i, LF_i] \tag{2.18}$$

Expression (2.14) sets the objective function to minimize the finish time of the dummy end activity which is the same as minimizing the project's duration, (2.18) defines x_{it} variables to be binary and, in conjunction with (2.15), forces each activity to have a single finish time, (2.16) ensures the precedence relations by constraining the start time of an activity to be greater or equal to the finish time of all its predecessors and (2.17) ensures resource constraints forcing resource consumption to be within their availability.

Additional ones can be found in Klein (2000) including the ones proposed by Kaplan (1988), Alvarez-Valdés and Tamarit (1993) and Mingozzi, Maniezzo, Ricciardelli and Bianco (1998) or a more recent one proposed by Bianco and Caramia (2013) that have been further commented on by Naber and Kolisch (2013) that jointly proposed an alternative formulation (Naber, Kolisch, Bianco, & Caramia, 2014).

Another approach to solving RCPSP to optimality, that is widely proposed, is to use branch-and-bound based algorithms which are well suited to solve combinatorial optimization problems. This tree solution enumeration technique varies mainly in the search/branching strategy and on the pruning techniques that are used to narrow the necessary search steps that needed to be evaluated to reach an optimal solution, which include proper upper and/or lower bounds.

RCPCP branch-and-bound algorithms can be classified according to their branching schemes. The most relevant ones are:

- Precedence tree

  The strategy consists of scheduling at each step of the search tree an activity whose predecessors have all already been scheduled. Thus, each node of the search tree corresponds to a resource and precedence feasible partial schedule PS built in chronological order and to a univocally defined set of eligible activities $E(PS)$ with no stored information regarding unscheduled activities. On backtracking, a new activity is chosen, so that each leave corresponds to a schedule.

  Examples of this approach can be found in Patterson, Słowiński, Talbot and Węglarz, (1989) and Talbot (1982);

- Minimal delaying alternatives

  Each node of the search tree is associated to a feasible partial schedule PS and a time instant m. The following sets at time m are defined - the set $D(PS, m)$ of completed activities, the set $P(PS, m)$ of activities in progress and the set $E(PS, m)$ of eligible activities for scheduling, which are activities whose predecessors have all completed execution. Then, an attempt is made to schedule all eligible activities by adding them to the set of activities in progress. This can cause a resource conflict that is solved by branching with the withdrawal of the so-called delay alternatives from execution, which are activities in progress that, if removed, no resource conflict remains. A delay alternative (B) is called minimal ($\underline{B}$) if none of its proper subsets is still a delay alternative. Branching on minimal delay alternatives is sufficient to explore the whole search space.

This method differs from the precedence tree based one in two main aspects: a) the process branches on sets of activities and b) activities are tentatively scheduled at a search node and can be removed later from execution if resource conflicts are detected.

Examples of this approach can be found in Christofides, Alvarez-Valdésand Tamarit (1987) and Demeulemeester and Herroelen (1992);

- <u>Extension alternatives</u>

  The approach is similar to delay alternatives in that each search node corresponds to a partial schedule PS and a time instant t, for which sets $D(PS, t)$, $P(PS, t)$ and $E(PS, t)$ (set of activities with all predecessors in $D(PS, t)$) are identified. The current partial schedule is extended by starting a subset of the eligible activities (and not all of them as in delay alternatives) without violating the resource constraints. If there are activities in process ($P(PS, t) \neq \emptyset$), the empty set is always an extension alternative which must be tested in order to guarantee optimality, otherwise it can be disregarded.

  An example of this approach can be found in Patterson et al. (1989);

- <u>Minimal forbidden sets</u>

  This approach uses the concept of disjunctive precedence constraints which requires a specific activity i in a minimal forbidden set to be executed after at least one other activity j in the set: the specific activity j causing the delay may be left undecided until all start times are assigned. Disjunctive precedence constraints are based on the same idea of delay alternatives, but enable one to consider conflicts in non-chronological order. With this branching strategy, a search node corresponds to a fictitious schedule rather than to a partial schedule. A fictitious schedule assigns a provisional start time to each activity of the project. In case of a conflict, each branch is a possible disjunctive precedence constraint that resolves the conflict.

  An example of this approach can be found in Igelmund and Radermacher (1983).

As already mentioned, besides the search strategy, the branch-and-bound algorithm performance depends on how effective the algorithms' pruning or fathoming rules are, which should quit exploring a branch (a path) as quickly as possible but with the certainty that it does not lead to a better solution.

Some branch-and-bound algorithms have good performance when compared with MIP techniques, for example the new DH-procedure, as it is named in Demeulemeester and Herroelen (1995). This new

DH-procedure, which is an enhanced version of the DH-B&B procedure mentioned as the last example in the minimal delaying alternatives branching strategy (Demeulemeester & Herroelen, 1992), is the first to claim to have solved all 480 KSD[7] instances (Kolisch, Sprecher, & Drexl, 1995) to optimality.

Nevertheless, given the NP-hard nature of the problem, optimal solution algorithms are always slow and have nondeterministic performance behaviour. They are too slow to be applicable in general project scheduling tools, which require the development of faster algorithms,, justifying the vast research effort in finding suboptimal schedules. Some approaches propose to relax MIP formulations or just to truncate the branch-and-bound procedure, taking the best solution found so far.

Another approach proposes to use heuristic and meta-heuristic based scheduling algorithms.

Heuristic methods can be divided in two major groups:

- Constructive heuristics: Heuristic methods that build or construct a schedule from scratch;
- Improvement heuristics: Heuristic methods that take an existing schedule, which can be built with a constructive heuristic, and try to create a better one.

Constructive heuristics boil down to a scheduling scheme and a priority rule. A scheduling scheme determines the way or the steps to be followed in building the schedule. The priority rule determines the next activity to be scheduled within each step. A description and a computational analysis of these methods is available in Kolisch (1996b).

There are two main scheduling schemes:

- SSGS - Serial Scheduling Generation Scheme;
- PSGS - Parallel Scheduling Generation Scheme.

The SSGS method consists of n scheduling stages ($g = 1, \dots, n$), one per activity, in which an activity is selected from the unscheduled activities according to its priority, and is scheduled as soon as possible regarding the up to then partial schedule, respecting the precedence and resource constraints. To do this, two sets are defined:

- S_g - scheduled set: contains the already scheduled activities up to stage g;

---

[7] KSD stands for Kolisch, Sprecher and Drexl. The 480 KSD instances are currently part of the PSPLIB.

- D_g - decision set: contains the non-scheduled activities having all their predecessors already scheduled, i.e., all their predecessors belong to S_g.

  This set can be defined as $D_g = \{i \in V \mid i \notin S_g, Pred_i \subseteq S_g\}$.

When an activity is scheduled, it is added to the scheduled set and the decision set has to be updated accordingly (additional activities might have now all their predecessors scheduled).

The procedure ends at stage $g = n$.

Because an activity is scheduled at each stage, the method is considered activity-oriented and its complexity is $O(|K|n^2)$ where $|K|$ is the number of resource types used by the project. It is also worth mentioning that the method generates active schedules (schedules that do not allow any local or global left shifts) which mean that among their solutions there is the optimal one. With this regard Sprecher, Kolisch and Drexl (1995) provide formal definitions and detailed information.

On the other hand, the PSGS heuristic method consists of, at most, n scheduling stages ($g = 1, \dots, n$), in which a set of activities, that can be empty, is scheduled. Each stage corresponds to a time instant t_g such that if $g = a < g = b$ then $t_a < t_b$. In this procedure, three sets are considered:

- C_g - complete set: the set of already scheduled activities that have been concluded up to t_g.

  Can be defined as $C_g = \{i \in V \mid f_i \leq t_g\}$;
- A_g - active set: contains the active activities at t_g, i.e., scheduled activities still being executed.

  Can be defined as $A_g = \{i \in V \mid f_i - d_i \leq t_g < f_i\}$;
- D_g - decision set: contains the non-scheduled activities that are precedence and resource feasible if started at time instant t_g. Note that this set differs from the one defined for SGSS.

  Can be defined as $D_g = \{i \in V \backslash (C_g \cup A_g) \big| \, Pred_i \subseteq C_g \wedge r_{ik} \leq a_k - \sum_{j \in A_g} r_{jk}, k \in K\}$.

In each stage, the procedure selects a new t_g as the earliest finish time of all active activities, moving all activities finishing at t_g from the active set to the complete set. The complete set is updated to reflect the changes according to the priority rules. Then, all activities in the decision set are scheduled successively. The procedure ends when all activities are scheduled.

Because a time is selected at each stage to schedule a set of activities, the method is considered time oriented and its complexity is the same as the SGSS procedure, i.e., $O(|K|n^2)$. It is also worth mentioning that the method generates non-delay schedules (active schedules in which no resource is kept idle at a time when it could begin processing some activity), which means that the optimal one is not guaranteed to be within the possible solutions of the method. Again, Sprecher, Kolisch and Drexl (1995) provide formal definitions and detailed information.

Both scheduling schemes rely on a priority rule to decide which activity to schedule next which are also known as priority rule based methods.

As a result of applying a priority rule to the set of activities of a project, an ordered list is created, which should take into account the precedence relations, meaning that an activity cannot appear in the list before all its predecessors whatever the priority rule used.

Priority rules can be divided into some main categories according to the source of information they rely on, namely the ones based on:

- Activity: prioritize activities according to activity duration. Examples:
  - SPT - Shortest Processing Time;
  - LPT - Longest Processing Time.
- Network: prioritize activities according to the network structure. Examples:
  - MIS - Most Immediate Successors;
  - MTS - Most Total Successors;
  - LRNJ - Least Non Related Jobs.
- Resources: prioritize activities according to resources used. Examples:
  - GRWC - Greatest Resource Work Content;
  - GCRWC - Greatest Cumulative Resource Work Content.
- Schedule: prioritize activities according to the CPM schedule. Examples:
  - EST - Earliest Start Time;
  - EFT - Earliest Finish Time;
  - LST - Latest Start Time;
  - LFT - Latest Finish Time;
  - MSLK - Minimum Slack.

These rules are defined statically (completely defined before scheduling starts), but some rules can also be defined dynamically (can change during the scheduling process) and can also be made as a combination of the basic ones, which are known as composite priority rules. A study on priority rules can be found in Kolisch (1996a).

Constructive heuristic methods, as described, are said to be performed in forward planning due to the start to end planning direction used. The reverse direction (from end to start) can also be employed in what is called backward planning, by making the necessary adjustments (reversing precedence and priorities) and solving the question of where is the end time by using an upper bound for the project's duration and left-shifting all activities by the offset of the dummy start activity.

This approach can also be extended to the bidirectional planning by using a combination of both directions. To perform this, an additional rule is necessary to select in which direction an activity will be scheduled. Demeulemeester and Herroelen (2002) adequately describe these procedures.

But the drawback of such easy and fast methods is that they may increase a project's duration well beyond their optimal value. This can be critical because typically the optimal value is unknown, therefore there is no way to find out the quality of the solution that is obtained. An easy way to improve such solution is to systematically apply all combinations of scheduling schemes and priority rules and select the best one.

The described approach does not alter any of the schedules, it just picks the best. Alternatively, one may pick one, probably the best, and then try doing some changes that still produce a feasible schedule, i.e., a schedule that does not violate any precedence and resource constraints. If the new schedule is better, keep it, otherwise continue the search until some predefined condition is met. This is the basic behaviour of the improvement heuristic scheduling approach.

Improvement heuristics typically do not work on schedules but rather on representations of schedules, which are indirect ways to univocally represent a schedule. An example of such representations, which is in the sequel of the constructive heuristic description, is the activity list representation. In this approach, a schedule is represented by an ordered list of activities that respects precedence constraints, that is decoded into a unique schedule by the SGSS procedure. Additional examples of schedule representation approaches like "priority rule", "random key" or "shift vector" can be found in Kolisch and Hartmann (1999).

Improvement heuristics consist then in picking one or more feasible schedules in a suitable representation and applying improvement algorithms like neighbourhood operators or meta-heuristic based algorithms. Based on this approach, a vast set of algorithms are proposed by several researchers. Because this line of research is not directly related to the work being presented, only an enumeration of the main related topics will be made, instead of a more detailed presentation. An overview and a computational analysis of some of these methods can be found in Hartmann and Kolisch (2000) and Kolisch and Hartmann (2006).

Generically the following operators can be used, classified by the number of schedules they use as a starting point:

- Unary operators: Pair wise interchange, Shift and Change;
- Binary (crossover) operators: One point, Two points and Uniform.

And the following meta-heuristics:

- Tabu search;
- Simulated annealing;
- Genetic algorithms;
- Electromagnetic algorithm;
- Ant colony;
- Particle swarm;
- GRASP (Greedy Randomized Adaptive Search Procedure).

RCPSP takes various assumptions to simplify the (real life) problem, but at the same time makes it difficult to solve to optimality. Some sub-optimal techniques that allow RCPSP instances to be solved for realistic problems in realistic time frames were presented. But RCPSP also serves as a starting point to other problem formulations or the so called RCPSP related problems and extensions. Some among them present concepts that are related to this thesis research work:

- TCPSP (Time-Constrained Project Scheduling Problem): instead of a resource-contained problem that minimizes the project's makespan, the constraint here is the project's completion time or deadline. Cumulatively, resource constraints and overtime may be considered so that costs are minimized while still meeting the project's deadline;
- RLP (Resource Levelling Problem): the problem of levelling as much as possible the resource usage while executing a schedule. Here the objective is to minimize resource fluctuation using a suitable function like $\sum_{k \in K} c_k \sum_{t=1}^{\delta} r_{kt}^2$, where c_k is the cost of resource k, δ is the project's deadline and r_{kt} are the required resources of type k during the time interval t. Note that the deadline can assume the minimum value in which case it has to be previously calculated (as a standard RCPSP solution) or a multi-objective function can be used;

- RACP (Resource Availability Cost Problem): similarly, the problem of minimizing the cost of resource availability aims to reduce as much as possible the resources that have to be available so that the project meets a predefined deadline. Here the function to be minimized is $\sum_{k \in K} c_k a_k$, where a_k is the necessary availability of resource k;
- DTCTP (Discrete Time-Cost Tradeoff Problem): in this case activities within a project can be executed in several ways, which are characterized by the time they require to be executed and its, assumed to be non-decreasing, associated cost. The problem is then to determine a schedule that minimises the project's costs, limited by a due date by selecting each activity's durations. Typically, the duration of each activity is limited by a minimum (crashed duration) and a maximum (normal duration) value, which must be integers, otherwise the problem becomes continuous rather than discrete. De, Dunne, Ghosh and Wells (1995) presents a general overview of this problem while in Demeulemeester and Herroelen (2002) an in-depth overview of this and related topics can be found. Demeulemeester, Herroelen and Elmaghraby (1996) provide some exact methods to solve this problem;
- MRCPSP (Multi-mode RCPSP): similarly to the DTCTP, the MRCPSP model assumes each activity can be executed in one of several distinct modes and the objective is to select a mode for each activity such that the resulting schedule is precedence and resource feasible, and its duration is minimal. Contrary to the DTCTP, several resource types can be used and the trade-off is not limited to time-cost. It comes to no surprise that, mathematically, MRCPSP is a generalisation of DTCTP and also RCPSP. A MIP formulation was presented by Talbot (1982), based on binary (0-1) variables, similar to the ones presented for the RCPSP MIP formulation using start times instead of finish times.

  Such variables are defined as:

$$x_{imt} = \begin{cases} 1, & \text{if activity } i \text{ is performed in mode } m \text{ and started at time } t \\ 0, & \text{otherwise} \end{cases} \tag{2.19}$$

The formulation is then:

$$minimize \sum_{m=1}^{M_n} \sum_{t=ES_n}^{LS_n} tx_{nmt} \tag{2.20}$$

$$subject\ to:$$

$$\sum_{m=1}^{M_i} \sum_{t=ES_i}^{LS_i} x_{imt} = 1, \qquad i \in V \tag{2.21}$$

$$\sum_{m=1}^{M_i} \sum_{t=ES_i}^{LS_i} (t + d_{im})x_{imt} \leq \sum_{m=1}^{M_j} \sum_{t=ES_j}^{LS_j} tx_{jmt}\,, \qquad (i,j) \in A \tag{2.22}$$

$$\sum_{i=1}^{n} \sum_{m=1}^{M_i} \sum_{q=\max\{t-d_{im},ES_i\}}^{\min\{t-1,LS_i\}} r_{imk}x_{imq} \leq a_k\,, \qquad t = 1, \ldots, T \text{ and } k \in K \tag{2.23}$$

$$x_{imt} \in \{0,1\}, \qquad i \in V, m = 1, \ldots, M, t \in [EF_i, LF_i] \tag{2.24}$$

where m is the mode in which an activity is executed, M_i is the last mode of the possible modes of activity i and d_{im} is the duration of activity i being executed in mode m. The ES_i/LS_i are the CPM computed values for Early/Late Start time for activity i when executed in their smallest duration mode with the exception that for the backwards pass an upper bound of the project duration, when all activities are executed at their longest durations, is used.

Expression (2.20) sets the objective function to minimize the finish time of the dummy end activity, which is the same as minimizing the project's duration, which should have only a single mode of execution making the outer summation redundant, (2.24) defines x_{imt} variables to be binary and, in conjunction with (2.21), forces each activity to have a single finish time, (2.22) ensure the precedence relations and (2.23) ensures resource constraints, forcing resource consumption to be within their availability.

In this formulation only renewable resource constraints are considered.

The addition of the following constraint extends the model to also cope with non-renewable resource constraints (and consequently to doubly constraint resources):

$$\sum_{i=1}^{n}\sum_{m=1}^{M_i}\sum_{q=ES_i}^{LS_i} r_{imk}^{N} x_{imq} \leq a_k^N, \qquad t = 1, \dots, T \text{ and } k \in K^N \tag{2.25}$$

As a generalization of RCPSP, MRCPSP is NP-hard in the strong sense. Its proposed solution methods are vast and can be classified as exact or heuristic procedures. An overview of the exact procedures can be found in Hartmann and Drexl (1998). Again, exact procedures do not solve real size instances in acceptable times, which leads to the development of heuristic procedures, some of which are identified in Van Peteghem and Vanhoucke (2014);

- GRCPSP (Generalized RCPSP): RCPSP makes several assumptions that can be extended to cope with more complex project modelling. The following can be considered:
  - Generalized Precedence Relations: RCPSP considers that precedence relations are of the type FS=0, i.e., an activity can start as soon as all their predecessors have finished. This means that neither lag nor lead time between activities is considered. If such limitation no longer exists, the problem is named as RCPSP-GPR (RCPSP under generalized precedence relations). Such problem can be modelled by replacing constraint (2.16) by

$$\sum_{t=EF_j}^{LF_j} (t - d_j)x_{jt} - \lambda_{ji} \geq \sum_{t=EF_i}^{LF_i} t_i x_{it}, \qquad (i,j) \in A, \lambda_{ij} \in \mathbb{Q} \tag{2.26}$$

    where λ_{ij} is the generalized precedence relation value between activity i and j.

    Branch-and-bound algorithms to solve this problem can be found in De Reyck and Herroelen (1998, 1996) and Dorndorf, Pesch and Phan-Huy (2000);

  - Release and Due dates: the concepts of Release date (time instant from where an activity can start) and Due date (time instant where an activity is due) can be incorporated in RCPSP by adding the correspondent constraints as well as activity's deadlines;

- Variable resource availability: when resource availability varies during project execution, constraint (2.17) is no longer valid and is replaced by the following in GRCPSP:

$$\sum_{i=1}^{n} \sum_{q=\max\{t,EF_i\}}^{\min\{t+d_i-1,LF_i\}} r_{ik}x_{it} \leq a_{kt}\,, \qquad t = 1, \dots, T \text{ and } k \in K \tag{2.27}$$

  where a_{kt} is the availability of resource k at time instant t;

- Pre-emption: no pre-emption is allowed in RCPSP. If this generalization alone is allowed, i.e., if activities are allowed to stop while being executed and restart or resume execution later, it becomes a Pre-emptive RCPSP (PRCPSP). Demeulemeester & Herroelen (1996) present a branch-and-bound optimal solution.

Descriptions on GRCPSP and their solution methods can be found in De Reyck, Demeulemeester and Herroelen (2001).

Additional surveys on RCPSP and its (deterministic) variants can be found in Hartmann and Briskorn (2010) and Kolisch and Padman (2001).

In the following subsection, the RCPSP deterministic model will be combined with PERT like stochastic activity durations so that this stochastic effect can be taken into account along with resource limitations.

2.2.5 SRCPSP - Stochastic Project Scheduling

At the planning phase, activity durations are estimated so that a schedule can be established. Depending on the framework in which these activities will be performed (e.g. kind of work, resources used, organizational environment and other constraints) these estimations can be appropriately modelled by a deterministic value or may be subject to deviations. In the deterministic case, durations can be defined by the vector $d = \{d_1, d_2, \dots, d_n\}$, being all its elements deterministic variables. In the non-deterministic case, and when the pattern of such deviations is known, durations can be defined by the vector $D = \{D_1, D_2, \dots, D_n\}$, being its elements random variables. Projects modelled under this approach typically assume these random variable are independent with known probability distribution and assume only integer values. Each random variable D_i may be defined by a discrete set ($D_i = \{d_{i1}, d_{i2}, \dots, d_{|D_i|}\}$) or an interval ($D_i = \left[d_i^{min}, d_i^{max}\right] \cap \mathbb{Q}$) being each sample or realization denoted by the deterministic vector $d = \{d_1, d_2, \dots, d_n\}$. Note that if dummy activities are considered $D_{dummy} = \{0\}$ and therefore $\mathrm{P}\left[D_{dummy} = 0\right] = 1$, where $\mathrm{P}[e]$ is the probability of event e. It will be assumed that dummy start and dummy end activities do exist, so $\mathrm{P}[D_1 = 0] = \mathrm{P}[D_n = 0] = 1$ and also $\mathrm{P}[D_i < 0] = 0$.

Similarly to the deterministic case, project scheduling under the non-deterministic durations case can be done without considering resource constraints (as is the PERT case) or under resource limitations. The focus here will be on the resource constrained case but other approaches regarding the non-resource constrained case, besides PERT, do exist which can be further explored in Demeulemeester and Herroelen (2002).

Then, how to generate a project schedule in such conditions, i.e., how to schedule activities with unknown (stochastic) durations so that the project's duration is minimized and precedence and resource constraints are not violated? This is the problem addressed by the SRCPSP methodology which can easily be identified as a generalization of the RCPSP problem and is therefore of exponential complexity (NP-hard in the strong sense).

To handle the problem’s increased complexity, SRCPSP does not define a schedule at the planning phase (before project execution starts) but rather builds the schedule as a multi-stage decision process (while the project is being executed) so that the information made available meanwhile is taken into account. A major drawback can immediately be identified that is the absence of a planned or baseline schedule which can be useful for synchronization with outside (the project) processes like resource plans or stakeholders communication. Nevertheless, the model is effective in managing such projects if one can live with this limitation.

The general solution procedure for the SRCPSP is then to determine the actions that should be taken when certain events occur so that the scheduling objective is achieved. This scheduling strategy is called a policy and the typical meaning of the relevant concepts is:

- Event: end of some activities;
- Action: decide which activities to execute next;
- Objective: minimize the project duration (or more generically to minimize the project's costs).

In summary, SRCPSP can be defined as the problem of finding a policy Π that minimizes a project's expected cost. The project is defined by a set of activities V that have random durations D_i subject to a set of precedence constraints A and requires r_{ik}, from a set of K resources, which are constrained by their limited availability a_k ($i \in V, k \in K$).

Using the project duration to represent the project cost, the objective function to be minimized can then be expressed as:

$$\mathrm{E}[s_n(D, \Pi)] \tag{2.28}$$

where $\mathrm{E}[\cdot]$ is the expected value operator.

This means that the expected value of the start time of the dummy end activity is to be minimized given a vector of random activity durations executed according to a policy Π.

Other objective function of interest can be e.g. the service level ($max\ \mathrm{P}[s_n \leq \delta]$) or the overrun ($min\ \mathrm{E}[max(0, s_n - \delta)]$).

The duration vector D can be obtained e.g. by simulation and the goal, i.e., the solution is to select the best policy Π.

Several scheduling policies have been proposed in the existing literature among which are the class of preselective policies, presented by Igelmund and Radermacher (1983).

In the literature, the following classes/subclasses of policies are considered of interest:

- Priority policies: based on a priority list where at each decision time (event) as many as possible activities are started according to the priority list. This approach schedules activities by using a PSGS decoding strategy, which does not guarantee to achieve optimality;
- ES (early start) policies: based on the concept of minimal forbidden sets (already mentioned as the basic concept for one of the branch-and-bound solution procedures for RCPSP) and breaking them by adding precedence relations (i, j);
- PRS (preselective) policies: also based on the concept of minimal forbidden sets, but preselecting an activity to be delayed in each minimal forbidden set (similar to the minimal delaying alternative approach referred to as the basic concept for another of the branch-and-bound solution procedures for RCPSP). The representation of this policy by AND/OR (waiting) precedence constraints can be further analysed in (Stork, 2001);
- LIN (linear preselective) policies: this is a special PRS policy in which the priority list ordering ensures that the preselected waiting activity is the one with the smallest priority, i.e., in the priority list, a preselected activity to be delayed for each minimal forbidden set will always succeed the remaining activities in the set;
- ABP (activity based) policies: this policy, also known as job-based, is also of the preselective type in which a priority list can be used to determine the start sequence of activities, i.e., an

activity once selected is started as early as possible but always at or after the already started activities. The scheduling scheme is a modified version of the SSGS (also known as Stochastic SSGS) which guarantees that optimality can be reached (using the adequate list).

A general analysis on scheduling strategies can be found in Möhring, Radermacher and Weiss (1984, 1985).

Similarly to RCPSP, solution methods for the SRCPSP can be optimal, typically using stochastic programming or the branch-and-bound approach, or heuristic procedures.

SRCPSP is further analysed in Ballestín and Leus (2009) and Stork (2001) and recent additional policies are presented in Ashtiani, Leus and Aryanezhad (2008). Ballestín (2007) analyses SRCPSP under its suitability perspective.

The purely reactive or online approach of SRCPSP scheduling can be relaxed, with the so-called pre-processing techniques that use a priori information to make sequencing decisions before project execution starts, and complete the scheduling decision process later, while the project is being executed. Ashtiani, Leus and Aryanezhad (2011, 2008) developed examples of this approach.

2.3 State of the art

After presenting the basic concepts that are relevant to this research, some scheduling approaches that address the problem of dealing with uncertainty in project scheduling, will be presented. Uncertainty is being used until now in its common sense form which is referring to something that is not known, it can happen or not, it is uncertain or non-deterministic. This means that no characterization is being made about the inevitability of it happening or not, neither about the degree of that inevitability. There are several ways in which uncertainty can be defined but the one considered in decision theory (French, 1993) seems to be the most helpful in making decisions within the project management process (Artigues, Leus, & Talla Nobibon, 2013). It makes a distinctive definition of uncertainty or non-certainty as it is named as in the theory, according to what is known about the possible outcomes:

- Risk: the distribution of the outcomes is known with certainty;
- Uncertainty: the distribution of the outcomes is not known or is not meaningful. It is also referred to as unmeasurable uncertainty or non-measurable uncertainty;
- Ignorance: the outcomes are not known. It can also be called unawareness or unk unks (unknown unknowns).

The concept of uncertainty will be used in its wider sense or as non-certainty unless otherwise is stated. As was made clear previously, SRCPSP assumes that the distribution of the outcomes, i.e., the possible values and probabilities of the activities durations, are known, which will enable a predefinition of the adequate scheduling policy so that the project's cost (or duration) is minimized. If the project parameters (activities and resources) are unknown in advance, then the only possible scheduling approach is to just start activities as they are defined, i.e., a purely on-line scheduling model. In between lies the most realistic cases that the probabilistic distribution on problem parameters, including activity durations, is not fully known but (1) a set of scenarios can be established and/or (2) an initial estimation is possible but unexpected changes may arise like a change in scope with impact in activities duration (activity insertion/deletion in the extreme case). The first case can be handled with "Proactive scheduling", also known as "Robust scheduling", while the second can be dealt with "Reactive scheduling".

Before going into details about proactive/reactive scheduling a preliminary introduction of the concept of schedule robustness and its relation to schedule flexibility is required.

When assuming deterministic activity durations (and implicitly assuming deterministic project parameters), 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. An example of such compact schedule can be seen in Herroelen and Leus (2004), originally published in Wiest and Levy (1977).

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 activities' start time and (2) the objective function value while flexibility has to do with its capacity to be repaired.

Next, the proactive/reactive scheduling approach will be presented and its main concepts will be further detailed. Also in this section, an emerging technique that can increase schedule flexibility by using a more flexible model in resource allocation will be mentioned as well as other state-of-the-art work that is related to this research.

2.3.1 Proactive/Reactive Project Scheduling

Contrary to the RCPSP approach where a single schedule $S = \{s_1, \ldots, s_n\}$ is sufficient, the proactive/reactive approach may need more than one schedule. Therefore, it is helpful to identify some schedule types that help explaining proactive/reactive scheduling:

- Baseline schedule S^B: a schedule defined before the project starts that is the base for the project execution. Can also be referred to as pre-schedule, predictive-schedule or S^0;
- Initial schedule S^I: an existing schedule that can be used to generate a baseline schedule. It can be the deterministic schedule S or other non-robustly generated schedule;
- Realized schedule S^R: the actual realized schedule.

The following additional schedule types can also be mentioned because they are used in the remainder of this work:

- Working schedule S^W: a schedule being defined at execution time. Note that it is different from a SRCPSP schedule because the first is always a complete schedule in which some activities may already be terminated, while the latter is a partial schedule that is only completed when the last ending activity is concluded;
- Projected schedule S^t: similar to a working schedule but is defined for each time $0 < t < T$. This type of schedule will be detailed later within the Critical Chain scheduling methodology;
- Ex-post schedule S^*: a virtual schedule (not to be used directly or indirectly for activity execution) that is obtained as if full information about the realized activity durations would have been available before the project execution started. It can be used to compare the scheduling decisions made (having past information only) with the ones that could have been made if full information (including future one) was available.

Proactive scheduling is about generating the best possible robust baseline schedules, therefore they have to be optimized according to some measures that quantify their robustness. Several robustness measures exist of which the most typical ones (Herroelen & Leus, 2004) can be classified as:

- Quality robustness: a schedule's insensitivity to disruptions regarding its solution value (its performance);

- Solution robustness: a schedule's insensitivity to disruptions regarding its solution (the verified difference from the baseline schedule and the realized one).

Similarly to the SRCPSP objective functions, quality robustness can be defined for the objective function considering e.g. their expected value ($\mathrm{E}[s_n]$) or the probability that a certain goal will be reached ($\mathrm{P}[s_n \leq \delta]$).

Solution robustness boils down to measuring the distance from the baseline schedule to the realized schedule. It can be expressed in several ways among which some of the ones identified by Van de Vonder (2006) in his PhD thesis can be used as examples:

- The weighted sum of all absolute deviation between planned and realized activity start times:

  $$\Delta(S^B, S^R) = \sum_{i \in V} \omega_i \left| s_i^B - s_i^R \right| \tag{2.29}$$

  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 that can represent the unitary cost of such deviation;

  Note that the considered cost is equally penalizing for overruns as it is for underruns which means that this model favours schedule stability.

  Note also that in stochastic environments activity start times are better modelled as stochastic variables which lead to the following objective function to be minimized:

  $$\sum_{i \in V} \omega_i E\left[\left| s_i^B - s_i^R \right| \right] \tag{2.30}$$

- Alternatively, solution robustness can be measured by considering only the maximal deviations $\max_I \Delta(S^B, S^R)$ which has to be minimizing over a set of execution scenarios I.

These measures rely on the realized schedule S^R which implies that they must be calculated using a set of scenarios, either predefined or obtained by simulation.

Proactive scheduling can be performed as a single step procedure or as a two step procedure. In the first case, all foreseeable scenarios are taken into account in a holistic procedure to establish the best solution (a baseline robust schedule) to the scheduling problem, being the best solution the one that maximizes the defined robustness measures. In the second, a first step is used to establish a schedule that does not take uncertainty into account, typically a RCPSP like solution and then, at a second step, the previously defined schedule is enhanced so that it can better cope with uncertainty, which is

achieved, again, by maximizing the defined robustness measures. The underlying objective function that is a combination of the project's objective (e.g. minimizing costs or makespan) and an assurance that it will be achieved (in the form of a robustness measure) is used as such (combined) in the single phase case and is spitted in the two phase approach. This latter case directly exposes the potential consequences on the project's objective in introducing robustness into the schedule.
Examples of proactive scheduling methods include the ones based on the:

- Robust resource allocation principle as the method proposed by Leus and Herroelen (2004) which forces individual resource units to minimize the number of activities to which they are allocated, restraining thus the impact of their failure or MABO (myopic activity-based optimization) proposed by Deblaere, Demeulemeester, Herroelen and Van de Vonder (2007);
- Time buffering techniques as the Critical Chain methodology or the STC (Starting Time Criticality) heuristic procedure which is presented in Van de Vonder, Demeulemeester and Herroelen (2007) and Van de Vonder, Demeulemeester, Leus and Herroelen (2006). Other exact and suboptimal procedures are described in Van de Vonder (2006).

More recently, Lamas and Demeulemeester (2014) developed a proactive scheduling procedure that does not rely on a reactive procedure to evaluate its robustness (it is a purely proactive procedure).

Reactive scheduling is required whenever disruptions occur that have an impact on the current schedule, which can be seen as a disruption management process. Contrary to proactive scheduling, reactive scheduling is a multi-stage process parallel to project execution. This process is similar to the SRCPSP and other purely reactive scheduling processes. This reactive scheduling process, the repairing multi-stage process done on a baseline schedule, is also called predictive-reactive scheduling, whereas the purely reactive can also be named as on-line or real-time scheduling.
Reactive scheduling procedures can range from simple schedule repairs such as right-shifting activities to full rescheduling. While the first approach has very limited capacity to cope with uncertainty (without compromising the solution) the latter tends to generate instability or nervousness.

To solve this problem regarding time uncertainty, sampling and WET (Weighted Earliness-Tardiness) procedures were proposed by Van de Vonder, Ballestín, Demeulemeester and Herroelen (2007) and Lambrechts (2007). Resource uncertainty reactive exact and heuristic procedures were proposed by Lambrechts, Demeulemeester and Herroelen (2007).

Besides schedule repair and full rescheduling, the following approaches can also be used as reactive scheduling techniques for which additional description and references can be found in Herroelen and Leus (2004):

- Contingent scheduling: when the schedule is manually changed during project execution. The goal is to develop algorithms that help in establishing several equally good solutions which can be switched to at project execution time without losing performance;
- Activity crashing: the forced decrease in activity duration, with increasing costs. Can be seen as a reactive multi-mode approach to overcome disruptions;
- Fast tracking: enabling some parallel execution of precedence related activities by relaxing the FS=0 (no lag nor lead time in precedence relations), that may be possible by rearranging activity internal execution or that might become possible at project execution time, may result in solving disruptions in the schedule.
- Sensitivity analysis: the "what if" applied to reactive schedule can help determining limits and consequences on project parameters changes.

Only single-mode scheduling is being considered so far but an obvious approach to cope with deviations in a multi-mode context is to switch activity execution modes so that deviations are absorbed.

As a closing remark, it should be clear that Proactive and reactive scheduling methods are not necessarily alternative approaches. They can be combined by establishing a robust baseline schedule, i.e., a schedule that is feasible in all foreseeable scenarios and, if it becomes unfeasible due to unforeseeable conditions, make use of reactive techniques so that it can be efficiently repaired.

Table 4 puts into perspective the possible approaches to scheduling when projects face uncertainty.

Table 4: Project scheduling methods under uncertainty

Proactive phase (Before project start)	**Reactive phase (During project execution)**
None (No baseline schedule)	Dynamic scheduling (scheduling policies)
Deterministic scheduling (No anticipation of variability) Proactive (robust) scheduling: - Single step; - Two steps. - Robust resource allocation; - Time buffer insertion; - Resource buffers.	None (No variation occurs) Reactive scheduling: - Schedule repair; - Full reschedule; - Contingency scheduling; - Activity crashing; - Fast tracking; - Sensitivity analysis.

Several strategies and algorithms were proposed to maximize the schedule stability or the schedule robustness, minimizing the project's makespan or the project's cost. While some aim for optimality, others will settle for "good enough solutions". One should mention two alternative methodologies that can be a basis for these algorithms: the railway scheduling and the roadrunner scheduling (Van de Vonder, Demeulemeester, Herroelen, & Leus, 2005). Railway scheduling always starts activities at their scheduled start time or later while the roadrunner approach will always start activities as soon as possible. The first favours schedule stability (don't start earlier than scheduled because that unnecessarily messes with the schedule) while the latter is defensive regarding the project's makespan (do not miss the opportunity to gain some additional slack time). Tian and Demeulemeester (2010) argued that the roadrunner methodology does not reduce the project's expected makespan, as might seem empirically clear, and analysed the interaction between railways scheduling and resource flow (Tian & Demeulemeester, 2012).

Critical Chain Project Management (CCPM) is a methodology that is related to this line of research that should be mentioned. The CCPM method (Goldratt, 1997), derived from the Theory of Constraints management paradigm (Goldratt & Cox, 1984), which is a well-known and a widely used method with a tool, ProChain (Newbold, 2008), that facilitates its practical use by project managers. CCPM simplifies the uncertainty problem by focusing on the Critical Chain (CC) that is the longest chain (path) of activities that are precedent and resource dependent in the schedule, i.e., that defines the project's duration. This chain is to be protected in disregard of the others, even if they are marginally not selected as CC. Time buffers are concentrated into Feeding Buffers (FB) and Project Buffers (PB). Simplistic FBs are inserted whenever a non CC activity meets the CC, protecting the CC from delays coming from that chain. PB are inserted immediately before the last (dummy) activity in order to protect the project's due date. Time buffers (FB and PB) are usually set at 50% of the duration of the chain they are inserted to (note that the project makespan is determined by the overall duration of the CC). This 50% buffer size rule does seem baggy and should take into account other resource, activity and project characteristics. CCPM also uses Resource Buffers (RB) that mainly serve as a warning system and are inserted when an activity in the CC uses a different resource from the previous activity. It also relies on Buffer Management (BM) to act as a proactive warning mechanism and uses the roadrunner scheduling methodology.

Several authors, e.g. Herroelen and Leus (2001), criticize the feasibility orientation of CCPM in disregard to optimality which can be critical in highly competitive markets (as are globalised markets) especially regarding large projects.

An extensive presentation of Proactive/Reactive scheduling methods and their applicability regarding distinct project management scenarios can be found in Demeulemeester and Herroelen (2009).

2.3.2 FRCPSP

The RCPSP 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 $k \in K$. This assumption is still valid if fluctuations in resource consumption exist within an activity but are not relevant for the remaining ones. In this case, the activity resource consumption can be adequately modelled by the activity's average resource consumption $r_{ik} = (\sum_{t=s_i}^{f_i} r_{ikt})/d_i$. The first difficulty comes from the fact that there is no guarantee that the value will be integer, which if not, invalidates it as a valid RCPSP model. Moreover, there are cases 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 RCPSP with Flexible resource profiles problem. It can be described as the problem of finding a precedence and resource feasible schedule of minimal duration subject to resource limited availability. The resource profile, i.e., the distribution of resource consumption per time period within an activity, can also be restricted. The solution to this problem will be to determine the start time and the resource profile of each activity in the project. Published research results on FRCPSP also assume that activity duration is not set, being part of the problem to be solved. Instead, the total resource consumption for each activity is used, which is also referred to as the work content ω_{ik}.

FRCPSP is a generalization of the RCPSP, which means it is also NP-hard in the strong sense and, due to the additional constraints, it is hard to model. This is a reason why the available literature is small if compared to the existing one regarding RCPSP and other RCPSP based models. This problem was initially addressed by Kolisch, Meyer, Mohr, Schwindt and Urmann in 2003, with a publication in the English language in Kolisch and Meyer (2006) to model the selection and scheduling of pharmaceutical research projects. The term RCPSP-FWP (RCPSP with Flexible Work Profiles) is used by Ranjbar and Kianfar (2010) with the same meaning as FRCPCP in their proposed solution, using a genetic algorithm. Fündeling and Trautmann (2010) use the term "work content constraints" to denote FRCPSP in their presentation of a priority rule based heuristic solution to the problem. Baumann and Trautmann

(2013) used the same terminology to present an MIP formulation that they solved using a commercial MIP solver (Gurobi 5.5). Another heuristic solution based on a genetic algorithm is proposed in Tritschler, Naber and Kolisch (2014) while Rokou, Dermitzakis and Kirytopoulos (2014) proposed an ant colony based heuristic to deal with a multi-project generalization of this problem.

Naber and Kolisch (2014) presented several MIP formulations for the problem and compare them by solving the models using a commercial MIP solver (IBM ILOG CPLEX 12.5) on a test set of instances, using pre-processing and priority-based heuristic methods to predefine solution bounds for the solver to use. They concluded that the so called "variable-intensity-based" model presents the best solution being also the one whose solution is computed the fastest. This formulation is based on the RCPSP formulation proposed in Bianco and Caramia (2013). In spite of the problem hardness, recent advances in commercial available solvers and the increase in computational capacity enabled these formulations to be solved but still only for small instances making it difficult for them to be used in practice.

2.3.3 Stochastic work content

The optimization of resource allocation in projects, considering stochastic work contents, was first addressed by Tereso in (2002), considering a single resource with unlimited availability. Two models were developed, one using Dynamic Programming (DP) (Tereso, Araújo, & Elmaghraby, 2004; Tereso, Mota, & Lameiro, 2006) and the other using the Electromagnetism like Mechanism (EM) (Birbil & Fang, 2003; Tereso, Novais, Araújo, & Elmaghraby, 2009). Next an Evolutionary Algorithm was used (Tereso, Costa, Novais, & Araújo, 2007) with better results than the DP model but similar to the EM model. The goal of these models was to decide on the amount of resources to allocate to each activity, considering the activities would start as soon as possible, minimizing the total cost, considering the cost of the resources and the cost of lateness.

Then this problem was further studied considering multiple resources (Moutinho & Tereso, 2014, 2015). In their work the authors addressed the stochastic multi-mode resource-constrained project scheduling problem in the case where the project activities require multiple renewable resources, constrained by total availability. The uncertainty was represented using a known distribution for the work content of each required resource within each activity. The objective was to determine the start times and the mode (defined by the resource allocations) which minimize the expected total project cost, evaluated as the sum of the resource allocation cost with the tardiness cost or earliness bonus, in case the project finished after or before the due date, respectively. The method used to solve the problem was based on two global optimization metaheuristics - the electromagnetic-like mechanism and the evolutionary algorithm.

In this line of work, a model was proposed by Elmaghraby and Morgan (2007) using a combination of Geometric Programming (GP) methodology with Sample Path Optimization (SPO). The authors aimed to extend the applicability of "resource allocation in activity networks under stochastic conditions" to large activity networks, i.e., large projects.

Godinho and Branco (2012) also proposed an adaptive model for multi-mode project scheduling under uncertainty. They assumed a due date for concluding the project and a tardiness penalty for failing to meet this due date, and that several distinct modes could be used to undertake each activity. They defined scheduling policies based on a set of thresholds where the starting time of each activity was compared with those thresholds in order to define their execution mode. The proposed procedure for choosing a scheduling policy was also based on the electromagnetism heuristic.

2.3.4 Other related work

There are some other studies worth mentioning since they dwell also around similar objectives as this thesis, 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 minimizing the project duration with the maximization 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 and presented two alternative ones based on the same approach that (1) maximizes the minimum of slacks and (2) maximizes the minimum of the ratio slacks/duration. Hazır, Haouari and Erel (2010) followed the same approach to solve the DTCTP (Discrete Time-Cost Trade-off Problem), which is a sub-problem of the MRCPCP, by using several (seven) slack based enhanced robustness measures, a critical path based and a project buffer size based one. 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 schedules robustness by using a slack based objective function but did not foresee the use of such slack in conjunction with resource flexibility which is the primary innovation presented in this thesis.

As in other areas of research, different research trends have been appearing and fading. "Agile Project Management" (Cobb, 2015), "Lean Project Management" (Ballard & Howell, 2003) and "Extreme Project Management" (DeCarlo, 2004) are some of the current research trends in project management. There are alternative ways to envision project management evolution like the one mentioned in Garel (2013) which is mainly concerned with the standardization process. As an example of the impact that standardization can have in project management, the study by Milosevic and Patanakul (2005) can be mentioned in which they conclude that standardization may be important but should be applied with flexibility.

2.4 Conclusion

In spite of these techniques, examples of projects with budget overruns and delays well beyond their promised delivery dates are countless, due to several reasons not the least important of which is poor planning and control (Couto & Teixeira, 2007; Couto, 2012). The Standish Group's report (2009) shows a disturbing projects success rate, with 32% of all projects succeeding, 44% being late, over budget, and/or with less than the required features and functions and 24% failing (cancelled or never used). In the last years, project failures do not tend to decrease (The Standish Group, 2014). Complex projects are normally performed in dynamic environments characterized by uncertainty and risk (Schatteman, Herroelen, Van de Vonder, & Boone, 2008). It is believed that the use of specific models designed to address these concerns would contribute to a more efficient use of the resources while keeping the risk controlled, particularly in large and complex projects, enabling an increase in project success rates.

Two aspects stand out as crucial to the successful adherence to budgetary and time constraints: the proper allocation of the resources and the explicit recognition of the stochastic nature of the undertakings.

As explained, there are several possibilities to be explored within these two lines of work: a combination of the resource allocation problem considering stochastic work contents and multimodal activities with the proactive/reactive techniques, being the driver of this research project, certainly is a challenging one. Nevertheless the belief that this combination is possible and that it will enable a better project management tool, in that it can deal with uncertainty within certain boundaries while keeping the duration of the project, makes this challenge worthwhile.

3. PROBLEM DESCRIPTION

As described in the previous chapters, 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 to commercial use due to their computational hardness and their complexity in modelling day-to-day projects (Demeulemeester & Herroelen, 2009). On the other hand, more elaborated resource allocation techniques, like FRCPSP (RCPSP with Flexible resource profiles), that can model the increasing need to have flexibility in resource usage, are being studied (Naber & Kolisch, 2014; Rokou et al., 2014; Tritschler et al., 2014). These techniques usually establish an *a priori* (before the project starts) flexible resource allocation in the sense 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 techniques, the proposal made here is to assume that projects should start with a stable and constant plan, i.e. a deterministic baseline schedule, but the plan should cope as much as possible with uncertainties. However, in this thesis, 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 (i.e. the activity slack) which, in conjunction with resource flexibility, can accommodate eventual increases in activity durations (i.e. uncertainty).

3.1 Problem definition

The problem is then how to transform a given schedule into a more robust one in the sense that the resulting one 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 a minimum deviation in 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 (as the ones described in previous chapters), 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 activities' 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 can assume distinct "forms" but for now it is assumed 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.

Consider the following definitions:

- a_k^{nom}: the nominal resource availability of type k (usually referred as a_k that has a constant value per project);
- α_k^-: the negative resource flexibility (maximum percentage decrease of $a_k$from nominal);
- α_k^+: the positive resource flexibility (maximum percentage increase of $a_k$from nominal);
- a_k^-: Minimal resource availability of type k;
- a_k^+: Maximal resource availability of type k.

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:

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

In this expression, a_k is the effective resource availability of resource k, which can vary between the defined values when allocated to a specific project's activity.

The minimal and maximal value for a_k can also be defined by the following expressions, respectively:

$$a_k^- = a_k^{nom}(1-\alpha_k^-) \tag{3.2}$$

$$a_k^+ = a_k^{nom}(1+\alpha_k^+) \tag{3.3}$$

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 $\alpha_k^- = 25\%$ and

$\alpha_k^+ = 25\%$, the effective resource unitary availability can assume a value in the interval defined by $(1 - 0.25) \leq a_k \leq (1 + 0.25)$. Considering a_k to be continuous ($a_k \in \mathbb{R}$), it results in $a_k \in [0.75,1.25]$.

If the model has to ensure that the project required resources are kept constant, i.e., the increase in the demand of some activities must be compensated by the decrease in the demand of some others, the following condition has to be respected:

$$\sum_{i \in V} d_i r_{ik}(S^w) \leq \sum_{i \in V} d_i r_{ik}(S^b), \qquad for\ all\ k \in K \tag{3.4}$$

where S^b is an established baseline schedule (before the project starts) for a project with a set V of activities each denoted as i, S^w is any working schedule (a modified version of the baseline schedule) and d_i represents the duration of activity i while $r_{ik}(S)$ represents the resources required by activity i of resource type k when executed according to a schedule S.

If the model can be extended to assume that all resources are kept available to the project during its execution time, the condition to be respected is then:

$$\sum_{i \in V} d_i r_{ik}(S^w) \leq T^w . a_k^{nom}, \qquad for\ all\ k \in K \tag{3.5}$$

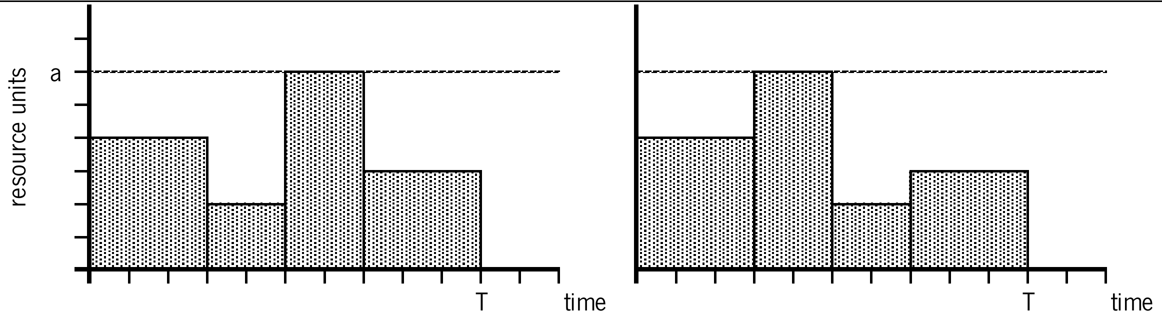
where T^w is the project duration when executed according to a schedule S^w which, in this case, should be the same as when the project is executed according schedule S^b ($T^w = T^b$).

The previous condition assumes that the model is limited to a constant resource allocation per activity, as in the MRCPSP model. If the resource allocation per activity can vary from time period to time period, as in the FRCPSP model, the condition to be respected is then:

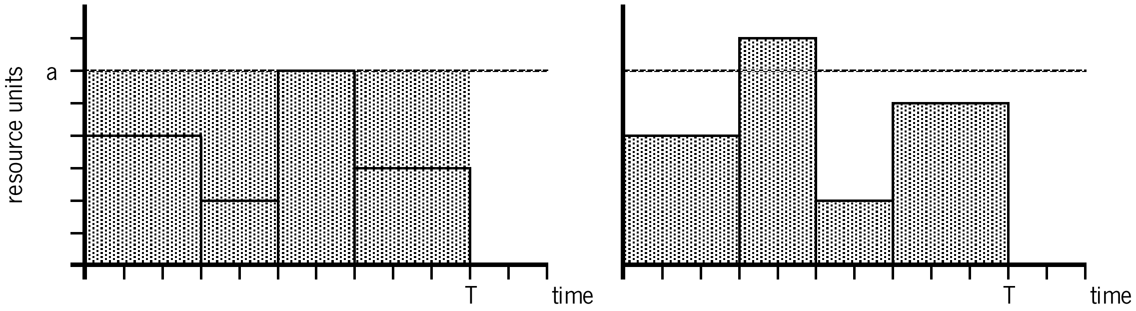
$$\sum_{i \in V} \sum_{t = s_i}^{s_i + d_i - 1} r_{ikt}(S^w) \leq T^w . a_k^{nom}, \qquad for\ all\ k \in K \tag{3.6}$$

where s_i is the start time of activity i and r_{ikt} represents the resources allocated to activity i of resource type k at time instant t.

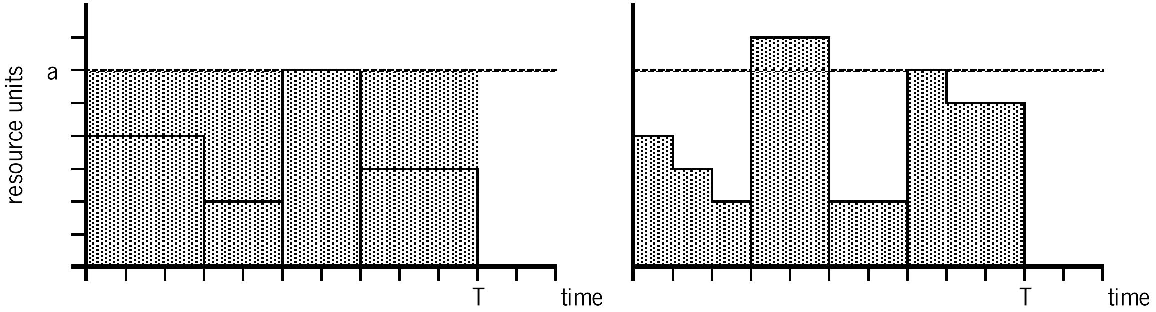
Figure 4 puts these conditions into perspective.



a) Baseline schedule and working schedule having the same resource demand (dotted areas)



b) Resource demand of a working schedule does not exceed resource availability for the duration of the baseline schedule



c) Same as b) but allowing flexible resource profile (FRCPSP)

Figure 4: Example a) for conditions (3.4), b) for condition (3.5) and c) for condition (3.6)

Baselines schedules are on the left side while working schedules are on the right

Baselines schedule dotted area must be respected by working schedules dotted area

In the remaining of this chapter, the model that complies with the constraints imposed by condition (3.4) will be explored further using an example.

3.2 The model

Consider the project that is represented in the network of Figure 5. To avoid overloading the example and still cover all relevant cases, the project example has 14 activities including the dummy start and dummy end ones.

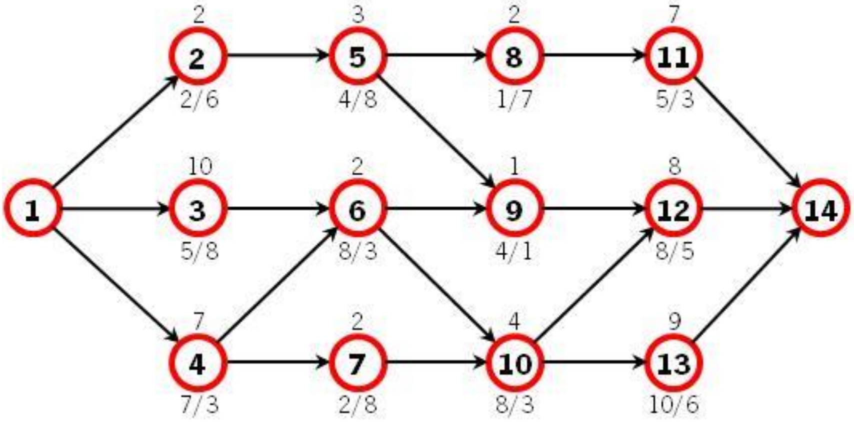


Figure 5: Project example with 14 activities and 2 resource types in AoN format

Above the nodes identifying each activity its duration (d_i) is presented and below are the resource requirements for each of the two resource types (r_{i1}/r_{i2}). Assuming that the resource availability is $a_1 = 14$ and $a_2 = 10$, an optimal solution for the RCPSP problem (minimal project makespan) is obtained with a project duration of $T = 45$. Figure 6 shows the schedule's representation in Microsoft Project 2013 (MSProject). In the chart, bold numbers indicate the activity number and italic characters indicate resource consumption per activity in the format:

resource1[consumption];resource2[consumption] (consumption is 1 if *[consumption]* is omitted).

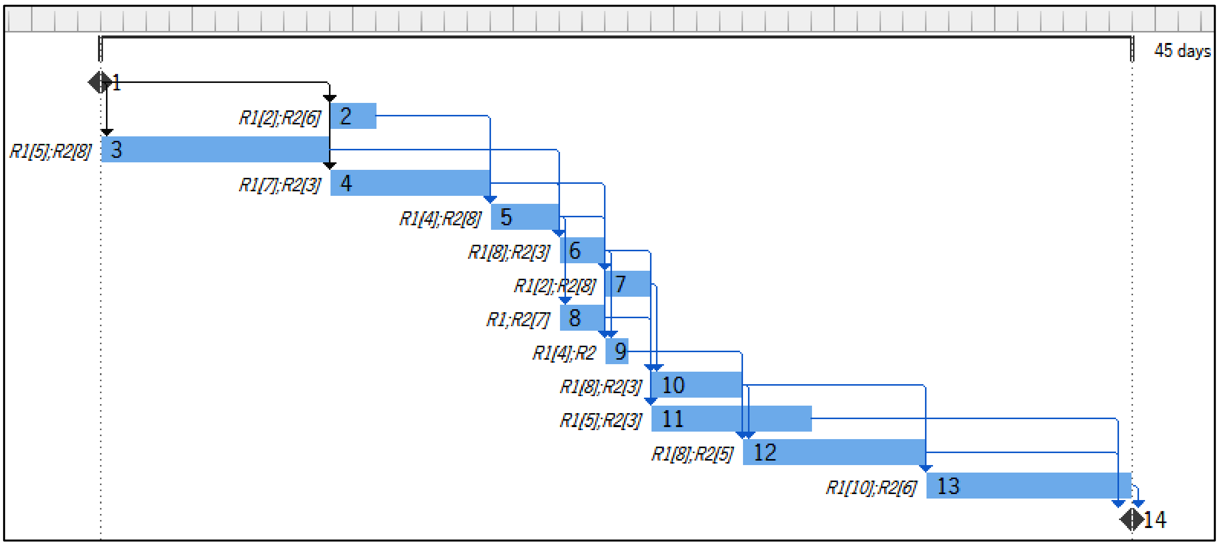
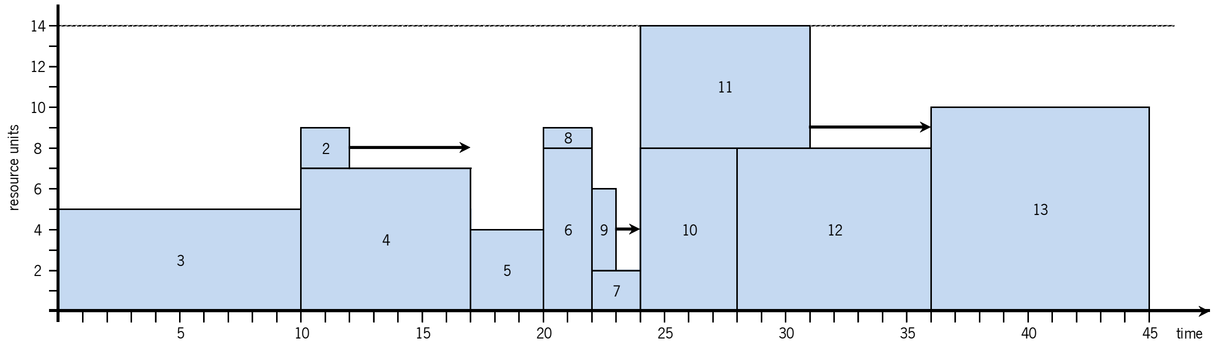


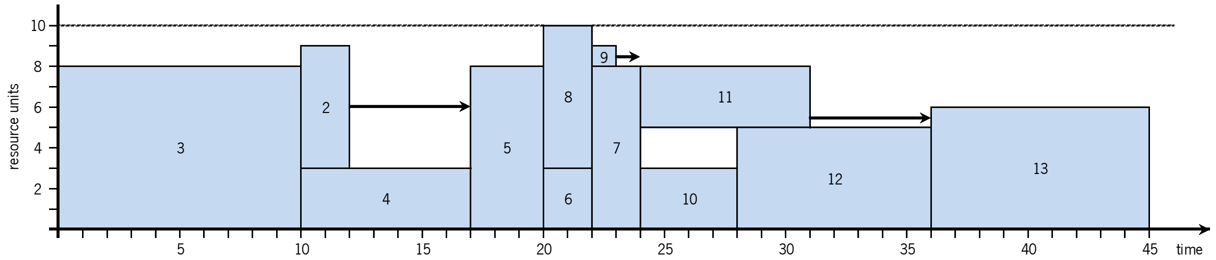
Figure 6: Gantt chart for minimal makespan baseline schedule (representation in MSProject)

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 7 shows the corresponding resource profiles for a) resource type 1 and b) resource type 2.



a) Resource 1



b) Resource 2

Figure 7: Resource profile for optimal baseline schedule: a) for resource 1 b) for resource 2

3.2.1 Capacity

With this type of chart it is easier to identify the unused capacity per resource, the 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.

In this context, the unused resource capacity of resource type k at time instant t can be defined as:

$$u_{kt} = a_k - \sum_{i \in P_t} r_{ik}, \qquad for\ each\ k \in K, t \in [0, T] \tag{3.7}$$

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

Graphically, it corresponds to the area above the stacked activities' resource profile and below the line that denotes the resource availability (the dashed lines in Figure 7). Note that the bars in the charts are not fully stacked which becomes evident in chart b) with activities 10 and 11. This option is used to enhance the perception of each individual activity. The relevant part of Chart b) (from t=22 to t=36) is displayed in fully stacked format (no vertical gaps between bars) in Figure 8 which enhances the perception of the unused resource capacity.

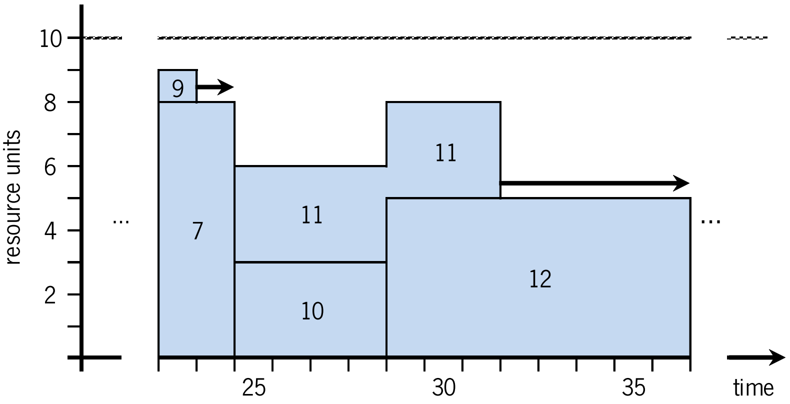


Figure 8: Fully stacked resource profile example

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.

$$\begin{gathered} slack_{ik} = \max\left(\tau \in \left\{\left[0, \min_{j \in Succ_i}\left(s_j\right) - s_i - d_i\right] \cap \mathbb{N}\right\}\right) \quad | \\ r_{ik} \leq a_k - \sum_{h \in P_t} r_{hk}\,, \quad t = s_i + d_i, \dots, s_i + d_i + \tau, \quad for\ each\ k \in K, i \in V \end{gathered} \tag{3.8}$$

where $Succ_i$ is the set of successor activities of i and P_t is the set of activities in progress at time instant t.

This definition is made per resource type. If resources are independent, each one can be managed separately that is to say the analysis explained here can be made for each resource type given that the baseline schedule is maintained.

One can explore further the case where, regardless of resources being dependent or independent, they must comply with the activity duration or, more precisely, they must be executed within each activity at the same rate (or pace) as 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:

$$slack_i = \min_{k \in K}(slack_{ik}), \qquad for\ each\ i \in V \tag{3.9}$$

The activity slack is illustrated in Figure 7 as a straight arrow beginning in the activity finish time and with a length corresponding to the activity's slack. According to the definition (3.9), their length is equal on both resource charts.

In Table 5 the calculated values for slack are presented 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.

Table 5: Example slack values

$\boldsymbol{i}$	$\boldsymbol{s_i}$	$\boldsymbol{d_i}$	$\boldsymbol{f_i}$	$\boldsymbol{slack_{i1}}$	$\boldsymbol{slack_{i2}}$	$\boldsymbol{slack_i}$
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

If within the scope of this project resources are flexible in terms described in (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 the resource flexibility is again bounded by $\alpha_k^- = 25\%$ and $\alpha_k^+ = 25\%$, the effective resource unitary availability is $0.75 \leq a_k \leq 1.25$. These values can now be applied to the activity durations to determine the duration span for each activity. The results are presented in Table 6 where:

- $d_i^{nom} = d_i$ (rate=100%): the nominal duration equals the initial (deterministic) duration d_i;
- $\overline{d_i} = d_i^{nom}/(1 - \alpha_k^-)$: the maximal activity duration is its duration when it is executed at its slowest rate (minimal resource unitary availability). In this case the minimal rate=75%;
- $\underline{d_i} = d_i^{nom}/(1 + \alpha_k^+)$: the minimal activity duration is its duration when it is executed at its fastest rate (maximal resource unitary availability). In this case the maximal rate=125%.

Obviously $\overline{d_i}$ and $\underline{d_i}$ can assume non-integer values which are neither practical nor commonly used values for activities' durations. Therefore their corresponding limit integer values are considered (which are also presented in Table 6):

- $d_i^{max} = \lfloor \overline{d_i} \rfloor$: the maximal integer activity duration (largest integer not greater than $\overline{d_i}$);
- $d_i^{min} = \lceil \underline{d_i} \rceil$: the minimal integer activity duration (smallest integer not less than $\underline{d_i}$).

Table 6: Possible duration span

i	**100%**	**75%**		**125%**	
	d_i^{nom}	$\overline{d_i}$	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

When distinct values of α_k^-, α_k^+ are considered for distinct resources, the most demanding ones have to be used to determine d_i^{max}, d_i^{min}. These α^-, α^+ are the α_k^-, α_k^+ that are closer to zero.

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:

$$d_i^{nom} \geq \frac{1}{\alpha^-} - 1 \,, \qquad \alpha^- \in \mathbb{R}_0^+ \tag{3.10}$$

$$d_i^{nom} \geq \frac{1}{\alpha^+} + 1 \,, \qquad \alpha^+ \in \mathbb{R}_0^+ \tag{3.11}$$

Understanding these expressions might be easier if written in the following form for which proofs are presented in Appendix I

$$d_i^{max} \geq d_i^{nom} + 1 \quad if\ \alpha^- \geq \frac{1}{d_i^{nom} + 1} \,, \qquad \alpha^- \in \mathbb{R}_0^+ \tag{3.12}$$

$$d_i^{min} \leq d_i^{nom} - 1 \quad if\ \alpha^+ \geq \frac{1}{d_i^{nom} - 1} \,, \qquad \alpha^+ \in \mathbb{R}_0^+ \tag{3.13}$$

Nevertheless, it is not at all mandatory that these criteria are met. Resource flexibility is more dependent on the kind of resources and on the organization that they are inserted in. These criteria only mean that extra care should be taken with activities that have "small" durations, as defined by expressions (3.10) and (3.11), because they will not participate and therefore will not take advantage of this methodology (and not because they have any harmful effect).

The above expressions in fact quantify the term "small" duration as a function of each of the resource flexibility parameters α^- and α^+.

The possible activity duration interval, as described here, 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 (whatever the reason) will never have an impact on the remaining activities, that is to say, it will not alter the schedule besides the finishing time of the activity in consideration. This means that the minimal duration of an activity is not affected by the schedule used to execute the project.

On the other hand, allowing an activity's 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 (3.14). Expression (3.15) defines the corresponding minimal duration variable d_{iS}^{min} that emphasizes the fact that minimal durations are not schedule dependent.

$$d_{iS}^{max} = \min(d_i^{max}, d_i^{nom} + slack_i) \tag{3.14}$$

$$d_{iS}^{min} = d_i^{min} \tag{3.15}$$

Table 7 shows the values resulting from these expressions when applied to the project example.

It can be seen that activities 3, 4, 12 and 13 (critical activities identified in light 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, i.e., with smaller durations (like critical activities), 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 as defined previously.

Table 7: Allowed duration span

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

In Figure 9, the resource profile is displayed again with the durations trimmed with the schedule dependent values. Each activity is displayed with its nominal duration and their possible alternative durations' are identified by diagonal filling lines (positive slope for decreased durations and negative slope for increased durations).

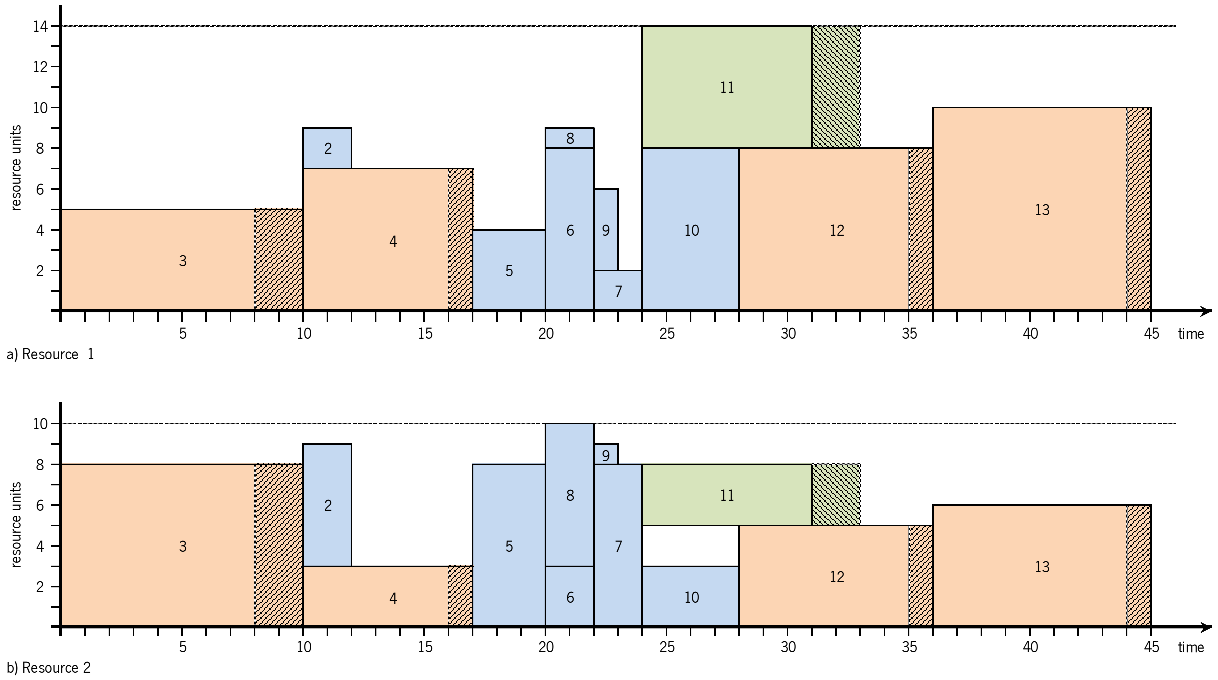
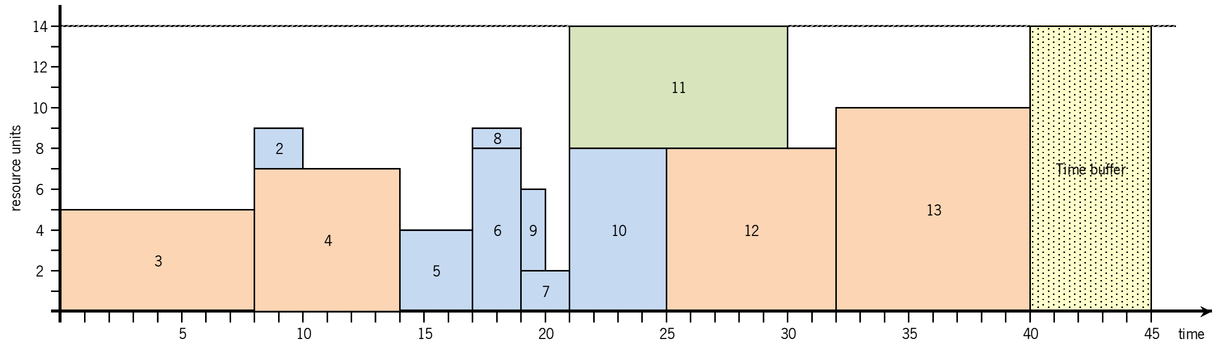


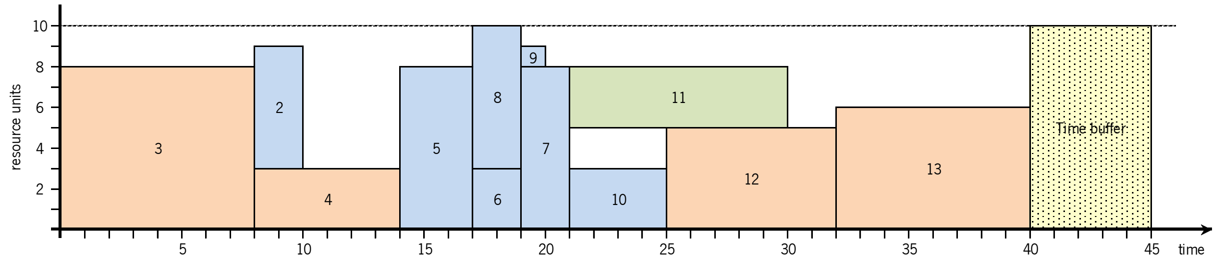
Figure 9: Resource profile for buffered schedule: a) for resource 1 b) for resource 2

This new schedule can be interpreted as having time buffers inserted 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 (in this case the only new constraint to be respected is the resource flexibility).

To emphasize this view, and because the "critical sequences" of this project/schedule always involve the activities that have time buffers (if this were not the case this view would be more difficult to be visualized), the schedule can be displayed with the time buffers joined together at the end of the project. The purpose of this method is not to alter the start time of activities but rather the opposite. The intent here is only to emphasize the effect on dealing with uncertainty, enabling an easier comparison with other time buffer insertion mechanisms. Figure 10 shows this effect.



a) Resource 1



b) Resource 2

Figure 10: Resource profile for end buffered schedule: a) for resource 1 b) for resource 2

One might observe that it is possible to use this technique to decrease the project makespan. Starting with an optimal RCPSP schedule, a new one can be obtained using resource flexibility that has smaller or equal duration. In the example, the reduction being greater than 10% ($\Delta T = (45\text{-}5)/45 \cong 11.1\%$) is quite interesting but, although correct, this is not the effect that this study is aiming for.

3.2.4 Score

The definition of the possible duration that each activity might have, provided the resources that are allocated to execute them are flexible, assumes that the estimation made regarding the work content 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 (deterministic) schedule[8]. Additional actions are needed otherwise.

To proceed, the concept of work content will be necessary. It is defined as the product of the activity's duration and its resource requirements or, when more than one resource type is considered, as given by expression (3.16), where w_{ik} is the work content of activity i for resource type k.

$$w_{ik} = d_i r_{ik}, \qquad for\ each\ i \in V, k \in K \tag{3.16}$$

---

[8] This is based on the assumption that the resource availability is constant (deterministic). The possibility that one or more resource units are absent (decreasing the resource availability) will be mentioned later on.

As stated before, 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 $score_{ir}$ measures the weighted working time deviation (under or over) per activity i and per resource type k and is formally defined by expression (3.17).

$$score_{ik} = r_{ik}(d_i - d_i^{nom}), \qquad for\ each\ i \in V, k \in K \tag{3.17}$$

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

The proof of this expression can be checked in Appendix I.

The previous expression denotes that $score_{ik}$ can be seen as a measure of the deviation of the work content since rearranging (3.17) the following expression is obtained: $score_{ik} = r_{ik}d_i - w_{ik}^{nom}$.

The extreme values of $score$ for the project example are presented in Table 8. As with the assumptions made to derive Figure 9, extreme values for $score$ consider the maximal contribution of each activity considering:

- Non-critical activities: the maximal duration. This determines the maximal positive score.
- Critical activities: the minimal duration. This determines the maximal negative score.

Table 8: Example score extreme values

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
				$\sum_i score_{ik},\ for\ each\ k \in K$:	-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, the $\sum_i score_{ik}$ will not be 0. The following cases might occur:

a) $\sum_i score_{ik} > 0$: It is not necessary to slow down all possible activities to enable a faster rate for all possible critical activities;

b) $\sum_i score_{ik} = 0$: Slowing down all possible activities enables a faster rate for all possible critical activities (the easy case);

c) $\sum_i score_{ik} < 0$: It is not possible to enable a faster rate for all possible critical activities even if all possible non-critical ones are slowed down.

Case 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 the "consumption" of it must be done until it reaches 0. The order in which activities can use this buffer should be established by using a predefined priority rule. An immediate priority rule is to order activities according to the smallest activity number but, while it is the easiest to deal with, it is more reasonable to select the activity with the smallest starting time to define the priority because it makes more sense to use buffers as they are needed chronologically than to prevent an activity from making use of an available buffer because a later starting activity with a smaller activity number may eventually need it. Nevertheless, the priority rule should be selected according to the specific project's needs.

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

$$score = \min_k \left(\sum_i score_{ik} \right), \qquad for\ all\ i \in V, k \in K \tag{3.18}$$

Alternatively, resource management might be done individually for each resource type.

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. Whenever a priority rule is used, the indexing should reflect the resulting order. For instance, for time related priority rules, a time indexed variable should be used ($balance_{tk}$) while for activity related priorities it should be activity indexed ($balance_{ik}$).

To apply this concept to the project example, the activity index will be used which leads to the following definition:

$$balance_{ik} = \sum_{j=1}^{i} score_{jk}\,, \qquad for\ each\ i, j \in V, k \in K \tag{3.19}$$

Using this expression, a resource independent value for balance is defined as $balance_n = score$.

The resulting values for the example are presented in Table 9. 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 9: Example balance values

$\boldsymbol{i}$	**Resource 1**		**Resource 2**	
	$\boldsymbol{score_{i1}}$	$\boldsymbol{balance_{i1}}$	$\boldsymbol{score_{i2}}$	$\boldsymbol{balance_{i2}}$
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 for all resource types, i.e., $balance_{nk} \geq 0$ for all $k \in K$, where n is the dummy end activity, 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:

$$balance_k^+ = \sum_{i=1}^{n} score_{ik} \mid score_{ik} > 0, i \in V, k \in K \tag{3.20}$$

and to set it to the schedule's initial balance, that can be defined as $balance_{0k}$ (which is 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:

$$score_{ik}^+ = 0 \mid score_{ik} > 0, \quad for\ each\ i \in V, k \in K. \tag{3.21}$$

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 $balance_{nk} \geq 0$ [9].

3.2.6 More on durations

Expression (3.14) can be written as (3.22). The maximal activity duration is schedule dependent besides being dependent on the activity's nominal (deterministic) duration d_i^{nom} and the resource flexibility α_k^-.

$$d_{iS}^{max} = \min\left(\min_k \left\lfloor \frac{d_i^{nom}}{(1-\alpha_k^-)} \right\rfloor, d_i^{nom} + slack_i\right), \qquad for\ each\ k \in K, i \in V \tag{3.22}$$

Similarly, expression (3.15) can be written as (3.23) to emphasize that the minimal activity duration is not schedule dependent, depending only on the activity's nominal (deterministic) duration d_i^{nom} and the resources flexibility α_k^+.

---

[9] This analysis is made considering extreme (up to) values wherein this expression assumes that activities with positive score are deterministic (active phase). When projects are executed (reactive phase) the balance should be calculated and managed in "real time" for all activities. This will be detailed later on.

$$d_{iS}^{min} = \max_k \left\lceil \frac{d_i^{nom}}{(1+\alpha_k^+)} \right\rceil, \qquad k \in K, i \in V \tag{3.23}$$

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 $= \min_k \left\lceil \frac{d_i^{nom}}{(1-\alpha_k^-)} \right\rceil$: 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).

Or, in plain words:

Capacity to absorb uncertainties = f(Schedule flexibility, Resource flexibility)

The schedule flexibility can also be identified in score. It is defined as $score_{ir} = r_{ik}(d_i - d_i^{nom})$ 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" can be defined as:

$$SIF_k = \sum_{i=1}^{n} r_{ik} slack_i\,, \qquad for\ each\ k \in K \tag{3.24}$$

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

$$SIF = \min_k \left(\sum_{i=1}^{n} r_{ik} slack_i \right), \qquad for\ all\ k \in K \tag{3.25}$$

Note that the SIF_k can be regarded as the $balance_k^+$ considering the schedule's flexibility only (when resource flexibility is not taken into account).

From the above, it is clear that slack is the key variable to be improved and consequently the selection of the baseline schedule is critical in order to take advantage of resource flexibility. Therefore, slack should be maximized to increase the capacity of a project being executed to cope with uncertainties that may arise.

3.2.7 Alternative schedules

In line with the RCPSP model, the impact of project duration on slack, and thus on flexibility, will be considered. The impact on slack can also be analysed for other schedule optimization objectives, like cost or time/cost trade-offs, but will be limited to project duration.

Consider that $slack_{(S)} = \sum_{i=1}^{n} slack_i$ of a schedule S.

Theorem 1: Given a schedule S with duration T, there is a schedule S' with duration $T' > T$ that has $slack_{(S')} > slack_{(S)}$.

Proof: Consider schedule S of a given project with duration T. Consider a schedule $S' = S$ except for the dummy end activity n which is scheduled to start one time period later $(s_n' = s_n + 1)$. The duration of S' is then $T' = T + 1$ and all predecessors of n $(i \mid i \in \{Pred(n)\})$ have their $slack$ increased by one unit $(slack_i' = slack_i + 1)$. As there must be at least one predecessor of dummy end activity n and all remaining activities are not affected in their schedule, then $slack_{(S')} \geq slack_{(S)} + 1$ or $slack_{(S')} > slack_{(S)}$

The following conclusion can be derived from theorem 1:

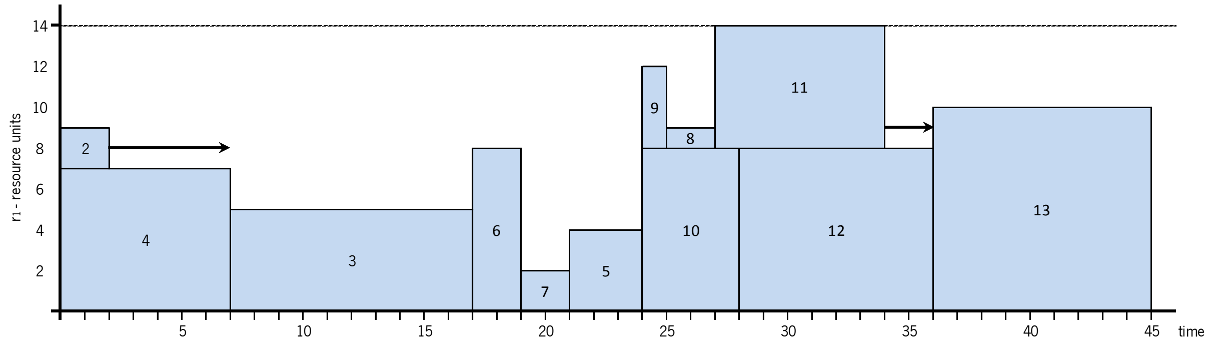
Corollary 1: A project's $slack_S$ resulting from a schedule S can be increased by increasing its duration.

But, on the other hand:

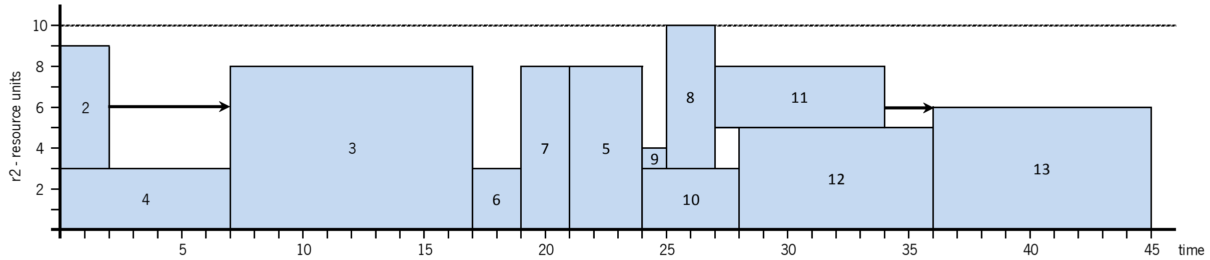
Theorem 2: Given a schedule S with duration T there does not always exist a schedule S' with duration $T' \leq T$ that has $slack_{(S')} \leq slack_{(S)}$.

Proof: To proof this theorem, consider the equivalent statement that given a schedule S with duration T there exists a schedule S' with duration $T' \leq T$ that has $slack_{(S')} > slack_{(S)}$. An example is sufficient to prove this statement and therefore to prove the theorem.

a) The $T' = T$ case: Consider a new schedule S for the project example shown in Figure 11, that has the same duration as the optimal schedule S' shown in Figure 7 ($T = T' = 45$). Activity 2 has the same slack $slack_2 = slack'_2 = 5$ while activities 9 and 11 have more slack ($slack'_9 = slack_9 + 1$ and $slack'_{11} = slack_{11} + 3$). Therefore in spite of having the same duration ($T' = T$), S' has more slack ($slack_{(S')} > slack_{(S)}$), which proves the case for $T' = T$.



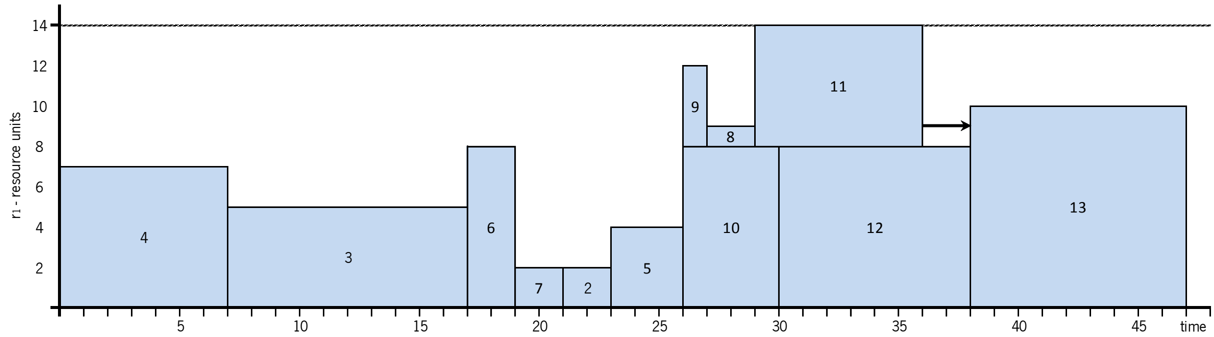
a) Resource 1



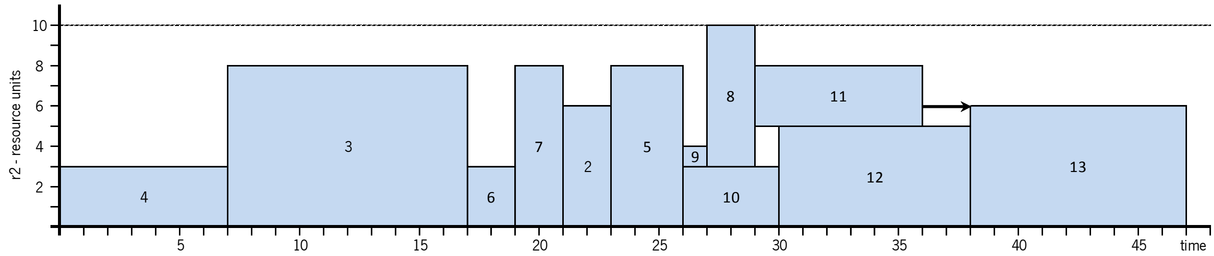
b) Resource 2

Figure 11: Resource profile for alternative optimal schedule: a) for resource 1 b) for resource 2

b) The $T' < T$ case: Consider now the schedule S' to be the one of Figure 11 and a new schedule S shown in Figure 12, that has a larger duration than S' ($T = 47$). Only activity 2 has its slack reduced to 0 ($slack'_2 > slack_2$) while all remaining activities maintained the same slack. Therefore in spite of having less duration ($T' < T$), S' has more slack ($slack_{(S')} > slack_{(S)}$), which proves the case for $T' < T$.



a) Resource 1



b) Resource 2

Figure 12: Resource profile for non-optimal schedule: a) for resource 1 b) for resource 2

The following conclusion can be derived from theorem 2:

Corollary 2: An RCPSP optimal schedule S can be non-minimal regarding its $slack_{(S)}$.

Applying the flexible resources' methodology to the aforementioned schedules (S', S''), quite distinct values are obtained comparing to the ones calculated for S (see Table 22 in Appendix II).

It becomes evident that the selected baseline schedule is of crucial importance when applying the proposed methodology to deal with uncertainty. Using an optimal duration baseline schedule seems to be a good starting point as it assures the minimal project duration, but this is only assured if the project is deterministic. When there is uncertainty within the project, there might be better ones to deal with uncertainty because, according to theorem 1, there are certainly schedules with increased slack. Even if only optimal duration schedules are considered, there may be some better than others.

3.2.8 Extreme cases

In the examples, all resource flexibility parameters α_k^- and α_k^+ were set to 25%. That means that a resource should "work" between 6h to 10h considering a day as the project time unit with a nominal working time of 8h per day. As proven before, the flexibility parameters will determine to which extent the intrinsic schedule flexibility can be taken advantage of. In spite of being highly dependent on the organization in which resources are embedded, it is interesting to understand to which extent they can be set.

Thereby, remembering that $a_k^{nom}(1-\alpha_k^-) \leq a_k \leq a_k^{nom}(1+\alpha_k^+)$, the unitary values ($a_k^{nom} = 1$) for a_k, are limited to:

- $a_k > 0$: otherwise no work would be done;
- $a_k \leq \frac{\text{all subunits per time unit}}{\text{nominal subunits per time unit}}$: no resource can work more than the existing time allows. As an example of this upper limitation is 24 hours/day or 7 days/week.

Accordingly,

- α_k^- is bounded by: $0 \leq \alpha_k^- < 1$ and
- α_k^+ is bounded by: $0 \leq \alpha_k^+ \leq \frac{\text{all subunits per time unit}}{\text{nominal subunits per time unit}} - 1$.

where 0 means no resource flexibility and the maximum value for α_k^+ depends on the selected time unit and its subunits.

Hence, it immediately follows that this method is not applicable to resources working non-stop (nominal=all subunits per time unit).

When a single project is considered, it is also not possible to compensate work (either more or less) if $\alpha_k^- = 0$ even if $\alpha_k^+ > 0$ or the other way around, i.e., if $\alpha_k^+ = 0$ even if $\alpha_k^- > 0$.

Another possibility that is worth mentioning is that this method can be used to execute activities with more or fewer resources which is a problem traditionally addressed by the Multi-mode RCPCP (Hartmann & Drexl, 1998). This method has no modes in the sense of the MRCPSP because the amount of resources used by an activity remains constant. What changes is the rate at which they work. Nevertheless, the model can be extended by keeping the work to be done in a time period and allowing the amount of resources to be used to vary within their predefined flexibility. A simple example is that 4 resource units working at a 75% rate are equivalent to 3 resource units working at a 100% rate if both modes are allowed by the activity and the rates are allowed for the resource.

Generically, an activity being executed at a lower rate can also be executed with fewer resources if the rate of execution is increased. As an extreme example, a resource might be missing for a time period which can be compensated by resource units working at a higher rate.

Similarly, an activity being executed at a higher rate can also be executed with more resources if the rate of execution is decreased.

The possibilities to explore can be greatly increased if resource flexibility is combined with FRCPSP (Flexible Resource Profiles).

These possibilities might seem far away from reality or hard to model and manage, but consider the following scenario:

- Time unit = week
- Subunit = hour (h)
- $\alpha_k^- = \alpha_k^+ = 25\%$

This will mean that a resource can work from 30h up to 50h per week. If flexible resource profiles are allowed, a top level schedule might be set to a weekly basis while a more detailed daily level one (or several partitioned ones) can be used to manage resources. At this level, resources would be allowed to work differently (non uniformly) within the time (a week), i.e., work differently from one day (the detailed time unit) to the other which, in extreme cases, could be not working at all (in which case $\alpha_k^- = 1$).

As long as the flexibility parameters, and other limits imposed by the resource characteristics (like their dependency or other characteristic that enable multi-mode) are respected, there are several other possibilities that can be explored.

It is even possible to extend the model to cope with activity insertion/deletion if flexibility parameters are adequately defined (for example by allowing resources to be idle) and allowing the schedule's flexibility to be used in such extreme cases. This possibility will not be explored further.

3.2.9 Additional remarks

Some additional remarks have to be made regarding the proposed model:

a) Project costs are not considered in the model, which means that they are assumed to be proportional (linear non-decreasing) to durations and resources. That is to say that the under/over work costs have no special cost function over the nominal work costs. This also means that there is no advantage in using one resource or another within the same type, nor is there a preference (cost related) in optimizing the use of one resource type over another. If this is not the case, additional modelling is required.

b) The model considers that when an activity requires additional effort (work content) it can be executed at a faster rate in order to still be able to finish in the scheduled finish time. The model was described considering that the additional effort is executed at nominal rate. It is possible to execute it at a faster rate, up to the resources' flexibility. This approach has the advantage of enabling additional effort to be allowed in the model and enables a uniform resource distribution within the activity (exempting the use of a more complex flexible resource profiles model). The additional effort that can be incorporated is up to the value resulting from the following expression:

$$\Delta w_{ik} = r_{ik} d_i^{nom} \frac{(\alpha_k^+)^2}{(1 + \alpha_k^+)} , \qquad for\ each\ i \in V, k \in K \tag{3.26}$$

The proof of this expression is presented in Appendix I.

c) It might occur that the actual rate of execution of an activity leads to a non-integer value regarding the subunit of time.
Recall the example where resources have their flexibility bounded by $\alpha_k^+ = 25\%$ and $\alpha_k^- = 25\%$, leading to a resource availability of $0.75 \leq a_k \leq 1.25$.
Consider the time unit is a day and that the nominal working time is 8h per day. A resource has to work 10h a day to execute the work at a 125% rate. Similarly, it works 6h in order to reduce the working rate to 75%. Consider also an activity with a single resource requirement of 1 unit and a nominal duration of 8 days. When executed at a maximal rate (125%) it should take 6.4 days to be completed. The model in use only allows integer activity durations that will set the activity duration to $\lceil 6.4 \rceil = 7$days. The effective rate of execution is then 114.3% which results in a 9.14h per day. This is in the allowable interval of $[6\text{h}, 10\text{h}]$ assuming that this interval is continuous ($\subset \mathbb{R}$). For organizational or administrative reasons, the working time per day might be restricted to integer values. For instance, daily working time might be limited to values from the set $\{$6h,7h,8h,9h,10h$\}$ ($\subset \mathbb{N}$) or some other finite set (like quarters of an hour or integer minutes) meaning that the working time per day will be in some $\{$discrete set$\}$ $\subset \mathbb{Q}$. If this is the case, the problem can be solved, or at least minimized, by using flexible resource profiles. In the example, the activity can be processed at the defined rate (114.3%) if the resource works 10h the first day and 9h for the remaining 6 days.

3.3 MIP formulations

To describe and apply this methodology any existing feasible schedule can be used as a baseline schedule. Hence, any scheduling technique can be used to set the baseline schedule so it is not required to search for more suitable ones regarding this methodology. But, as stated previously, the selected baseline schedule does have an impact when projects face uncertainty, according to this model. The remaining analysis, including computational studies, will rely on existing scheduling techniques, but it is important to keep in mind that some schedules are better suited than others in taking advantage of this methodology.

To highlight how to find schedules that are more capable to take advantage of the presented methodology and consequently, to better deal with uncertainty, some MIP (Mixed Integer Programming) formulations are presented. For this reason, only conceptual formulations are presented. Figuring out computable ones can be done by analogy with the RCPSP computable formulations as the ones presented in (Bianco & Caramia, 2013; Klein, 2000; Mingozzi et al., 1998).

3.3.1 Resource flexibility

From a strict schedule intrinsic flexibility point of view, an objective function to optimize flexibility involves only the maximization of its slack (r_{ik} is not schedule dependent). Conceptually the mathematical model can be expressed by:

Formulation 1:

$$maximize \left(\min_k \left(\sum_{i=1}^{n} r_{ik} \tau_{ik} \right) \right) \tag{3.27}$$

$subject\ to$:

$$s_j \geq s_i + d_i, \qquad for\ all\ (i,j) \in A \tag{3.28}$$

$$s_1 = 0 \tag{3.29}$$

$$\sum_{i \in P_t} r_{ik} \leq a_k\ , \qquad t = 1, \ldots, s_n, k \in K, i \in V \tag{3.30}$$

$$0 \leq \tau_{ik} \leq \min_{j \in Succ_i} (s_j) - s_i - d_i\ , \qquad \tau_{ik} \in \mathbb{N}, k \in K, i \in A \tag{3.31}$$

$$r_{ik} \leq a_k - \sum_{h \in P_t} r_{hk}\ , \qquad t = s_i + d_i, \ldots, \tau_{ik} + s_i + d_i, k \in K, i \in V \tag{3.32}$$

where (3.27) maximizes the SIF (Schedule Intrinsic Flexibility) considering that resources are not managed independently, τ and constraints (3.31) and (3.32) represent the slack as defined by (3.8) and the constraints defined by (3.28), (3.29) and (3.30) are the well-known RCPSP (conceptual) constraints: precedence constraints, dummy start set at time 0 and resource constraints respectively.

Alternatively, the objective function can be expressed as:

Formulation 1a:

$$maximize \left(\sum_{k\in K}\left(\sum_{i=1}^{n} r_{ik}\tau_{ik}\right)\right) \tag{3.33}$$

which would optimize the overall resource contribution to flexibility and not only the most demanding one. This equal weight resource contribution approach can be further detailed if resources are to be managed independently in which case the objective function has to be transformed into a multi-objective (per resource type) one which is much harder to deal with.

As stated in theorem 1, slack can be increased by increasing the schedule's duration, therefore slack can be maximized by increasing the project's duration, i.e., the project's completion time ($s_n \to \infty$). This is possible because there is no constraint that limits the project's completion time.

3.3.2 Slack management

Additional limitations have to be set to properly manage the project's available slack. Among the possibilities, three will be presented:

- Deadline: Restricts the project duration while maximizing slack (use the best possible schedule's flexibility).
- Bi-objective: Favours the trade-off between minimizing duration and maximizing slack.
- Target flexibility: Defines the required schedule's flexibility while minimizing duration.

3.3.3 Deadline

An immediate way to limit the solution space of this optimization problem is to impose a project's maximal duration by adding the following constraint:

Formulation 2:

$$s_n \leq \delta, \qquad \delta \in \mathbb{N} \tag{3.34}$$

This forces the dummy end activity to start (and end) before or at a predefined time instant. By theorem 1, this maximization problem will increase its result (the slack) with the increase of δ (the deadline) and that if $\delta <$ optimal (minimum) duration, the problem will have no solutions.

3.3.4 Bi-objective

Another way to limit the solution space is not focusing exclusively on the optimization of the slack but also on the project's duration. In this case, the objective function will include the maximization of the slack (converted to a minimization problem) combined with the minimization of the project duration (s_n).

Formulation 3:

$$minimize \left(-\left(\min_k \left(\sum_{i=1}^{n} r_{ik} \tau_{ik} \right) \right) \text{ , } s_n \right) \tag{3.35}$$

$subject\ to$: (3.28), (3.29), (3.30),(3.31), (3.32)

The notation of this bi-objective function is left abstract because only conceptual formulations are of concern here. To solve this problem, multi-criteria techniques (T'kindt & Billaut, 2002) can be used such as Scalarization (Ehrgott, 2006) and Pareto (Zhu, Isac, & Zhao, 2005) approaches.

A key factor in using this formulation is the adequate balance between the two objective functions.

3.3.5 Target flexibility

A distinct approach is to "invert" the logic behind the "deadline" formulation (presented in 3.3.3) where slack was optimized and a project time limit was imposed. The idea is then to optimize the project duration while imposing a minimum schedule's flexibility of ω (target flexibility).

Formulation 4:

$$minimize\ s_n \tag{3.36}$$

$subject\ to$: (3.28), (3.29), (3.30),(3.31), (3.32)

$$\min_k \left(\sum_{i=1}^{n} r_{ik} \tau_{ik} \right) \geq \omega, \qquad for\ all\ k \in K \tag{3.37}$$

In this case, it is easier to include the possibility that resources are managed independently which can be done by replacing constraint (3.37) with per resource ones ω_k as defined by:

Formulation 4a:

$$\sum_{i=1}^{n} r_{ik}\tau_{ik} \geq \omega_k, \qquad for\ all\ k \in K \tag{3.38}$$

3.3.6 Notes on formulations

While formulation 1 is only a first step to get to the remaining ones, formulation 3 is hard to manage because there is no easy way to balance the importance between flexibility and project duration. From a practical point of view, formulations 2 and 4 are the most interesting ones that can be recommended for the following specific situations:

- Formulation 2 (Deadline): If a project has to be delivered in a predefined time, force it and explore the best possible schedule to deal with uncertainty. A special case of this approach is the minimal duration case in which the maximal flexibility schedule is selected among the ones with minimal duration.
- Formulation 4 (Target flexibility): If a project's uncertainty can be quantified, establish a schedule that can cope with it at the minimal possible project duration.

3.4 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 organizational 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.

3.4.1 General procedure

The procedure to apply this methodology to project management can be summarized in the following steps:

- STEP -1: Define resource flexibility parameters (α_k^-, α_k^+).

  Typically this step should be made before the project starts and should be defined for the entire organization or within a project portfolio scope. Project specific tuning is possible if needed.

- STEP 0: Define (basic) project data and determine possible durations (d_i^{min}, d_i^{max}).

  The calculations should be made just before scheduling.

- STEP 1: Establish the baseline schedule ($S^b = \{s_1, \ldots, s_n\}$) using an adequate scheduling technique/tool in order to achieve the established objectives (that should include the maximization of slack flexibility) 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 the baseline schedule but with finish times defined in the intervals resulting from the possible activity durations.

  This is done within the project scheduling stage, just before project execution.

- STEP 2: If deviations occur, check if they can be absorbed by the schedule's flexibility (the activities subject to deviations can be performed within the defined intervals) and that there is enough balance left. If this is possible, update the working schedule accordingly. If not, rescheduling is necessary (or other actions to further cope with deviations have to be defined like buffering techniques).

  This step is performed during project execution by monitoring and control processes.

It should be noted that some RCPSP scheduling techniques, especially implicit enumeration branch-and-bound ones, rely on pruning rules to speed up the solution search process that can be expressed by the following pseudocode:

```
...
if $s_n(S_{new}) \geq s_n(S_{best})$ then discard $S_{new}$
else $S_{best} = S_{new}$                    // $s_n(S_{new}) < s_n(S_{best})$
...
```

If these techniques are used in STEP 1 to establish the baseline schedule, it should be enhanced to:

```
...
if $s_n(S_{new}) > s_n(S_{best})$ then discard $S_{new}$                    // i.e., do nothing
else if $s_n(S_{new}) == s_n(S_{best})$ then
        if $SIF(S_{new}) \leq SIF(S_{best})$ then discard $S_{new}$          // i.e., do nothing
        else $S_{best} = S_{new}$
else $S_{best} = S_{new}$                                                    // $s_n(S_{new}) < s_n(S_{best})$
...
```

It may also be interesting to analyse the effect of priority rules, as the ones used by Serial (SGSS) and Parallel (PGSS) Schedule Generation Schemes and (Kolisch, 1996b) regarding the generated schedule flexibility as defined here. Moreover, in addition to test which existing priority rules show better results, other ones can be searched to further improve the generation of schedules regarding their flexibility.

3.4.2 Multi-project management

Although a project is a one-time endeavour it is, more often than not, integrated in a multi-project environment as organizations tend to become more and more project oriented. Typically a project is planned using some historical data gathered from previous projects and ends with a closing activity that includes the collection and archival of data for future use. In general, projects will be competing for resources and time with other projects that can be executed concurrently, sequentially or not executed at all. As globalization increases, the pressure to cut costs and meet deadlines while resources are more heterogeneous also increases. This emphasizes the need for an integrated management environment that deals with resource management within a multi-project context. Portfolio management (Banerjee & Hopp, 2001; Martinsuo, 2013) fulfils this goal and can benefit from resource flexibility and the proposed methodology. In fact, to take full advantage of this methodology, a portfolio management context is mandatory because each project's unused capacity ($balance_{nk}$) can be "exported" to other projects that can start with a balance that results from their initial balance plus some of the exported one. This unused balance will be denoted as $\Sigma balance_k$ and can be generically defined as:

$$\Sigma balance_k = \sum_{e=1}^{m} balance_{nk}(p_e) \tag{3.39}$$

where $balance(p_e)$ is the final balance of a previously executed project p_e of a set of m projects.

A possible drawback of this analysis might be that "all projects tend to have delays" and therefore will never have capacity left to be exported. While uncertainty tends to always be present, it should go both ways, even if not symmetrically, otherwise the portfolio/project management process is probably not being efficient. The idea is that if some capacity remains unused by one activity, it can be used by other critical ones even if they lie in another project. To achieve this, resources have to be flexible and the schedule flexibility has to be managed.

A correct balance of projects in a portfolio between, "critical and less-critical" ones or "with a high degree of uncertainty and well controlled (deterministic)" ones, should tend to compensate one another if resource flexibility is possible and adequately managed.

Nevertheless, this practice is certainly usual in a very informal way and the proposed methodology aims to identify a model to assist in managing it, which can be integrated in standard portfolio/project management practices.

3.4.3 Modes

The concept of transferring balance between projects leads to the definition of project execution modes. Execution modes consist in allowing a project to "consume" more or less balance than it can "generate" during its execution and therefore in contributing positively or negatively to the overall (inter-project) available balance. Four basic modes can be identified:

- CONSERVATIVE: The project's execution generates balance. Activities cannot consume any balance, and therefore can only be executed at their nominal or slower rates.

- NORMAL: The project is executed within the limits of its internal flexibility.
  Activities can be executed at any rate allowed by the flexibility parameters as long as the balance remains non-negative;

- RESTRICTED: Same as NORMAL except that an initial additional balance is added that can be used by the project;

- OPEN: Activities can be executed at any rate allowed by the flexibility parameters. There are no restrictions regarding the final balance.

Table 10 shows the relation between modes, the initial balance to be set and the allowed operations in the working schedule.

In the table, $balance_k$ is the balance per resource assigned to the project from the balance made available from previously executed projects: $balance_k \in [0, \Sigma balance_k]$

Table 10: Project execution modes

Project type	**Allowed consumption**	**MODE**	$\boldsymbol{balance_{0k}} =$ (schedule initial balance)
Non-critical	no balance consumption	CONSERVATIVE	0
Normal	$balance_{nk} \geq balance_{0k}$	NORMAL	$balance_k^+$
Critical	$balance_{nk} < balance_{0k}$	RESTRICTED	$balance_k^+ + balance_k$
Urgent	balance consumption is unlimited	OPEN	unlimited

In other words, modes are assigned according to the project's criticality classification, where criticality means its importance to the organization and/or its degree of uncertainty, and determines how flexibility can be used in project execution.

3.4.4 Scheduling procedure

The general procedure presented previously can now be completed and detailed in order to cope with a multi-project environment and to take advantage of it.

The project scheduling process to apply this methodology should take the following steps:

- STEP -1 – Before project starts:

  Define resource flexibility parameters α_k^-, α_k^+ for each resource $k \in K$.

- STEP 0 – Before project scheduling:

  Define project data $G = (N, A)$, determine resource availability and assign resources.

  Determine a_k^-, a_k^+ for each resource $k \in K$.

$$\left| \begin{array}{l} a_k^- = a_k^{nom}(1 - \alpha_k^-) \\ a_k^+ = a_k^{nom}(1 + \alpha_k^+) \end{array} \right.$$

  Determine possible durations (d_i^{min}, d_i^{max}) for each activity $i \in V$:

$$d_i^{min} = \max_k \left\lceil \frac{d_i^{nom}}{(1+\alpha_k^+)} \right\rceil, \qquad k \in K$$

$$d_i^{max} = \min_k \left\lfloor \frac{d_i^{nom}}{(1-\alpha_k^-)} \right\rfloor, \qquad k \in K$$

Define the project $mode \in \{$CONSERVATIVE, NORMAL, RESTRICTED, OPEN$\}$ (according to its criticality).

- <u>STEP 1</u> – At project scheduling stage (before project starts execution):

  Select a scheduling technique/tool according to availability, project dimension, objectives (objective function) and establish the baseline schedule $S_0^b = \{s_1, \dots, s_n\}$.

  Determine $slack_i$ for each activity ($i \in V$):

$$slack_{ik} = \max\left(\tau \in \left\{\left[0, \min_{j \in Succ_i}(s_j) - s_i - d_i\right] \cap \mathbb{N}\right\}\right) \quad | \quad r_{ik} \le a_k - \sum_{h \in P_t} r_{hk}\,, \;\; t = s_i + d_i, \dots, s_i + d_i + \tau, \;\; k \in K, \;\; i \in V$$

  Determine the schedule specific minimal d_{iS}^{min} and maximal durations d_{iS}^{max} for each activity $i \in V$:

$$d_{iS}^{min} = d_i^{min}$$

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

  Determine the initial $score_{ik}$ for each activity $i \in V$ with $slack_i > 0$ and each resource type $k \in K$

$$score_{ik} = r_{ik}(d_{iS}^{max} - d_i^{nom})$$

  Determine the schedule flexibility $balance_k^+$ for each resource type $k \in K$ (the more general independent resource management is considered):

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

  Set the initial project balance $balance_{0k}$, according to the execution mode, for each resource type $k \in K$, updating $\Sigma balance_k$ if necessary:

```
if $mode ==$ CONSERVATIVE then
        $balance_{0k} = 0$
else if $mode ==$ NORMAL then
        $balance_{0k} = balance_k^+$
```

```
else if $mode ==$ RESTRICTED then
    set $balance_k$ from the interval $[0, \Sigma balance_k]$
    update $\Sigma balance_k = \Sigma balance_k - balance_k$
    $balance_{0k} = balance_k^+ + balance_k$
```

Establish a working schedule S^w having the same activity start times as the baseline schedule but with finish times defined as late as possible (resulting from the possible activities' durations):

```
if $slack_i > 0$ then $d_i = d_{is}^{max}$
else $d_i = d_i^{nom}$
```

- STEP 2 – During project execution (monitoring and control processes)

  While there are unfinished activities and according to the working schedule S^w:

  Execute activities with assigned duration d_i and corresponding execution rate (effective working time per time unit):

```
$rate_i = d_i^{nom}/d_i$
```

If deviations occur (as an example, only deviations in work content are considered), check if they can be absorbed by the schedule's flexibility (the activities subject to deviations can be performed within the defined intervals) and check if there is enough balance left:

```
$adjust$ = FALSE
if $w_i < w_i^{nom}$ then
    update $d_i$
    $adjust$ = TRUE
if $w_i > w_i^{nom}$ and $mode \neq$ CONSERVATIVE then
    update $d_i$
    if $d_i \in [d_i^{nom}, d_i^{max}]$ then
        if $mode ==$ OPEN then $adjust$ = TRUE
        else if $balance_k \geq \Delta w_{ir}$ for all $k \in K$ then $adjust$ = TRUE
```

If this is possible, i.e., deviations can be dealt with ($adjust$ is TRUE), do:

```
    update $rate_i = d_i^{nom} / d_i$
    update $score_{ik} = r_{ik}(d_i - d_i^{nom})$
    update $balance_k = balance_k - \Delta w_{ir}$ for all $k \in K$
    update working schedule $S^w$
    repeat STEP 2
```

If not ($adjust$ is FALSE), reschedule: repeat scheduling process from STEP 0 (updates project data and set a new baseline schedule S^b).

- STEP 3 — After project execution end

  Update $\Sigma balance_k$ for each resource type $k \in K$:

```
    $\Sigma balance_k = \Sigma balance_k + balance_{nk}$
```

An important factor in applying methodologies that rely on reactive or online procedures is the readiness in which deviations are perceived. Regarding this methodology, it is relevant to identify three distinct cases about when deviations are known:

- Before project starts: there is no impact on the reactive part of the procedure because there are no activities in progress yet and therefore the baseline schedule can still be recalculated;
- Before the activity subject to deviation starts: the procedure can be "fully" applied, which means that it can be applied in the way that has been described;
- During the execution of an activity subject to deviation, at time instant t: the procedure can be "fully" applied to other non-started activities but can only be "partially" applied to the activity subject to the deviation, as well as to other activities already in progress. This means that only the remaining activity durations can be taken into account in applying the procedure. This can be done by splitting each selected ongoing activity into two sub-activities, (1) the finished part and (2) the remaining part, and setting a strict (must start) FS=0 precedence relation between them. Sub-activity 1 started at $s_{i_1} = s_i$ and finished at $f_{i_1} = t$ with a duration of $d_{i_1} = t - s_i$ while sub-activity 2 starts at $s_{i_2} = f_{i_1}$ and finishes at $f_{i_2} = f_i$ with a duration of $d_{i_2} = d_i - d_{i_1}$. The procedure can then be "fully" applied to sub-activity 2.

3.5 Comparative notes

The described methodology aims to protect the defined schedule by protecting, as much as possible, the start time of the individual activities while maintaining the project's costs. Such deviations on start times have a great potential to propagate and to have impact on internal and external project interfaces such as resources, other projects and control points.

The methodology can be regarded as a score management process that can dynamically allocate time buffers to critical activities. Note that activities can change from "non-critical" to "critical" or vice-versa as the project is being executed due to eventual deviations in their duration which can consume the slack that the non-critical had, or add slack that the critical ones did not have. This can be compared with some of the well-known stochastic time uncertainty scheduling techniques. Table 11 presents some distinguishing aspects of PERT (Malcolm, Roseboom, Clark, & Fazar, 1959) and Critical Chain (Goldratt, 1997) methodologies compared to the proposed one. SRCPSP methodology (Stork, 2001) is not considered in this comparison because, while its ability to cope with stochastic uncertainty is its main strength, it fails the basic assumption that a baseline schedule is mandatory, especially in a multi-project environment, for the reasons explained before.

Table 11: Methodology comparison

	PERT	**Critical Chain**	**Proposed**
Time buffers	Per activity	Per project and chain	None added
Buffer management	Static	Deterministic	Dynamic
Schedule stability	Probabilistic	No	Yes
Ability to deal with uncertainty	Unlimited	Limited (buffer dimension dependent)	Limited (schedule dependent)

Additionally, the ability to deal with uncertainty within the context of the proposed methodology also depends on the characteristics of resources which have to be flexible. They can be regarded as renewable resources with variable capacities that are also limited in a project wide base. Therefore, they can be classified as doubly constrained with variable capacity.

Besides assuming that resources are flexible (which can be impossible in certain contexts), the main limitation of this model is that its upper flexibility limit is static for a given schedule. The only way to overcome a balance overrun is to reschedule. Correct scheduling parameters can mitigate this problem but it can also be solved by extending or combining this method with other ones. This is certainly a path to be explored further to make the proposed model more suitable to cope with a boarder range of possibilities but, in this context, will not be analysed further.

4. COMPUTATIONAL STUDY

In the previous chapter, the model was presented by using a quite simple project example, having twelve real activities and two resource types. The example was designed to be as simple as possible yet allowing to explain the concepts and their application in a variety of concrete situations.

In fact, it was not really designed, but was selected from a few instances created to test and debug the code that implemented some basic RCPSP solution methods. Therefore, it was not tuned to achieve "good" (neither "bad") flexibility results. It is just an example that resulted from picking a PSPLIB (Kolisch & Sprecher, 1997) instance with 32 activities (j30) and, randomly, eliminating two of the four resource types and eighteen activities, performing the necessary precedence adjustments, in order to debug and test the code with a simple instance. For the same purpose, simpler and more complex ones were generated and used, but for explaining the flexibility concepts, this example revealed to be the simplest one encompassing all cases of interest.

The main goal of this chapter is then to make a deeper evaluation of this model using an extensive set of project examples and scheduling techniques, under predefined conditions, whose description follows.

4.1 Test environment

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[10] 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 (Trautmann & Baumann, 2009). As most tools base their scheduling method on heuristic ones, 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. In spite of theorem 2, stating that there is no guarantee that a longer duration schedule has more slack, by theorem 1 the slack can be increased by increasing the schedule's duration. It also seems (empirically) reasonable that as the project's duration decreases,

---

[10] "Professional" is used in the sense that the aim of the tool is to be used in a productive environment in opposition of a test or research environment and is, in general, commercially available to be purchased.

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, as described in chapter 2, is a computationally demanding task.

There are some educational[11] scheduling tools available, like RESCON (Deblaere, Demeulemeester, & Herroelen, 2011) or Pro-Track[12] that could be used but they lack the necessary flexibility to, for instance, import the resulting schedules allowing bulk processing of a test set.

Therefore, the decision was to implement from scratch the necessary scheduling methods.

4.1.1 Development environment

As stated before, solving RCPSP instances to optimality is computationally demanding. That demanded for a development tool that produces fast runtime applications.

Accordingly, all developed code was implemented in Microsoft Visual Studio 2012® using Microsoft Visual C++ which, besides producing fast runtime applications, also encompasses some performance analysis tools that have revealed quite helpful to achieve the required results in time.

4.1.2 Test algorithms/tools

The scheduling algorithms implemented to obtain the baseline schedule are:

- For the time optimal solutions: Demeulemeester and Herroelen branch and bound (DH-B&B) algorithm (1992, 1997).
- To represent a heuristic scheduling method: A Serial Scheduling Generation Scheme (SSGS) with the following typical priority rules (Kolisch, 1996b):
    - LJN (Lowest Job Number);
    - RND (Random);
    - SPT (Shortest Processing Time);
    - LPT (Longest Processing Time);
    - MIS (Most Immediate Successors);
    - MTS (Most Total Successors);

---

[11] "Educational" is used in the sense that the aim of the tool is to be used in an educational or research environment.

[12] Available at www.protrack.be.

- LNRJ (Least Number of Related Jobs);
- GRPW (Greatest Rank Positional Weight);
- EST (Earliest Start Time);
- EFT (Earliest Finish Time);
- LST (Latest Start Time);
- LFT (Latest Finish Time);
- MSLK (Minimum Slack);
- GRWC (Greatest Resource Work Content);
- GCRWC (Greatest Cumulative Resource Work Content).

The algorithms used in the implementation are presented in Annex I.

Additionally, Microsoft Project 2013 (MSProject) was used to generate a baseline schedule in order to include one of the most popular project management software tools.

In order to achieve typical values for MSProject scheduling, the following parameters were set (all other parameters remain at their default values):

- "Saturday" and "Sunday" were set to "working time" with the same working hours as the other days (this was done for easier Gantt chart visualization and comparison);
- "Levelling Options" were set in order not to allow an activity split.

Each instance was scheduled within MSProject by performing the following procedure:

- Import activity data (activity name, their precedence relations and their required resources) into MSProject;
- Import resources data (resource name and availability) into MSProject;
- Set "Task Mode" to "Automatic Schedule" for all activities;
- Execute the procedure "Level All".

All durations (project instances and their activities) were considered in days.

It is possible to improve MSProject generated schedules using its embedded scheduling algorithms and some additional programming with VBA (Visual Basic for Applications) (Trautmann & Baumann, 2010). Although possible, this is not typically used and therefore, was not considered.

4.1.3 Test set

Tests will be performed using RCPSP instances. These instances should be large enough to provide significance to the capacity transfer process but not so large that one can state that the method only works for large instances. Also, larger instances tend to be harder to solve as was already mentioned: the time to reach an optimal solution increases exponentially with the number of activities and even the constructive heuristic solution increases linearly with the number of activities.

Instances should also not be tuned to take advantage of this method in order to demonstrate the potential of the method for real life projects.

Consequently, the RCPSP instance test set PSPLIB J30 (Kolisch & Sprecher, 1997) will be used. This test set consists of 480 project examples (or instances)[13] with 30 real activities plus a dummy start and a dummy end one.

4.2 Preliminary study

Before analysing the schedules' flexibility and the potential of resource flexibility to take advantage of it to absorb uncertainties, the impact of the scheduling model in the resources allocated to a project will be assessed. For that purpose the previously defined parameters will be used.

4.2.1 Results

In Figure 13, a graphical view for all 480 PSPLIB J30 instances is shown. The x axis represents each instance and the y axis the corresponding project duration (t). To highlight the increase from the optimal values, durations are displayed having the negative part as the project optimal duration and the positive part as the deviation from the optimum. Accordingly, the values for the solution methods are represented as:

- A bar for the optimal ("Opt") duration, with the finish time corresponding to t=0 and the absolute negative start time corresponding to the project optimal duration;
- A red dot (dots are connected with a red line) for the MSProject project ("MSP") duration with the positive part representing the deviation from the optimum value. The overall project duration is then the sum of this value and the correspondent optimal one;

---

[13] In this context, the term "project example" and "instance" will be used interchangeably.

- A vertical line presenting the dispersion in the duration regarding all priority rules SSGS. The upper limit of each vertical line represents the maximal deviation from optimal of all durations computed with each priority rule and the lower limit represents the minimal one. Again, the overall project duration is the sum of these values and the correspondent optimal one.

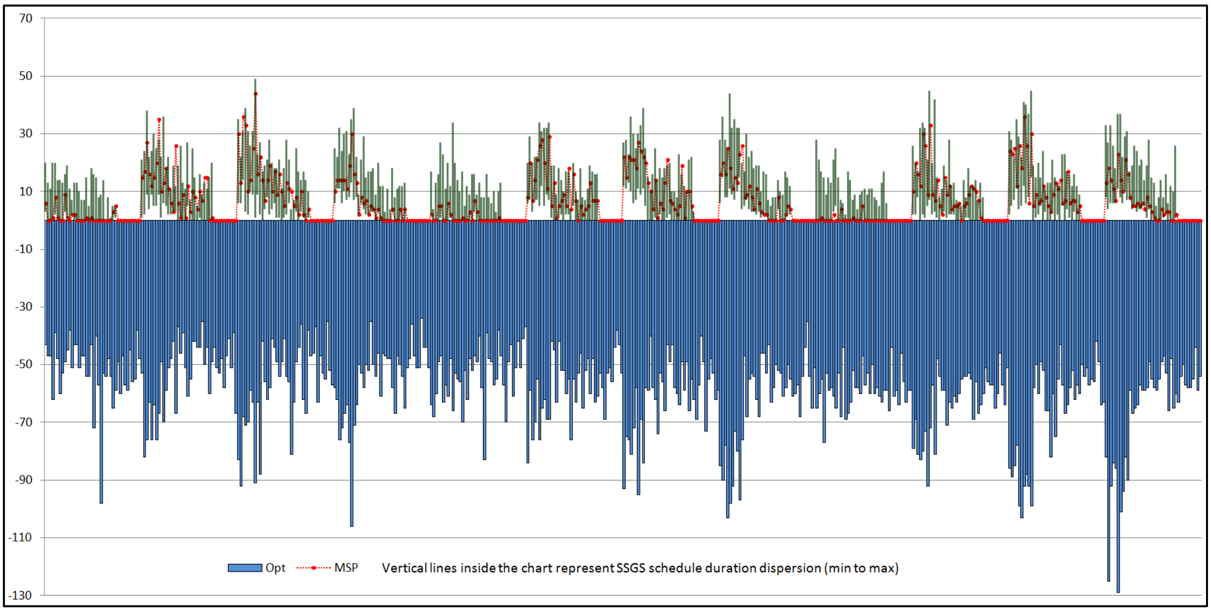


Figure 13: Project duration for all 480 J30 instances

An amplified detail of Figure 13 is presented bellow (Figure 14) to make it clearer.

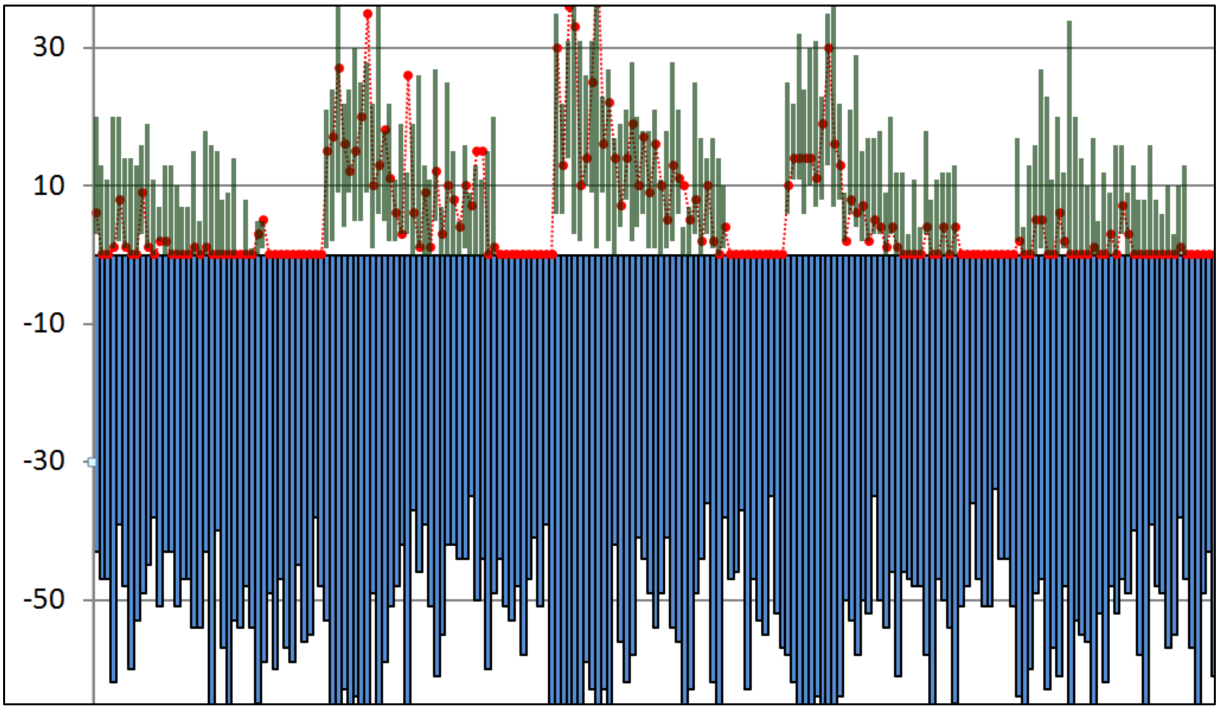


Figure 14: Amplified detail of project duration chart

In Table 12, a summary of all 480 psplib J30 instances regarding their scheduled durations are presented. Again, the optimal duration (T_{opt}) of each instance or example (e) is used as reference to emphasize the potential for improvement and therefore is presented as a plain value in the highlighted column. The other values are shown as absolute deviations from the optimum, for each remaining solution method, regarding:

- "Max": defined as $max_e(T_e - T_{opt})$;
- "Average": defined as $avg_e\left(T_e - T_{opt}\right) = \frac{\sum_e (T_e - T_{opt})}{480}$;
- "Min": defined as $min_e(T_e - T_{opt})$.

Corresponding relative deviations are also considered which are calculated by replacing $(T_e - T_{opt})$ in the previous formulas by $\left(\frac{T_e - T_{opt}}{T_{opt}}\right)$.

Table 12: J30 project duration summary

Note: Optimal duration is used as reference	**Optimal duration** (T_{opt})	**Deviation from optimal duration** ($T_e - T_{opt}$)															
		SSGS with defined priority rule															**MSProject**
		LJN	**RND**	**SPT**	**LPT**	**MIS**	**MTS**	**LNRJ**	**GRPW**	**EST**	**EFT**	**LST**	**LFT**	**MSLK**	**GRWC**	**GCRWC**	
Max	129	37	45	49	39	36	26	34	36	31	37	30	30	39	39	35	44
Average	59	5.96	7.83	10.55	7.71	6.11	4.22	6.71	6.50	5.74	7.25	3.31	3.67	6.12	7.39	6.72	6.13
Min	34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Max (%)		44%	63%	57%	51%	48%	32%	49%	52%	44%	46%	33%	34%	49%	60%	57%	53%
Average (%)		9%	13%	17%	12%	10%	7%	11%	10%	9%	12%	5%	6%	9%	12%	11%	9%
Min (%)		0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Based on the results presented above, the resources that must be available for the duration of each project can be calculated by the expression $a_k T_e$, for each resource type k and each project example e, where a_k is the availability of resource k.

The average values for all instances are presented in Table 13 for each resource type k and considering:

- The optimal solution;
- The best schedule resulting from SSGS (among all enumerated priority rules);
- The MSProject schedule.

Also presented are the values for:

- The required resources (that is the total work content of the project), which are independent from the schedule, given by the expression $\sum_i d_i r_{ik}$, for each resource type k and each project example e, and
- The percentage of unused resources, given by the generic expression $\frac{\text{Available} - \text{Required}}{\text{Available}}$.

Table 13: J30 average resources

	Required ($avg_e(\sum_i d_i r_{ik})$)				**Available** ($avg_e(a_k T_e)$)				**%Unused** ($\frac{\text{Available} - \text{Required}}{\text{Available}}$)			
	r1	r2	r3	r4	r1	r2	r3	r4	r1	r4	r3	r4
Optimal	570.66	583.46	574.56	581.99	1160.78	1171.60	1161.13	1161.61	52.00%	51.50%	51.96%	50.78%
Best SSGS	570.66	583.46	574.56	581.99	1191.42	1202.88	1192.46	1192.24	53.54%	53.05%	53.48%	52.36%
MSProject	570.66	583.46	574.56	581.99	1263.46	1276.71	1265.40	1264.40	56.42%	55.97%	56.40%	55.27%

4.2.2 Conclusions

The majority of projects are modelled as this type of optimization problem (minimize the project duration) and most commonly, costs are a non-decreasing function of its duration. As the presented results show, the scheduling solution method will greatly influence the project's duration and the most common scheduling techniques used present poor results, even considering small projects (less than one hundred activities) like the problem instances used in this analysis.

Additional efforts to develop and make available tools with better scheduling techniques are increasingly necessary. These tools should provide schedule durations closer to optimal and should be, as much as possible, independent of the problem instance to achieve such optimality (as presented in this preliminary study) as well as in the time they need to reach a solution (not covered in this preliminary study).

But, even using these non-optimal schedules, projects do, more often than not, overrun their estimated duration and costs. This means that additional efforts are needed to, given a schedule (better or worse regarding its duration), make it more resistant to failure, i.e., make it more robust.

Several techniques were studied to achieve these goals, starting with PERT (Program Evaluation and Review Technique) where simplistic project duration estimations, beyond deterministic ones, are calculated, to increasingly enhanced versions of RCPSP. As mentioned in chapter 2, some of these enhancements are:

- SRCPSP (Stochastic RCPSP) whose lack of a base schedule hinders its use (see Ballestín and Leus (2009) as an example);
- MRCPSP (Multi-mode RCPSP) (see Peteghem and Vanhoucke (2010) as an example);
- Proactive/Reactive Scheduling (see Demeulemeester and Herroelen (2009));

These techniques are still being the subject of additional research as is a recent topic designated as FRCPSP (see Naber and Kolisch (2014) as an example) which can be seen as a generalization of MRCPSP.

This preliminary study served as a starting point to the development of a method to address the problem of transforming a given schedule into a more robust one, attempting to attain a better behaviour when unscheduled events occur during project execution.

4.3 Flexibility Data: schedule flexibility

Having identified the high dependency of project duration on the scheduling method and its variation regarding each instance, it is now time to gather the respective flexibility data in order to evaluate to which extent these factors (the scheduling technique and the project instance) influence them.

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

Using the schedules obtained previously for the 480 psplib RCPSP instances, the schedule flexibility data was computed for the same scheduling techniques which are:

- An optimal schedule, obtained with the DH B&B procedure, identified by "Optimal duration" or "Opt";
- The best (minimal T) suboptimal schedule, obtained with the SSGS using all priority rules, identified by "Best SGSS" or "SSS";
- The professional schedule, obtained using MSProject, identified by "MSProject" or "MSP".

4.3.1 Slack

First of all, slack related values are calculated for the schedules under consideration that are expressed in the following two distinct ways, where V_e is the set of all activities of the project example e:

- $\#NC$: The number of activities that have positive slack, i.e., the number of non-critical activities of the project example (instance) according to the selected schedule. This value is calculated 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 (instance) which is equal to the sum of the slack of the non-critical activities because the slack of the critical ones is zero by definition. This value is calculated by the expression $\Sigma slack = \sum_{i \in V_e} slack_i$.

Due to the large number of values, results will be presented in more than one way with different degrees in detail. This will make an overall perception of them easier while still not disabling an individual per instance view.

a) Results for $\#NC$:

Figure 15 presents the values of $\#NC$ calculated for each instance regarding each scheduling method. The radar chart type used enables a clearer view of all values when compared to the more conventional linear x-y chart type due to the relatively small values of the y axis.

Values presented in the chart were also added to Table 24 which is included in Appendix IV and in Figure 16 the same data is displayed again in a linear x-y chart to facilitate the comparison with charts that follow and are expressed in this way.

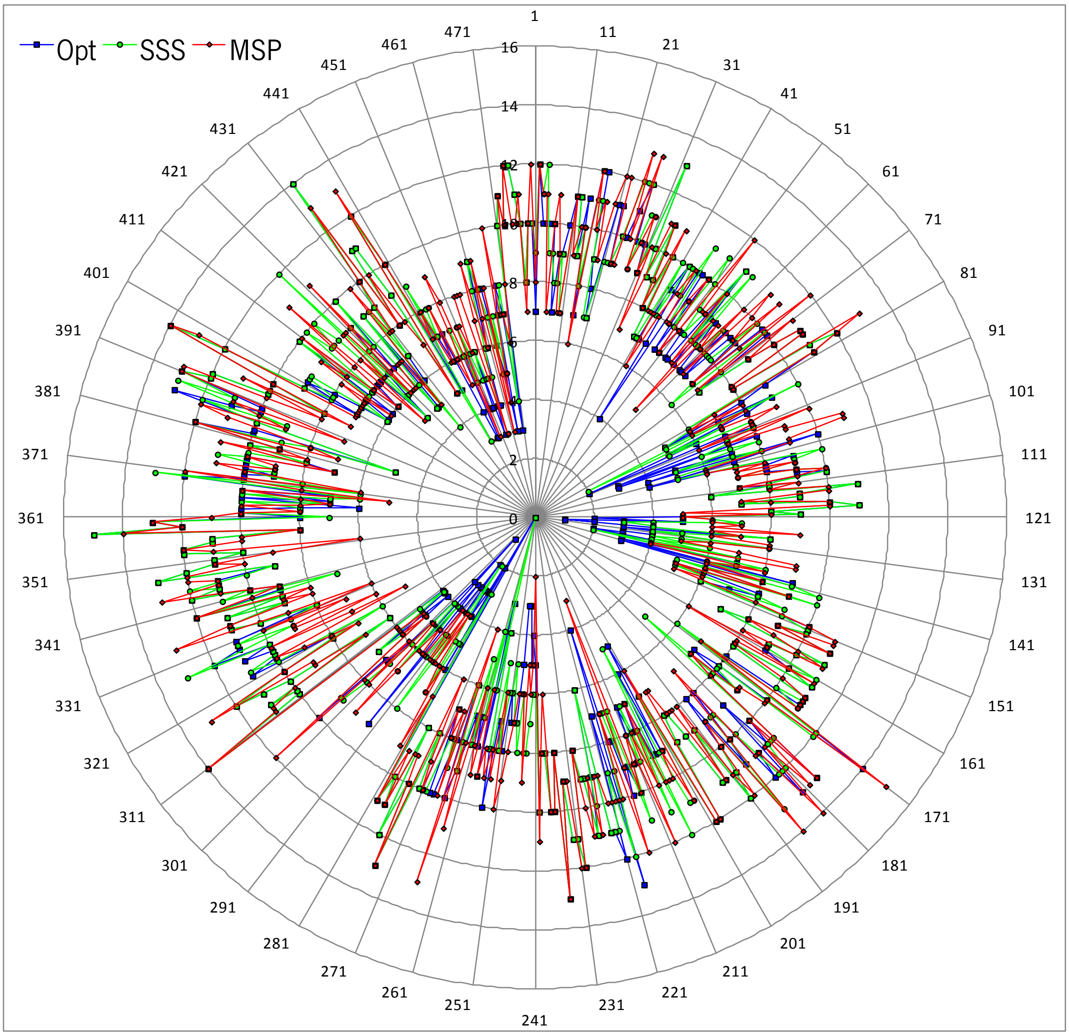


Figure 15: #NC for each scheduling method for all J30 instances (radar chart)

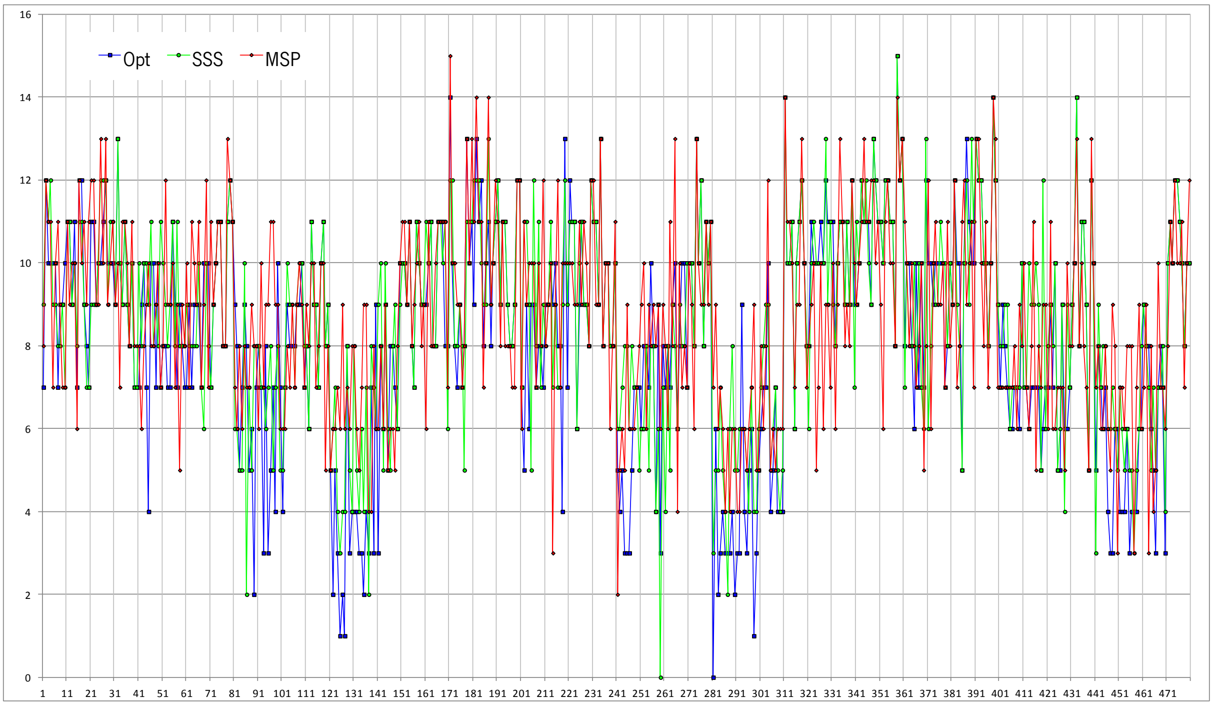


Figure 16: #NC for each scheduling method for all J30 instances (x-y chart)

From the chart, it can be seen that only in two instances there is no activity with slack and even in these cases, it happens with only one of the scheduling methods. On the other end, again, two instances have half their real activities with slack. In this case it occurs for distinct scheduling methods. All remaining cases lie between $0 < \#NC < 15$.

Figure 17 presents the frequency distribution of $\#NC$, normalized to 100%. It can be seen that the number of activities with slack lie in the interval [4,13] for more than 90% of the cases. This means that, in most cases, the number of activities with slack is between about 13.3% and 43.3% of the total number of real activities in a project, whatever the scheduling technique used.

"Opt" values also present a significant value of 5.4% of instances with $\#NC = 3$ which is equivalent to saying that a relevant amount of instances have 10% of their activities with slack. Nevertheless the previous conclusion remains valid.

However, it is worth recalling that the techniques used are optimised for the only objective of minimal duration and, therefore, are not tuned for maximizing slack.

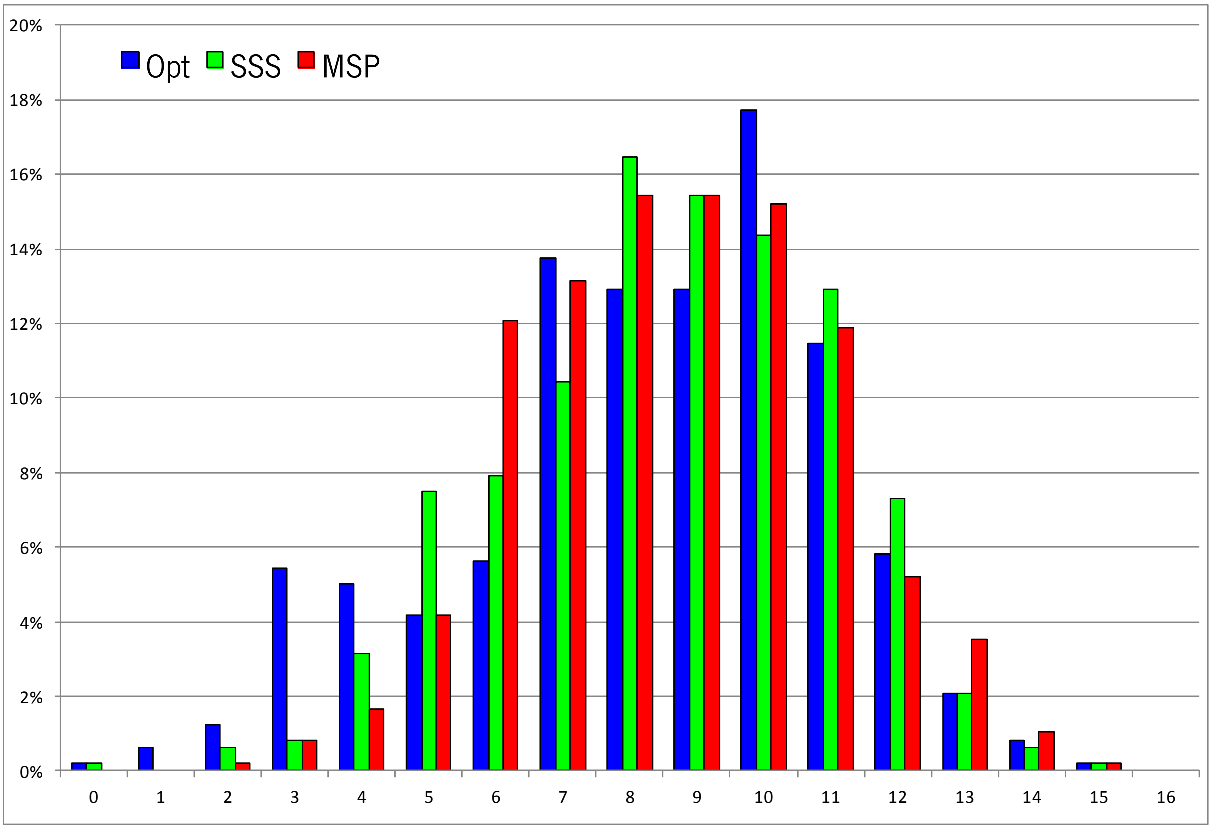


Figure 17: Normalized frequency of #NC for each scheduling method

J30 instances consist of ten randomly generated project examples for each triplet of the form $<NC, RF, RS>$ where NC, RF, RS are variable parameters set for the random project generator

"ProGen"[14]. A full description of the project generator and their parameters, including these ones, can be found in Kolisch, Sprecher and Drexl (1995). Generically they can be 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 J30 instances, they assume the following values:

- $NC \in \{1.5,\ 1.8,\ 2.1\}$: 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.

These parameters intend to generate project examples that cover a wide range of possibilities for real life projects regarding their constraints: precedence and resources.

Therefore, an intermediate aggregation of the results for $\#NC$ by computing its average value regarding each subset of instances with the same set of <NC, RF, RS> may help to establish the relation, if any, of $\#NC$ with these parameters. Each of the 48 calculated average values is presented in Table 23 of Appendix IV and all are shown in Figure 18. In the chart, each point in the x axis corresponds to a distinct triplet <NC, RF, RS>, one for each distinct combination of their established values (which are shown below the x axis).

---

[14] Available at http://www.om-db.wi.tum.de/psplib/files/progen-sfx.exe.

While the previous chart focuses on determining the relationship between $\#NC$ and the $<NC, RF, RS>$ parameters, analysing each individual subset of schedules generated for instances having the same $<NC, RF, RS>$ parameters enables to determine their independence.

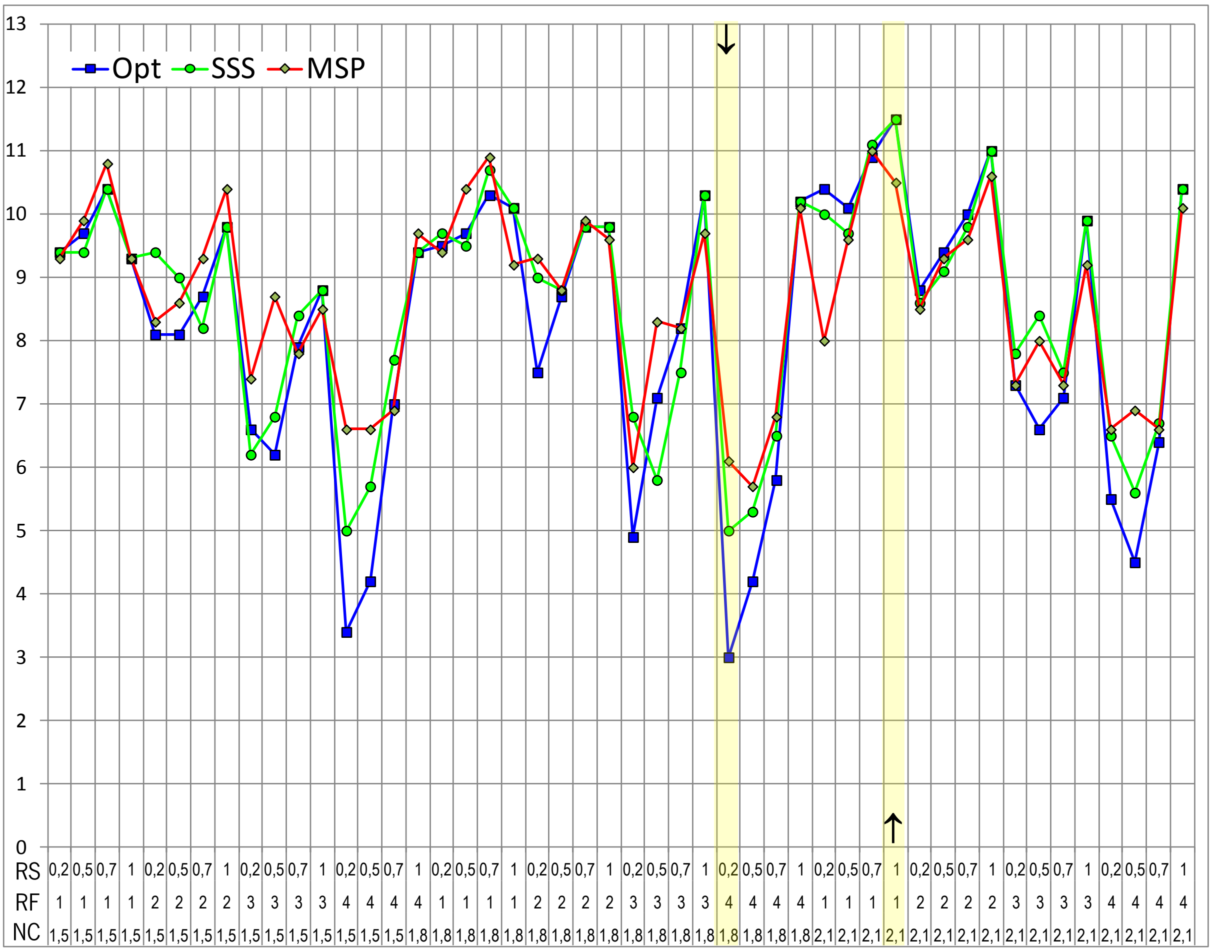


Figure 18: Average #NC for J30 instances with same <NC,RF,RS>

It would be fastidious to show all 48 charts so only two examples with detailed values are presented: the set whose average value is the lowest (minimal) and the set whose average value is the highest (maximal). These values are considered for all scheduling techniques and were obtained for:

- Minimal $avg(\#NC) = 3$ for $<NC = 1.8, RF = 4, RS = 0.2>$ using the optimal scheduling technique (Opt). This value is highlighted in Figure 18 and is identified with the symbol "↓". Figure 19 a) shows the detailed values of $\#NC$ for this set of instances;
- Maximal $avg(\#NC) = 11.5$ for $<NC = 2.1, RF = 1, RS = 1>$ using either the optimal scheduling technique (Opt) or the best SSGS (SSS). This value is highlighted in Figure 18 and is identified with the symbol "↑". Figure 19 b) shows the detailed values of $\#NC$ for this set of instances. In this case the values for "Opt" are the same as for "SSS".

All other charts can be obtained using the values presented in Table 24 included in Appendix IV.

It can be seen that, while the average values of each extreme case are quite different (around 5 and 11 respectively whatever the scheduling method used), each instance generated with the same parameters varies a lot, such that it is possible to find higher values for the most demanding case (lowest average) than some others in the less demanding case (highest average). The same can be verified the other way around, i.e., lower values can be found in the less demanding instances (highest average) than some others in the most demanding ones (lowest average).

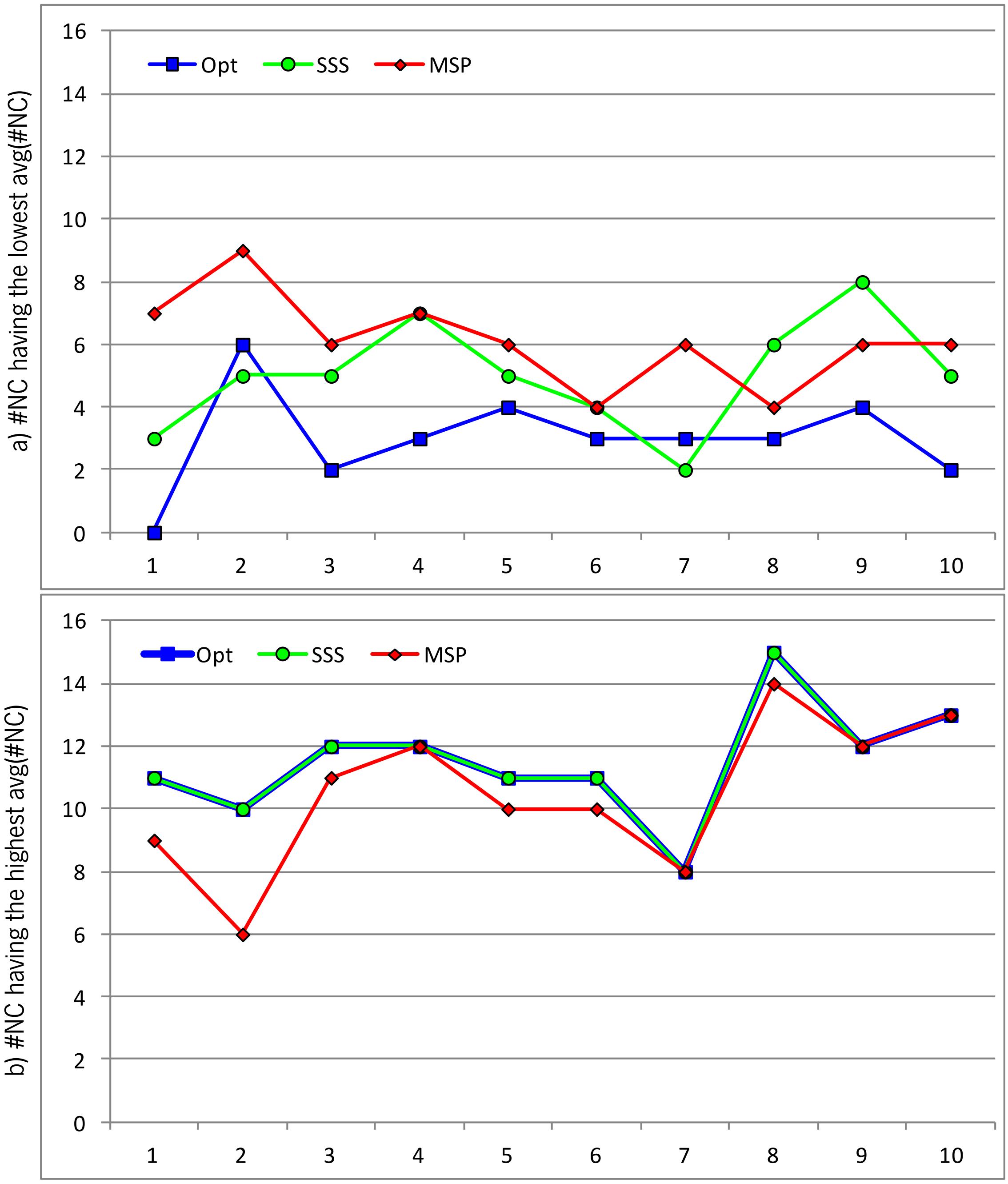


Figure 19: #NC for instances with same <NC,RF,RS> having avg(#NC) a) lowest and b) highest

b) Results for $\Sigma slack$:

Having presented the values regarding the number of activities that have slack ($\#NC$), the amount of slack that each project has when executed according to a specific schedule will be presented ($\Sigma slack$).

The chart of Figure 20 presents $\Sigma slack$ for each instance and scheduling technique. The option used for plotting the $\#NC$ values (a radar type chart) is not equally suitable for plotting the $\Sigma slack$ values because of the large number of possible values in the y axis. A larger chart is reproduced in Figure 34 in Appendix II (charts with higher legibility).

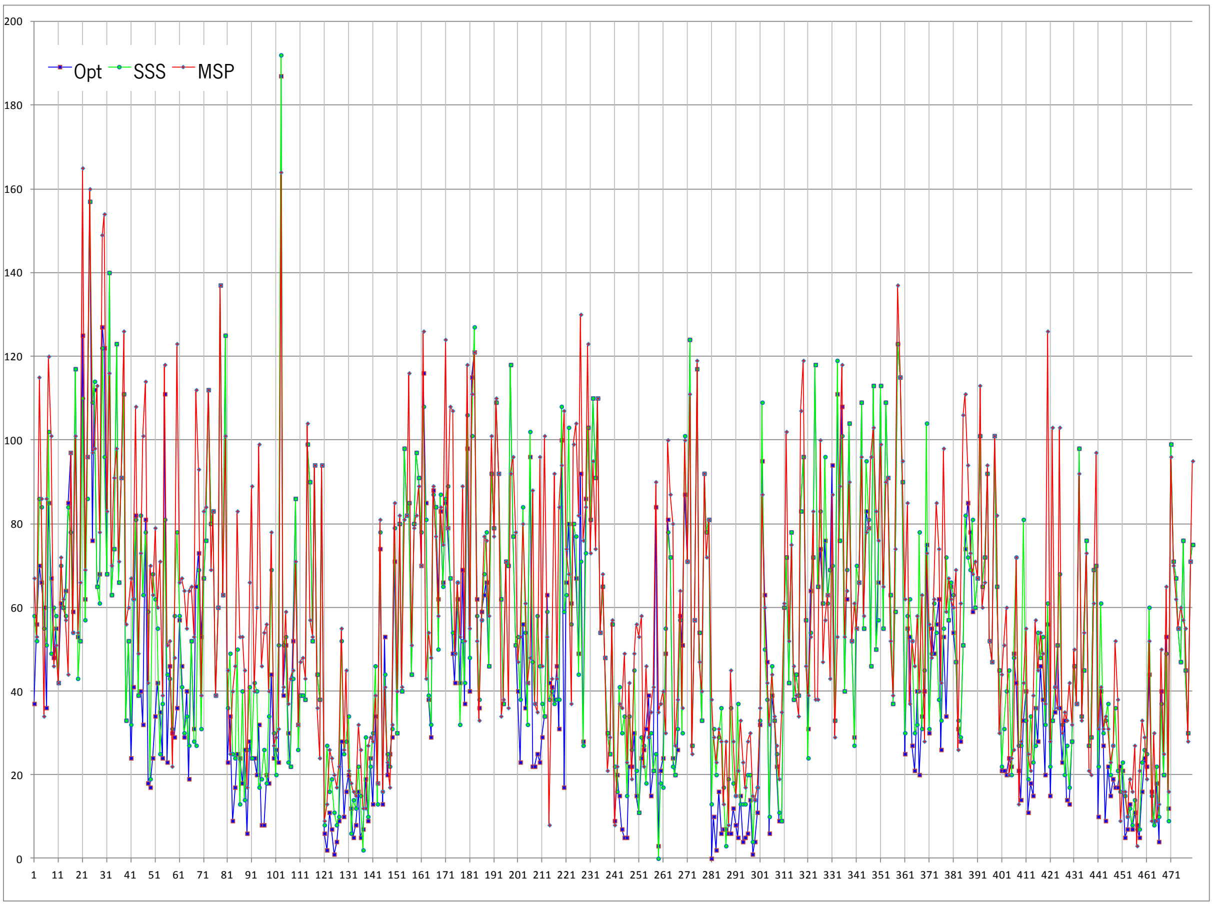


Figure 20: Σslack for each scheduling method for all J30 instances

Similarly to the previous case, an intermediate aggregation of the results for $\Sigma slack$ will be calculated by computing its average value regarding each subset of instances with the same set of $<NC, RF, RS>$ to help to establish the relation, if any, of $\Sigma slack$ with these parameters. The results are presented in Table 23 of Appendix IV and are shown in Figure 21. Again, each point on the x axis corresponds to a distinct triplet $<NC, RF, RS>$, one for each distinct combination of their established values.

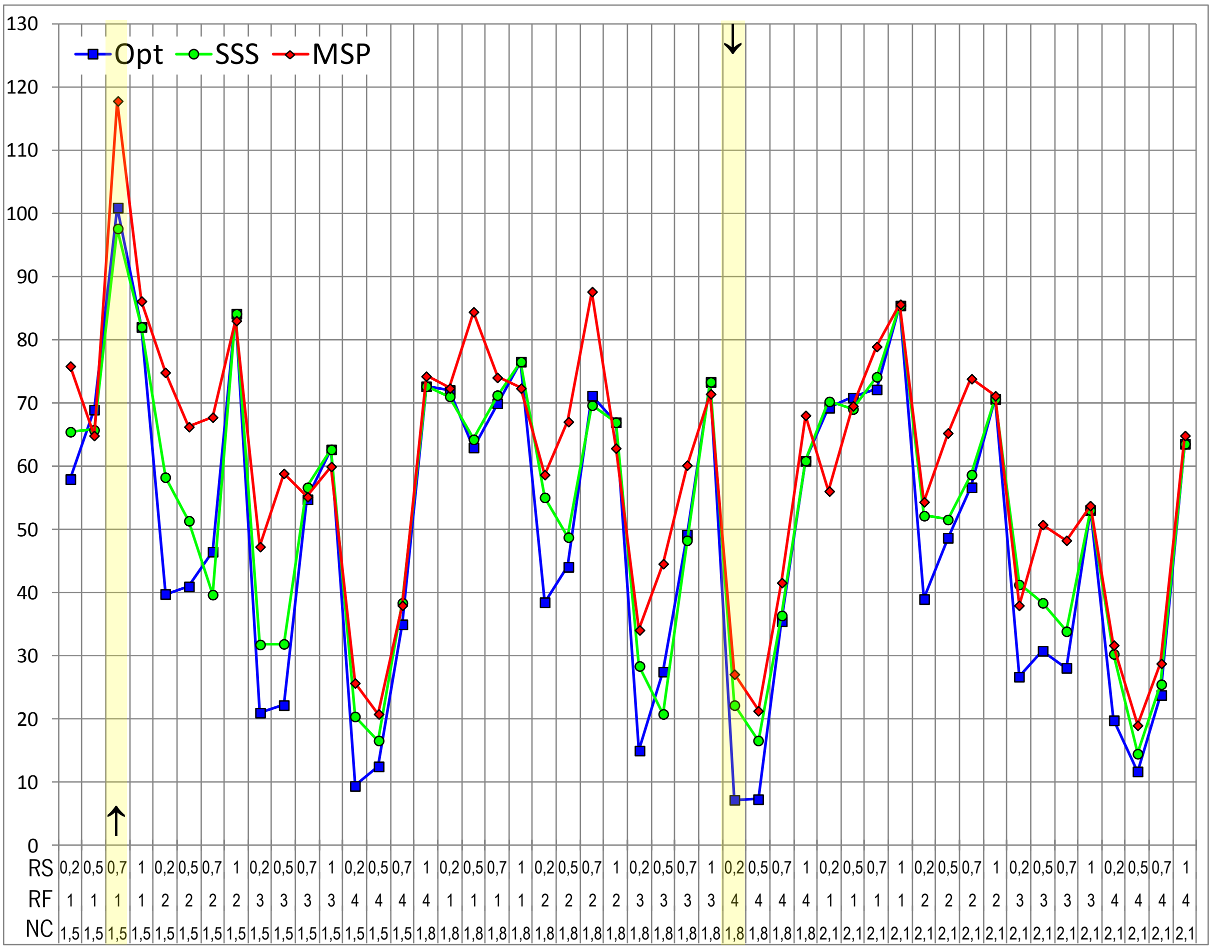


Figure 21: Average Σslack for J30 instances with same <NC,RF,RS>

To analyze the independence of $\Sigma slack$ regarding each set of schedules generated for instances having the same <NC, RF, RS> parameters, two examples with detailed values are presented: the set whose average value is the lowest (minimal) and the set whose average value is the highest (maximal). These values are considered for all scheduling techniques and were obtained for:

- Minimal $avg(\ \Sigma slack) = 7.2$ for <$NC = 1.8, RF = 4, RS = 0.2$> using the optimal scheduling technique (Opt). This value is highlighted in Figure 21 and is identified with the symbol "↓". Figure 22 a) shows the detailed values of $\Sigma slack$ for this set of instances;

- Maximal $avg(\ \Sigma slack) = 117.9$ for <$NC = 1.5, RF = 1, RS = 0.2$> using the MSProject scheduling technique (MSP). This value is highlighted in Figure 21 and is identified with the symbol "↑". Figure 22 b) shows the detailed values of $\Sigma slack$ for this set of instances. The average values for "Opt" and "SSS" are also maximal within their respective category (scheduling technique).

Once again, all other charts can be obtained using the values presented in Table 24 included in Appendix IV).

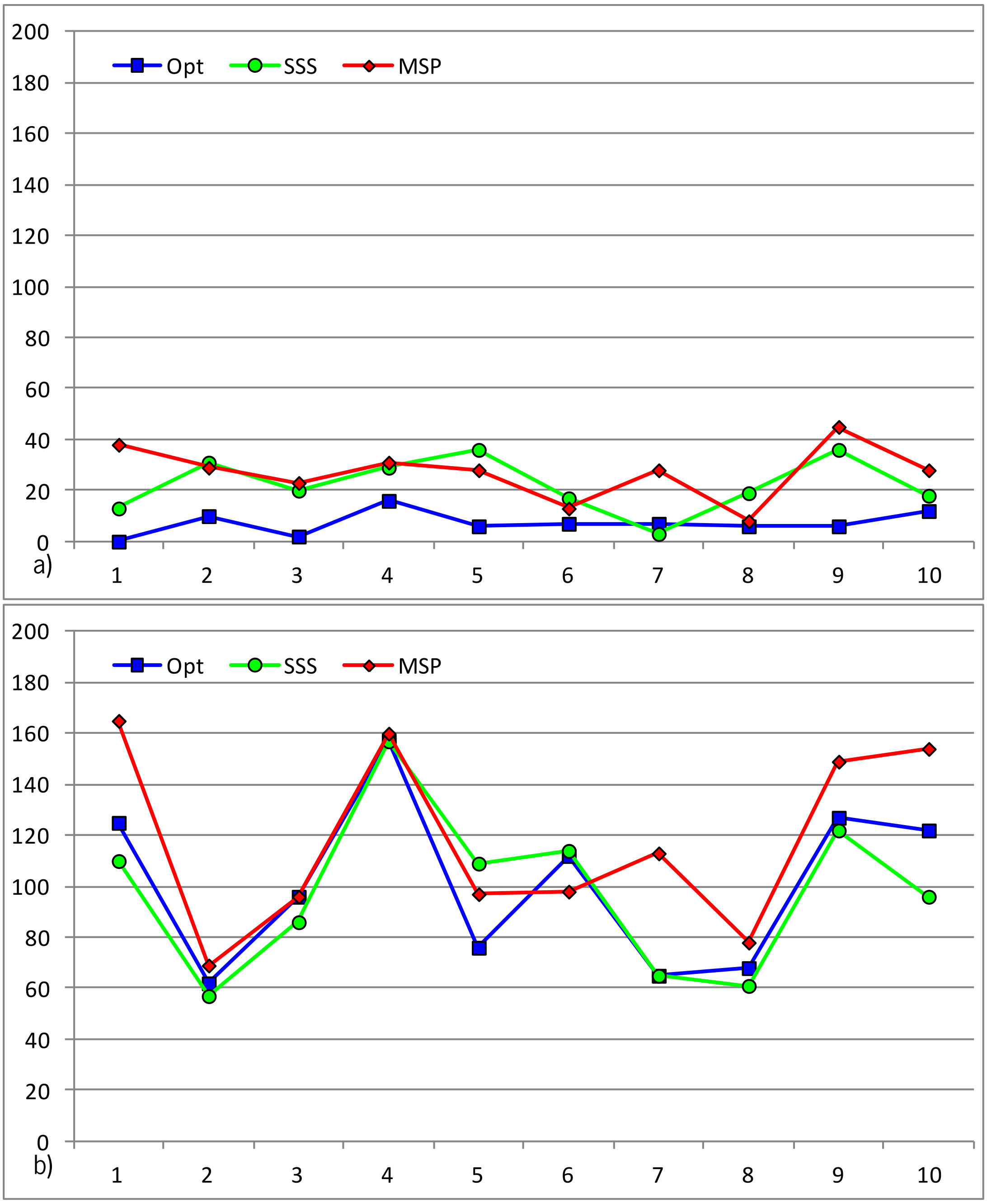


Figure 22: Σslack for instances with same <NC,RF,RS> and avg(Σslack) a) Lowest b) Highest

In this case, it can be seen that the average value of each extreme case is still quite different and each instance generated with the same parameters still varies but values for the more demanding case (lowest average) never reach values within the range of the less demanding ones (highest average). The same can be verified the other way around, i.e., values in the less demanding instances (highest average) never reach values in the range of the more demanding ones (lowest average).

c) Aggregated results for $\#NC$ and $\Sigma slack$:

Finally, aggregated values for $\#NC$ and $\Sigma slack$ for all instances, regarding their overall minimum (min), their average (avg) and their maximum (max) are calculated and presented in Table 14.

Table 14: Slack aggregated values for J30 instances

	$\#NC$			$\Sigma slack$		
	min	avg	max	min	avg	max
Optimal duration (Opt)	0	8.25	15	0	49.19	187
Best SSGS (SSS)	0	8.57	15	0	52.91	192
MSProject (MSP)	2	8.66	15	3	59.50	165

Additionally, relative aggregated values of all instances are also presented in Table 15, for:

- $\#NC/30\ (\%)$: Percentage of $\#NC$ over the number of real activities in the project (which are always 30);
- $(\Sigma slack)/T\ (\%)$: Percentage of $\Sigma slack$ over the project's duration when executed with the schedule generated by the considered scheduling method. Values for T (project duration) are presented for each instance and scheduling method in Table 24 of Appendix IV).

Table 15: Relative slack aggregated values for J30 instances

	$\#NC/30$ (%)			$(\Sigma slack)/T$ (%)		
	min	avg	max	min	avg	max
Optimal duration (Opt)	0.00%	27.49%	50.00%	0.00%	87.49%	268.63%
Best SSGS (SSS)	0.00%	28.56%	50.00%	0.00%	91.61%	268.63%
MSProject (MSP)	6.67%	28.86%	50.00%	4.76%	98.67%	268.63%

It can be seen that while some instances have no activities with slack (which are only two and for distinct scheduling techniques as identified before), on average, more than 25% of these project's activities do have slack whatever the scheduling technique. The number of activities with slack in the project example used in chapter 3, just by chance, was precisely 25% (3/12). Also, for all scheduling techniques, the upper limit (maximum) of the percentage of $\#NC$ is 50%.

As might be expected, the values for optimal schedules (Opt) have a tendency (but not always) to be lower than the values obtained for the other scheduling techniques. This remains valid for the corresponding weighted values as is the case of $\Sigma slack$ when weighted by $1/T$.

4.3.2 SIF - Schedule Intrinsic Flexibility

Having presented the results for plain sums of slack per project, the weighted version can be computed and presented. SIF, the schedule intrinsic flexibility, as defined in (3.24) and (3.25), denotes such a weighted version which is a better measure for the schedule's flexibility than slack for the reasons explained when the concept was explained.

In order to provide more detailed values, results will be presented for the first expression, which is reproduced here: $SIF_k = \sum_{i=1}^{n} r_{ik} slack_i\ ,\ for\ each\ k \in K, for\ all\ i \in V.$

In this case, the detailed values for each instance are presented only in Table 27 of Appendix IV since their presentation in a chart or a set of charts would not be easy to read.

Therefore, computed values are presented, once again considering their average values regarding each distinct triplet <NC, RF, RS>, in Figure 23. It comprises four charts, one for each resource type, in which average values are shown for each scheduling technique.

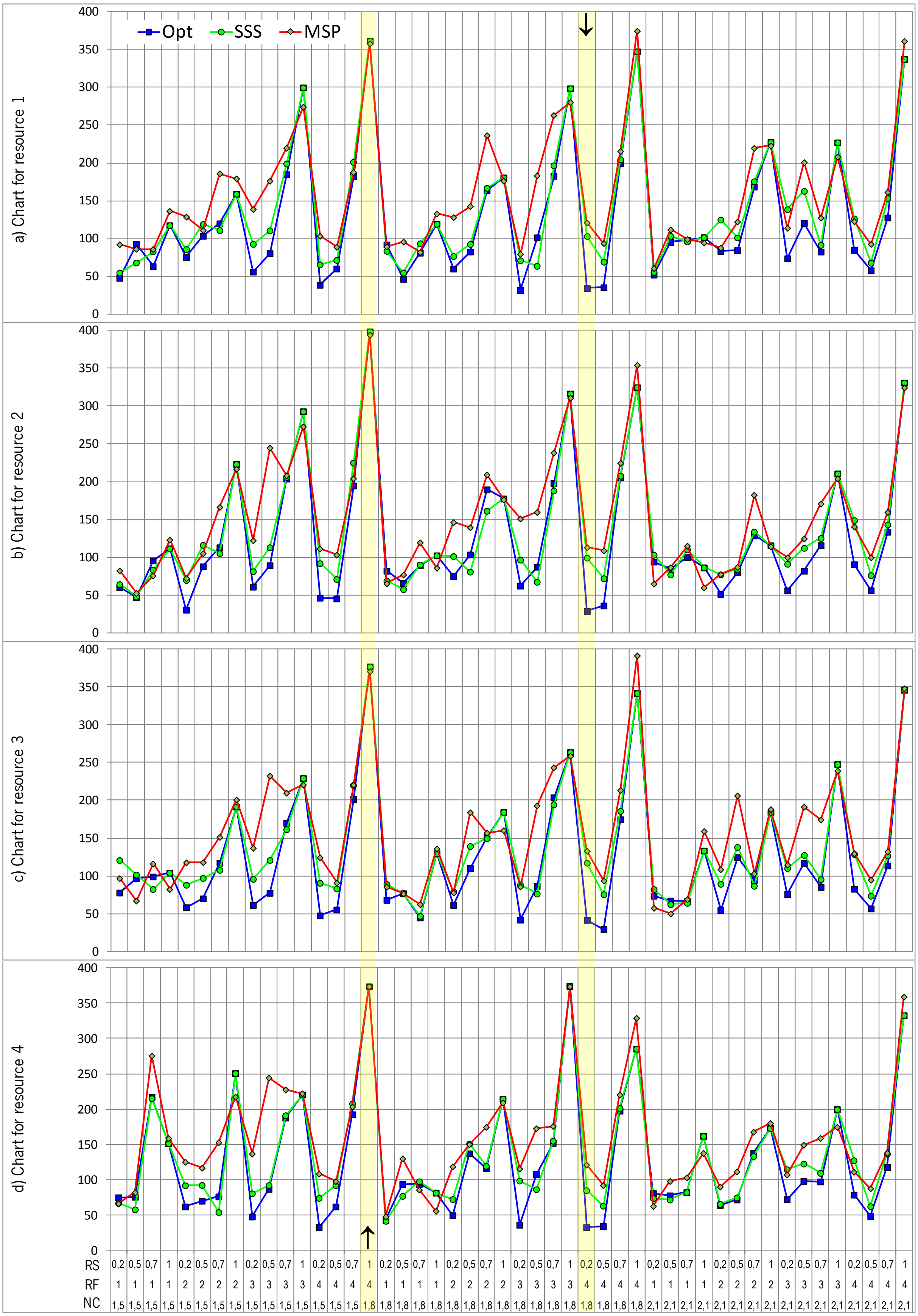


Figure 23: Average SIF for instances with same <NC,RF,RS> for each resource type

To complement the aggregated view, two examples detailing each of the values for instances having the same $<NC, RF, RS>$ parameters are presented. Once again, the selected instances are the ones that belong to the set that has the instance with the lowest (Figure 24) and the highest (Figure 25) value of $avg(SIF_k)$.

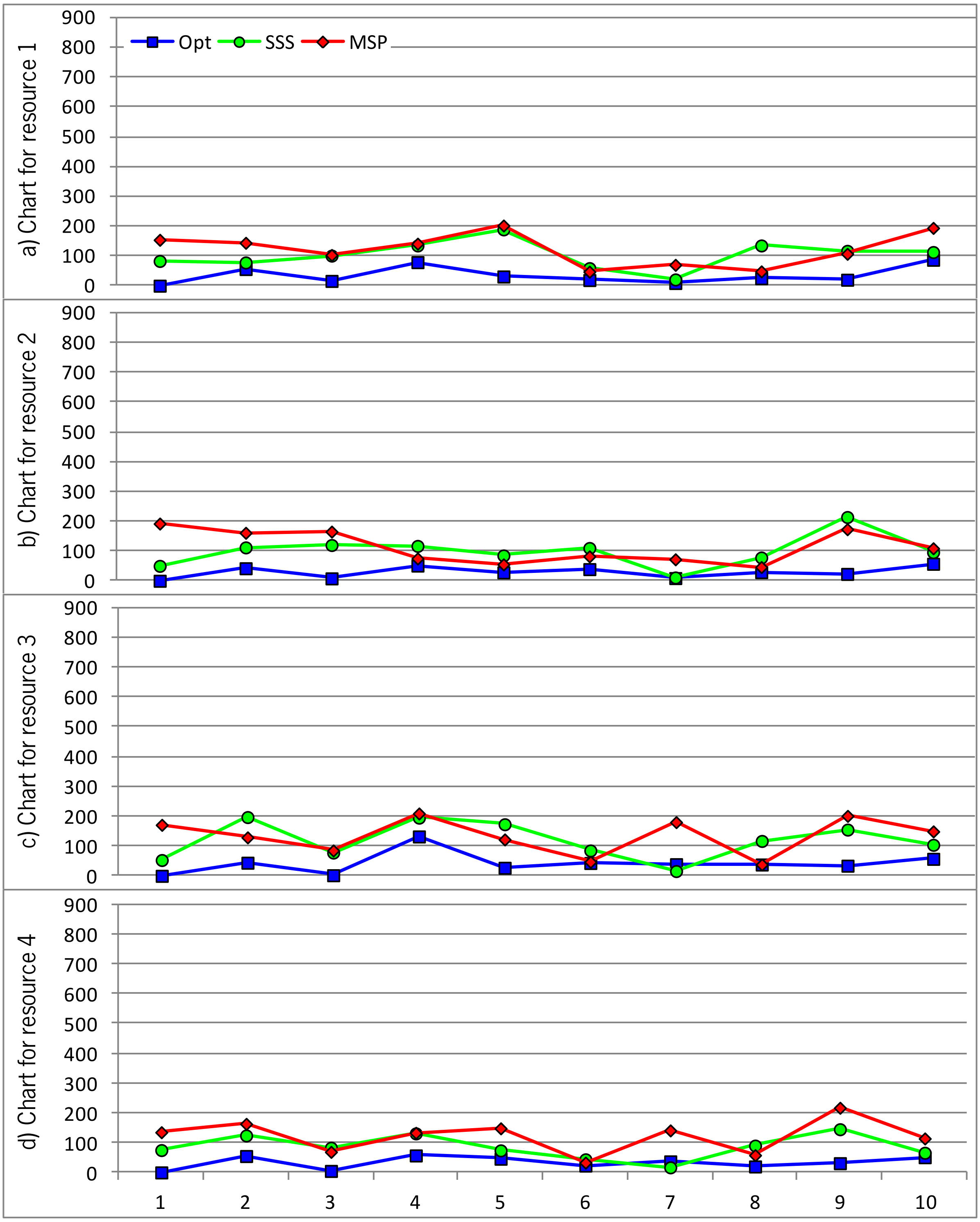


Figure 24: SIF for instances with same <NC,RF,RS> for each resource type with lowest avg(SIF)

The lowest $avg(SIF_k)$ is 28.6 corresponding to resource 2 and $<NC = 1.8, RF = 4, RS = 0.2>$ instances (filenames of the form J3029_*) obtained with the optimal scheduling technique (Opt).

The highest $avg(SIF_k)$ is 398.6 corresponding to resource 2 and $<NC = 1.8, RF = 4, RS = 1>$ instances (filenames of the form J3016_*) obtained with both the optimal (Opt) and heuristic (SSS) scheduling techniques.

Figure 25: SIF for instances with same <NC,RF,RS> for each resource type with highest avg(SIF)

Next, aggregated values for SIF are presented in Table 16 for minimum, average and maximum values for each resource type and scheduling technique.

Table 16: SIF aggregated vales for each resource type and scheduling technique

	r_1			r_2			r_3			r_4		
	min	avg	max	min	avg	max	min	avg	max	min	avg	max
Optimal duration	0	134.6	667	0	132.0	689	0	130.6	655	0	130.8	670
Best SSGS	0	147.7	755	0	140.5	735	0	142.8	655	0	139.9	670
MSProject	0	165.5	673	0	157.1	846	0	160.3	818	0	157.0	670

Finally, a relative measure concerning the ratio of SIF and the sum of the required resources is considered.

Values are calculated using the expression $SIF_k / \sum_i (d_i r_{ik})\ ,\ for\ each\ k \in K, for\ all\ i \in V$ and are presented as a percentage in:

- Figure 26: for its average values for each <NC, RF, RS> and each scheduling technique, using a chart for each resource type;
- Table 17: minimum (min), average (avg) and maximum (max) values aggregated for each resource type and each scheduling technique.

This analysis does not affect the relative position regarding each scheduling technique once it is kept due to the fact that the weighting factor $1/\sum_i (d_i r_{ik})$ is the same for each instance. The purpose of this analysis is to evaluate the SIF's magnitude when compared to the overall resource consumption of the project, expressed in its global work content $w_k = \sum_i (d_i r_{ik})$.

It can be seen that these values have a significant variation that can, in some cases, exceed 100%.

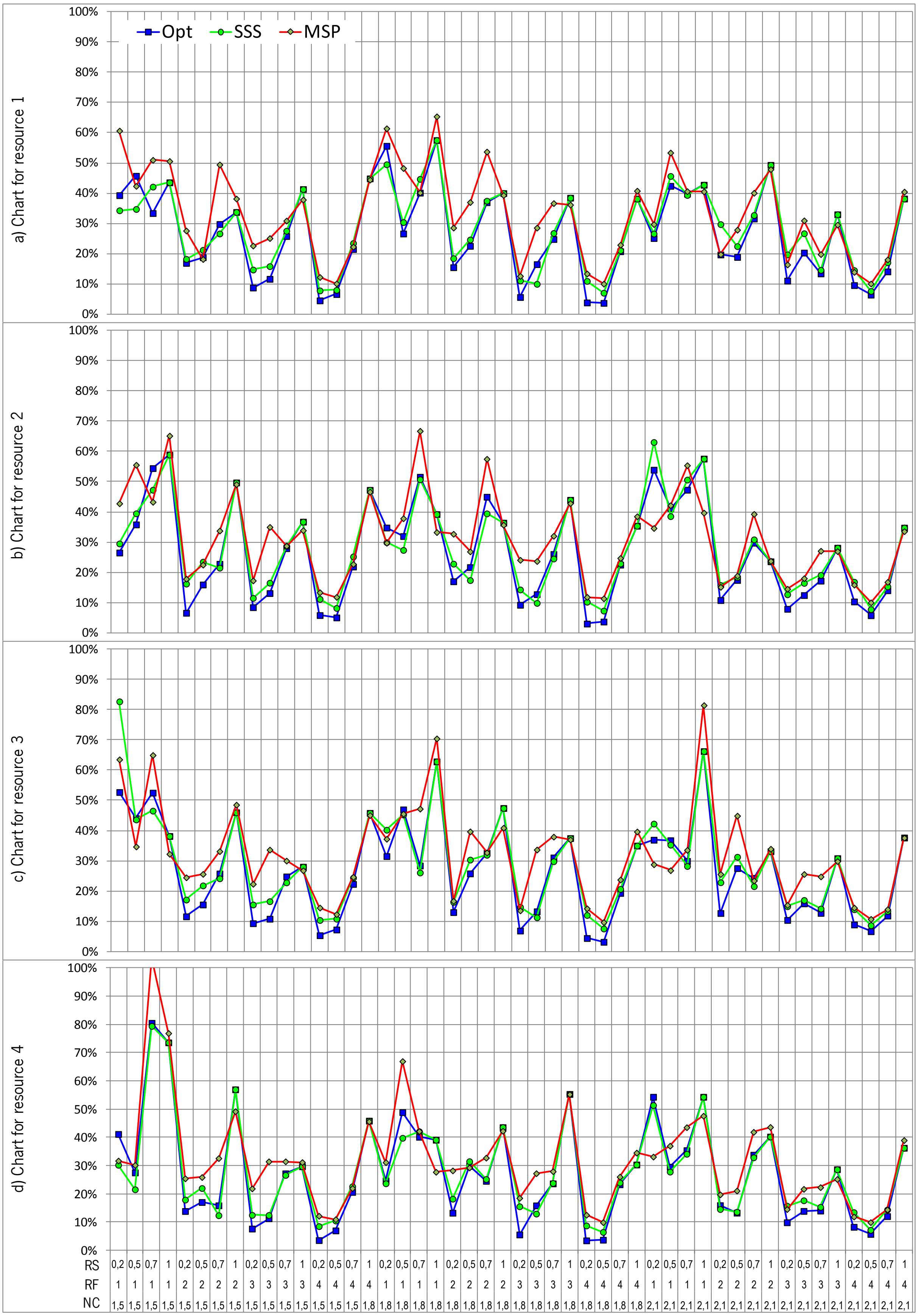


Figure 26: Average SIF/Σ(dr) for instances with same <NC,RF,RS> for each resource type

It is significant that the global average values are between 26.6% and 33.3% with extreme (max) cases near 300% (MAP has a max value for r_1 of 298.6%).

Table 17: SIF/Σ(dr) aggregated values for each resource type and scheduling technique

	r_1			r_2			r_3			r_4		
	min	avg	max	min	avg	max	min	avg	max	min	avg	max
Optimal duration	0.0%	27.5%	225.7%	0.0%	27.1%	191.7%	0.0%	26.9%	176.4%	0.0%	26.6%	209.3%
Best SSGS	0.0%	29.4%	202.7%	0.0%	28.1%	191.7%	0.0%	28.7%	176.4%	0.0%	27.4%	209.3%
MSProject	0.0%	33.3%	298.6%	0.0%	30.5%	240.4%	0.0%	32.0%	238.5%	0.0%	30.4%	153.7%

4.3.3 Analysis

In this section the focus is on flexibility regarding schedules that are generated by scheduling techniques that primarily focus on minimizing the projects' makespan. No concern is given to the limitation imposed by resources other than the deterministic availability and estimated requirements. This means that:

- On the one hand, their combination with the resource flexibility parameters can limit the possibility to take advantage of the previously presented values and
- On the other hand, better schedules can be searched for, using new or modified scheduling techniques that can further enhance the presented values.

Then, both these statements are addressed.

4.4 Flexibility data: resource flexibility

Flexibility existing in a given deterministic baseline schedule resides in slacks that some activities may have. In the previous section, the amount of slack that a test set of projects, scheduled with typical scheduling techniques, was computed. Almost all instances had available slack to be explored to cope with uncertainties that might arise during project execution. But, to be able to explore the flexibility resulting from the existence of slack while not changing the project's duration, and also not changing the activity start times, resources have to be flexible. Next, the impact of setting some resource flexibility parameters on constraining schedule flexibility will be studied.

4.4.1 Optimal α^-, α^+

As was established before, the resource flexibility parameters α^- and α^+ have optimal minimal values regarding activity durations in the sense that those values allow that all activities can benefit from resource flexibility.

To enable all activities to contribute with their slack, from expression (3.12) it is set that α^- has to comply with the condition $\alpha^- \geq \frac{1}{d_i^{nom}+1}$.

Similarly, to enable all activities to benefit from slack, from expression (3.13) it is set that α^+ has to comply with the condition $\alpha^+ \geq \frac{1}{d_i^{nom}-1}$.

The test set used (J30) is generated according to several parameters from which the group of "variable parameters" were already explained. There are two other groups, the "fixed parameters" group and the "base parameters" group. One of the "base parameters", d_j, the possible duration of any non-dummy activity of any instance in the test set, deserves special mention because of its great impact in this analysis. According to Kolisch and Sprecher (1997), $d_j \in [1,10] \cap \mathbb{N}$, that is, besides being an integer, must lie in the interval [1,10].

If one considers that durations are expressed in days, the following immediate conclusions can be 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 5 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 following optimal resource flexibility parameters are required:

- $\alpha^- \geq 16{,}7\%$;
- $\alpha^+ \geq 25\%$.

A more detailed analysis will follow considering the first scenario which is the more demanding case.

4.4.2 Impact of varying α^-, α^+

As previously stated, the time unit to be considered is a day. Furthermore, the test will consider flexible resource parameters values that lead to integer working hours per day. Table 18 presents such possible values.

Table 18: Resource flexibility parameters

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

While α^- cannot assume values greater than 100%, α^+ can assume values up to 200%. However, regarding this analysis, α values to be considered are limited to more reasonable ones (if human resources are considered) that are highlighted in Table 18 ($0 < \alpha \leq 50\%$).

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

Figure 27 presents the values based on the number of activities that have slack and whose duration satisfies expression (3.10), that is: $y = \left|i \in V, slack_i > 0, d_i^{nom} \geq \left\lceil \frac{1}{\alpha^-} - 1 \right\rceil\right|$.

More specifically, to make reading easier, the presented values are the average of the previously computed values for all instances having the same <NC, RF, RS>.

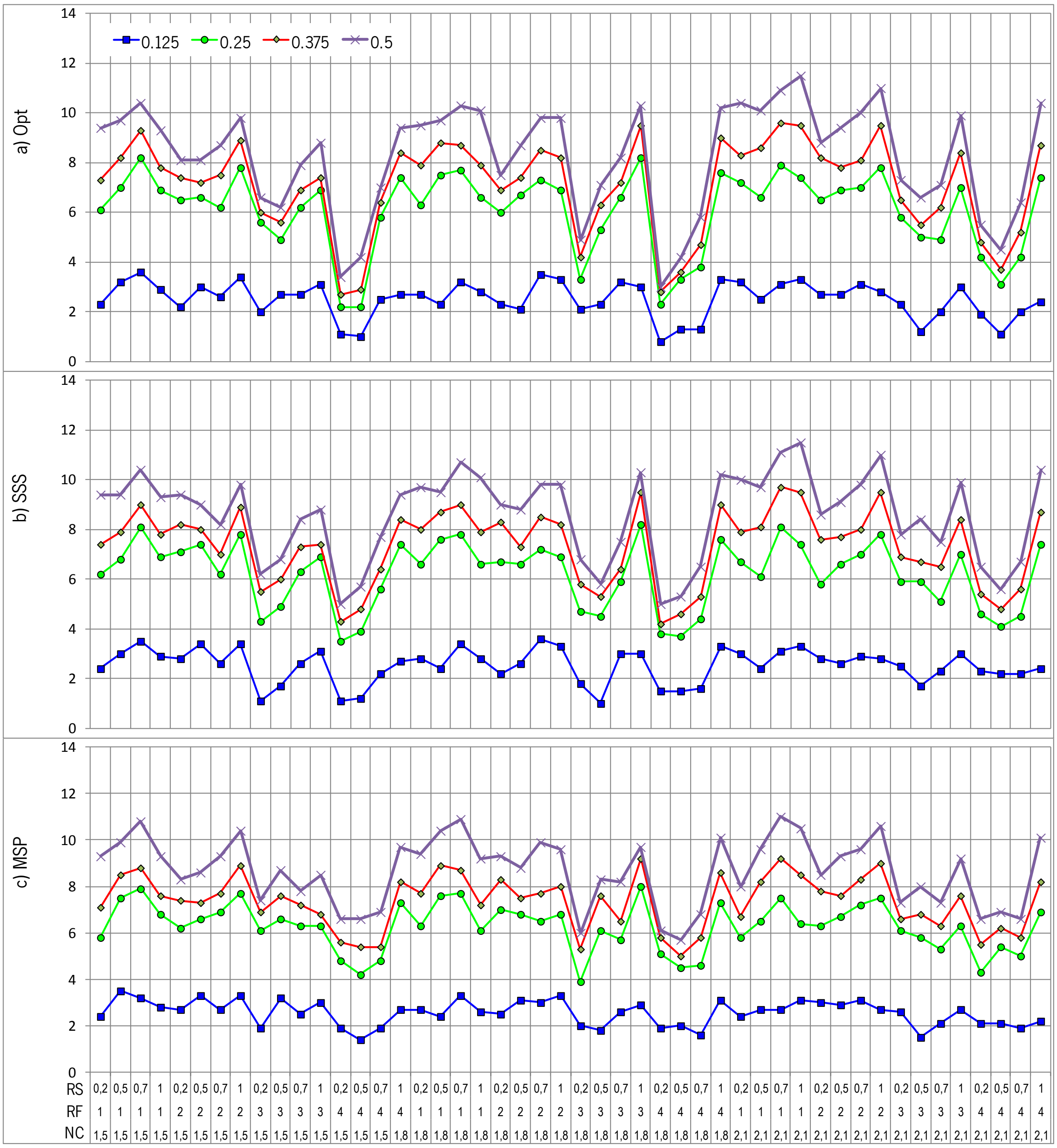


Figure 27: Number of activities with slack according to α^- for each scheduling technique

Keeping in mind that values for $\alpha^- = 50\%$ are the same as $\#NC$ values, because for this value of α^- all non-critical activities can participate with slack, the curve representing that value in the charts is the upper limit of all the other ones.

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 distribution on the test set (randomly distributed) which, combined with the values of minimal durations according to α^-, have that effect. The values of such minimal durations are presented in Table 19.

Table 19: Activity minimal durations regarding α^-

Values for 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%	$d_i^{nom} \geq 1$

The same conclusion can be drawn when analysing average values for all instances as can be seen in the following table:

Table 20: Average number of activities with slack according to α^- for each scheduling technique

$\alpha^-=$	**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 as identified in a baseline schedule, it can also be applied to non-critical ones if, for some reason, they become critical. The goal is then to determine which non-dummy 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 28 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 <NC, RF, RS>, plotted for each selected value of α^+. These values are computed, for each instance, according to expression (3.11), that is $y = \left|i \in V, d_i^{nom} \geq \left\lceil \frac{1}{\alpha^+} + 1 \right\rceil\right|$.

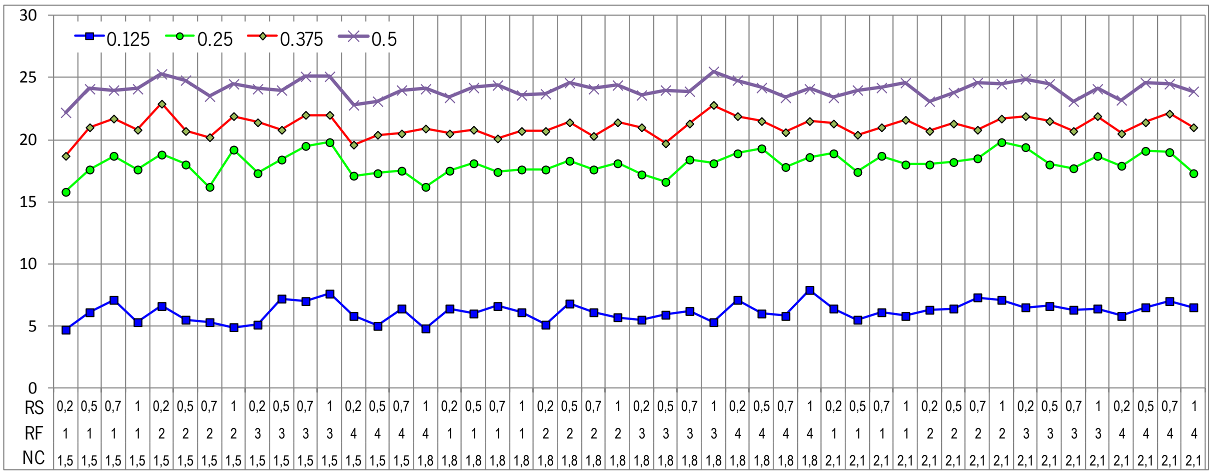


Figure 28: Number of activities that can benefit from resource flexibility according to α^+

The upper limit for any of the plots (the theoretical maximum value) is, obviously, 30 (all non-dummy activities in the project) 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 21.

Table 21: Activity minimal durations regarding α^+

Values for 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 (the relevant variable to $\sum slack$ and SIF), 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.

4.5 Enhance schedule flexibility

Having presented values for all relevant flexibility variables using some of the most relevant scheduling techniques applied to the well-known and widely used PSPLIB J30 test set, a final topic will be studied in order to analyse if the schedule flexibility can be enhanced by using the scheduling techniques at hand.

Actually, MSP (Microsoft Project 2013) will not be considered in this topic because, although being possible (Trautmann & Baumann, 2010), to change MSP behaviour regarding scheduling would contradict the goal that led to its choice in the first place, which is to use a "standard professional" scheduling tool.

Therefore, the implemented scheduling algorithms, minimal project duration using SSGS (Serial Generation Scheduling Scheme) with 15 priority rules, as defined in topic 3 of Annex I, and DH-B&B (Demeulemeester and Herroelen Branch and Bound), as defined in topic 1 and 2 of Annex I, were changed so that better schedules could be found regarding their flexibility as defined in this thesis.

Only one measure of schedule flexibility was selected to be enhanced to demonstrate the potential of this approach but, similarly, any other one can be used.

The selected measure is $\Sigma slack$ as defined previously: $\Sigma slack = \sum_{i \in V_e} slack_i$ where V_e is the set of activities of the instance (project example) e belonging to the J30 test set.

The study will start with the SGSS based procedure followed by the DH-B&B based procedure.

4.5.1 Heuristic procedure

The SGSS based procedure used so far determined a minimal duration schedule for each predefined priority rule and selected the best one, that is, the minimal duration one.

Now, the procedure will be the same but the best one to be selected is the one with the maximal value for $\Sigma slack$. In short, the selected schedule is the one with maximal $\Sigma slack$ from the minimal duration ones (which is not necessarily the minimal duration one).

Figure 29 presents the computed values regarding the SGSS based procedure as described above. Once more, not to overload the chart, all values are presented with their average values for sets having instances with the same <NC, RF, RS> parameters.

The chart presents averages calculated from the following values:

- T1: The minimal project duration, represented as a bar (the lower bar);

- T2: The project duration for the schedule with maximal $\Sigma slack$, represented as a bar (only the upper part is visible over T1);
- S1: $\Sigma slack$ for the schedule with minimal duration, represented as squares over a line (the lower line);
- S2: $\Sigma slack$ for the schedule with maximal $\Sigma slack$, represented as dots over a line (the upper line);

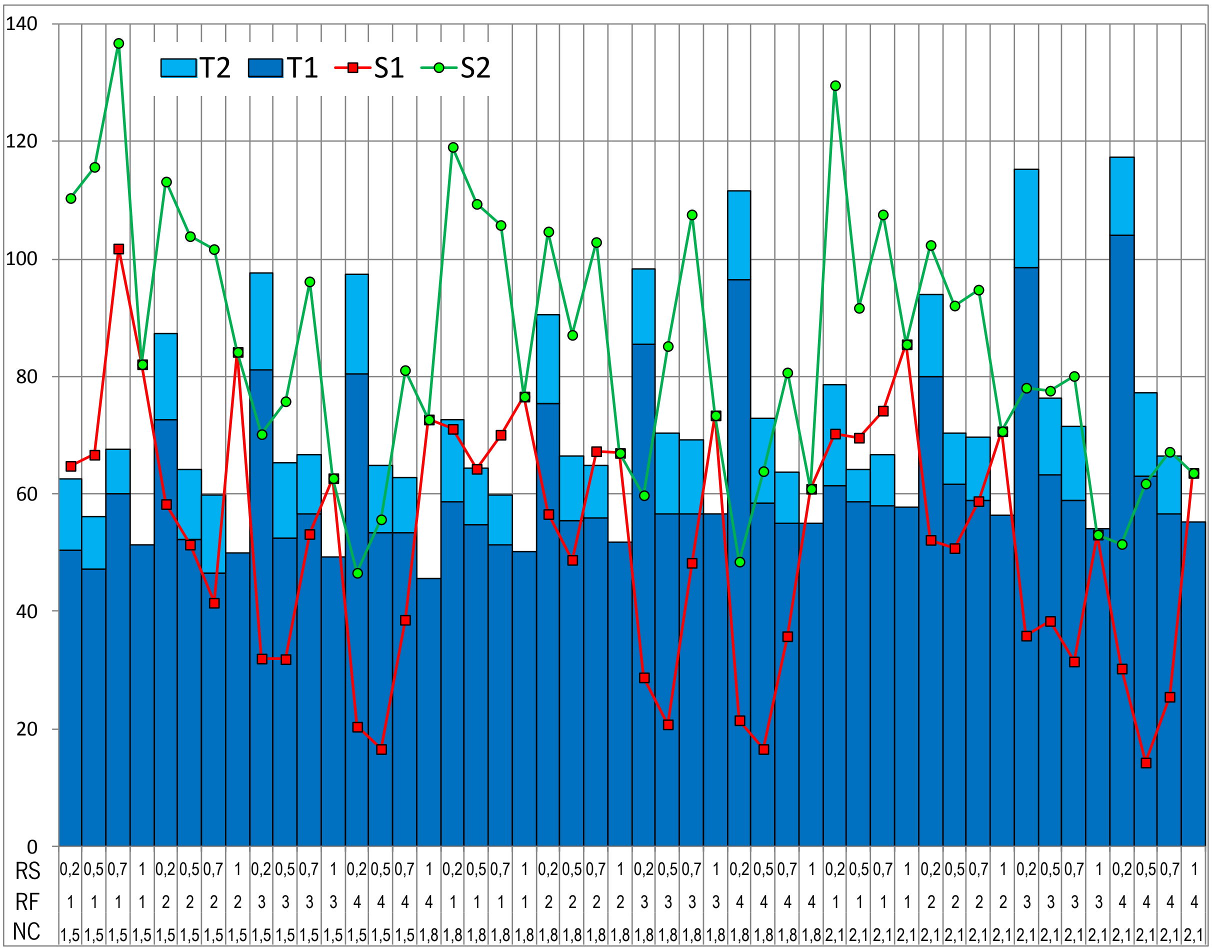


Figure 29: SGSS based enhanced schedule flexibility

Analysing these averages, one can conclude that to enhance $\Sigma slack$ always means an increase in project duration and that some increase in project duration greatly improves $\Sigma slack$. In fact, an average increase in project duration of less than 15% (14.9%) resulted in an average increase of more than 60% (60.7%) in $\Sigma slack$. Nevertheless, there can be additional priority rules that can lead to better results that are worth looking for.

Regarding this, Figure 30 presents an overview of which priority rules contributed to the best solutions. The chart presents the number of schedules that were generated for each priority rule that have:

- minT: minimal project duration, represented with diamonds over a line;
- maxSLK: maximal $\Sigma slack$, represented as squares over a line.

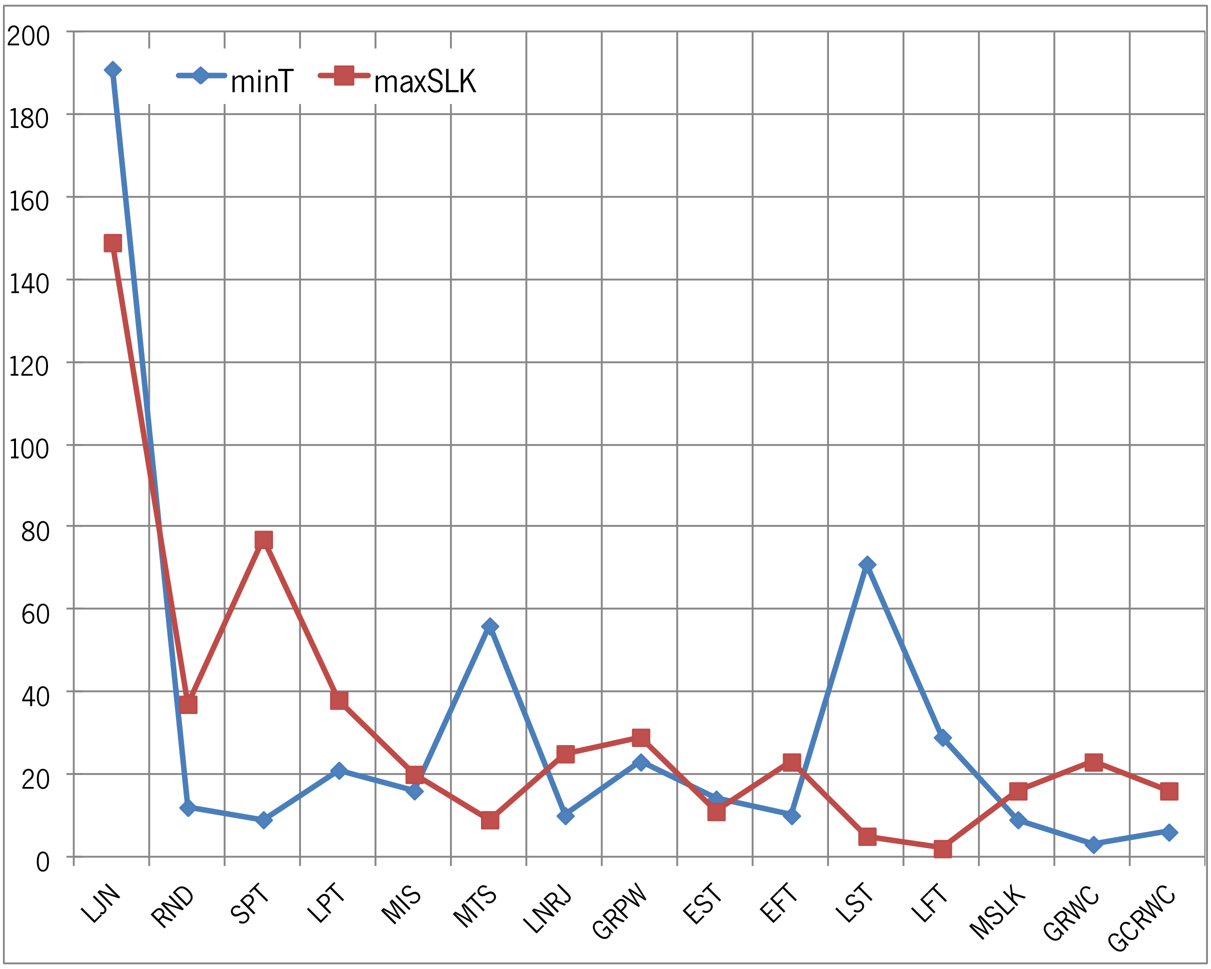


Figure 30: Priority rules' relation with enhanced schedule flexibility

Notice that, while still the major contributor, LJN (Lowest Job Number) loses importance when comparing the sole objective of minimal duration, while SPT (Shortest Processing Time) becomes important. On the other end, MTS (Most Total Successors) and especially LST (Latest Starting Time) also lose importance in searching for better values of $\Sigma slack$.

A final note to remind that the "scheduling information" based priority rules refer to CPM schedules (no resource constraints) which justifies the irrelevance of the MSLK (Minimum Slack).

4.5.2 Modified DH B&B

The DH-B&B procedure will be modified to store not the first optimal candidate solutions it finds, but the one, among all the optimal candidates it finds, that has the maximal $\Sigma slack$. This modification involves changes similar to the ones generically described in 3.4.1, more specifically:
Typical branch-and-bound (B&B) scheduling methods search for minimum project duration schedules that are optimal for this objective which means that once the optimal solution is achieved they stop the search. Generically, they go through all possible solutions that respect the precedence and resource constraints, saving the best up to date solution which, when the process finishes, is guaranteed to be optimal.

This process, whatever the method used to generate each solution, is exponential regarding the number of activities in the project leading to an exponential computational time to reach the optimum. To speed up the search process, although not in a deterministic way, B&B methods use some pruning rules that try to discard, as soon as possible, solutions that cannot be optimal.

The DH-B&B method uses such pruning rules which discard solutions that are as good as the best one so far. In this way, potential alternative (to the existing one) optimal solutions may be discarded. Among these discarded optimal solutions might be some with better flexibility characteristics. To overcome this limitation, the procedure has to be modified.

Accordingly, the following modifications were made:

- The cutset rule was disabled;
- When a new schedule is complete, the modifications described in 3.4.1 were made;
- For all pruning rules having the comparison "*if* $LB(p) \geq T$ *then discard current solution* ", change to "*if* $LB(p) > T$ *then discard current solution* ".

An immediate obvious implication of these modifications is that one of the most attractive features of the algorithm will be lost, which is the potential gain in processing speed that comes from the pruning rules that cut off "equally good" solutions. In this context, the main goal is not so much to reduce the computational speed, but more to generate all optimal duration schedules to be able to select the one with the maximal $\Sigma slack$.

Naturally, the procedure is much slower than the original one and will take too long to solve some of the harder instances. Therefore, the procedure will be executed with a processing time limit that might not always reach an optimal solution. Even though, such a truncated procedure is set in order to always generate some good solutions (for the testset the maximum deviation from the optimal duration is 25% and the overall average deviation from the optimum is 2%).

The modified procedure was executed with a time limit of 10s for each instance and the resulting values are presented in Figure 31. Using a greater time limit would allow additional instances to examine the complete solution space enabling the selection of the best solution regarding flexibility and other instances, that did not reach optimality could proceed to eventually reach it. Nevertheless, not all of them would do so even if the time limit is extended to 3.600s. The evaluated scenario models a more realistic case and does not compromise the conclusion to be taken regarding flexibility.

Again, all values are presented with their average values for sets having instances with the same $<NC, RF, RS>$ parameters.

The chart presents averages calculated from the following values:

- T1: The optimal project duration, represented as a bar (the lower bar). To serve as reference to the project duration obtained with the modified procedure;
- T2: The minimal project duration obtained by the modified procedure, represented as a bar (only the upper part is visible over T1);
- S1: Minimum value of $\Sigma slack$ for the schedule with minimal duration obtained by the modified procedure, represented as dots over a line (the lower line);
- S2: Maximal value of $\Sigma slack$ for the schedule with minimal duration obtained by the modified procedure, represented as squares over a line (the upper line). This corresponds to the best schedule, that is, the schedule of minimal duration with the highest $\Sigma slack$;

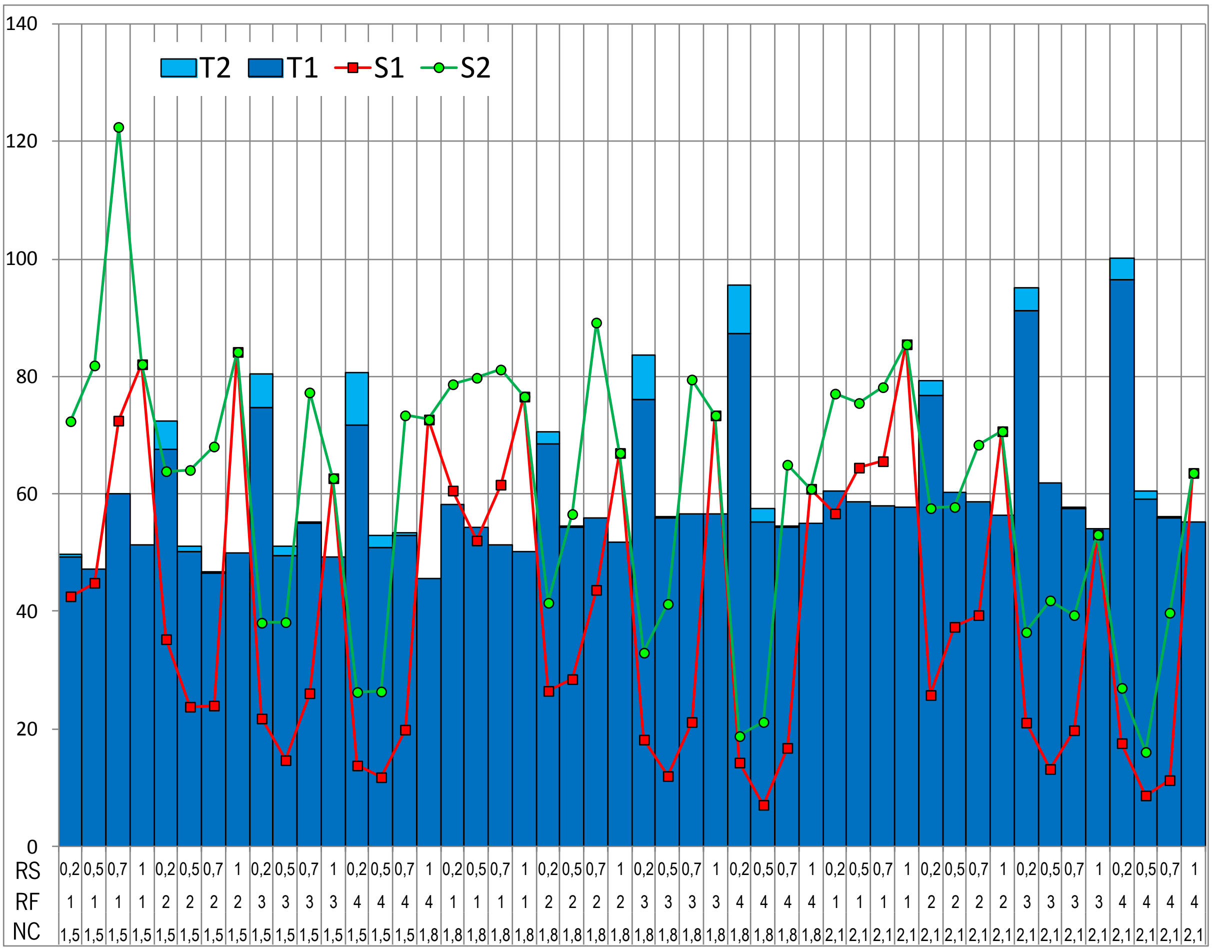


Figure 31: DH-B&B based enhanced schedule flexibility

From the chart, it can be seen that the obtained solution should be quite near the optimal duration, in spite of the levelling effect of averages that hides some duration peaks. As already mentioned, the maximal deviation on the optimal duration is 25% and 121 instances (25.2%) did not reach duration optimality.

On the other hand, 204 (42.5%) instances had their processing truncated by the 10s timer, which means that potentially not all time minimal schedules were processed, some of which could have greater $\Sigma slack$ than the already processed ones.

Obviously, there is no guarantee that $\Sigma slack$ would be the same if instances that were not processed to duration optimality were indeed allowed further processing until such optimality was reached.

Anyway, and contrary to what happened with the previous procedure (SGSS), schedules having the same duration can have substantially distinct $\Sigma slack$ which validates the use of this approach. In fact, the best value exceeds the minimum by 65% on average, despite 140 instances that do not have distinct values for $\Sigma slack$.

The methodology used in this case is different from the one followed with the SGSS based procedure. Here, the procedure selects the best solution regarding the maximal $\Sigma slack$ from all solutions that are optimal in duration. As for the SGSS based procedure, from the 15 generated solutions that are unique and duration minimal, for each priority rule, the one with the best $\Sigma slack$ is selected. This is a contributing factor to justify the distinct conclusions reached.

The main reason to use a different methodology is that SGSS, executed with a selected priority rule or priority list, generates a unique schedule which is the reason why this can be used to represent a schedule by decoding a solution from, for example, an activity list (Kolisch & Hartmann, 1999; Moumene & Ferland, 2009) or an activity set list (Moumene & Ferland, 2008). This prevents the use the of SGSS procedure in the same way as for the DH-B&B based one.

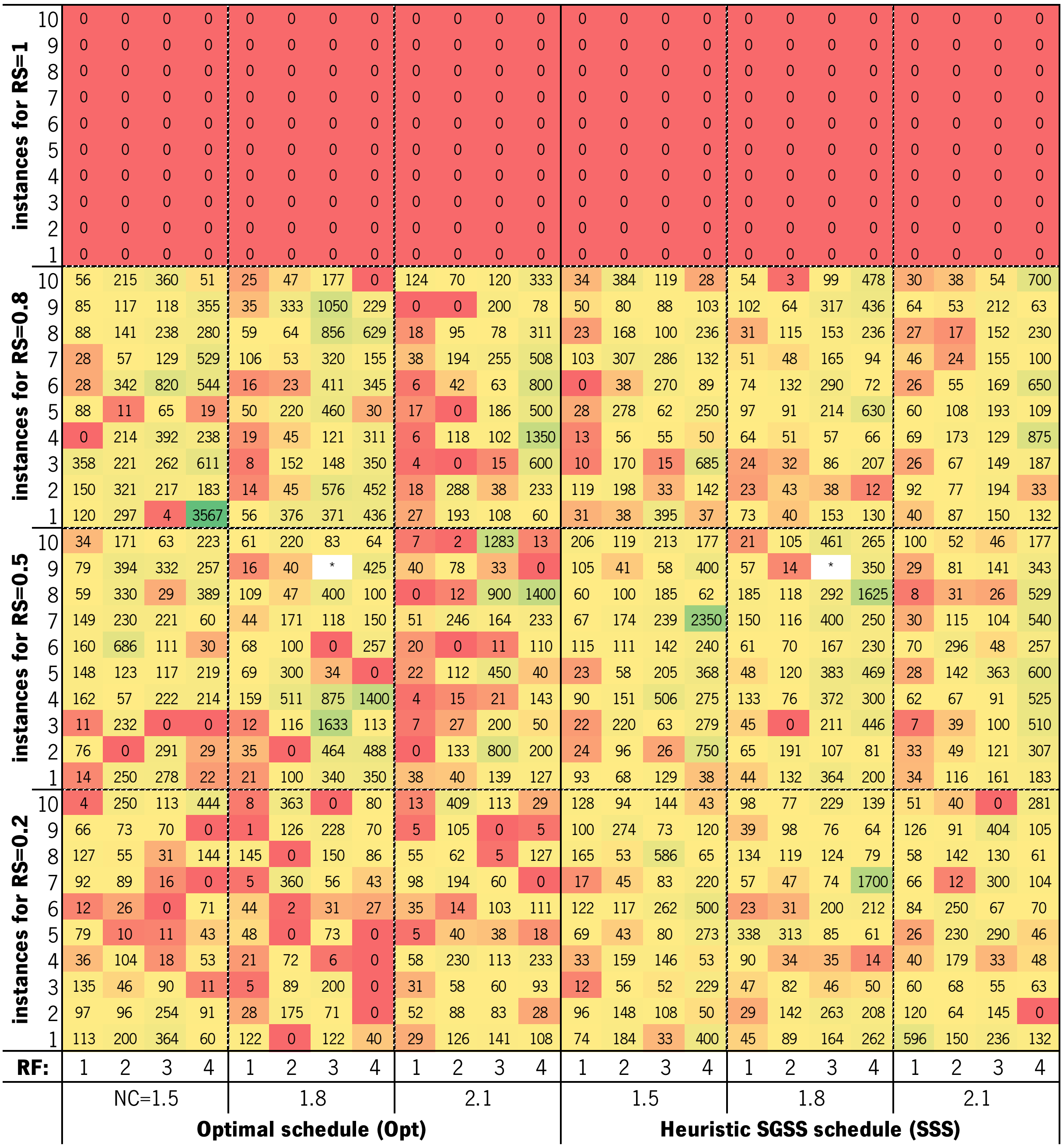
4.5.3 Results overview

An overall view of the results of applying both enhanced flexibility scheduling procedures to the testset is presented in Figure 32. The presented values are the relative deviations of total slack between the schedule with the highest slack and the schedule with the lowest slack, which is given by the expression:

$$\Delta slack\% = 100 * \frac{\Sigma slack_e(S_h) - \Sigma slack_e(S_l)}{\Sigma slack_e(S_l)},$$ where:

- e is an instance of the testset,
- $\Sigma slack_e(S_h)$ is the values of $\Sigma slack_e$ for the schedule with the highest $\Sigma slack_e$ and
- $\Sigma slack_e(S_l)$ is the values of $\Sigma slack_e$ for the schedule with the lowest $\Sigma slack_e$.

In the chart, each value's cell is coloured according to the magnitude of its value ranging from 🟥 which means that no increases was achieved to xxx meaning that a very high increase was achieved (more than 3000%). In between there several intermediate colours, representing the 🟨 the case where an intermediate increase was achieved.
The symbol $\boxed{*}$ means that the schedule for the project has no slack ($\Sigma slack_e(S_l) = 0$).



instances for RS=1	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
instances for RS=0.8	10	56	215	360	51	25	47	177	0	124	70	120	333	34	384	119	28	54	3	99	478	30	38	54	700
	9	85	117	118	355	35	333	1050	229	0	0	200	78	50	80	88	103	102	64	317	436	64	53	212	63
	8	88	141	238	280	59	64	856	629	18	95	78	311	23	168	100	236	31	115	153	236	27	17	152	230
	7	28	57	129	529	106	53	320	155	38	194	255	508	103	307	286	132	51	48	165	94	46	24	155	100
	6	28	342	820	544	16	23	411	345	6	42	63	800	0	38	270	89	74	132	290	72	26	55	169	650
	5	88	11	65	19	50	220	460	30	17	0	186	500	28	278	62	250	97	91	214	630	60	108	193	109
	4	0	214	392	238	19	45	121	311	6	118	102	1350	13	56	55	50	64	51	57	66	69	173	129	875
	3	358	221	262	611	8	152	148	350	4	0	15	600	10	170	15	685	24	32	86	207	26	67	149	187
	2	150	321	217	183	14	45	576	452	18	288	38	233	119	198	33	142	23	43	38	12	92	77	194	33
	1	120	297	4	3567	56	376	371	436	27	193	108	60	31	38	395	37	73	40	153	130	40	87	150	132
instances for RS=0.5	10	34	171	63	223	61	220	83	64	7	2	1283	13	206	119	213	177	21	105	461	265	100	52	46	177
	9	79	394	332	257	16	40	*	425	40	78	33	0	105	41	58	400	57	14	*	350	29	81	141	343
	8	59	330	29	389	109	47	400	100	0	12	900	1400	60	100	185	62	185	118	292	1625	8	31	26	529
	7	149	230	221	60	44	171	118	150	51	246	164	233	67	174	239	2350	150	116	400	250	30	115	104	540
	6	160	686	111	30	68	100	0	257	20	0	11	110	115	111	142	240	61	70	167	230	70	296	48	257
	5	148	123	117	219	69	300	34	0	22	112	450	40	23	58	205	368	48	120	383	469	28	142	363	600
	4	162	57	222	214	159	511	875	1400	4	15	21	143	90	151	506	275	133	76	372	300	62	67	91	525
	3	11	232	0	0	12	116	1633	113	7	27	200	50	22	220	63	279	45	0	211	446	7	39	100	510
	2	76	0	291	29	35	0	464	488	0	133	800	200	24	96	26	750	65	191	107	81	33	49	121	307
	1	14	250	278	22	21	100	340	350	38	40	139	127	93	68	129	38	44	132	364	200	34	116	161	183
instances for RS=0.2	10	4	250	113	444	8	363	0	80	13	409	113	29	128	94	144	43	98	77	229	139	51	40	0	281
	9	66	73	70	0	1	126	228	70	5	105	0	5	100	274	73	120	39	98	76	64	126	91	404	105
	8	127	55	31	144	145	0	150	86	55	62	5	127	165	53	586	65	134	119	124	79	58	142	130	61
	7	92	89	16	0	5	360	56	43	98	194	60	0	17	45	83	220	57	47	74	1700	66	12	300	104
	6	12	26	0	71	44	2	31	27	35	14	103	111	122	117	262	500	23	31	200	212	84	250	67	70
	5	79	10	11	43	48	0	73	0	5	40	38	18	69	43	80	273	338	313	85	61	26	230	290	46
	4	36	104	18	53	21	72	6	0	58	230	113	233	33	159	146	53	90	34	35	14	40	179	33	48
	3	135	46	90	11	5	89	200	0	31	58	60	93	12	56	52	229	47	82	46	50	60	68	55	63
	2	97	96	254	91	28	175	71	0	52	88	83	28	96	148	108	50	29	142	263	208	120	64	145	0
	1	113	200	364	60	122	0	122	40	29	126	141	108	74	184	33	400	45	89	164	262	596	150	236	132
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
		NC=1.5				1.8				2.1				1.5				1.8				2.1			
		Optimal schedule (Opt)												**Heuristic SGSS schedule (SSS)**											

Figure 32: Enhanced schedule flexibility results

4.6 Study summary

In this chapter, a computational study was performed to evaluate the potential of the proposed method. Using a benchmark testset and typical scheduling algorithms, the following studies were made:

- A preliminary study that shows that duration minimal solutions deviate 6% to 9% on average from optimum and that increases in project duration can exceed 50%, depending on the scheduling technique used.

- The flexibility of the generated schedules was evaluated regarding:
  - The number of activities with slack ($\#NC$), which are, on average, 27% to 29% of the total number of non-dummy activities in the projects, with a maximum of 50% activities with slack;
  - The schedule's total slack ($\Sigma slack$), which is, on average, 87% to 99% of the schedule's duration, ranging up to more than 200%;
  - The schedule's intrinsic flexibility (SIF), which is, on average, 27% to 33% of the total project's resource consumption, ranging up to about 300%.
- The impact of resource flexibility in limiting the use of the schedule flexibility, concluding that:
  - The optimal value of the α^- parameter should be at least 50% and α^+ should be set as high as possible, if activity durations of the testset 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%;
  - 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 impact of selecting schedules with the highest flexibility from the generated schedules. Two procedures were developed to generate a set of duration minimal schedules and selecting the one with the highest slack:
  - The SGSS scheduling algorithm that generated the set of solution using with several priority rules;
  - A modified version of the DH-B&B algorithm which generated the set of solutions by not discarding equally (in project duration) good ones. The procedure included a processing time limit, not ensuring that the set of solutions are minimal in duration.

5. CONCLUSIONS

In the last two chapters of this thesis, a new approach to deal with uncertainties that might arise when a project is being executed is presented and its applicability is evaluated when applied to a benchmark test set. It is the result of the work done to fulfil the objectives defined in the first chapter that were "to improve project management practices through the development of new models and methods for project scheduling and resource allocation that can be reflected in a tool to assist in solving some of the decision problems faced by projects managers". More specifically, these models and methods should be a result from a mix of two lines of research: (1) the resource allocation problem considering stochastic work contents and (2) the proactive/reactive scheduling techniques.

In other words the goal is to answer the research question:

> "*Can Project Management in Resource Constrained Projects be improved by using a combination of the assumption that activities have stochastic work contents with the use of proactive/reactive scheduling techniques?*"

Therefore, to fully fulfil the defined goals, the procedure comprising the developed model and methods must meet the following topics:

- It is of proactive/reactive nature;
- It can deal with activities with stochastic work content;
- It can be reflected in a (software) tool;
- It improves project management practices.

<u>The proactive /reactive nature of the procedure</u> is twofold:

a) When used with an existing baseline schedule, it can be viewed as a reactive scheduling technique that aims to keep the activity start times allowing their finish times to vary within their slack. When deviations in the work content of an activity occur, leading to finish times outside available slack, resource flexibility is used to cope with the increase in work content so that the finish time does not exceed the available slack. The additional rate used by resources can be compensated in by lower rate usage in activities with slack;

b) When a baseline schedule is selected among a set of optimal or sub-optimal schedules, it can be viewed as a single step proactive scheduling approach which then uses the same reactive approach as defined in a). This is also the case where a dedicated multi-objective scheduling method is used to generate the baseline schedule;

c) Alternatively, an initial schedule can be used to generate a more robust baseline schedule, i.e., a schedule with increased slack and/or SIF (use theorem 1). This case can be viewed as a two-step proactive scheduling approach which then uses the reactive approach defined in a).

Regarding the work contents, the procedure can deal with any non-deterministic or uncertain work content that lies between schedule dependent and resource flexibility boundaries. Therefore, it does cope with work contents that are stochastic in nature, but does not require it, as long as the variables comply with the stated limitations.

Reflecting it in a software tool is assured to be possible because it is already put into a computer program form. The developed form does not meet the requirements for professional use but does prove that this can be done.

Improving project management practices is the ultimate goal of this research. On the one hand, none of the analysed literature uses the proposed approach and therefore does not propose similar procedures. On the other hand, existing scheduling techniques generate schedules for the used testset that, with a few exceptions, can use this procedure to cope with uncertainties because they have activities with slack. Therefore, if resources are flexible in the proposed sense, 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.

While some limitations are identified and put into perspective, results show the potential of the presented approach that 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 and the objectives have been achieved.

5.1 Concluding remarks

The presented methodology relies on the ability of resources being flexible in the sense that they can work at varying rates within a time unit according to the project needs. This assumption can be regarded as a limitation of the model because of the inherent characteristic of resource flexibility but also due to the necessary ability of the project manager/project management process to promptly identify deviations and react to them by applying the adequate actions proposed by the model.

Besides these procedural limitations, on a more technical level limitations concern the intrinsic schedule flexibility (SIF) and the added constraints due to the resource flexibility and activity durations. The former, being well defined and limited, imposes a budget to the project that can be used by this methodology. This limitation can be overcome, or at least mitigated, either by increasing the project duration (thus increasing SIF) or by allowing the project's $balance$ to be less than zero (which can be compensated within other projects).

The second type of limitations constrains the methodology in taking full advantage of SIF when deviations arise, leading, in the worst case, to the exclusion in applying it to activities with small durations, which is a concept related to the resource flexibility parameters (α^- and α^+). An adequate selection of such parameters mitigated and even fully overcame this issue. Also, defining activities such that their durations take into account the limitations caused by the flexibility of resources, can lead to activity durations that are "long enough" regarding resource flexibility. Considering activities with integer durations that can be executed in one time unit can lead to errors in estimations that can go up to 50% which might not be acceptable.

Furthermore, some limitations of this approach can be eliminated by combining it with other scheduling techniques that increase schedule robustness including the "time buffer" based ones.

Either way, 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 as prescribed in this methodology, or in a similar way, resources can be, just to mention some:

- Idle, which means that no work will be done;
- Set to work in other projects or tasks, in which case setup and re-setup events may be necessary resulting in inefficiency and possible disturbance on the project;

- When resources are persons, they might try to be helpful and add non agreed upon features and, by doing so, changing the project's scope with unplanned and potentially unnecessary functionalities that will increase complexity and have to be maintained and that might induce unforeseen side effects, just to mention some dangers 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".

With the proposed methodology, "consuming" slack is not mandatory once it is used according to what is needed, that includes compensating for activities being executed in faster rates within other projects. Thus, the above mentioned resource status for "slacked" periods may also be necessary.

On the other hand, activities without slack, which are schedule critical, face additional challenges when deviations to plan occur.

If deviations tend to decrease their duration or their work content, they might have slack becoming non-critical thus falling in the previous description. Some might be tempted to anticipate, if possible, the start of dependent activities (either by precedence or resource constraints) in an approach known as "Roadrunner mentality" as is the case of "Prochain" (Herroelen & Leus, 2001). This will obviously collide with the premise of protecting as much as possible the start time of activities which follows the opposing scheduling approach known as "railways scheduling". The stability of the plan is preferred over reducing the project's duration either at the planning phase (before project starts) or at the execution phase (while it is being done). Note also that there is no guarantee that the "Roadrunner mentality" always leads to a shorter project duration (Tian & Demeulemeester, 2010).

But, more often than not, deviations tend to increase activity work content and duration. With the proposed methodology, the flexibility existing in the schedule can be used to compensate such deviations. This means that, as far as that schedule flexibility exists, there is a mechanism to prevent such activities to propagate its deviations to other activities that depend directly (via precedence constraints) or indirectly (via resource constraints) on it. Such deviations propagate in the form of activities starting later than planned, jeopardizing the schedule thereafter. Some alternatives to the proposed approach are identified and studied like "activity crashing" and "activity overlapping" or "fast tracking" (Gerk & 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 thesis, 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 (also executors) because:

- Planning by excess (more than really needed) will not result in work capacity being thrown away, rather it can be used within the project (or other projects) if required;
- Planning by default (less than really needed) will not result in (eventually with additional costs) overtime but rather can mean working at a faster rate, compensating other lower rate executed activities.

In the latter case, when human resources are considered, even if overtime does not represent additional costs, it will certainly represent dissatisfaction and bad management perception if this happens on a regular basis.

In short, the proposed methodology tends to promote responsibility over estimations and provides mechanisms to deal with the consequences of "bad" planning within the context of the project (or groups of projects) by having flexibility in the resources used.

Finally, one cannot fail to mention that resource flexibility is frequently used in an empiric way. When projects tend to be delayed, it is not uncommon that project managers try to lead resources to work at faster rates to overcome perceived delays. The proposed methodology can also be used to manage such *ad-hoc* practice[15] in a planned and integrated way such that, from the project planning phase, it is perceived:

- Till what extent is this possible;
- In which timing;
- With which resources; and
- With what impact on the project.

All these are relevant questions depend on the selected schedule and on the impact of uncertainties that must be handled according to the defined and accepted resource flexibility. This is a complex problem to solve which cannot be handled efficiently in an empiric way.

---

[15] It remains to be seen if project managers are willing to give up this hidden backup tool to solve project delays and use it in an explicitly managed way

5.2 Future/Open Work

Initially, the goal of this work was to produce a software application that could help project managers to better deal with uncertainties that might arise or to implement such functionalities in existing software tools. It was then well understood that this goal would go beyond the scope of this work and that the most important task was to establish a methodology to accomplish that final goal and to develop a prototype in order to validate, as much as possible, the methodology.

The application was not developed nor was the functionality integrated in any existing tool but a software module was developed that enabled a computational study to validate the principles underlying the methodology. This cannot be called a prototype in the sense that it implies that the full procedure to apply the methodology is supported in order to enable anyone to test it. It rather should be seen as a "proof of concept" that is an earlier step from a prototype that still fulfils the purpose of validating the methodology.

Therefore, the task of building a prototype from here is seen as an immediate future work to be done, followed by the final application or its integration in existing software project management tools. It should be noted that, due to the possibility to work over any feasible deterministic schedule, this methodology should not be too hard to be integrated into project management software tools that support scheduling and that provides an open programming interface.

Two additional subjects are very important in complementing the validating analysis on the proposed methodology.

To start, the methodology can be further validated by simulating its operation in order to demonstrate that the working schedule (S^w) resists better to such deviations than the baseline schedule (S^b). Simulation can take into account that the working schedule (S^w) can be extended to be a set of schedules ($S^w = \{S_1^w, S_2^w, \ldots, S_g^w\}$) instead, depending on the quantity (how many and to which extent) and the timing (when they happen and when they are perceived) of deviations. This validation can also be extended to include other scheduling techniques that address this problem which it, to 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, specifically resource requirements. There is no consideration for additional costs regarding schedule deviations, i.e., earliness or tardiness (Van de Vonder, Ballestín, et al., 2007; Vanhoucke, Demeulemeester, &

Herroelen, 2001) or distinct costs for each resource type. Considerations about flexibility costs are also assumed to be merely proportional to durations but other models are possible like assuming an increase in the resource usage cost when deviations from nominal work rate happen or, more generally, assuming a generic function for the relation of the cost with effectively used flexibility, that can be expressed by the execution rate: $cost_i = f(exec.rate_i), exec.rate_i = d_i^{nom}/d_i$ (costs are considered by activity i as an example), or even a cost for the availability of a resource (k) to be flexible: $cost_k = f(\alpha_k^-, \alpha_k^+)$.

On the other hand, during the exposition, several topics were identified and discussed that require further research and analysis.

As has already been referred to, another topic that can benefit from further research respects the development of slack oriented scheduling procedures. The ones used in this thesis always consider the minimal project duration as the main objective to be achieved. As already mentioned, it can be of interest to search for solution whose aim is to maximize the schedule's flexibility, i.e. its slack as defined here, in combination with duration minimization or deadline. The procedures to be developed can be based on existing ones, like branch-and-bound or constructive heuristics, which in the latter example should include further testing and the search for more adequate (better in the flexibility sense) priority rules.

Research on this subject can also focus on developing mathematical programming models of which some conceptual hints were already mentioned regarding MIP models that could be developed. Some interesting solution may arise here.

On the other hand, the possibility to use this procedure to decrease ("crash") project duration that was mention in sub-section 3.2.3, can be an interesting research topic. The problem of minimizing the project duration by using this technique can be modelled as an MRCPSP problem by using a pre-processing phase where all possible activity durations are enumerated, taking into account the constraints imposed by the resource flexibility parameters and also the additional constraints that might exist due to the selection of a project execution mode (MODE). The resulting possible activity durations can be regarded as activity execution modes that can then be used to feed the MRCPSP model.

Also, the problem of activity insertion and/or deletion and the possibility that activities are executed with fewer resources without impact on the schedule were also mentioned before as a problem that can have an easier solution when resource flexibility is considered. Further analysis of with which conditions (which flexibility parameters) and to what extent (how many activities can be inserted, with which resource requirements, which durations ...) and with which impacts (on the schedule, on the final balance...) can be of interest.

Another subject that might be worth of further study concerns the variable $\#NC$ that can be of more interest than it probably seems. For instance, two schedules with the same SIF have quite distinct $\#NC$: which one is better once they have the same flexibility? The one with lower $\#NC$ has its slack concentrated in fewer activities than the other. Is it better to have flexibility concentrated in fewer activities or is it better to spread it among several more? What about their time distribution: is it better that activities with slack are distributed along the project's time window or not? Whatever the answers to these questions, another one arises: how to create such optimized schedules?

Finally, some subjects were considered to be further studied in the context of this thesis but were disregarded because they deviated from the main focus on flexibility, which was defined in the meantime.

One is the integration of the flexibility concepts with time buffers and also with risk management. It can be interesting to evaluate the impact of flexibility on risk management and to understand to which extent the complementary effect of the introduction of time buffers regarding flexibility can have a positive effect on dealing with project uncertainty. In this later case it is clear that the starting point can be a time buffer management technique that is complemented with resource flexibility.

Another effect of resource flexibility that can be explored is to use it to shorten project duration. As already mentioned, this is not the option within the context of this thesis although some clues on how that can be done were already given.

To end this already long list, an alternative approach on using resource flexibility might start with multi-mode RCPSP (MRCPSP) rather than RCPSP. It should be harder to deal with as MRCPSP is a generalization of the RCPSP problem (Hartmann & Drexl, 1998; Talbot, 1982; Van Peteghem & Vanhoucke, 2014), but it can lead to better results. Just to give a hint, the procedure could be something like this:

a) define modes;

b) set the baseline schedule with slower modes (d_i^{max});

c) when deviations arise, reschedule within modes combined with resource flexibility.

In c) several possibilities could be explored like considering only slower modes for critical activities (the ones without slack) that resulted from a classical MRCPSP scheduling (minimal project duration).

All these subjects and questions were raised during the research underlying this thesis. Some of them were considered to be further researched and some among them were in fact dealt with but, for some reason, were not included in this final document.

Hopefully, some of these can be elected for further research.

BIBLIOGRAPHY

Al-Fawzan, M. A., & Haouari, M. (2005). A bi-objective model for robust resource-constrained project scheduling. *International Journal of Production Economics*, *96*(2), 175–187. http://doi.org/10.1016/j.ijpe.2004.04.002

Alvarez-Valdés, R., & Tamarit, J. M. (1993). The project scheduling polyhedron: Dimension, facets and lifting theorems. *European Journal of Operational Research*, *67*(2), 204–220. http://doi.org/10.1016/0377-2217(93)90062-R

Anbari, F. T. (2003). Earned value project management method and extensions. *Project Management Journal*, *34*(4), 12–23. http://doi.org/10.1109/EMR.2004.25113

Anbari, F. T., Bredillet, C. N., Turner, J. R., Anbari, F. T., & Turner, J. R. (2008). Perspectives on Research in Project Management. *Academy of Management Proceedings*, *2008*(1), 1–6. http://doi.org/10.5465/AMBPP.2008.33660270

Artigues, C., Leus, R., & Talla Nobibon, F. (2013). Robust optimization for resource-constrained project scheduling with uncertain activity durations. *Flexible Services and Manufacturing Journal*, *25*(1-2), 175–205. http://doi.org/10.1007/s10696-012-9147-2

Ashtiani, B., Leus, R., & Aryanezhad, M.-B. (2008). A Novel Class of Scheduling Policies for the Stochastic Resource-Constrained Project Scheduling Problem. *SSRN Electronic Journal*. http://doi.org/10.2139/ssrn.1368714

Ashtiani, B., Leus, R., & Aryanezhad, M.-B. (2011). New competitive results for the stochastic resource-constrained project scheduling problem: Exploring the benefits of pre-processing. *Journal of Scheduling*, *14*(2), 157–171. http://doi.org/10.1007/s10951-009-0143-7

Ballard, G., & Howell, G. (2003). Lean project management. *Building Research & Information*, *31*(2), 119–133. http://doi.org/10.1080/09613210301997

Ballestín, F. (2007). When it is worthwhile to work with the stochastic RCPSP? *Journal of Scheduling*, *10*(3), 153–166. http://doi.org/10.1007/s10951-007-0012-1

Ballestín, F., & Leus, R. (2009). Resource-Constrained Project Scheduling for Timely Project Completion with Stochastic Activity Durations. *Production and Operations Management*, *18*(4), 459–474. http://doi.org/10.1111/j.1937-5956.2009.01023.x

Banerjee, S., & Hopp, W. J. (2001). The Project Portfolio Management Problem. *Department of Industrial Engineering and Management Sciences, Northwestern University*.

Baumann, P., & 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., & Caramia, M. (2013). A new formulation for the project scheduling problem under limited resources. *Flexible Services and Manufacturing Journal*, 6–24. http://doi.org/10.1007/s10696-011-9127-y

Blazewicz, J., Lenstra, J. K., & Kan, A. H. G. R. (1983). Scheduling subject to resource constraints: classification and complexity. *Discrete Applied Mathematics*, *5*(1), 11–24. http://doi.org/10.1016/0166-218X(83)90012-4

Borrego, M., Douglas, E. P., & Amelink, C. T. (2009). Quantitative, Qualitative, and Mixed Research Methods in Engineering Education. *Journal of Engineering Education*, *98*(1), 53–66. http://doi.org/10.1002/j.2168-9830.2009.tb01005.x

Chapman, C., & Ward, S. (2003). *Project risk management: processes, techniques and insights* (2nd ed.). John Wiley & Sons Ltd.

Christofides, N., Alvarez-Valdés, R., & Tamarit, J. M. (1987). Project scheduling with resource constraints: A branch and bound approach. *European Journal of Operational Research*, *29*(3), 262–273. http://doi.org/10.1016/0377-2217(87)90240-2

Cobb, C. G. (2015). *The Project Manager's Guide to Mastering Agile: Principles and Practices for an Adaptive Approach*. Wiley. Retrieved from https://books.google.com/books?id=ouIWBgAAQBAJ&pgis=1

Couto, J. (2012). Identifying factors for the poor public institutions performance in portuguese construction industry. *International Journal of Academic Research, Vol. 3, Issue 3, May, Part 1, Pp. 252-257.* http://doi.org/http://hdl.handle.net/1822/16346

Couto, J., & Teixeira, J. (2007). The Evaluation of the Delays in the Portuguese Construction. *CIB World Building Congress*. Cape Town, South Africa. Retrieved from http://hdl.handle.net/1822/8327

De, P., Dunne, J. E., Ghosh, J. B., & Wells, C. E. (1995). The discrete time-cost tradeoff problem revisited. *European Journal of Operational Research*, *81*(2), 225–238. http://doi.org/10.1016/0377-2217(94)00187-H

De Reyck, B., Demeulemeester, E., & Herroelen, W. (2001). Algorithms for Scheduling Projects with Generalized Precedence Relations. In J. Weglarz (Ed.), *Project Scheduling: Recent Models, Algorithms and Applications* (pp. 77–105). Springer US.

De Reyck, B., & Herroelen, willy. (1998). A branch-and-bound procedure for the resource-constrained project scheduling problem with generalized precedence relations. *European Journal of Operational Research*, *111*(1), 152–174. http://doi.org/10.1016/S0377-2217(97)00305-6

De Reyck, B., & Herroelen, W. (1996). Computational experience with a branch-and-bound procedure for the resource-constrained project scheduling problem with generalized precedence relations. Retrieved from https://lirias.kuleuven.be/handle/123456789/219465

Deblaere, F., Demeulemeester, E., & Herroelen, W. (2011). RESCON: Educational project scheduling software. *Computer Applications in Engineering Education*, *19*(2), 327–336. http://doi.org/10.1002/cae.20314

Deblaere, F., Demeulemeester, E., Herroelen, W., & Van de Vonder, S. (2007). Robust Resource Allocation Decisions in Resource-Constrained Projects. *Decision Sciences*, *38*(1), 5–37. http://doi.org/10.1111/j.1540-5915.2007.00147.x

DeCarlo, D. (2004). *eXtreme Project Management: Using Leadership, Principles, and Tools to Deliver Value in the Face of Volatility*. Wiley. Retrieved from http://books.google.pt/books?id=Z5SSIsA4M-0C

Demeulemeester, E., & Herroelen, W. (1992). A Branch-and-Bound Procedure for the Multiple Resource-Constrained Project Scheduling Problem. *Management Science*, *38*(12), 1803–1818. http://doi.org/doi:10.1287/mnsc.38.12.1803

Demeulemeester, E., & Herroelen, W. (1996). An efficient optimal solution procedure for the preemptive resource-constrained project scheduling problem. *European Journal of Operational Research*, *90*(2), 334–348. http://doi.org/10.1016/0377-2217(95)00358-4

Demeulemeester, E., & Herroelen, W. (1997). New Benchmark Results for the Resource-Constrained Project Scheduling Problem. *Management Science*, *43*(11), 1485–1492.

Demeulemeester, E., & Herroelen, W. (2002). *Project Scheduling: A Research Handbook*. Springer. http://doi.org/10.1007/b101924

Demeulemeester, E., & Herroelen, W. (2009). *Robust Project Scheduling. Foundations and Trends® in Technology, Information and Operations Management* (Vol. 3). now Publishers Inc. http://doi.org/10.1561/0200000021

Demeulemeester, E., Herroelen, W., & Elmaghraby, S. (1996). Optimal procedures for the discrete time/cost trade-off problem in project networks. *European Journal of Operational Research*, *88*(1), 50–68. http://doi.org/10.1016/0377-2217(94)00181-2

Dorndorf, U., Pesch, E., & Phan-Huy, T. (2000). A Time-Oriented Branch-and-Bound Algorithm for Resource-Constarined Scheduling with Generalised Precedence Constraints. *Management Science*, *46*(10 (Oct., 2000)), 1365–1384. http://doi.org/10.1063/1.1746492

Ehrgott, M. (2006). A discussion of scalarization techniques for multiple objective integer programming. *Annals of Operations Research*, *147*(1), 343–360. http://doi.org/10.1007/s10479-006-0074-z

Elmaghraby, S. (2005). On the fallacy of averages in project risk management. *European Journal of Operational Research*, *165*(2), 307–313. http://doi.org/10.1016/j.ejor.2004.04.003

Fleming, Q. W., & Koppelman, J. M. (2010). *Earned value project management* (3rd ed). Newtown Square, Pa.: Project Management Institute.

Flyvbjerg, B., Bruzelius, N., & Rothengatter, W. (2003). *Megaprojects and Risk: An Anatomy of Ambition*. Cambridge University Press. Retrieved from http://books.google.pt/books?id=RAV5P-50UjEC

French, S. (1993). *Decision Theory: An Introduction to the Mathematics of Rationality*. Ellis Horwood. Retrieved from https://books.google.com/books?id=HcxBPwAACAAJ&pgis=1

Fündeling, C.-U., & Trautmann, N. (2010). A priority-rule method for project scheduling with work-content constraints. *European Journal of Operational Research*, *203*(3), 568–574. http://doi.org/10.1016/j.ejor.2009.09.019

Garel, G. (2013). A history of project management models: From pre-models to the standard models. *International Journal of Project Management*, *31*(0), 663–669. http://doi.org/http://dx.doi.org/10.1016/j.ijproman.2012.12.011

Gerk, J. E. V., & Qassim, R. Y. (2008). Project Acceleration via Activity Crashing, Overlapping, and Substitution. *IEEE Transactions on Engineering Management*, *55*(4), 590–601. http://doi.org/10.1109/TEM.2008.927786

Goldratt, E. M. (1997). *Critical chain*. Great Barrington, MA: North River Press.

Goldratt, E. M., & Cox, J. (1984). *The Goal: Excellence in Manufacturing*. Croton-on-Hudson, NY: North River Press. Retrieved from https://books.google.com/books?id=rp1ZwCiPMzoC&pgis=1

Golenko-Ginzburg, D. (1989). PERT assumptions revisited. *Omega*, *17*(4), 393–396. http://doi.org/10.1016/0305-0483(89)90053-4

Gul, S. (2011). Critical realism and project management: Revisiting the noumenal and phenomenal. *African Journal of Business Management*, *5*(31), 12212–12221. http://doi.org/10.5897/ajbm11.741

Hällgren, M. (2012). The construction of research questions in project management. *International Journal of Project Management*, *30*(7), 804–816. http://doi.org/http://dx.doi.org/10.1016/j.ijproman.2012.01.005

Hartmann, S., & Briskorn, D. (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, *207*(1), 1–14. http://doi.org/http://dx.doi.org/10.1016/j.ejor.2009.11.005

Hartmann, S., & Drexl, A. (1998). Project scheduling with multiple modes: A comparison of exact algorithms. *Networks*, *32*(4), 283–297. http://doi.org/10.1002/(SICI)1097-0037(199812)32:4<283::AID-NET5>3.0.CO;2-I

Hartmann, S., & Kolisch, R. (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. http://doi.org/10.1016/S0377-2217(99)00485-3

Hazır, Ö., Haouari, M., & Erel, E. (2010). Robust scheduling and robustness measures for the discrete time/cost trade-off problem. *European Journal of Operational Research*, *207*(2), 633–643. http://doi.org/10.1016/j.ejor.2010.05.046

Herrerías-Velasco, J. M., Herrerías-Pleguezuelo, R., & Van Dorp, J. R. (2011). Revisiting the PERT mean and variance. *European Journal of Operational Research*, *210*(2), 448–451. http://doi.org/10.1016/j.ejor.2010.08.014

Herroelen, W., & Leus, R. (2001). On the merits and pitfalls of critical chain scheduling. *Journal of Operations Management*, *19*(5), 559–577. http://doi.org/10.1016/S0272-6963(01)00054-7

Herroelen, W., & Leus, R. (2004). Robust and reactive project scheduling: a review and classification of procedures. *International Journal of Production Research*, *42*(8), 1599–1620. http://doi.org/10.1080/00207540310001638055

Igelmund, G., & Radermacher, F. J. (1983). Preselective strategies for the optimization of stochastic project networks under resource constraints. *Networks*, *13*(1), 1–28. http://doi.org/10.1002/net.3230130102

Jia, J., Fan, X., & Lu, Y. (2007). System Dynamics Modeling for Overtime Management Strategy of Software Project. *2007, International System Dynamics Conference*. Boston, Massachusetts, USA. Retrieved from http://www.systemdynamics.org/conferences/2007/proceed/papers/JIA341.pdf

Kaplan, L. A. (1988). *Resource-constrained project scheduling with preemption of jobs*. University of MICHIGAN. Retrieved from https://books.google.pt/books/about/RESOURCE_CONSTRAINED_PROJECT_SCHEDULING.html?id=PPDZNAAACAAJ&pgis=1

Kastor, A., & Sirakoulis, K. (2009). The effectiveness of resource levelling tools for Resource Constraint Project Scheduling Problem. *International Journal of Project Management*, *27*(5), 493–500. http://doi.org/10.1016/j.ijproman.2008.08.006

Kerzner, H. (2013). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. John Wiley & Sons. Retrieved from https://books.google.com/books?id=CSHhsh2TpLwC&pgis=1

Klein, R. (2000). *Scheduling of Resource-Constrained Projects*. Kluwer Academic. Retrieved from http://books.google.pt/books?id=_j_YUUAcvs0C

Kobylański, P., & 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*, *107*(2), 496–501. http://doi.org/10.1016/j.ijpe.2006.07.012

Kolisch, R. (1996a). Efficient priority rules for the resource-constrained project scheduling problem. *Journal of Operations Management*, *14*(3), 179–192. http://doi.org/10.1016/0272-6963(95)00032-1

Kolisch, R. (1996b). Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. *European Journal of Operational Research*, *90*(2), 320–333. http://doi.org/http://dx.doi.org/10.1016/0377-2217(95)00357-6

Kolisch, R. (1999). Resource allocation capabilities of commercial project management software packages. *Interfaces*, *29*(4), 19–31. http://doi.org/10.1287/inte.29.4.19

Kolisch, R., & Hartmann, S. (1999). Heuristic Algorithms for the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis. In J. Węglarz (Ed.), *Project Scheduling SE - 7* (Vol. 14, pp. 147–178). Springer US. http://doi.org/10.1007/978-1-4615-5533-9_7

Kolisch, R., & Hartmann, S. (2006). Experimental investigation of heuristics for resource-constrained project scheduling: An update. *European Journal of Operational Research*, *174*(1), 23–37. http://doi.org/10.1016/j.ejor.2005.01.065

Kolisch, R., & Meyer, K. (2006). Selection and Scheduling of Pharmaceutical Research Projects. In J. Józefowska & J. Węglarz (Eds.), *Perspectives in Modern Project Scheduling* (p. Chapter 13).

Springer Science+Business Media, LLC. Retrieved from http://link.springer.com/chapter/10.1007/978-0-387-33768-5_13#page-1

Kolisch, R., & Padman, R. (2001). An integrated survey of deterministic project scheduling. *Omega*, *29*(3), 249–272. http://doi.org/10.1016/S0305-0483(00)00046-3

Kolisch, R., & Sprecher, A. (1997). PSPLIB - A project scheduling problem library. *European Journal of Operational Research*, *96*(1), 205–216. http://doi.org/10.1016/S0377-2217(96)00170-1

Kolisch, R., Sprecher, A., & Drexl, A. (1995). Characterization and Generation of a General Class of Resource-Constrained Project Scheduling Problems. *Management Science*, *41*(10), 1693–1703. Retrieved from http://www.jstor.org/stable/2632747?seq=1&cid=pdf-reference#references_tab_contents

Lamas, P., & Demeulemeester, E. (2014). A purely proactive scheduling procedure for the resource-constrained project scheduling problem with stochastic activity durations. *Journal of Scheduling*, 1–20. http://doi.org/10.2139/ssrn.2464056

Lambrechts, O. (2007, November 27). *Robust project scheduling subject to resource breakdowns*. Katholieke Universiteit Leuven, Belgium. Retrieved from https://lirias.kuleuven.be/handle/1979/1009

Lambrechts, O., Demeulemeester, E., & Herroelen, W. (2007). Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities. *Journal of Scheduling*, *11*(2), 121–136. http://doi.org/10.1007/s10951-007-0021-0

Leus, R., & Herroelen, W. (2004). Stability and resource allocation in project planning. *IIE Transactions*, *36*(7), 667–682. http://doi.org/10.1080/07408170490447348

MacCrimmon, K. R., & Ryavec, C. A. (1964). An Analytical Study of the PERT Assumptions. *Operations Research*, *12*(1), 16–37. http://doi.org/10.1287/opre.12.1.16

Malcolm, D., Roseboom, J., Clark, C., & Fazar, W. (1959). Application of a Technique for Research and Development Program-Evaluation. *Operations Research*, *7*(5), 646–669. http://doi.org/10.1287/opre.7.5.646

Martinsuo, M. (2013). Project portfolio management in practice and in context. *International Journal of Project Management*, *31*(6), 794–803. http://doi.org/10.1016/j.ijproman.2012.10.013

Meredith, J. R., & Mantel, S. J. (2011). *Project Management: A Managerial Approach*. John Wiley & Sons. Retrieved from https://books.google.ca/books/about/Project_Management.html?id=xGRtQetWjNsC&pgis=1

Milosevic, D., & Patanakul, P. (2005). Standardized project management may increase development projects success. *International Journal of Project Management*, *23*(3), 181–192. http://doi.org/http://dx.doi.org/10.1016/j.ijproman.2004.11.002

Mingozzi, A., Maniezzo, V., Ricciardelli, S., & Bianco, L. (1998). An Exact Algorithm for the Resource-Constrained Project Scheduling Problem Based on a New Mathematical Formulation. Retrieved December 29, 2015, from http://pubsonline.informs.org/action/showCitFormats?doi=10.1287%2Fmnsc.44.5.714

Möhring, R., Radermacher, F. J., & Weiss, G. (1984). Stochastic scheduling problems I — General strategies. *Zeitschrift Für Operations Research*, *28*(7), 193–260. http://doi.org/10.1007/BF01919323

Möhring, R., Radermacher, F. J., & Weiss, G. (1985). Stochastic scheduling problems II-set strategies-. *Zeitschrift Für Operations Research*, *29*(3), 65–104. http://doi.org/10.1007/BF01918198

Moumene, K., & Ferland, J. A. (2008). New representation to reduce the search space for the resource-constrained project scheduling problem. *RAIRO - Operations Research*, *42*(2), 215–228. Retrieved from https://eudml.org/doc/250390

Moumene, K., & Ferland, J. a. (2009). Activity list representation for a generalization of the resource-constrained project scheduling problem. *European Journal of Operational Research*, *199*(1), 46–54. http://doi.org/10.1016/j.ejor.2008.10.030

Naber, A., & Kolisch, R. (2013). *Comments on the RCPSP - Model of Bianco and Caramia (2013)*. Retrieved from http://ssrn.com/abstract=2286718

Naber, A., & Kolisch, R. (2014). MIP models for resource-constrained project scheduling with flexible resource profiles. *European Journal of Operational Research*, *239*(2), 335–348. http://doi.org/http://dx.doi.org/10.1016/j.ejor.2014.05.036

Naber, A., Kolisch, R., Bianco, L., & Caramia, M. (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. http://doi.org/10.1007/s10696-014-9197-8

Newbold, R. (2008). *The Billion Dollar Solution: Secrets of Prochain Project Management*. *Hospitals & Health Networks* (Vol. May). Prochain Press.

Olsen, B., & Swenson, D. (2011). Overtime Effects on Project Team Effectiveness. *Midwest Instruction and Computing Symposium*. April, Duluth, MN - USA. Retrieved from http://www2.css.edu/mics/Submissions/submissions/Overtime Effects on Project Team Effectiveness.pdf

Patterson, J. H., Słowiński, R., Talbot, F. B., & 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

Peteghem, V. Van, & Vanhoucke, M. (2010). A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem. *European Journal of Operational Research*, *201*(2), 409–418. http://doi.org/http://dx.doi.org/10.1016/j.ejor.2009.03.034

PMI. (2013). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)-Fifth Edition*. *Project Management Journal* (Vol. 44). http://doi.org/10.1002/pmj.21345

Pritsker, A. A. B., Waiters, L. J., & Wolfe, P. M. (1969). Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach. *Management Science*, *16*(1), 93–108.

http://doi.org/10.1287/mnsc.16.1.93

Ranjbar, M., & Kianfar, F. (2010). Resource-Constrained Project Scheduling Problem with Flexible Work Profiles: A Genetic Algorithm Approach. *Transaction E: Industrial Engineering*, *17*(1), 25–35. Retrieved from http://www.sid.ir/en/VEWSSID/J_pdf/95520101E05.pdf

Rokou, E., Dermitzakis, M., & 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* (pp. 642–646). IEEE. http://doi.org/10.1109/IEEM.2014.7058717

Schatteman, D., Herroelen, W., Van de Vonder, S., & Boone, A. (2008). Methodology for Integrated Risk Management and Proactive Scheduling of Construction Projects. *Journal of Construction Engineering and Management-Asce*, *134*(11), 885–893. http://doi.org/10.1061/(asce)0733-9364(2008)134:11(885)

Schwindt, C. (2006). *Resource Allocation in Project Management*. Springer Science & Business Media.

Slowinski, R. (1980). Two approaches to problems of resource allocation among project activities—A comparative study. *Journal of the Operational Research Society*. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-0019047699&partnerID=tZOtx3y1

Sprecher, A., Kolisch, R., & Drexl, A. (1995). Semi-active, active, and non-delay schedules for the resource-constrained project scheduling problem. *European Journal of Operational Research*, *80*(1), 94–102. http://doi.org/10.1016/0377-2217(93)E0294-8

Stork, F. (2001). *Stochastic Resource-Constrained Scheduling*. *PhD Thesis*. Technische Universität Berlin. Retrieved from papers2://publication/uuid/0E0D6199-4A7F-4527-832B-D160769539A6

T'kindt, V., & Billaut, J.-C. (2002). Multicriteria scheduling problems: a survey. *RAIRO - Operations Research*, *35*(2), 143–163. http://doi.org/10.1051/ro:2001109

Talbot, F. B. (1982). Resource-Constrained Project Scheduling with Time-Resource Tradeoffs: The Nonpreemptive Case. *Management Science*, *28*(10), 1197–1210. http://doi.org/10.1287/mnsc.28.10.1197

Tereso, A., Araújo, M., & Elmaghraby, S. (2004). Adaptive resource allocation in multimodal activity networks. *International Journal of Production Economics*, *92*(1), 1–10. http://doi.org/10.1016/j.ijpe.2003.09.005

Tereso, A., Mota, J. R., & Lameiro, R. (2006). Adaptive resource allocation to stochastic multimodal projects: a distributed platform implementation in Java. *Control and Cybernetics*, *35*(3), 661–686. Retrieved from <Go to ISI>://WOS:000245200600009

Tereso, A., Novais, R., Araújo, M., & Elmaghraby, S. (2009). Optimal resource allocation in stochastic activity networks via the electromagnetism approach: a platform implementation in Java. *Control and Cybernetics*, *38*(3). Retrieved from http://pessoais.dps.uminho.pt/anabelat/objectos/CC2009tereso-et-al.pdf

The Standish Group. (2009). Chaos Summary 2009, 1–4. Retrieved from http://www1.standishgroup.com/newsroom/chaos_2009.php

The Standish Group. (2014). Chaos Summary 2014.

Tian, W., & Demeulemeester, E. (2010). Railway scheduling reduces the expected project makespan. *Katholieke Universiteit Leuven KBI*, *1004*.

Tian, W., & Demeulemeester, E. (2012). On the Interaction Between Railway Scheduling and Resource Flows. *Flexible Services and Manufacturing Journal*, *25*(1-2), 145–174. http://doi.org/10.1007/s10696-012-9145-4

Trautmann, N., & Baumann, P. (2009). Resource-allocation capabilities of commercial project management software: An experimental analysis. *2009 International Conference on Computers & Industrial Engineering*, 1143–1148. http://doi.org/10.1109/ICCIE.2009.5223881

Trautmann, N., & Baumann, P. (2010). An iterative backward/Forward Technique for the scheduling of resource-constrained projects within Microsoft Project. In *Industrial Engineering and Engineering Management (IEEM), 2010 IEEE International Conference on* (pp. 1558–1562). http://doi.org/10.1109/IEEM.2010.5674171

Tritschler, M., Naber, A., & 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*, (2003), 230–233.

Van de Vonder, S. (2006). *Proactive-reactive procedures for robust project scheduling*. *Applied Economics*.

Van de Vonder, S., Ballestín, F., Demeulemeester, E., & Herroelen, W. (2007). Heuristic procedures for reactive project scheduling. *Computers & Industrial Engineering*, *52*(1), 11–28. http://doi.org/10.1016/j.cie.2006.10.002

Van de Vonder, S., Demeulemeester, E., & Herroelen, W. (2007). A classification of predictive-reactive project scheduling procedures. *Journal of Scheduling*, *10*(3), 195–207. http://doi.org/10.1007/s10951-007-0011-2

Van de Vonder, S., Demeulemeester, E., Herroelen, W., & Leus, R. (2005). The use of buffers in project management: The trade-off between stability and makespan. *International Journal of Production Economics*, *97*(2), 227–240. http://doi.org/10.1016/j.ijpe.2004.08.004

Van de Vonder, S., Demeulemeester, E., Leus, R., & Herroelen, W. (2006). Proactive-reactive project scheduling trade-offs and procedures. In *Perspectives in modern project scheduling* (pp. 25–51). Springer.

Van Peteghem, V., & Vanhoucke, M. (2014). An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances. *European Journal of Operational Research*, *235*(1), 62–72. http://doi.org/10.1016/j.ejor.2013.10.012

Van Slyke, R. (1963). Uses of Monte Carlo in PERT. RAND Corporation. Retrieved from http://www.rand.org/pubs/research_memoranda/RM3367.html

Vanhoucke, M., Demeulemeester, E., & Herroelen, W. (2001). An Exact Procedure for the Resource-Constrained Weighted Earliness – Tardiness Project Scheduling. *Annals of Operations Research*, *102*, 179–196.

Wiest, J. D., & Levy, F. K. (1977). *A management guide to PERT/CPM: with GERT/PDM/DCPM and other networks*. Prentice-Hall. Retrieved from https://books.google.com/books?id=GAFUAAAAMAAJ&pgis=1

Wilson, J. (2003). Gantt charts: A centenary appreciation. *European Journal of Operational Research*, *149*(2), 430–437. http://doi.org/http://dx.doi.org/10.1016/S0377-2217(02)00769-5

Yang, Z., & Wang, Z. (2010). Comparison between AON and AOA Network Diagrams. *IEEE International Conference on Industrial Engineering and Engineering Management*, 1507–1509. Retrieved from 10.1109/ICIEEM.2010.5646036\n635076841633216431.pdf

Zhu, J., Isac, G., & Zhao, D. (2005). Pareto optimization in topological vector spaces. *Journal of Mathematical Analysis and Applications*, *301*(1), 22–31. http://doi.org/10.1016/j.jmaa.2004.07.003

APPENDIX I (PROOFS)

1. Proof for expression **(3.12)**

Expression

$$d_i^{max} \geq d_i^{nom} + 1 \quad if\ \alpha^- \geq \frac{1}{d_i^{nom} + 1}, \qquad \alpha^- \in \mathbb{R}_0^+$$

Proof

$$d_i^{max} = \frac{d_i^{nom}}{1 - \alpha^-}$$

$$(1 - \alpha^-) d_i^{max} = d_i^{nom}$$

For $d_i^{max} \geq d_i^{nom} + 1$ to be true, then

$$(1 - \alpha^-)(d_i^{nom} + 1) \leq d_i^{nom}$$

$$d_i^{nom} + 1 - d_i^{nom}\alpha^- - \alpha^- \leq d_i^{nom}$$

$$1 - d_i^{nom}\alpha^- - \alpha^- \leq 0$$

$$-\alpha^-(d_i^{nom} + 1) \leq -1$$

$$\alpha^- \geq \frac{1}{d_i^{nom} + 1}$$

This proofs the expression.

2. Proof for expression **(3.13)**

Expression

$$d_i^{min} \leq d_i^{nom} - 1 \quad if\ \alpha^+ \geq \frac{1}{d_i^{nom} - 1}, \qquad \alpha^+ \in \mathbb{R}_0^+$$

Proof

$$d_i^{min} = \frac{d_i^{nom}}{1 + \alpha^+}$$

$$(1 + \alpha^+) d_i^{min} = d_i^{nom}$$

For $d_i^{min} \leq d_i^{nom} - 1$ to be true, then

$$(1 + \alpha^+)(d_i^{nom} - 1) \geq d_i^{nom}$$

$$d_i^{nom} - 1 + d_i^{nom}\alpha^+ - \alpha^+ \geq d_i^{nom}$$

$$-1 + d_i^{nom}\alpha^+ - \alpha^+ \geq 0$$

$$\alpha^+(d_i^{nom} - 1) \geq 1$$

$$\alpha^+ \geq \frac{1}{d_i^{nom} - 1}$$

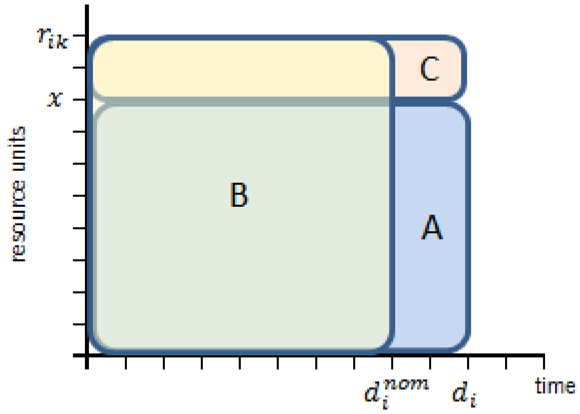
This proofs the expression.

3. Proof for expression (3.17)

Expression

$$score_{ik} = r_{ik}(d_i - d_i^{nom}), \qquad \text{with } d_i^{min} \leq d_i \leq d_i^{max}(S), i \in V, k \in K$$

Note: To simplify the expressions the index k will be omitted in this demonstration.



Proof

Whatever the duration of an activity, within the limits previously described, its work content should be the same as the initially estimated (area of A = area of B in the illustration above). Therefore:

$$xd_i = r_i d_i^{nom}$$

$$x = r_i \frac{d_i^{nom}}{d_i}$$

$score_i$ is the remaining capacity for the duration of the activity (area of C), that is:

$$score_i = (r_i - x)d_i$$

$$score_i = (r_i - r_i \frac{d_i^{nom}}{d_i})d_i$$

$$score_i = r_i d_i - r_i d_i^{nom}$$

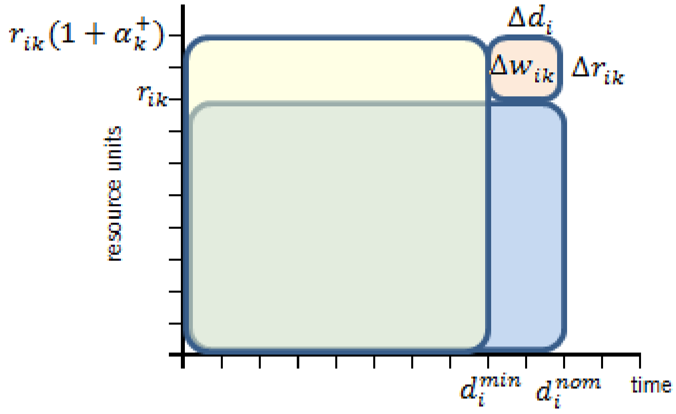
$$score_i = r_i(d_i - d_i^{nom})$$

This proofs the expression.

4. Proof for expression (3.26)

Expression

$$\Delta w_{ik} = r_{ik} d_i^{nom} \frac{(\alpha_k^+)^2}{(1+\alpha_k^+)}, \qquad i \in V, k \in K$$



Proof

Consider the illustration above that shows that:

$$\Delta w_{ik} = \Delta r_{ik} \Delta d_i$$

$$\Delta w_{ik} = (r_{ik}^{max} - r_{ik})(d_i^{nom} - d_i^{min})$$

Given that:

$$d_i^{min} = \frac{d_i^{nom}}{1+\alpha_k^+} \quad \text{and} \quad r_{ik}^{max} = r_{ik}(1+\alpha_k^+)$$

$$\Delta w_{ik} = (r_{ik}(1+\alpha_k^+) - r_{ik})(d_i^{nom} - \frac{d_i^{nom}}{1+\alpha_k^+})$$

$$\Delta w_{ik} = (r_{ik}\alpha_k^+)(d_i^{nom} - \frac{d_i^{nom}}{1+\alpha_k^+})$$

$$\Delta w_{ik} = r_{ik}\alpha_k^+ d_i^{nom}(1 - \frac{1}{1+\alpha_k^+})$$

$$\Delta w_{ik} = r_{ik} d_i^{nom}(\alpha_k^+ - \frac{\alpha_k^+}{1+\alpha_k^+})$$

$$\Delta w_{ik} = r_{ik} d_i^{nom} \frac{\alpha_k^+(1+\alpha_k^+) - \alpha_k^+}{1+\alpha_k^+}$$

$$\Delta w_{ik} = r_{ik} d_i^{nom} \frac{\alpha_k^+ + (\alpha_k^+)^2 - \alpha_k^+}{1+\alpha_k^+}$$

$$\Delta w_{ik} = r_{ik} d_i^{nom} \frac{(\alpha_k^+)^2}{1+\alpha_k^+}$$

This proofs the expression.

APPENDIX II (ADDITIONAL EXAMPLES)

1. Alternative schedule's slack and score

Table 22 presents slack values for schedules S, S' and S'' which are the examples from 3.2. Besides the project's basic data (i, d_i^{nom}, r_1, r_2), it displays calculated data for $slack_i$ that enables an activity wide comparison of the intrinsic schedule flexibility. It also displays calculated data for $score_{ir}^{INT} = slack_i . r_k$ that is the schedule specific score, defined as the score of an activity with positive slack and without taking into account the effect of resource flexibility limitations. This enables a project wide comparison using $\sum_i score_{ik}^{INT}$. Note that $\sum_i slack_i$ is not a valid comparison value (two distinct activities with the same slack and distinct resource requirements contribute differently to the schedule's flexibility) being the correct one $slack_i . r_{ik}$.

Table 22: Slack and score for distinct schedules (S, S', S'')

i	d_i^{nom}	$slack_i$			r_1	r_2	$score_{i1}^{INT}$			$score_{i2}^{INT}$		
		S	S'	S''			S	S'	S''	S	S'	S''
1	0	0	0	0	0	0	0	0	0	0	0	0
2	2	5	5	0	2	6	10	10	0	30	30	0
3	10	0	0	0	5	8	0	0	0	0	0	0
4	7	0	0	0	7	3	0	0	0	0	0	0
5	3	0	0	0	4	8	0	0	0	0	0	0
6	2	0	0	0	8	3	0	0	0	0	0	0
7	2	0	0	0	2	8	0	0	0	0	0	0
8	2	0	0	0	1	7	0	0	0	0	0	0
9	1	1	0	0	4	1	4	0	0	1	0	0
10	4	0	0	0	8	3	0	0	0	0	0	0
11	7	5	2	2	6	3	30	12	12	15	6	6
12	8	0	0	0	8	5	0	0	0	0	0	0
13	9	0	0	0	10	6	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0
$\sum_i score_{ik}^{INT}$							44	22	12	46	36	6

APPENDIX III (CHARTS)

Some charts presented in the text are unclear because of the large amount of data which they enclose. While this is not critical to the purpose of their presentation, they are reproduced here with higher clarity.

The following charts are reproduced:

a) Project duration for all 480 J30 instances;

b) $\sum slack$ of each scheduling method for all J30 instances.

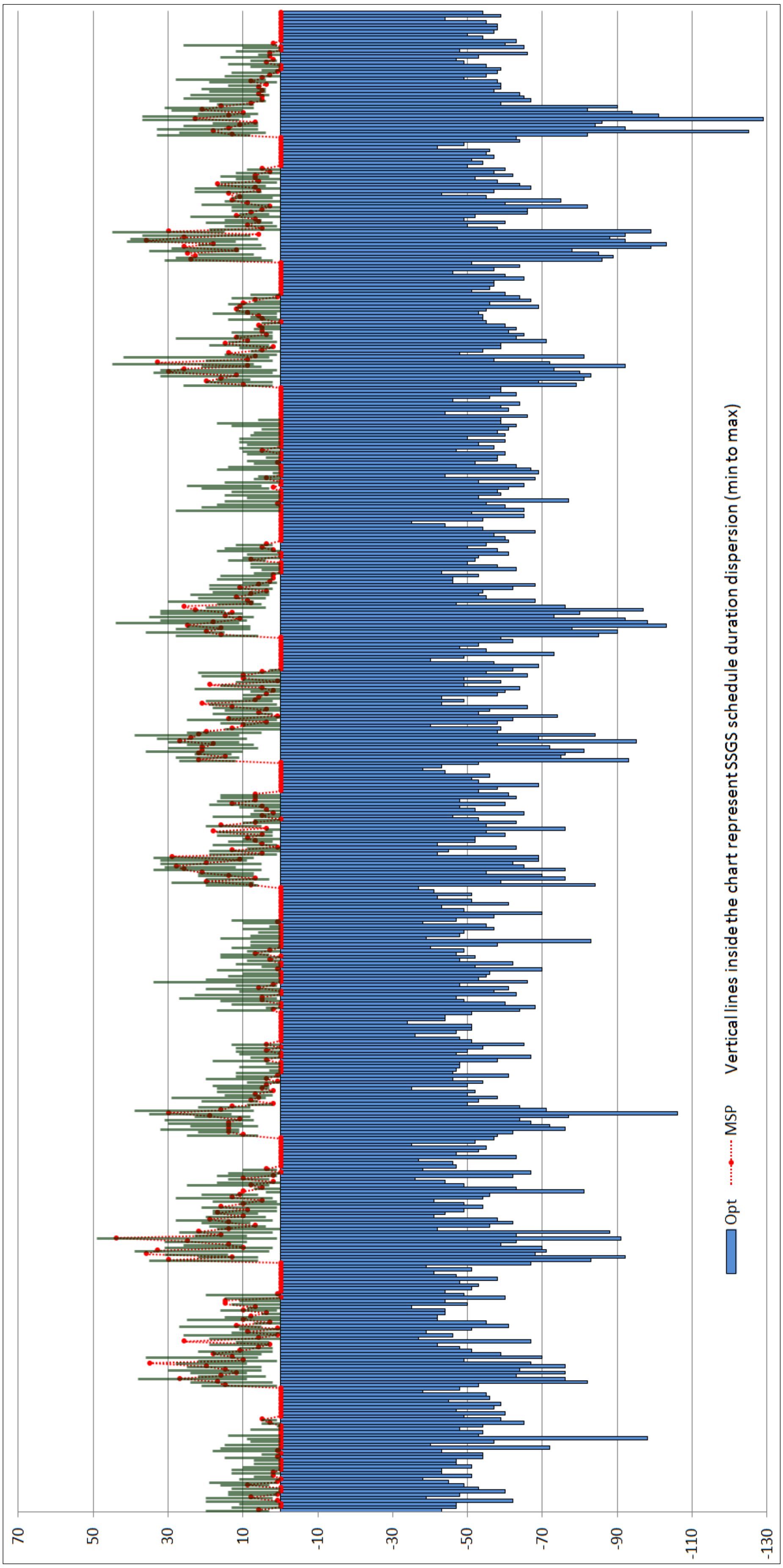


Figure 33: Project duration for all 480 J30 instances (large chart)

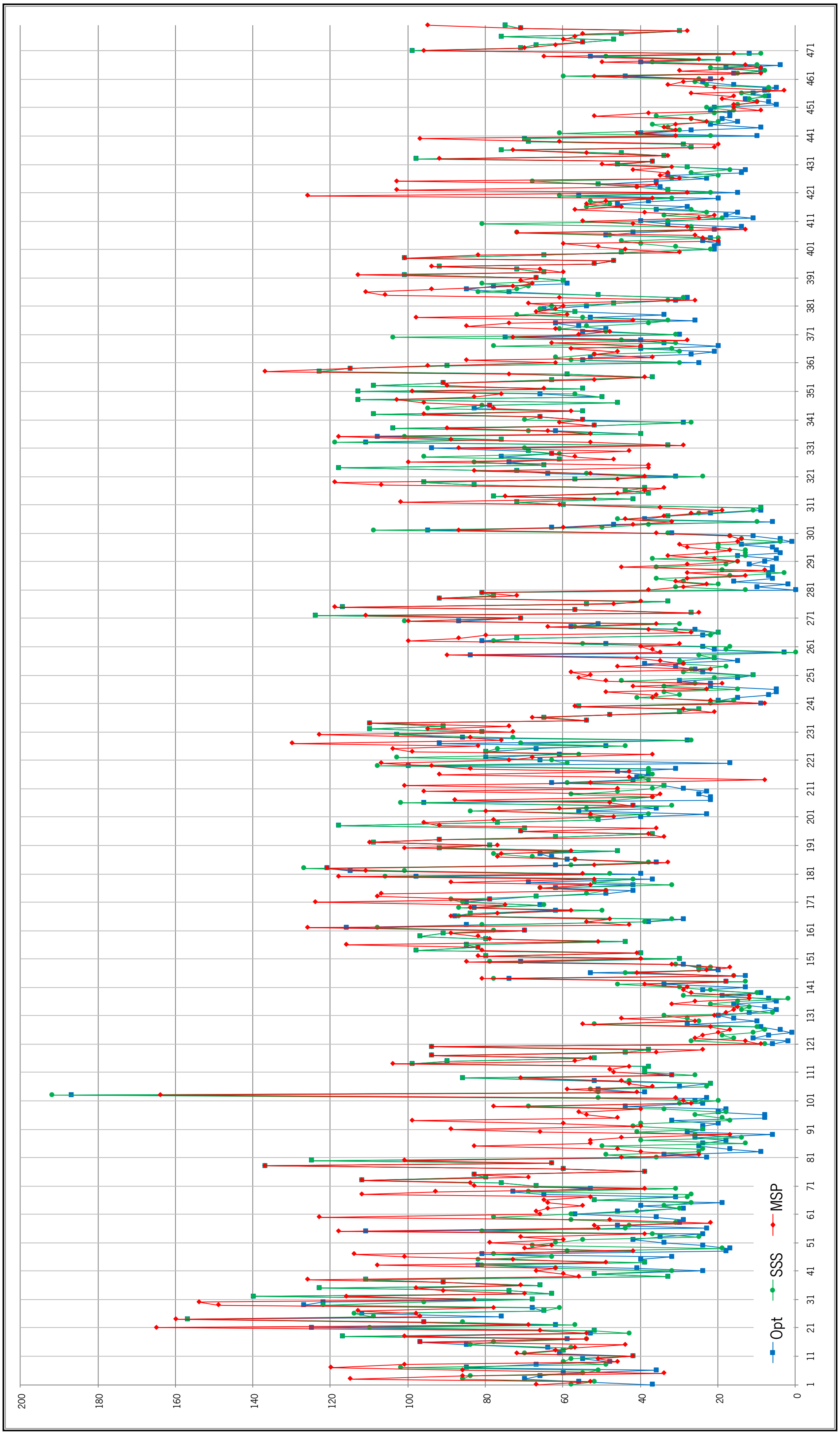


Figure 34: Σslack of each scheduling method for all J30 instances (larger chart)

APPENDIX IV (TABLES)

In order to not overload the main text with numbers, the values are presented there are preferentially in compact forms using charts and small tables. Larger tables having extensive data are presented in this appendix for reference.

The following tables are presented:

a) Table 23: Average #NC and Σslack for J30 instances with same <NC,RF,RS>;

b) Table 24: T, #NC and Σslack for all J30 instances;

c) Table 25: Average SIF for J30 instances with same <NC,RF,RS>;

d) Table 26: Average relative SIF for J30 instances with same <NC,RF,RS>;

e) Table 27: SIF.

Table 23: Average #NC and Σslack for J30 instances with same <NC,RF,RS>

NC	RF	RS	$\#NC$			$\Sigma slack$		
			Opt	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**
1.5	1	0.2	9.4	9.4	9.3	58.0	65.5	75.9
		0.5	9.7	9.4	9.9	69.0	65.8	64.9
		0.7	10.4	10.4	10.8	101.0	97.7	117.9
		1	9.3	9.3	9.3	82.1	82.1	86.2
	2	0.2	8.1	9.4	8.3	39.8	58.3	74.9
		0.5	8.1	9.0	8.6	41.0	51.4	66.3
		0.7	8.7	8.2	9.3	46.5	39.7	67.8
		1	9.8	9.8	10.4	84.2	84.2	83.1
	3	0.2	6.6	6.2	7.4	21.0	31.8	47.3
		0.5	6.2	6.8	8.7	22.2	31.9	58.9
		0.7	7.9	8.4	7.8	54.8	56.7	55.2
		1	8.8	8.8	8.5	62.7	62.7	60.0
	4	0.2	3.4	5.0	6.6	9.4	20.4	25.7
		0.5	4.2	5.7	6.6	12.5	16.6	20.8
		0.7	7.0	7.7	6.9	35.0	38.4	38.0
		1	9.4	9.4	9.7	72.7	72.7	74.3
1.8	1	0.2	9.5	9.7	9.4	72.1	71.1	72.4
		0.5	9.7	9.5	10.4	63.0	64.3	84.5
		0.7	10.3	10.7	10.9	70.0	71.3	74.1
		1	10.1	10.1	9.2	76.6	76.6	72.4
	2	0.2	7.5	9.0	9.3	38.5	55.1	58.7
		0.5	8.7	8.8	8.8	44.1	48.8	67.1
		0.7	9.8	9.8	9.9	71.2	69.7	87.7
		1	9.8	9.8	9.6	67.0	67.0	62.9
	3	0.2	4.9	6.8	6.0	15.0	28.4	34.1
		0.5	7.1	5.8	8.3	27.5	20.8	44.6
		0.7	8.2	7.5	8.2	49.2	48.3	60.2
		1	10.3	10.3	9.7	73.4	73.4	71.5
	4	0.2	3.0	5.0	6.1	7.2	22.2	27.1
		0.5	4.2	5.3	5.7	7.3	16.6	21.3
		0.7	5.8	6.5	6.8	35.5	36.4	41.6
		1	10.2	10.2	10.1	60.9	60.9	68.1
2.1	1	0.2	10.4	10.0	8.0	69.3	70.3	56.1
		0.5	10.1	9.7	9.6	70.9	69.1	69.6
		0.7	10.9	11.1	11.0	72.2	74.2	79.0
		1	11.5	11.5	10.5	85.5	85.5	85.7
	2	0.2	8.8	8.6	8.5	39.0	52.2	54.4
		0.5	9.4	9.1	9.3	48.7	51.6	65.3
		0.7	10.0	9.8	9.6	56.7	58.7	73.9
		1	11.0	11.0	10.6	70.7	70.7	71.2
	3	0.2	7.3	7.8	7.3	26.7	41.3	38.0
		0.5	6.6	8.4	8.0	30.8	38.4	50.8
		0.7	7.1	7.5	7.3	28.1	33.9	48.3
		1	9.9	9.9	9.2	53.1	53.1	53.8
	4	0.2	5.5	6.5	6.6	19.8	30.3	31.7
		0.5	4.5	5.6	6.9	11.7	14.5	19.0
		0.7	6.4	6.7	6.6	23.8	25.5	28.8
		1	10.4	10.4	10.1	63.6	63.6	64.9

Table 24: T, #NC and Σslack for all J30 instances

a) instances having < NC=1.5, RF=1, RS∈{0.2, 0.5, 0.7, 1} >

RS	e	T			$\#NC$			$\Sigma slack$		
		Opt	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**
0.2	J301_1	43	46	49	7	9	8	37	58	67
	J301_2	47	47	47	12	12	12	56	52	53
	J301_3	47	47	47	10	11	11	70	86	115
	J301_4	62	63	63	10	12	11	66	84	86
	J301_5	39	41	47	10	9	7	60	55	34
	J301_6	48	49	49	10	9	10	36	51	86
	J301_7	60	60	60	7	8	11	85	102	120
	J301_8	53	53	53	9	8	9	67	49	101
	J301_9	49	52	58	9	9	7	48	60	46
	J301_10	45	46	46	10	7	7	55	58	51
0.5	J302_1	38	38	38	11	11	11	42	42	42
	J302_2	51	53	53	11	11	9	61	70	72
	J302_3	43	43	45	9	9	10	60	60	62
	J302_4	43	43	43	11	10	10	64	58	57
	J302_5	51	51	51	7	8	6	85	84	44
	J302_6	47	47	47	12	11	12	97	78	97
	J302_7	47	47	47	12	11	10	59	54	54
	J302_8	54	54	55	9	9	11	117	117	101
	J302_9	54	54	54	8	7	9	53	43	54
	J302_10	43	43	44	7	7	11	52	52	66
0.7	J303_1	72	72	72	11	9	12	125	110	165
	J303_2	40	40	40	11	9	12	62	57	69
	J303_3	57	57	57	9	9	9	96	86	96
	J303_4	98	98	98	10	10	9	157	157	160
	J303_5	53	53	53	10	12	13	76	109	97
	J303_6	54	54	54	11	12	10	112	114	98
	J303_7	48	48	48	12	12	13	65	65	113
	J303_8	54	54	54	10	9	9	68	61	78
	J303_9	65	65	68	10	11	10	127	122	149
	J303_10	59	60	64	10	11	11	122	96	154
1	J304_1	49	49	49	9	9	9	68	68	83
	J304_2	60	60	60	13	13	10	140	140	116
	J304_3	47	47	47	10	10	7	63	63	70
	J304_4	57	57	57	9	9	11	74	74	91
	J304_5	59	59	59	11	11	11	123	123	98
	J304_6	45	45	45	9	9	10	66	66	71
	J304_7	56	56	56	8	8	8	91	91	91
	J304_8	55	55	55	10	10	11	111	111	126
	J304_9	38	38	38	7	7	8	33	33	56
	J304_10	48	48	48	7	7	8	52	52	60

b) instances having < NC=1.5, RF=2, RS∈{0.2, 0.5, 0.7, 1} >

RS	e	T			$\#NC$			$\Sigma slack$		
		Opt	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**
0.2	J305_1	53	54	68	7	10	8	24	32	67
	J305_2	82	84	99	9	8	6	41	62	62
	J305_3	76	85	103	10	10	8	82	81	108
	J305_4	63	67	79	7	10	9	39	39	49
	J305_5	76	85	88	4	9	10	40	82	73
	J305_6	64	69	79	10	11	8	32	63	101
	J305_7	76	81	96	10	8	8	81	78	114
	J305_8	67	76	102	7	9	9	18	59	42
	J305_9	49	50	59	10	8	10	17	19	70
	J305_10	70	76	83	7	11	7	24	68	63
0.5	J306_1	59	64	77	8	10	8	34	62	79
	J306_2	51	53	62	8	9	12	42	55	60
	J306_3	48	50	54	7	8	9	35	25	71
	J306_4	42	45	45	7	9	9	24	37	39
	J306_5	67	70	93	11	11	10	111	81	118
	J306_6	37	37	43	9	8	7	23	44	51
	J306_7	46	47	47	7	11	9	46	43	52
	J306_8	39	39	48	9	8	5	30	31	22
	J306_9	51	51	52	8	8	9	29	58	48
	J306_10	61	66	73	7	8	8	36	78	123
0.7	J307_1	55	55	58	7	9	10	57	58	66
	J307_2	42	42	52	9	8	7	46	41	67
	J307_3	42	42	50	7	8	11	29	30	64
	J307_4	44	44	48	9	8	10	40	34	55
	J307_5	44	45	54	9	8	9	19	27	64
	J307_6	35	35	42	9	10	11	52	52	65
	J307_7	50	50	65	7	7	7	31	28	53
	J307_8	44	44	59	10	6	9	65	27	112
	J307_9	60	60	60	10	10	12	73	69	93
	J307_10	49	49	50	10	8	7	53	31	39
1	J308_1	44	44	44	7	7	11	67	67	83
	J308_2	51	51	51	9	9	9	76	76	84
	J308_3	53	53	53	10	10	10	112	112	112
	J308_4	48	48	48	11	11	11	80	80	69
	J308_5	58	58	58	11	11	11	83	83	83
	J308_6	47	47	47	8	8	8	39	39	39
	J308_7	41	41	41	8	8	8	60	60	60
	J308_8	51	51	51	11	11	13	137	137	137
	J308_9	39	39	39	12	12	12	63	63	63
	J308_10	67	67	67	11	11	11	125	125	101

c) instances having < NC=1.5, RF=3, RS∈{0.2, 0.5, 0.7, 1} >

RS	e	T			$\#NC$			$\Sigma slack$		
		Opt	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**
0.2	J309_1	83	89	113	9	6	7	23	36	45
	J309_2	92	99	105	8	6	6	34	49	25
	J309_3	68	82	104	5	5	8	9	25	40
	J309_4	71	74	104	7	5	6	17	24	46
	J309_5	70	72	80	9	10	8	25	50	83
	J309_6	59	69	73	8	2	7	24	13	53
	J309_7	63	72	88	5	7	7	18	40	53
	J309_8	91	92	135	6	5	9	26	14	45
	J309_9	63	72	79	2	8	8	6	26	17
	J309_10	88	90	110	7	8	8	28	41	66
0.5	J3010_1	42	42	56	8	8	6	24	24	89
	J3010_2	56	60	63	7	7	10	24	42	40
	J3010_3	62	70	76	3	7	7	20	40	60
	J3010_4	58	60	77	8	6	9	32	17	99
	J3010_5	41	45	51	3	7	9	8	19	46
	J3010_6	44	51	61	5	8	11	8	26	54
	J3010_7	49	50	58	7	5	11	20	18	56
	J3010_8	54	55	70	4	7	9	18	34	40
	J3010_9	49	49	59	10	8	9	44	69	78
	J3010_10	41	42	46	7	5	6	24	30	27
0.7	J3011_1	54	56	67	4	5	6	26	20	29
	J3011_2	56	62	67	7	7	6	23	51	31
	J3011_3	81	81	91	9	10	8	187	192	164
	J3011_4	63	63	68	8	9	7	39	51	41
	J3011_5	49	52	57	8	9	9	51	53	59
	J3011_6	44	44	46	8	7	7	30	23	37
	J3011_7	36	39	46	9	9	9	22	22	43
	J3011_8	62	63	64	9	10	10	52	43	45
	J3011_9	67	67	67	10	10	9	86	86	71
	J3011_10	38	39	42	7	8	7	32	26	32
1	J3012_1	47	47	47	8	8	9	39	39	47
	J3012_2	46	46	46	6	6	8	39	39	48
	J3012_3	37	37	37	11	11	10	38	38	43
	J3012_4	63	63	63	9	9	10	99	99	104
	J3012_5	47	47	47	9	9	7	90	90	57
	J3012_6	53	53	53	7	7	8	52	52	53
	J3012_7	55	55	55	10	10	10	94	94	94
	J3012_8	35	35	35	11	11	10	44	44	36
	J3012_9	52	52	52	8	8	5	38	38	24
	J3012_10	57	57	57	9	9	8	94	94	94

d) instances having < NC=1.5, RF=4, RS∈{0.2, 0.5, 0.7, 1} >

RS	e	T			$\#NC$			$\Sigma slack$		
		Opt	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**
0.2	J3013_1	58	64	68	5	5	5	6	8	9
	J3013_2	62	72	76	2	6	6	2	27	13
	J3013_3	76	87	90	5	7	6	11	16	26
	J3013_4	72	78	86	3	4	7	7	19	24
	J3013_5	67	77	81	1	3	6	1	11	20
	J3013_6	64	71	75	2	4	9	4	8	17
	J3013_7	77	85	96	1	4	6	9	10	22
	J3013_8	106	119	136	8	8	7	28	52	55
	J3013_9	71	78	87	3	5	6	10	25	26
	J3013_10	64	72	77	4	4	8	16	28	45
0.5	J3014_1	50	52	52	4	8	8	20	34	21
	J3014_2	53	59	61	4	5	6	12	6	18
	J3014_3	58	62	64	3	4	5	5	14	16
	J3014_4	50	52	57	3	6	7	8	12	15
	J3014_5	52	54	54	2	4	9	16	22	32
	J3014_6	35	38	40	4	7	9	5	15	26
	J3014_7	50	53	54	3	2	4	7	2	12
	J3014_8	54	54	55	7	8	4	19	29	12
	J3014_9	46	49	50	3	7	8	9	10	27
	J3014_10	61	61	62	9	6	6	24	22	29
0.7	J3015_1	46	46	46	3	9	6	13	30	28
	J3015_2	47	47	47	8	10	8	34	46	39
	J3015_3	48	48	48	6	5	6	18	13	18
	J3015_4	48	48	48	9	10	9	74	78	81
	J3015_5	58	61	62	5	6	5	13	16	16
	J3015_6	67	67	67	8	5	6	53	44	41
	J3015_7	47	47	47	8	8	6	20	25	23
	J3015_8	50	50	54	7	9	5	25	22	17
	J3015_9	54	54	54	6	6	8	29	31	32
	J3015_10	65	65	69	10	9	10	71	79	85
1	J3016_1	51	51	51	10	10	11	30	30	40
	J3016_2	48	48	48	10	10	11	80	80	82
	J3016_3	36	36	36	9	9	9	40	40	41
	J3016_4	47	47	47	11	11	11	98	98	81
	J3016_5	51	51	51	8	8	8	82	82	82
	J3016_6	51	51	51	7	7	9	85	85	116
	J3016_7	34	34	34	11	11	10	44	44	51
	J3016_8	44	44	44	10	10	11	80	80	79
	J3016_9	44	44	44	9	9	8	97	97	82
	J3016_10	51	51	51	9	9	9	91	91	89

e) instances having < NC=1.8, RF=1, RS∈{0.2, 0.5, 0.7, 1} >

RS	e	T			$\#NC$			$\Sigma slack$		
		Opt	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**
0.2	J3017_1	64	66	66	9	11	6	70	78	70
	J3017_2	68	68	68	10	10	11	116	108	126
	J3017_3	60	60	60	11	11	8	85	81	43
	J3017_4	49	49	54	8	8	8	38	39	54
	J3017_5	47	48	52	8	8	10	29	32	48
	J3017_6	63	63	63	11	11	11	88	87	89
	J3017_7	57	57	57	11	11	11	84	84	77
	J3017_8	61	61	67	11	10	11	62	50	58
	J3017_9	48	49	50	8	11	11	83	87	84
	J3017_10	66	66	66	8	6	7	66	65	75
0.5	J3018_1	53	53	53	14	12	15	85	86	124
	J3018_2	55	55	55	10	12	10	79	89	79
	J3018_3	56	56	56	8	8	10	67	67	108
	J3018_4	70	70	71	7	8	9	49	54	107
	J3018_5	52	52	52	9	9	9	42	49	49
	J3018_6	62	62	62	7	8	7	62	66	66
	J3018_7	48	48	51	8	5	8	42	32	53
	J3018_8	52	52	52	13	11	13	69	52	89
	J3018_9	47	50	54	10	11	10	37	42	52
	J3018_10	49	49	52	11	11	13	98	106	118
0.7	J3019_1	40	40	40	9	12	11	40	48	55
	J3019_2	58	58	58	13	12	14	115	101	111
	J3019_3	83	83	83	11	12	11	121	127	121
	J3019_4	39	39	39	12	10	11	62	58	52
	J3019_5	48	48	48	8	9	7	36	38	33
	J3019_6	49	49	49	10	9	10	59	57	57
	J3019_7	57	57	57	11	13	14	63	68	77
	J3019_8	55	55	55	8	9	9	66	78	76
	J3019_9	38	38	39	10	10	10	46	46	58
	J3019_10	47	47	47	11	11	12	92	92	101
1	J3020_1	57	57	57	12	12	11	79	79	77
	J3020_2	70	70	70	9	9	8	109	109	110
	J3020_3	49	49	49	11	11	11	92	92	92
	J3020_4	43	43	43	11	11	8	62	62	34
	J3020_5	61	61	61	9	9	8	37	37	38
	J3020_6	51	51	51	8	8	8	71	71	71
	J3020_7	42	42	42	8	8	7	70	70	36
	J3020_8	51	51	51	9	9	7	118	118	92
	J3020_9	41	41	41	12	12	12	77	77	96
	J3020_10	37	37	37	12	12	12	51	51	78

f) instances having < NC=1.8, RF=2, RS∈{0.2, 0.5, 0.7, 1} >

RS	e	T			$\#NC$			$\Sigma slack$		
		Opt	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**
0.2	J3021_1	84	90	92	7	7	6	40	53	47
	J3021_2	59	66	79	5	11	11	23	38	53
	J3021_3	76	79	83	9	11	9	56	84	80
	J3021_4	70	76	84	6	9	10	36	54	61
	J3021_5	55	62	76	9	5	10	42	32	42
	J3021_6	76	82	102	9	12	10	96	102	48
	J3021_7	65	77	93	7	8	7	22	47	88
	J3021_8	62	67	82	8	11	10	22	37	37
	J3021_9	69	78	80	8	7	8	25	58	35
	J3021_10	69	74	98	7	9	12	23	46	96
0.5	J3022_1	42	43	47	9	8	8	29	37	46
	J3022_2	45	45	58	9	9	9	34	34	101
	J3022_3	63	63	64	10	11	10	63	59	53
	J3022_4	42	43	47	9	7	3	42	38	8
	J3022_5	52	54	59	10	9	9	41	40	43
	J3022_6	52	52	61	8	7	12	38	37	92
	J3022_7	60	62	65	8	7	7	46	38	43
	J3022_8	55	57	73	4	9	10	31	38	84
	J3022_9	76	76	80	13	12	10	100	108	94
	J3022_10	55	60	71	7	9	10	17	59	107
0.7	J3023_1	63	63	70	12	11	10	66	63	74
	J3023_2	53	53	53	11	11	10	80	103	68
	J3023_3	46	46	51	11	11	9	61	56	37
	J3023_4	65	65	67	6	6	9	80	80	99
	J3023_5	52	52	56	9	10	11	67	77	104
	J3023_6	48	48	53	11	11	9	49	44	82
	J3023_7	60	60	73	9	9	11	92	71	130
	J3023_8	48	48	55	9	9	10	28	27	76
	J3023_9	63	63	70	8	8	8	86	73	84
	J3023_10	61	61	68	12	12	12	103	103	123
1	J3024_1	53	53	53	11	11	12	81	81	73
	J3024_2	58	58	58	11	11	9	110	110	95
	J3024_3	69	69	69	9	9	9	91	91	74
	J3024_4	53	53	53	13	13	13	110	110	110
	J3024_5	51	51	51	8	8	8	54	54	54
	J3024_6	56	56	56	10	10	10	65	65	68
	J3024_7	44	44	44	10	10	10	48	48	48
	J3024_8	38	38	38	8	8	6	30	30	21
	J3024_9	43	43	43	8	8	8	25	25	29
	J3024_10	53	53	53	10	10	11	56	56	57

g) instances having < NC=1.8, RF=3, RS∈{0.2, 0.5, 0.7, 1} >

RS	e	T			$\#NC$			$\Sigma slack$		
		Opt	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**
0.2	J3025_1	93	105	115	5	6	2	9	22	8
	J3025_2	75	86	90	4	6	5	20	16	22
	J3025_3	76	89	98	5	7	6	15	41	37
	J3025_4	81	91	102	3	8	5	7	30	36
	J3025_5	72	78	93	3	8	9	5	34	49
	J3025_6	58	65	76	3	6	6	5	15	23
	J3025_7	95	106	122	5	8	6	22	34	42
	J3025_8	69	75	93	7	7	6	22	26	19
	J3025_9	84	95	106	7	7	7	30	45	49
	J3025_10	58	63	78	7	5	8	15	21	56
0.5	J3026_1	59	59	72	6	6	9	11	11	53
	J3026_2	40	40	50	8	6	10	24	29	58
	J3026_3	58	58	62	8	8	6	26	27	22
	J3026_4	62	63	76	7	5	9	31	18	46
	J3026_5	74	74	75	10	8	8	39	29	29
	J3026_6	53	55	59	8	8	9	15	30	35
	J3026_7	56	56	60	4	4	8	21	21	41
	J3026_8	66	66	79	9	6	9	84	25	90
	J3026_9	43	44	64	3	0	6	3	0	35
	J3026_10	49	51	56	8	7	9	21	18	37
0.7	J3027_1	43	43	49	7	4	8	24	17	40
	J3027_2	58	58	62	8	8	6	49	55	30
	J3027_3	60	60	62	7	5	11	81	78	100
	J3027_4	64	64	69	9	9	8	72	72	87
	J3027_5	49	49	68	10	8	13	24	22	80
	J3027_6	59	59	60	6	6	4	20	20	27
	J3027_7	49	49	59	8	9	10	26	31	38
	J3027_8	66	66	76	10	8	7	58	57	64
	J3027_9	55	55	60	10	8	8	51	30	36
	J3027_10	62	62	62	7	10	7	87	101	100
1	J3028_1	69	69	69	10	10	10	71	71	71
	J3028_2	57	57	57	10	10	9	124	124	111
	J3028_3	40	40	40	8	8	6	27	27	25
	J3028_4	49	49	49	13	13	13	57	57	57
	J3028_5	73	73	73	10	10	10	117	117	119
	J3028_6	55	55	55	12	12	9	54	54	47
	J3028_7	48	48	48	8	8	9	33	33	40
	J3028_8	53	53	53	11	11	11	92	92	92
	J3028_9	62	62	62	10	10	9	78	78	72
	J3028_10	59	59	59	11	11	11	81	81	81

h) instances having < NC=1.8, RF=4, RS∈{0.2, 0.5, 0.7, 1} >

RS	e	T			$\#NC$			$\Sigma slack$		
		Opt	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**
0.2	J3029_1	85	91	101	0	3	7	0	13	38
	J3029_2	90	104	110	6	5	9	10	31	29
	J3029_3	78	86	94	2	5	6	2	20	23
	J3029_4	103	111	128	3	7	7	16	29	31
	J3029_5	98	109	116	4	5	6	6	36	28
	J3029_6	92	101	103	3	4	4	7	17	13
	J3029_7	73	80	88	3	2	6	7	3	28
	J3029_8	80	90	93	3	6	4	6	19	8
	J3029_9	97	112	120	4	8	6	6	36	45
	J3029_10	76	80	102	2	5	6	12	18	28
0.5	J3030_1	47	52	55	3	5	4	8	15	15
	J3030_2	68	75	77	3	6	4	5	37	21
	J3030_3	55	59	67	9	6	6	15	13	33
	J3030_4	53	56	61	4	6	6	4	13	23
	J3030_5	54	57	58	3	5	5	5	13	17
	J3030_6	62	64	73	5	4	6	6	20	28
	J3030_7	68	71	74	6	7	7	14	20	30
	J3030_8	46	47	49	1	4	9	1	4	15
	J3030_9	46	48	48	3	4	5	4	14	14
	J3030_10	53	55	55	5	6	5	11	17	17
0.7	J3031_1	43	43	43	6	7	8	32	33	36
	J3031_2	63	63	63	7	8	6	95	109	87
	J3031_3	58	58	58	7	9	8	63	50	60
	J3031_4	50	50	50	10	9	12	47	38	42
	J3031_5	52	56	60	4	5	5	6	10	32
	J3031_6	53	53	53	5	6	6	39	46	44
	J3031_7	61	61	61	7	7	5	33	33	34
	J3031_8	58	58	60	4	5	6	22	25	27
	J3031_9	50	52	55	4	4	6	9	11	19
	J3031_10	55	57	59	4	5	6	9	9	35
1	J3032_1	61	61	61	14	14	14	60	60	61
	J3032_2	60	60	60	10	10	11	72	72	102
	J3032_3	57	57	57	10	10	11	42	42	52
	J3032_4	68	68	68	11	11	10	78	78	75
	J3032_5	54	54	54	6	6	7	38	38	46
	J3032_6	44	44	44	10	10	9	44	44	39
	J3032_7	35	35	35	11	11	9	39	39	34
	J3032_8	54	54	54	12	12	13	83	83	107
	J3032_9	65	65	65	10	10	10	96	96	119
	J3032_10	51	51	51	8	8	7	57	57	46

i) instances having < NC=2.1, RF=1, RS∈{0.2, 0.5, 0.7, 1} >

RS	e	T			$\#NC$			$\Sigma slack$		
		Opt	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**
0.2	J3033_1	65	65	65	8	6	8	31	24	39
	J3033_2	60	60	60	11	10	9	64	54	53
	J3033_3	55	55	56	10	11	10	72	72	83
	J3033_4	77	77	77	10	10	5	118	118	38
	J3033_5	53	53	53	10	10	7	65	65	38
	J3033_6	59	59	59	11	10	10	74	83	100
	J3033_7	58	58	58	10	10	6	61	61	47
	J3033_8	61	64	63	12	13	9	76	96	57
	J3033_9	65	70	65	11	9	9	63	61	63
	J3033_10	53	53	53	11	11	7	69	69	43
0.5	J3034_1	68	68	72	11	9	10	94	70	87
	J3034_2	44	44	44	8	8	6	33	33	29
	J3034_3	69	69	69	10	10	9	111	119	53
	J3034_4	67	67	67	11	11	13	76	76	89
	J3034_5	63	63	63	11	11	11	108	101	118
	J3034_6	52	52	53	9	9	8	40	40	53
	J3034_7	58	58	58	11	11	9	62	69	64
	J3034_8	58	58	58	9	9	8	104	104	90
	J3034_9	60	60	60	12	12	12	52	52	52
	J3034_10	47	47	52	9	7	10	29	27	61
0.7	J3035_1	57	57	57	9	11	9	55	70	55
	J3035_2	53	53	53	10	10	10	66	66	66
	J3035_3	60	60	60	12	12	11	109	109	96
	J3035_4	50	50	50	11	11	13	55	55	58
	J3035_5	60	60	60	11	12	11	83	95	78
	J3035_6	58	58	58	11	10	11	79	81	79
	J3035_7	61	61	61	9	9	12	46	46	96
	J3035_8	63	63	63	13	13	12	113	113	103
	J3035_9	59	59	59	12	12	10	50	50	83
	J3035_10	59	59	59	11	11	11	66	57	76
1	J3036_1	66	66	66	11	11	9	113	113	99
	J3036_2	44	44	44	10	10	6	55	55	65
	J3036_3	61	61	61	12	12	11	109	109	90
	J3036_4	59	59	59	12	12	12	91	91	91
	J3036_5	64	64	64	11	11	10	63	63	52
	J3036_6	46	46	46	11	11	10	37	37	39
	J3036_7	56	56	56	8	8	8	59	59	74
	J3036_8	63	63	63	15	15	14	123	123	137
	J3036_9	59	59	59	12	12	12	115	115	115
	J3036_10	59	59	59	13	13	13	90	90	95

j) instances having < NC=2.1, RF=2, RS∈{0.2, 0.5, 0.7, 1} >

RS	e	T			$\#NC$			$\Sigma slack$		
		Opt	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**
0.2	J3037_1	79	81	89	8	7	11	25	30	62
	J3037_2	69	71	89	10	10	10	55	58	85
	J3037_3	81	89	97	10	8	8	53	62	37
	J3037_4	83	83	95	8	8	10	27	52	52
	J3037_5	80	82	110	6	10	8	21	30	46
	J3037_6	73	74	99	10	7	7	40	32	58
	J3037_7	92	97	101	7	10	8	20	78	40
	J3037_8	72	79	105	10	7	10	34	31	63
	J3037_9	57	59	66	7	6	5	40	45	28
	J3037_10	81	85	88	12	13	8	75	104	73
0.5	J3038_1	48	49	62	10	6	12	30	31	56
	J3038_2	54	54	59	10	8	6	55	49	48
	J3038_3	59	61	61	9	10	10	49	61	62
	J3038_4	59	62	74	10	9	11	56	54	85
	J3038_5	71	72	80	10	9	9	62	38	74
	J3038_6	63	65	75	10	11	9	26	33	42
	J3038_7	65	67	69	10	10	10	53	55	98
	J3038_8	61	63	66	7	10	7	34	72	59
	J3038_9	63	63	68	8	8	11	57	57	67
	J3038_10	60	60	66	10	10	8	65	66	62
0.7	J3039_1	55	55	55	9	9	9	54	63	60
	J3039_2	54	54	59	12	12	12	47	47	69
	J3039_3	54	55	60	10	9	7	31	33	26
	J3039_4	53	53	62	10	8	8	28	29	61
	J3039_5	55	55	67	5	5	11	51	51	106
	J3039_6	69	69	80	10	10	12	74	82	111
	J3039_7	56	56	66	13	12	9	85	72	94
	J3039_8	67	67	74	11	9	10	78	69	73
	J3039_9	64	65	65	10	13	11	59	81	68
	J3039_10	60	60	60	10	11	7	60	60	71
1	J3040_1	51	51	51	13	13	13	67	67	67
	J3040_2	56	56	56	12	12	13	101	101	113
	J3040_3	57	57	57	12	12	10	65	65	60
	J3040_4	57	57	57	10	10	8	72	72	66
	J3040_5	65	65	65	10	10	11	92	92	94
	J3040_6	60	60	60	8	8	7	52	52	52
	J3040_7	46	46	46	10	10	10	47	47	47
	J3040_8	57	57	57	14	14	14	101	101	101
	J3040_9	64	64	64	12	12	13	65	65	82
	J3040_10	51	51	51	9	9	7	45	45	30

k) instances having < NC=2.1, RF=3, RS∈{0.2, 0.5, 0.7, 1} >

RS	e	T			$\#NC$			$\Sigma slack$		
		Opt	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**
0.2	J3041_1	86	88	110	7	8	7	21	22	44
	J3041_2	89	94	112	9	8	7	21	31	51
	J3041_3	85	92	110	8	9	7	20	40	60
	J3041_4	78	89	90	7	9	7	24	45	20
	J3041_5	99	103	125	6	6	7	22	20	24
	J3041_6	103	108	121	6	7	7	49	48	26
	J3041_7	92	104	128	7	7	8	42	72	72
	J3041_8	88	94	114	7	7	6	21	27	13
	J3041_9	92	100	98	6	7	9	14	27	28
	J3041_10	99	114	129	10	10	8	33	81	42
0.5	J3042_1	58	58	63	7	7	10	40	33	55
	J3042_2	50	51	59	7	7	7	11	19	25
	J3042_3	60	62	66	6	10	6	18	34	21
	J3042_4	49	54	56	7	9	8	15	23	39
	J3042_5	52	52	64	7	9	11	36	27	57
	J3042_6	66	69	74	7	10	5	28	54	45
	J3042_7	66	66	71	7	9	8	46	48	54
	J3042_8	82	82	85	5	5	7	38	53	49
	J3042_9	60	63	69	6	12	9	20	32	37
	J3042_10	75	75	88	7	6	9	56	61	126
0.7	J3043_1	55	57	66	7	8	6	15	22	28
	J3043_2	43	43	57	9	9	11	33	33	103
	J3043_3	57	62	63	7	8	6	35	41	41
	J3043_4	67	67	74	10	10	6	51	51	36
	J3043_5	64	68	81	5	5	7	36	68	103
	J3043_6	58	59	64	5	6	7	23	32	30
	J3043_7	52	52	59	9	9	7	33	20	35
	J3043_8	62	65	69	6	4	5	14	27	33
	J3043_9	57	57	60	6	9	10	13	17	42
	J3043_10	60	60	65	7	7	8	28	28	32
1	J3044_1	50	50	50	9	9	8	46	46	50
	J3044_2	54	54	54	10	10	10	37	37	37
	J3044_3	51	51	51	14	14	13	98	98	92
	J3044_4	57	57	57	8	8	8	34	34	33
	J3044_5	55	55	55	11	11	10	45	45	54
	J3044_6	56	56	56	11	11	8	76	76	73
	J3044_7	42	42	42	9	9	7	27	27	21
	J3044_8	49	49	49	5	5	5	29	29	20
	J3044_9	64	64	64	12	12	13	69	69	61
	J3044_10	63	63	63	10	10	10	70	70	97

l) instances having < NC=2.1, RF=4, RS∈{0.2, 0.5, 0.7, 1} >

RS	e	T			$\#NC$			$\Sigma slack$		
		Opt	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**	**Opt**	**SSS**	**MSP**
0.2	J3045_1	82	90	95	5	3	7	10	22	31
	J3045_2	125	129	143	8	9	8	40	61	41
	J3045_3	92	98	106	7	7	6	27	30	31
	J3045_4	84	93	95	6	8	6	9	33	34
	J3045_5	86	92	93	7	8	8	22	37	31
	J3045_6	129	140	152	4	6	6	15	20	23
	J3045_7	101	109	115	3	6	5	19	27	27
	J3045_8	94	100	104	3	7	9	17	36	52
	J3045_9	82	92	103	6	6	8	17	21	38
	J3045_10	90	97	106	6	5	3	22	16	9
0.5	J3046_1	59	66	67	4	7	7	21	23	16
	J3046_2	67	71	72	4	6	7	5	15	16
	J3046_3	65	70	70	4	5	6	7	10	10
	J3046_4	64	67	70	6	6	8	13	12	19
	J3046_5	57	61	62	3	5	8	7	8	16
	J3046_6	59	63	65	4	5	8	11	14	27
	J3046_7	59	63	63	4	3	3	8	7	3
	J3046_8	58	59	66	4	5	7	5	7	21
	J3046_9	49	53	54	6	6	9	16	23	33
	J3046_10	55	55	58	6	8	6	24	26	29
0.7	J3047_1	58	59	59	9	9	7	22	25	19
	J3047_2	59	59	59	8	8	9	44	60	52
	J3047_3	55	55	55	8	7	3	16	15	9
	J3047_4	49	49	53	6	5	8	9	8	30
	J3047_5	47	48	49	5	7	4	18	22	9
	J3047_6	53	56	56	3	5	5	4	10	13
	J3047_7	66	66	69	7	7	10	40	37	50
	J3047_8	48	48	48	8	7	7	20	20	25
	J3047_9	65	65	65	7	8	7	53	49	65
	J3047_10	60	61	62	3	4	6	12	9	16
1	J3048_1	63	63	63	10	10	8	99	99	96
	J3048_2	54	54	54	11	11	11	71	71	70
	J3048_3	50	50	50	10	10	10	67	67	62
	J3048_4	57	57	57	12	12	12	55	55	55
	J3048_5	58	58	58	12	12	10	47	47	60
	J3048_6	58	58	58	11	11	10	76	76	57
	J3048_7	55	55	55	10	10	11	45	45	55
	J3048_8	44	44	44	8	8	7	30	30	28
	J3048_9	59	59	59	10	10	10	71	71	71
	J3048_10	54	54	54	10	10	12	75	75	95

Table 25: Average SIF for J30 instances with same <NC,RF,RS>

NC	RF	RS	Opt				SSS				MSP			
			r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4
1.5	1	0.2	47.9	60.1	77.8	74.8	54.8	64.2	120.8	66.8	92.1	82.3	96.8	66.2
		0.5	92.5	46.7	96.9	75.7	68.1	47.8	101.6	57.7	86.2	52.4	67.2	82.0
		0.7	63.2	95.5	99.0	217.3	82.9	83.4	82.4	215.1	85.9	75.6	116.3	275.6
		1	117.4	111.5	104.1	151.3	117.4	111.5	104.1	151.3	136.4	123.0	82.4	158.6
	2	0.2	75.3	30.4	58.4	61.9	85.9	69.7	88.1	91.8	128.8	72.1	117.7	125.2
		0.5	103.5	87.8	70.0	70.0	119.0	116.1	97.1	92.3	111.0	105.1	118.0	116.6
		0.7	119.9	113.0	117.2	76.1	110.9	105.2	107.8	53.9	186.1	166.3	151.4	152.9
		1	159.1	223.2	191.5	250.6	159.1	223.2	191.5	250.6	179.4	216.9	200.8	217.4
	3	0.2	56.1	60.8	61.5	47.6	92.7	81.2	95.9	80.5	138.8	122.2	137.0	136.8
		0.5	80.5	89.2	77.5	86.8	110.6	113.1	120.8	92.1	176.2	244.7	232.4	244.4
		0.7	184.9	203.9	170.1	187.8	199.1	206.1	161.6	191.0	219.9	208.2	209.8	227.6
		1	299.7	292.9	229.2	220.4	299.7	292.9	229.2	220.4	274.3	272.9	220.7	222.1
	4	0.2	38.6	46.2	47.5	32.8	65.5	91.9	90.6	74.0	103.4	111.3	124.4	108.3
		0.5	60.2	45.3	55.3	61.9	71.5	70.9	83.1	92.0	88.9	103.1	90.8	97.7
		0.7	182.6	194.6	201.7	192.4	201.3	225.1	220.0	207.8	187.9	204.1	220.6	203.3
		1	361.5	398.6	376.9	373.6	361.5	398.6	376.9	373.6	358.0	394.3	370.7	373.7
1.8	1	0.2	91.9	82.0	68.1	42.8	83.5	68.9	89.1	41.5	89.4	65.3	85.5	47.8
		0.5	46.5	65.7	76.8	93.7	54.9	57.5	77.3	76.8	95.8	76.9	76.6	129.9
		0.7	81.1	89.0	44.8	93.9	93.5	90.1	47.0	97.5	81.9	119.5	62.4	85.7
		1	119.3	102.3	129.7	81.4	119.3	102.3	129.7	81.4	132.9	85.8	136.2	55.7
	2	0.2	60.1	74.9	61.4	49.4	76.6	101.3	79.3	72.3	128.1	146.2	77.9	118.7
		0.5	82.6	103.4	110.2	137.1	92.6	80.7	139.3	151.0	142.8	139.4	184.0	150.0
		0.7	163.9	189.6	151.3	115.9	166.8	161.2	149.8	119.5	236.6	209.3	157.0	174.6
		1	180.6	177.5	184.3	214.5	180.6	177.5	184.3	214.5	176.3	176.2	160.3	209.1
	3	0.2	31.9	62.2	42.0	36.0	71.0	96.6	88.4	98.5	79.5	151.0	85.8	115.5
		0.5	101.4	87.2	86.6	107.7	63.7	67.3	76.4	86.3	183.3	159.6	193.1	172.6
		0.7	182.8	198.1	203.8	151.9	196.7	188.1	194.5	155.0	263.2	238.3	243.5	175.8
		1	298.6	316.5	263.6	374.3	298.6	316.5	263.6	374.3	280.4	310.9	258.7	373.7
	4	0.2	34.2	28.6	41.6	32.5	103.1	99.3	117.3	84.9	121.0	113.0	133.1	121.1
		0.5	35.3	35.9	29.6	34.1	69.1	72.0	75.6	62.9	93.8	108.8	93.6	92.0
		0.7	199.8	205.5	174.7	197.6	204.6	207.7	185.9	201.2	215.7	224.8	213.4	220.3
		1	347.4	324.7	341.7	285.3	347.4	324.7	341.7	285.3	374.8	354.4	391.5	329.0
2.1	1	0.2	52.1	94.0	73.6	80.7	55.7	103.1	82.3	73.6	60.7	64.8	57.5	62.5
		0.5	95.2	84.3	67.2	77.6	103.3	76.9	62.4	71.6	111.8	86.6	50.0	97.9
		0.7	98.1	100.0	66.4	82.1	96.0	110.1	64.1	82.0	99.0	115.1	68.9	103.1
		1	101.5	86.3	133.3	161.9	101.5	86.3	133.3	161.9	94.4	59.8	159.2	137.6
	2	0.2	83.4	51.3	54.6	64.0	124.9	77.1	89.3	65.6	87.7	77.5	108.7	90.0
		0.5	84.5	80.3	124.6	71.7	101.2	84.3	138.1	74.7	122.3	86.9	206.2	111.4
		0.7	168.5	128.5	94.4	137.9	175.6	133.6	86.8	133.2	219.8	182.5	103.1	167.6
		1	227.5	115.5	183.9	172.9	227.5	115.5	183.9	172.9	222.8	114.2	188.2	179.9
	3	0.2	73.6	55.9	75.8	72.1	138.6	91.1	110.2	114.5	114.0	99.6	114.7	107.1
		0.5	120.5	82.0	116.9	98.0	162.9	112.2	127.5	122.6	200.8	124.6	191.5	149.1
		0.7	82.6	115.8	85.2	97.3	91.4	125.2	95.5	109.4	127.3	170.8	174.3	158.8
		1	227.0	210.6	247.8	199.6	227.0	210.6	247.8	199.6	208.3	204.1	239.2	174.7
	4	0.2	84.7	90.5	82.8	78.6	126.3	148.8	128.9	127.3	122.7	140.4	129.6	110.9
		0.5	57.8	56.0	56.8	48.2	67.8	76.0	73.6	62.2	92.7	99.7	94.8	87.6
		0.7	127.7	133.6	113.6	117.8	153.4	143.2	126.9	136.7	161.3	159.6	132.3	137.9
		1	337.4	331.0	346.1	332.7	337.4	331.0	346.1	332.7	361.3	324.4	348.0	359.0

Table 26: Average relative SIF for J30 instances with same <NC,RF,RS>

NC	RF	RS	**Opt**				**SSS**				**MSP**			
			r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4
1.5	1	0.2	39%	27%	53%	41%	34%	30%	83%	30%	61%	43%	64%	32%
		0.5	46%	36%	44%	28%	35%	39%	44%	22%	42%	56%	35%	30%
		0.7	33%	54%	53%	80%	42%	47%	47%	79%	51%	43%	65%	103%
		1	44%	59%	38%	74%	44%	59%	38%	74%	51%	65%	32%	77%
	2	0.2	17%	7%	12%	14%	18%	16%	17%	18%	28%	18%	24%	25%
		0.5	19%	16%	15%	17%	21%	23%	22%	22%	18%	22%	26%	26%
		0.7	30%	23%	26%	16%	27%	21%	24%	12%	49%	34%	33%	33%
		1	34%	50%	46%	57%	34%	50%	46%	57%	38%	49%	49%	49%
	3	0.2	9%	8%	9%	8%	15%	11%	15%	12%	23%	17%	22%	22%
		0.5	12%	13%	11%	11%	16%	17%	17%	12%	25%	35%	34%	31%
		0.7	26%	28%	25%	27%	28%	29%	23%	27%	31%	29%	30%	31%
		1	41%	37%	28%	30%	41%	37%	28%	30%	38%	34%	27%	31%
	4	0.2	5%	6%	5%	4%	8%	11%	10%	8%	12%	13%	14%	12%
		0.5	7%	5%	7%	7%	8%	8%	11%	10%	10%	12%	12%	11%
		0.7	21%	22%	22%	21%	23%	25%	24%	23%	22%	23%	25%	22%
		1	45%	47%	46%	46%	45%	47%	46%	46%	44%	47%	45%	46%
1.8	1	0.2	56%	35%	32%	25%	50%	30%	40%	24%	61%	30%	37%	31%
		0.5	27%	32%	47%	49%	30%	27%	45%	40%	48%	38%	46%	67%
		0.7	40%	52%	28%	40%	45%	51%	26%	42%	40%	67%	47%	42%
		1	58%	39%	63%	39%	58%	39%	63%	39%	65%	33%	70%	28%
	2	0.2	15%	17%	13%	13%	18%	23%	16%	18%	29%	33%	17%	28%
		0.5	23%	22%	26%	30%	25%	17%	30%	31%	37%	27%	40%	29%
		0.7	37%	45%	33%	25%	37%	39%	32%	25%	54%	58%	33%	33%
		1	40%	36%	47%	44%	40%	36%	47%	44%	39%	36%	41%	42%
	3	0.2	6%	9%	7%	6%	11%	14%	15%	16%	13%	24%	14%	19%
		0.5	16%	13%	13%	16%	10%	10%	11%	13%	29%	24%	34%	27%
		0.7	25%	26%	31%	24%	27%	25%	30%	24%	37%	32%	38%	28%
		1	38%	44%	37%	55%	38%	44%	37%	55%	36%	43%	37%	55%
	4	0.2	4%	3%	4%	3%	11%	10%	12%	9%	13%	12%	14%	13%
		0.5	4%	4%	3%	4%	7%	7%	8%	6%	10%	11%	10%	10%
		0.7	21%	23%	19%	23%	21%	23%	21%	24%	23%	25%	24%	26%
		1	38%	35%	35%	30%	38%	35%	35%	30%	41%	39%	40%	35%
2.1	1	0.2	25%	54%	37%	54%	27%	63%	42%	51%	30%	35%	29%	33%
		0.5	42%	41%	37%	30%	46%	39%	35%	28%	53%	42%	27%	37%
		0.7	40%	47%	30%	35%	39%	51%	28%	34%	41%	55%	34%	44%
		1	43%	58%	66%	54%	43%	58%	66%	54%	41%	40%	81%	48%
	2	0.2	20%	11%	13%	16%	30%	16%	23%	15%	20%	15%	26%	20%
		0.5	19%	17%	28%	13%	22%	18%	31%	14%	28%	19%	45%	21%
		0.7	32%	30%	24%	34%	33%	31%	22%	33%	40%	39%	23%	42%
		1	49%	24%	33%	40%	49%	24%	33%	40%	48%	24%	34%	44%
	3	0.2	11%	8%	10%	10%	20%	13%	15%	16%	16%	15%	15%	15%
		0.5	20%	12%	16%	14%	27%	16%	17%	18%	31%	18%	26%	22%
		0.7	13%	17%	13%	14%	15%	19%	14%	15%	20%	27%	25%	22%
		1	33%	28%	31%	29%	33%	28%	31%	29%	30%	27%	30%	25%
	4	0.2	10%	10%	9%	8%	14%	17%	14%	13%	14%	16%	14%	12%
		0.5	6%	6%	7%	6%	8%	8%	9%	7%	10%	10%	11%	10%
		0.7	14%	14%	12%	12%	17%	15%	13%	14%	18%	17%	14%	14%
		1	38%	35%	38%	36%	38%	35%	38%	36%	40%	34%	38%	39%

Table 27: SIF for each resource type for all J30 instances

a) instances having < NC=1.5, RF=1, RS∈{0.2, 0.5, 0.7, 1} >

RS	e	Opt				SSS				MSP			
		r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4
0.2	J301_1	65	7	46	0	113	28	64	12	137	28	70	14
	J301_2	34	128	19	55	32	138	21	60	120	132	21	11
	J301_3	0	50	78	149	0	35	78	199	45	37	118	216
	J301_4	80	104	38	102	45	102	223	105	45	102	212	100
	J301_5	0	34	35	138	0	49	57	43	63	8	35	78
	J301_6	32	36	93	19	35	51	199	25	142	165	20	32
	J301_7	48	117	215	117	48	70	215	117	108	133	215	117
	J301_8	103	8	91	35	54	8	88	3	113	48	124	3
	J301_9	5	107	80	98	86	131	170	89	5	140	65	83
	J301_10	112	10	83	35	135	30	93	15	143	30	88	8
0.5	J302_1	79	38	6	83	79	38	6	83	79	38	6	83
	J302_2	7	46	103	89	18	48	141	43	30	40	141	101
	J302_3	42	70	36	50	42	70	36	50	47	61	17	64
	J302_4	130	31	107	48	36	42	116	56	130	99	9	49
	J302_5	202	31	291	0	116	76	276	0	202	22	54	0
	J302_6	72	98	58	191	72	63	26	129	72	98	58	191
	J302_7	138	24	26	119	138	20	26	73	132	20	26	72
	J302_8	134	102	115	24	107	105	115	18	50	85	137	18
	J302_9	48	27	145	90	0	16	192	62	48	27	150	142
	J302_10	73	0	82	63	73	0	82	63	72	34	74	100
0.7	J303_1	114	209	74	159	0	224	65	159	114	209	134	299
	J303_2	88	19	18	114	88	12	13	104	145	19	18	168
	J303_3	79	131	16	336	99	120	16	294	79	136	21	328
	J303_4	56	98	78	116	56	98	78	116	0	108	90	100
	J303_5	0	6	41	370	132	38	41	412	62	38	41	412
	J303_6	66	130	46	307	72	138	40	307	90	118	38	334
	J303_7	58	26	222	53	58	26	222	53	110	10	332	117
	J303_8	0	11	257	125	0	13	210	120	0	65	229	155
	J303_9	159	320	77	325	222	160	112	309	197	48	128	530
	J303_10	12	5	161	268	102	5	27	277	62	5	132	313
1	J304_1	0	163	32	58	0	163	32	58	0	168	32	98
	J304_2	159	515	85	67	159	515	85	67	143	395	85	47
	J304_3	67	10	220	97	67	10	220	97	127	0	216	82
	J304_4	162	72	94	171	162	72	94	171	162	127	94	171
	J304_5	27	50	415	356	27	50	415	356	27	70	230	356
	J304_6	240	113	105	20	240	113	105	20	220	113	77	90
	J304_7	138	0	19	344	138	0	19	344	138	0	19	344
	J304_8	308	132	51	169	308	132	51	169	358	197	51	169
	J304_9	29	8	20	127	29	8	20	127	145	8	20	127
	J304_10	44	52	0	104	44	52	0	104	44	152	0	102

b) instances having < NC=1.5, RF=2, RS∈{0.2, 0.5, 0.7, 1} >

RS	e	Opt				SSS				MSP			
		r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4
0.2	J305_1	32	10	53	62	54	52	56	115	98	121	95	242
	J305_2	142	30	48	111	74	106	68	120	188	29	55	61
	J305_3	83	18	89	107	51	42	74	191	149	46	216	163
	J305_4	16	72	64	49	66	107	82	17	50	79	84	96
	J305_5	169	51	16	5	175	89	71	131	236	63	147	76
	J305_6	39	52	84	40	102	70	99	54	137	75	247	112
	J305_7	155	30	90	167	137	35	93	60	187	83	102	95
	J305_8	24	20	56	26	61	109	175	111	98	67	64	169
	J305_9	73	6	38	38	68	36	63	20	46	104	143	124
	J305_10	20	15	46	14	71	51	100	99	99	54	24	114
0.5	J306_1	114	9	79	56	146	74	173	48	234	68	188	158
	J306_2	6	116	52	74	27	122	100	142	52	54	108	97
	J306_3	121	48	165	31	136	70	119	36	57	13	182	141
	J306_4	28	24	96	69	70	22	152	98	105	52	52	128
	J306_5	416	359	24	64	306	308	64	50	153	205	272	107
	J306_6	31	76	40	47	24	100	92	42	27	111	150	126
	J306_7	30	114	61	45	50	124	70	64	62	128	93	25
	J306_8	55	83	67	90	55	78	65	113	68	69	19	64
	J306_9	40	45	103	109	159	88	133	191	102	109	116	163
	J306_10	194	4	13	115	217	175	3	139	250	242	0	157
0.7	J307_1	265	356	129	3	265	306	151	3	304	303	205	2
	J307_2	216	163	100	205	267	164	114	127	319	247	65	324
	J307_3	69	15	112	78	35	21	67	98	154	101	69	289
	J307_4	94	80	164	22	87	36	115	8	64	110	153	71
	J307_5	50	49	66	17	54	57	66	29	232	208	189	53
	J307_6	172	198	72	101	166	246	87	105	278	210	179	144
	J307_7	56	62	48	4	44	62	58	4	136	169	92	0
	J307_8	59	63	41	249	16	67	16	77	239	118	112	461
	J307_9	109	111	300	20	106	74	329	50	106	174	383	155
	J307_10	109	33	140	62	69	19	75	38	29	23	67	30
1	J308_1	328	257	113	180	328	257	113	180	343	324	180	236
	J308_2	98	449	88	244	98	449	88	244	258	289	268	284
	J308_3	62	154	60	370	62	154	60	370	122	109	60	275
	J308_4	70	320	231	411	70	320	231	411	70	350	231	313
	J308_5	195	218	197	100	195	218	197	100	195	218	197	100
	J308_6	85	180	45	20	85	180	45	20	85	180	45	20
	J308_7	66	96	140	264	66	96	140	264	66	96	140	264
	J308_8	352	120	310	297	352	120	310	297	320	165	342	296
	J308_9	92	103	194	168	92	103	194	168	92	103	194	168
	J308_10	243	335	537	452	243	335	537	452	243	335	351	218

c) instances having < NC=1.5, RF=3, RS∈{0.2, 0.5, 0.7, 1} >

RS	e	Opt				SSS				MSP			
		r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4
0.2	J309_1	68	116	112	18	196	38	197	39	118	105	209	104
	J309_2	178	71	113	46	168	36	32	54	65	47	54	75
	J309_3	8	39	28	38	88	114	134	100	191	153	98	118
	J309_4	41	20	59	37	97	86	35	41	192	188	83	101
	J309_5	99	65	65	123	206	123	251	169	288	164	347	262
	J309_6	99	59	63	61	52	37	6	75	196	129	113	234
	J309_7	8	71	33	23	51	97	127	48	94	134	95	71
	J309_8	28	54	47	27	7	60	31	26	140	168	115	76
	J309_9	4	40	12	24	45	118	47	78	59	52	49	80
	J309_10	28	73	83	79	17	103	99	175	45	82	207	247
0.5	J3010_1	115	56	89	66	98	54	86	49	228	302	439	428
	J3010_2	119	185	100	41	103	173	253	87	170	268	152	141
	J3010_3	1	148	64	8	97	182	157	81	177	308	251	211
	J3010_4	83	87	132	188	27	68	36	40	115	352	450	474
	J3010_5	20	45	45	32	57	71	74	35	89	153	126	220
	J3010_6	17	36	27	23	162	98	62	62	219	321	160	115
	J3010_7	101	74	26	81	59	86	50	134	243	177	97	180
	J3010_8	104	72	50	28	170	123	124	40	186	175	100	193
	J3010_9	187	161	232	270	246	250	350	272	297	340	452	347
	J3010_10	58	28	10	131	87	26	16	121	38	51	97	135
0.7	J3011_1	96	183	10	90	67	132	9	34	136	181	28	115
	J3011_2	39	45	182	50	137	164	248	115	154	74	202	131
	J3011_3	823	608	392	587	812	620	385	645	748	525	504	752
	J3011_4	79	75	156	89	99	137	160	86	162	70	213	108
	J3011_5	41	90	278	48	50	93	300	66	53	142	240	87
	J3011_6	92	101	72	175	37	79	54	162	47	137	139	190
	J3011_7	20	119	165	101	41	132	120	144	178	256	291	224
	J3011_8	157	169	103	91	194	61	53	118	230	101	142	147
	J3011_9	485	503	233	454	508	474	230	431	408	404	218	337
	J3011_10	17	146	110	193	46	169	57	109	83	192	121	185
1	J3012_1	215	291	111	80	215	291	111	80	223	371	127	112
	J3012_2	117	100	137	225	117	100	137	225	117	180	185	285
	J3012_3	209	155	77	98	209	155	77	98	233	153	78	91
	J3012_4	446	408	560	561	446	408	560	561	496	413	570	561
	J3012_5	446	360	392	350	446	360	392	350	358	202	271	258
	J3012_6	233	350	204	44	233	350	204	44	249	321	186	104
	J3012_7	478	487	306	218	478	487	306	218	478	487	306	218
	J3012_8	174	263	160	152	174	263	160	152	120	224	151	104
	J3012_9	174	159	123	128	174	159	123	128	54	22	101	70
	J3012_10	505	356	222	348	505	356	222	348	415	356	232	418

d) instances having < NC=1.5, RF=4, RS∈{0.2, 0.5, 0.7, 1} >

RS	e	Opt				SSS				MSP			
		r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4
0.2	J3013_1	37	34	23	22	38	41	48	30	48	44	48	39
	J3013_2	12	6	9	15	60	128	78	127	47	68	54	72
	J3013_3	59	48	60	61	90	65	95	89	130	83	62	86
	J3013_4	31	26	29	15	63	63	69	68	97	120	98	142
	J3013_5	5	9	9	2	23	60	42	43	73	137	103	103
	J3013_6	9	26	7	18	19	46	19	35	95	84	78	71
	J3013_7	45	36	54	18	33	75	35	41	136	93	162	117
	J3013_8	106	93	144	103	157	138	280	152	183	164	269	175
	J3013_9	34	43	95	28	59	106	148	105	91	120	158	85
	J3013_10	48	141	45	46	113	197	92	50	134	200	212	193
0.5	J3014_1	91	36	101	80	121	97	165	209	97	68	73	112
	J3014_2	58	29	27	36	35	21	48	28	48	83	65	54
	J3014_3	12	31	22	43	31	65	95	51	31	88	74	78
	J3014_4	47	52	36	39	47	75	69	76	72	82	87	91
	J3014_5	37	79	118	84	64	115	143	122	196	211	236	127
	J3014_6	26	23	28	28	74	87	70	85	119	142	97	151
	J3014_7	19	32	32	45	9	16	10	10	46	67	28	28
	J3014_8	110	63	108	141	137	102	136	220	38	59	66	74
	J3014_9	31	19	25	34	41	58	36	43	108	141	60	101
	J3014_10	171	89	56	89	156	73	59	76	134	90	122	161
0.7	J3015_1	113	45	127	117	217	157	234	208	172	110	216	218
	J3015_2	172	231	209	215	290	335	322	272	196	248	227	233
	J3015_3	127	107	135	138	88	77	102	84	127	102	138	136
	J3015_4	306	445	532	238	277	487	562	274	322	450	556	254
	J3015_5	92	86	51	50	96	116	70	59	50	122	87	77
	J3015_6	311	266	286	338	305	260	230	312	248	218	259	304
	J3015_7	161	84	116	128	155	158	120	130	161	87	172	90
	J3015_8	131	129	190	144	96	92	121	110	89	72	81	88
	J3015_9	115	151	92	218	150	146	131	243	138	139	134	246
	J3015_10	298	402	279	338	339	423	308	386	376	493	336	387
1	J3016_1	147	163	133	125	147	163	133	125	183	182	152	173
	J3016_2	269	368	496	571	269	368	496	571	279	374	510	587
	J3016_3	244	217	271	210	244	217	271	210	196	261	275	246
	J3016_4	442	557	386	390	442	557	386	390	306	429	333	333
	J3016_5	435	463	378	391	435	463	378	391	435	463	378	391
	J3016_6	328	648	374	506	328	648	374	506	564	846	447	650
	J3016_7	244	305	208	275	244	305	208	275	220	253	222	242
	J3016_8	465	291	391	366	465	291	391	366	445	291	385	363
	J3016_9	493	516	537	596	493	516	537	596	416	396	420	454
	J3016_10	548	458	595	306	548	458	595	306	536	448	585	298

e) instances having < NC=1.8, RF=1, RS∈{0.2, 0.5, 0.7, 1} >

RS	e	Opt				SSS				MSP			
		r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4
0.2	J3017_1	158	106	55	0	119	38	189	0	209	16	71	0
	J3017_2	17	92	180	82	47	43	172	80	77	92	180	82
	J3017_3	0	163	35	152	0	183	35	152	0	83	0	140
	J3017_4	81	0	0	23	75	0	0	29	96	14	0	35
	J3017_5	60	10	48	26	41	6	104	12	33	24	168	52
	J3017_6	161	184	66	42	161	174	66	42	161	154	114	42
	J3017_7	127	60	83	40	127	60	83	40	127	20	83	40
	J3017_8	120	36	155	15	62	36	155	14	54	45	168	0
	J3017_9	131	98	30	48	139	82	58	46	73	118	42	87
	J3017_10	64	71	29	0	64	67	29	0	64	87	29	0
0.5	J3018_1	8	54	177	169	32	24	162	106	188	45	195	173
	J3018_2	42	66	44	114	60	56	64	114	42	66	44	114
	J3018_3	11	77	20	74	11	77	20	74	18	97	40	74
	J3018_4	7	48	50	133	7	48	55	131	99	18	25	269
	J3018_5	24	24	44	131	30	24	79	131	30	24	79	131
	J3018_6	27	75	272	16	27	90	207	16	27	135	116	16
	J3018_7	16	28	39	114	0	28	19	72	40	28	71	135
	J3018_8	191	93	58	41	197	29	52	5	371	93	58	41
	J3018_9	46	101	21	78	100	104	30	36	88	150	30	78
	J3018_10	93	91	43	67	85	95	85	83	55	113	108	268
0.7	J3019_1	97	81	27	66	128	102	14	86	69	82	52	186
	J3019_2	76	75	94	157	70	75	86	157	74	75	110	101
	J3019_3	88	187	43	62	88	187	73	62	88	187	43	62
	J3019_4	86	12	84	103	89	0	88	105	92	12	84	53
	J3019_5	120	2	23	79	120	2	74	40	120	2	34	40
	J3019_6	101	99	0	143	101	99	0	132	89	99	0	143
	J3019_7	48	68	66	34	54	110	31	53	30	203	87	23
	J3019_8	155	88	24	117	245	48	17	162	155	88	124	117
	J3019_9	40	53	87	32	40	53	87	32	75	60	81	56
	J3019_10	0	225	0	146	0	225	0	146	27	387	9	76
1	J3020_1	15	184	240	72	15	184	240	72	7	184	240	72
	J3020_2	151	18	105	181	151	18	105	181	151	18	155	100
	J3020_3	6	155	126	110	6	155	126	110	6	155	126	110
	J3020_4	120	21	66	137	120	21	66	137	124	21	0	41
	J3020_5	89	89	0	6	89	89	0	6	89	88	0	6
	J3020_6	164	90	72	16	164	90	72	16	164	90	72	16
	J3020_7	189	174	12	15	189	174	12	15	135	54	12	15
	J3020_8	377	92	249	91	377	92	249	91	371	4	239	91
	J3020_9	82	0	333	128	82	0	333	128	282	10	330	48
	J3020_10	0	200	94	58	0	200	94	58	0	234	188	58

f) instances having < NC=1.8, RF=2, RS∈{0.2, 0.5, 0.7, 1} >

RS	e	**Opt**				**SSS**				**MSP**			
		r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4
0.2	J3021_1	48	88	45	22	104	135	66	6	121	121	54	0
	J3021_2	18	84	52	42	55	73	73	57	168	143	16	139
	J3021_3	32	182	50	74	70	203	93	104	75	179	64	107
	J3021_4	62	45	72	118	87	116	75	123	64	217	88	70
	J3021_5	17	90	47	14	12	68	0	0	107	169	46	66
	J3021_6	193	119	162	3	169	113	144	67	8	79	169	73
	J3021_7	77	15	22	104	75	132	74	187	281	249	2	323
	J3021_8	23	28	14	29	60	38	14	81	86	93	31	110
	J3021_9	84	52	94	27	62	35	183	55	95	85	172	75
	J3021_10	47	46	56	61	72	100	71	43	276	127	137	224
0.5	J3022_1	52	75	18	38	76	101	15	66	180	16	78	89
	J3022_2	34	56	143	33	34	56	143	33	73	289	238	72
	J3022_3	98	131	163	145	79	104	138	125	110	160	175	14
	J3022_4	26	61	63	193	67	81	76	164	3	25	21	8
	J3022_5	110	54	139	128	184	0	151	96	245	27	169	57
	J3022_6	42	66	93	74	47	64	101	71	225	225	298	119
	J3022_7	93	127	144	72	24	101	39	43	62	115	13	34
	J3022_8	110	0	62	110	113	32	59	163	185	120	68	443
	J3022_9	253	413	242	472	266	208	397	485	203	248	349	409
	J3022_10	8	51	35	106	36	60	274	264	142	169	431	255
0.7	J3023_1	149	281	158	273	111	230	177	156	51	332	134	124
	J3023_2	223	162	84	30	223	240	88	65	223	138	90	16
	J3023_3	142	255	98	94	122	184	105	84	20	106	82	20
	J3023_4	117	266	163	66	117	266	163	66	248	362	160	150
	J3023_5	97	205	24	131	219	189	14	244	198	371	14	272
	J3023_6	102	109	92	39	77	103	95	39	170	157	198	91
	J3023_7	290	48	197	325	270	59	122	206	333	132	141	669
	J3023_8	58	65	107	84	52	88	80	96	312	256	146	94
	J3023_9	198	450	81	51	214	198	145	173	406	102	117	245
	J3023_10	263	55	509	66	263	55	509	66	405	137	488	65
1	J3024_1	74	187	165	169	74	187	165	169	74	187	167	153
	J3024_2	207	81	391	154	207	81	391	154	156	63	361	130
	J3024_3	370	270	410	321	370	270	410	321	370	270	240	321
	J3024_4	327	475	235	514	327	475	235	514	327	475	235	514
	J3024_5	53	180	67	244	53	180	67	244	53	180	67	244
	J3024_6	267	273	96	235	267	273	96	235	279	273	96	244
	J3024_7	309	47	127	286	309	47	127	286	309	47	127	286
	J3024_8	59	53	108	61	59	53	108	61	55	25	63	26
	J3024_9	10	62	97	65	10	62	97	65	10	90	97	77
	J3024_10	130	147	147	96	130	147	147	96	130	152	150	96

g) instances having < NC=1.8, RF=3, RS∈{0.2, 0.5, 0.7, 1} >

RS	***e***	**Opt**				**SSS**				**MSP**			
		r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4
0.2	J3025_1	25	20	41	12	56	66	110	76	17	45	15	9
	J3025_2	94	29	7	28	33	46	28	54	119	37	37	26
	J3025_3	52	47	56	14	110	69	182	139	42	41	194	140
	J3025_4	39	39	18	17	74	123	137	56	82	237	56	36
	J3025_5	6	24	5	17	147	66	48	183	195	174	83	193
	J3025_6	7	2	9	7	22	51	29	58	94	88	56	16
	J3025_7	7	172	99	59	33	273	109	121	28	370	78	147
	J3025_8	40	126	99	29	89	112	107	30	58	85	90	35
	J3025_9	16	88	61	147	102	45	43	193	61	124	44	311
	J3025_10	33	75	25	30	44	115	91	75	99	309	205	242
0.5	J3026_1	31	49	49	17	31	49	49	17	226	121	276	278
	J3026_2	37	91	102	112	100	90	109	130	194	314	302	280
	J3026_3	43	186	73	116	41	177	82	119	146	177	65	20
	J3026_4	143	43	109	165	71	32	113	106	207	160	143	207
	J3026_5	241	119	169	111	123	103	131	133	149	76	85	71
	J3026_6	36	45	47	72	51	86	97	143	74	119	114	145
	J3026_7	71	19	21	76	71	19	21	76	213	136	211	240
	J3026_8	295	268	220	353	95	72	87	73	334	161	372	298
	J3026_9	1	13	0	2	0	0	0	0	163	173	224	121
	J3026_10	116	39	76	53	54	45	75	66	127	159	139	66
0.7	J3027_1	43	27	74	51	54	18	33	27	108	41	95	106
	J3027_2	268	277	106	206	292	308	91	164	130	140	52	114
	J3027_3	507	252	333	223	514	239	340	233	673	398	377	330
	J3027_4	122	262	166	131	122	262	166	131	253	316	228	173
	J3027_5	107	101	102	116	96	85	88	111	257	298	446	259
	J3027_6	81	82	103	68	81	82	103	68	69	70	169	88
	J3027_7	83	89	111	31	67	84	127	58	133	194	169	81
	J3027_8	247	188	302	328	247	192	312	321	430	270	183	220
	J3027_9	190	236	248	121	208	125	170	128	231	175	118	139
	J3027_10	180	467	493	244	286	486	515	309	348	481	598	248
1	J3028_1	384	314	247	267	384	314	247	267	384	314	247	267
	J3028_2	667	445	329	670	667	445	329	670	550	367	329	670
	J3028_3	105	33	165	142	105	33	165	142	95	36	158	122
	J3028_4	145	166	314	338	145	166	314	338	145	166	314	338
	J3028_5	321	509	347	554	321	509	347	554	321	525	357	558
	J3028_6	204	229	162	234	204	229	162	234	140	195	128	208
	J3028_7	174	218	65	113	174	218	65	113	174	253	100	169
	J3028_8	398	430	420	441	398	430	420	441	398	430	420	441
	J3028_9	208	454	326	635	208	454	326	635	217	456	273	615
	J3028_10	380	367	261	349	380	367	261	349	380	367	261	349

h) instances having < NC=1.8, RF=4, RS∈{0.2, 0.5, 0.7, 1} >

RS	e	Opt				SSS				MSP			
		r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4
0.2	J3029_1	0	0	0	0	83	50	53	75	154	192	171	135
	J3029_2	56	42	44	55	78	112	197	124	144	161	129	163
	J3029_3	16	8	2	5	101	120	78	83	103	165	85	69
	J3029_4	79	51	132	57	135	117	195	131	141	75	209	133
	J3029_5	31	28	27	46	189	85	173	74	202	55	122	148
	J3029_6	19	39	43	23	59	110	85	44	47	82	47	33
	J3029_7	9	9	39	37	21	11	16	17	70	72	180	141
	J3029_8	25	29	38	20	135	78	117	90	48	46	38	58
	J3029_9	20	23	34	31	117	214	155	145	107	173	201	217
	J3029_10	87	57	57	51	113	96	104	66	194	109	149	114
0.5	J3030_1	49	41	49	18	94	50	58	31	58	93	49	72
	J3030_2	19	31	25	27	171	156	220	95	108	126	72	82
	J3030_3	87	55	48	64	63	48	38	43	193	128	130	120
	J3030_4	16	20	13	17	58	66	87	57	100	155	167	133
	J3030_5	26	31	32	27	30	82	80	44	33	120	105	42
	J3030_6	28	29	30	35	64	53	49	116	139	114	84	132
	J3030_7	61	53	53	84	48	82	92	109	93	122	131	172
	J3030_8	5	4	6	3	21	10	16	19	96	62	75	54
	J3030_9	22	25	11	24	65	64	52	41	50	68	54	40
	J3030_10	40	70	29	42	77	109	64	74	68	100	69	73
0.7	J3031_1	262	141	178	185	233	175	199	204	273	183	222	206
	J3031_2	658	689	467	610	755	735	504	661	583	632	428	556
	J3031_3	221	349	369	385	151	271	316	325	185	322	344	394
	J3031_4	311	239	104	284	236	176	68	240	258	223	92	291
	J3031_5	33	17	12	27	58	30	31	48	115	145	102	93
	J3031_6	293	222	157	187	353	239	230	218	314	201	215	187
	J3031_7	99	205	237	140	99	205	237	140	85	187	217	109
	J3031_8	64	88	130	98	81	109	149	113	125	118	167	125
	J3031_9	23	65	32	41	35	95	71	40	63	108	131	114
	J3031_10	34	40	61	19	45	42	54	23	156	129	216	128
1	J3032_1	278	341	310	266	278	341	310	266	295	344	319	256
	J3032_2	421	279	578	272	421	279	578	272	461	429	818	482
	J3032_3	193	292	119	182	193	292	119	182	253	332	129	222
	J3032_4	556	382	425	481	556	382	425	481	553	367	413	451
	J3032_5	281	268	271	147	281	268	271	147	297	241	329	250
	J3032_6	236	247	293	232	236	247	293	232	196	217	253	182
	J3032_7	294	188	187	146	294	188	187	146	257	166	155	129
	J3032_8	370	412	430	432	370	412	430	432	532	564	548	582
	J3032_9	563	555	561	356	563	555	561	356	626	657	732	451
	J3032_10	282	283	243	339	282	283	243	339	278	227	219	285

i) instances having < NC=2.1, RF=1, RS∈{0.2, 0.5, 0.7, 1} >

RS	e	Opt				SSS				MSP			
		r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4
0.2	J3033_1	83	21	2	19	83	21	2	0	88	0	22	19
	J3033_2	42	43	116	62	26	43	129	50	20	11	128	62
	J3033_3	13	92	10	111	10	92	15	94	122	96	34	67
	J3033_4	50	153	270	88	50	153	270	88	35	22	99	24
	J3033_5	33	185	28	180	33	185	28	180	12	150	8	70
	J3033_6	121	206	28	75	121	241	28	90	121	226	117	39
	J3033_7	36	166	24	28	36	166	24	28	54	124	0	0
	J3033_8	70	31	89	52	56	82	138	72	73	0	128	43
	J3033_9	51	11	46	89	126	23	50	24	60	1	15	206
	J3033_10	22	32	123	103	16	25	139	110	22	18	24	95
0.5	J3034_1	16	99	135	278	16	27	93	242	42	87	201	290
	J3034_2	0	30	21	57	0	30	21	57	0	30	21	42
	J3034_3	36	128	147	87	36	152	147	87	12	92	33	87
	J3034_4	152	151	0	30	152	151	0	30	152	121	0	174
	J3034_5	176	185	96	41	176	129	96	41	176	185	96	51
	J3034_6	10	48	30	144	10	48	30	144	18	128	31	146
	J3034_7	113	83	0	43	197	113	0	19	189	48	0	43
	J3034_8	275	60	161	10	275	60	161	10	264	60	8	60
	J3034_9	110	35	67	86	117	35	67	86	110	35	67	86
	J3034_10	64	24	15	0	54	24	9	0	155	80	43	0
0.7	J3035_1	10	129	164	35	19	185	164	0	10	129	164	35
	J3035_2	82	234	21	123	82	234	21	123	82	234	21	123
	J3035_3	77	12	194	162	77	12	194	162	75	12	134	152
	J3035_4	72	106	36	58	72	106	36	58	72	113	48	58
	J3035_5	144	90	29	0	164	140	9	0	140	72	49	8
	J3035_6	91	51	33	103	91	51	30	122	91	51	33	103
	J3035_7	68	56	52	54	68	56	52	54	61	56	112	336
	J3035_8	286	71	2	254	286	71	2	254	242	101	2	184
	J3035_9	56	60	53	0	56	60	53	0	72	192	46	0
	J3035_10	95	191	80	32	45	186	80	47	145	191	80	32
1	J3036_1	110	139	140	113	110	139	140	113	48	89	108	35
	J3036_2	38	39	163	104	38	39	163	104	0	24	327	98
	J3036_3	177	230	58	173	177	230	58	173	147	70	108	158
	J3036_4	102	20	314	178	102	20	314	178	102	20	314	178
	J3036_5	60	45	41	223	60	45	41	223	60	45	41	109
	J3036_6	34	106	46	72	34	106	46	72	102	26	38	36
	J3036_7	59	194	5	74	59	194	5	74	71	194	90	58
	J3036_8	252	12	110	342	252	12	110	342	231	52	110	354
	J3036_9	150	0	298	249	150	0	298	249	150	0	298	249
	J3036_10	33	78	158	91	33	78	158	91	33	78	158	101

j) instances having < NC=2.1, RF=2, RS∈{0.2, 0.5, 0.7, 1} >

RS	e	**Opt**				**SSS**				**MSP**			
		r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4
0.2	J3037_1	11	88	36	52	4	71	20	86	134	207	92	143
	J3037_2	47	45	43	99	56	62	49	108	83	109	138	158
	J3037_3	171	48	36	131	302	58	70	30	118	31	86	57
	J3037_4	26	56	47	30	127	156	78	61	49	79	21	92
	J3037_5	46	28	36	30	52	31	70	50	81	77	112	114
	J3037_6	103	0	58	64	81	0	47	20	123	16	76	41
	J3037_7	40	70	23	40	97	158	281	57	8	72	171	58
	J3037_8	53	77	105	30	105	106	57	52	44	116	177	103
	J3037_9	60	48	15	84	78	48	9	117	7	18	57	88
	J3037_10	277	53	147	80	347	81	212	75	230	50	157	46
0.5	J3038_1	24	35	104	45	16	58	100	15	70	76	270	97
	J3038_2	10	54	148	89	19	51	180	44	26	39	118	78
	J3038_3	63	73	34	169	51	81	40	273	59	112	64	187
	J3038_4	211	123	152	25	194	94	155	13	213	211	409	105
	J3038_5	19	108	146	84	41	93	98	36	55	97	230	127
	J3038_6	100	58	81	87	149	64	106	134	179	55	167	121
	J3038_7	123	46	231	68	158	58	267	78	275	52	312	228
	J3038_8	18	77	34	7	109	108	123	11	95	42	106	7
	J3038_9	200	132	144	99	200	132	144	99	168	150	204	102
	J3038_10	77	97	172	44	75	104	168	44	83	35	182	62
0.7	J3039_1	58	124	79	138	44	127	75	98	55	133	55	159
	J3039_2	229	138	53	131	229	138	53	131	274	215	115	84
	J3039_3	59	101	89	37	75	155	58	40	103	65	70	48
	J3039_4	23	92	27	76	49	46	95	57	26	192	171	137
	J3039_5	10	146	146	302	10	146	146	302	23	315	190	499
	J3039_6	317	41	196	58	401	52	184	65	602	127	241	107
	J3039_7	243	252	182	228	193	250	132	188	248	222	44	232
	J3039_8	486	216	86	77	436	145	30	92	495	211	35	121
	J3039_9	75	73	68	115	129	187	76	144	90	169	52	148
	J3039_10	185	102	18	217	190	90	19	215	282	176	58	141
1	J3040_1	75	206	160	142	75	206	160	142	75	206	160	142
	J3040_2	249	267	298	208	249	267	298	208	355	275	316	304
	J3040_3	272	136	86	312	272	136	86	312	246	98	81	215
	J3040_4	158	143	309	149	158	143	309	149	148	129	284	149
	J3040_5	374	122	206	107	374	122	206	107	390	122	206	107
	J3040_6	110	82	249	142	110	82	249	142	65	122	304	113
	J3040_7	148	55	150	121	148	55	150	121	148	55	150	121
	J3040_8	342	15	221	341	342	15	221	341	342	15	221	341
	J3040_9	338	43	79	123	338	43	79	123	320	34	79	243
	J3040_10	209	86	81	84	209	86	81	84	139	86	81	64

k) instances having < NC=2.1, RF=3, RS∈{0.2, 0.5, 0.7, 1} >

RS	e	Opt				SSS				MSP			
		r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4
0.2	J3041_1	79	67	76	84	83	55	139	71	115	110	125	85
	J3041_2	63	71	39	47	97	66	68	80	172	106	103	131
	J3041_3	53	64	67	53	118	104	146	73	119	188	200	76
	J3041_4	51	28	107	71	183	60	78	185	87	56	56	94
	J3041_5	43	94	85	72	10	93	78	99	28	146	119	79
	J3041_6	188	34	64	41	170	76	92	109	123	48	62	74
	J3041_7	56	66	118	111	107	145	158	155	170	117	173	263
	J3041_8	26	45	79	66	82	27	154	49	43	61	69	88
	J3041_9	16	17	26	9	62	100	54	7	51	36	70	62
	J3041_10	161	73	97	167	474	185	135	317	232	128	170	119
0.5	J3042_1	74	103	230	102	80	63	174	114	222	78	221	208
	J3042_2	45	47	43	27	76	79	28	31	68	159	103	93
	J3042_3	65	105	124	50	101	157	215	103	121	91	116	96
	J3042_4	33	41	50	43	106	89	41	102	97	109	71	56
	J3042_5	142	45	109	25	111	67	90	68	260	178	215	122
	J3042_6	92	24	40	119	224	63	46	140	166	85	215	95
	J3042_7	236	145	129	188	243	151	144	198	109	101	103	215
	J3042_8	221	133	144	153	267	210	245	235	261	105	192	167
	J3042_9	59	75	81	30	128	94	169	88	143	119	207	76
	J3042_10	238	102	219	243	293	149	123	147	561	221	472	363
0.7	J3043_1	23	62	52	70	51	73	66	77	42	54	107	121
	J3043_2	89	100	85	115	89	100	85	115	339	500	401	218
	J3043_3	91	130	123	78	115	83	158	110	124	87	152	117
	J3043_4	189	266	119	173	189	266	119	173	73	158	75	214
	J3043_5	60	143	97	42	118	232	139	72	180	343	361	163
	J3043_6	48	85	68	127	90	103	113	162	73	141	61	190
	J3043_7	163	134	96	152	89	122	101	82	78	114	210	75
	J3043_8	43	42	60	101	42	12	44	180	95	21	65	243
	J3043_9	15	79	37	40	39	96	40	41	121	125	157	125
	J3043_10	105	117	115	75	92	165	90	82	148	165	154	122
1	J3044_1	158	195	142	132	158	195	142	132	158	176	154	90
	J3044_2	115	150	126	117	115	150	126	117	115	150	126	117
	J3044_3	446	442	655	607	446	442	655	607	371	481	536	460
	J3044_4	96	90	167	128	96	90	167	128	96	69	155	98
	J3044_5	223	281	233	178	223	281	233	178	290	291	223	173
	J3044_6	350	217	429	209	350	217	429	209	279	187	489	160
	J3044_7	111	70	103	117	111	70	103	117	73	58	77	91
	J3044_8	111	163	63	80	111	163	63	80	57	91	36	80
	J3044_9	254	293	280	194	254	293	280	194	190	239	286	222
	J3044_10	406	205	280	234	406	205	280	234	454	299	310	256

l) instances having < NC=2.1, RF=4, RS∈{0.2, 0.5, 0.7, 1} >

RS	e	**Opt**				**SSS**				**MSP**			
		r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4
0.2	J3045_1	32	52	46	35	85	106	84	49	136	113	124	105
	J3045_2	144	161	130	201	213	297	257	272	161	167	112	134
	J3045_3	147	127	112	61	163	155	163	64	113	101	73	83
	J3045_4	40	28	15	53	115	94	106	184	98	134	194	217
	J3045_5	101	89	98	144	160	138	139	239	136	129	101	168
	J3045_6	52	42	69	71	66	76	63	93	94	45	68	92
	J3045_7	66	148	57	31	95	186	107	62	100	188	94	56
	J3045_8	61	110	102	59	155	267	203	162	120	271	254	105
	J3045_9	101	57	99	36	115	104	92	55	210	227	235	120
	J3045_10	103	91	100	95	96	65	75	93	59	29	41	29
0.5	J3046_1	38	36	94	80	100	89	99	51	42	36	53	64
	J3046_2	36	18	19	28	69	49	34	73	100	64	59	98
	J3046_3	37	26	46	34	35	74	48	32	39	71	57	41
	J3046_4	54	98	58	51	56	78	47	66	66	121	82	72
	J3046_5	49	36	44	39	61	43	47	43	100	110	96	108
	J3046_6	64	61	38	47	63	83	77	52	173	142	130	125
	J3046_7	68	58	25	18	35	57	59	10	19	18	20	11
	J3046_8	30	26	32	27	30	28	28	32	94	93	94	89
	J3046_9	54	81	98	100	67	119	165	190	109	199	222	192
	J3046_10	148	120	114	58	162	140	132	73	185	143	135	76
0.7	J3047_1	101	72	71	124	142	111	96	172	121	85	55	144
	J3047_2	303	210	235	242	439	278	284	274	386	271	257	275
	J3047_3	82	118	89	119	71	110	74	107	66	63	45	49
	J3047_4	42	70	44	29	50	64	59	57	180	176	133	157
	J3047_5	53	113	72	63	147	150	102	133	25	55	65	67
	J3047_6	11	27	24	20	24	67	60	52	35	79	62	57
	J3047_7	251	280	139	187	227	236	163	215	297	341	214	237
	J3047_8	122	82	111	131	122	76	103	127	165	124	120	143
	J3047_9	283	337	263	238	254	286	246	206	279	343	270	210
	J3047_10	29	27	88	25	58	54	82	24	59	59	102	40
1	J3048_1	462	333	406	335	462	333	406	335	560	250	320	583
	J3048_2	453	429	415	259	453	429	415	259	448	421	413	258
	J3048_3	377	267	375	385	377	267	375	385	291	277	331	351
	J3048_4	277	244	356	253	277	244	356	253	277	244	356	253
	J3048_5	255	246	202	311	255	246	202	311	356	285	233	417
	J3048_6	380	481	524	489	380	481	524	489	322	287	454	287
	J3048_7	279	299	227	227	279	299	227	227	349	369	247	277
	J3048_8	105	179	235	201	105	179	235	201	104	179	235	197
	J3048_9	360	502	329	496	360	502	329	496	360	502	329	496
	J3048_10	426	330	392	371	426	330	392	371	546	430	562	471

ANNEX I (ALGORITHMS)

1. The B&B DH algorithm

The Branch-and-bound procedure according to Demeulemeester and Herroelen (1992):

The following symbols are used:

d_i = duration of activity i
r_{ik} = required number of units of resource type k by activity i
a_k = per period available number of units of resource type k
$RCPL_i$ = remaining critical path length of activity i
$LB(p)$ = lower bound at level p of the search tree
C = set of all cutset activities
s_x = earliest start time of cutset activity x
H = set of precedence relations
D_q = a minimal delaying alternative
$D(p)$ = set of delaying alternatives at level p of the search tree
G_q = the set of delaying constraints corresponding with delaying alternative D_q
L_q = lower bound for delaying alternative D_q
DS = set of all activities belonging to the delaying alternative that have been started earlier than the current decision point
S = set of active activities
E = set of eligible activities
p = level of the branch and bound tree
t_i = completion time of activity i
T = project duration of the current best solution
m = decision point
PS = partial schedule

Remarks:

For simplicity of notation the subscripts m for the sets PS, S, E and C are omitted.

The save operation performed in Step 2 below should be distinguished from the store operation performed in Step 5:

- The save operation saves cutset information needed in order to apply the cutset dominance rule.
- The store operation saves information which is restored during backtracking.

The DH B&B algorithm :

Step 1: Initialization

- Let $T = 9999$ be an upper bound on the project duration.
- Set the level of the branch-and-bound tree $p = 0$.
- Initialize $m = 0$.
- For every activity i compute the remaining critical path length $RCPL_i$.
- Initialize the activity completion times $f_i = 9999$.
- Schedule the dummy start activity:
  - Set $f_1 = 0$,
  - Update the partial schedule $PS = \{1\}$,
  - Update the set of activities in progress $S = \{1\}$.
- Compute the lower bound as $LB(0) = RCPL_1$.
- Update the cutset:
  - $C = \{x \mid x \text{ has activity 1 as a single predecessor}\}$,
  - Set the early start times of the cutset activities $s_x = 0$.

Step 2: Incrementation

- Compute the next decision point m as the earliest completion time of all activities in progress: $m = \min\{f_i, i \in S\}$.
- For all activities j in S for which $f_j = m$ update the set of activities in progress: $S = S - \{j\}$.
- If the last scheduled activity is the dummy ending activity n, the schedule is completed.
- Update the schedule length $T = f_n$.
- If T is equal to $LB(0)$, then stop (with the optimal solution), else go to Step 7 (backtrack).
- → Check if the current cutset C is dominated by a previously saved cutset.
  - If so, go to Step 7 (backtrack),
  - Else save the current cutset C, save the set S of activities in progress together with their finishing times f_j and save the current decision point m.
- Construct the set of eligible activities: $E = \emptyset$ and for each activity i in C with $s_i = m$ update the eligible set $E = E + \{i\}$.
- If there are no eligible activities (i.e., if $E = \emptyset$), go to Step 2.
- If there are still some activities in progress, go to Step 4; else continue with Step 3.

Step 3: Separation

- *For each eligible activity $i \in E$ count the number of unscheduled activities j (not necessarily elements of E) which can be processed simultaneously with activity i without violating the precedence and resource constraints.*
- *If none can be ongoing con-currently with the eligible activity i then put the eligible activity in progress:*
  - $PS = PS + \{i\}$,
  - $S = \{i\}$,
  - $f_i = m + d_i$ *and*
  - *Update the cutset:* $C = C - \{i\} + \{x \mid x$ is an immediate successor of i with all its predecessors in $PS\}$.
- *For all cutset activities x in C update the early start times $s_x = f_i$.*
- *If the eligible activity i can be scheduled concurrently with only one other activity j which is eligible and does not have a larger duration than activity i, put both activities i and j in progress and update the cutset:* $C = C - \{i, j\} + \{x \mid x$ is an immediate successor of i or j with all its predecessors in the partial schedule$\}$.
- *For all x in C update the early start times $s_x = f_i$.*
- *If an eligible activity was scheduled during this step, go to Step 2, else go to Step 4.*

Step 4: Scheduling

- *Temporarily put all eligible activities in progress:*
  - $PS = PS + E$,
  - $S = S + E$,
  - *Set* $f_i = m + d_i$, *for all* $i \in E$.
- *Update the cutset:* $C = C - E + \{x \mid x$ is an immediate successor of $i \in E$ with all its predecessors in $PS\}$ *and*
- *Set the early start times of the cutset activities as follows:* $s_x = \max\{f_a \mid (a, x) \in H\}$.
- *For each resource type k check if* $\sum_{i \in S} r_{ik} \leq a_k$.
- *If there is at least one resource type k for which the sum of the resource requirements of all activities in progress exceeds the resource availability (resource conflict): go to Step 5, else go to Step 2.*

Step 5: Resource conflict

- *Update the branch level in the search tree:* $p = p + 1$.
- *Determine for each resource type* k *how many units have to be freed to resolve the resource conflict, i.e., for each* k *set* $c_k = \sum_{i \in S} r_{ik} - a_k$.
- → *Define the delaying set* $D(p) = \{D_q \subset S \mid \sum_{i \in D_q} r_{ik} \geq c_k$ *for all* k *and* D_q *does not contain other* $D_v \in D(p)$ *as a subset}.*
- *For every* $D_q \in D(p)$ *determine the earliest finishing task* j, *that is in progress and that is not delayed (i.e.,* $j \in (S - D_q)$*).*
- *Define the corresponding set of tuples* $G_q = \{(j, i)\}$ *for all* $i \in D_q$.
- *Compute for each delaying alternative* D_q *the critical sequence bound* L_q.
- *Select the* $D_q^* \in D(p)$ *with the smallest* L_q^* *(ties are broken arbitrarily).*
- *Update the delaying set:* $D(p) = D(p) - D_q^*$.
- *Set* $LB(p) = \max\{LB(p-1), L_q^*\}$.
- *If* $LB(p) \geq T$, *go to Step 7. Otherwise store:*
    - *The activity completion times,*
    - *The partial schedule,*
    - *The set of activities in progress,*
    - *The cutset activities with their early start times and*
    - *The decision point* m.

Step 6: Delay (Branch into a new node)

- *Define* DS *as the set of all activities that started earlier than m but must be delayed:* $DS = \{i \in D_q^* \mid f_i < m + d_i\}$.
- *Add the extra precedence relations at this level:* $H = H + G_q^*$.
- *Update:*
    - $PS = PS - D_q^*$,
    - $S = S - D_q^*$.
- *For all* $i \in D_q^*$ *set* f_i *equal to 9999.*
- *Update the cutset:* $C = C + D_q^* - \{r \mid x \in D_q^* \; and \; (x, r) \in H\}$.
- *For* $i \in D_q^*$ *set the early start times:* $s_i = f_j \mid (j, i) \in G_q^*$.
- *If* DS *is not empty, invoke the left-shift dominance rule as follows:*
    - *If the precedence relations which were added at previous levels of the search tree forced an activity* i *to become eligible at time* m, *if the current decision was to start that activity at time* m *and if delaying activity set* DS *would allow this task* i *to be left-shifted without causing a resource conflict,*
    - *Then this schedule is dominated: go to Step 7 (backtrack); else, go to Step 2.*

Step 7: Backtracking

- *If the branching level* $p = 0$, *then STOP. Else, delete the extra precedence relations which were added at this branching level:* $H = H - G_q^*$. *If* $D(p) = \emptyset$ *set* $p = p - 1$ *and repeat Step 7.*
- *Select the* $D_q^* \in D(p)$ *with the smallest* L_q^*.
- *Update the delaying set:* $D(p) = D(p) - D_q^*$.
- *Compute the lower bound* $LB(p) = \max\{LB(p-1), L_q^*\}$.
- *If* $LB(p) \geq T$, *decrease the branching level:* $p = p - 1$ *and repeat Step 7.*
- *Restore:*
  - *The activity completion times,*
  - *The partial schedule,*
  - *The set of activities in progress,*
  - *The cutset activities,*
  - *The early start times of the cutset activities and*
  - *The decision point* m.
- *Go to Step 6.*

Cutset Dominance Rule:

The cutset dominance rule used in step 2 of the DH B&B algorithm (the line identified by the arrow →) is defines as follows:

If(a cutset C_m at time m contains the same activities as a previously saved cutset C_k of another path),

if(time k was not greater than time m) and

if(all activities in progress at time k did not finish later than the maximum of (m, finish time of the corresponding activities in PS_m)) then

- The current partial schedule PS_m is dominated.

2. Minimal Delaying Alternatives algorithm

This algorithm is used in step 5 of the DH B&B algorithm to define the delaying set (the line identified by the arrow →), i.e., the set of minimal delaying alternatives.

The algorithm is based on the one presented and explained in algorithm 1.4 of the book "Resource Allocation in Project Management" (Schwindt, 2006).

```
MinimalDelayingAlternatives $(A, i)$
Given: a project with resource requirements $r_{ik}$ and resource availability $a_k$ with $k \in K$.
Input: a forbidden set $A$, an index i.
Output: the set of all minimal delaying alternatives $B$.
Ensure: $B$ contains all minimal delaying alternatives $A' \subseteq A$ for $F$ with $\min(A/A') > i$.

if $A$ satisfies $\sum_{i \in F \setminus A} r_{ik} \leq a_k,\ for\ all\ k \in K$ then
    // A is delaying alternative in F
    if $A$ satisfies $\sum_{i \in F \setminus A} r_{ik} + min_{j \in A} r_{jk} > a_k, for\ some\ k \in K$ then
        // A is minimal delaying alternative
        $B = B + \{A\}$
    else
        for all $j \in A\ with\ j > i$ do MinimalDelayingAlternatives $(A/\{j\}, j)$
```

The algorithm is a recursive procedure that starts with the call *MinimalDelayingAlternatives* $(F, 1)$ where:

F: is the forbidden set for which all delaying alternatives are to be computed;

1: is the starting index (the dummy start activity);

and the result B from this first call contains all delaying alternatives for F.

3. Serial SGS and priority rules

3.1 SSGS algorithm

The algorithm is based on the one presented and explained in algorithm "Serial SGS" of Kolisch and Hartmann (1999).

```
$f_1 = 0, S_1 = \{1\}$
for $g = 2$ to $n - 1$ do
    $D_g = \{i \mid i \notin S_g, Pred_i \subseteq S_g\}$
    $F_g = \{f_i \mid i \in S_g\}$
    $r'_k(t) = a_k - \sum_{i \in P_t} r_{ik}, k \in K, t \in F_g$
    select one $i \in D_g$                    $// according\ to\ the\ priority\ rule$
    $EF_i = max_{h \in Pred_i}\{f_h\} + d_i$
    $f_i = \min\{t \in [EF_i - d_i, LF_i - d_i] \cap F_g \mid r_{ik} \leq r'_k(t), k \in K, \tau \in [t, t + d_i[ \cap F_g\} + d_i$
    $S_g = S_{g-1} \cup \{i\}$
$f_n = max_{h \in Pred_n}\{f_h\}$
```

Where:

S_g: is the scheduled set, i.e., the set of activities that are already scheduled at stage g;

D_g: is the decision set, i.e., the set of non-scheduled activities that can be scheduled because all their predecessors have already been scheduled at stage g;

F_g: is the set of finish times at stage g;

$r'_k(t)$: is the remaining resource capacity of resource k at time instant t;

LF_i: is the late finish time of activity i (calculated by a backward recursion using an upper bound for the project's finish time).

3.2 Priority rules

Table 28 presents some priority rules (Kolisch, 1996a) that can be used in constructive heuristic scheduling like Serial Schedule Generation Scheme (SGSS) and Parallel Schedule Generation Scheme (PGSS) to determine the next activity to be scheduled within the scheduling process (Kolisch, 1996b).

Table 28: Priority rules

Priority Rule		**Formula (higher priority)**
LJN	Lowest Job Number	$\min_{i\in E}(i)$
RND	Random	$\mathrm{rand}_{i\in E}(i)$
SPT	Shortest Processing Time	$\min_{i\in E}(d_i)$
LPT	Longest Processing Time	$\max_{i\in E}(d_i)$
MIS	Most Immediate Successors	$\max_{i\in E}\lvert Succ_i\rvert$
MTS	Most Total Successors	$\max_{i\in E}\lvert Succ_i^*\rvert$
LNRJ	Least Non Related Jobs	$\min_{i\in E}(NRJ_i)$
GRPW	Greatest Rank Positional Weight	$\max_{i\in E}(d_i + \sum_{j\in Succ_i} d_j)$
EST	Earliest Start Time	$\min_{i\in E}(EST_i)$
EFT	Earliest Finish Time	$\min_{i\in E}(EFT_i)$
LST	Latest Start Time	$\min_{i\in E}(LST_i)$
LFT	Latest Finish Time	$\min_{i\in E}(LFT_i)$
MSLK	Minimum Slack	$\min_{i\in E}(LFT_i - EFT_i)$
GRWC	Greatest Resource Work Content	$\max_{i\in E}(d_i.\sum_{k\in K} r_{ik})$
GCRWC	Greatest Cumulative Resource Work Content	$\max_{i\in E}(d_i.\sum_{k\in K} r_{ik} + \sum_{j\in Succ_i}(d_j\,.\sum_{k\in K} r_{jk}))$

Where:

$Succ_i$: is the set of immediate (direct) successors of activity i;

$Succ_i^*$: is the set of all (direct or indirect) successors of activity i;

NRJ_i: is the set of activities that have no precedence (direct or indirect) relation with activity i;

EST_i (Early Start Time), EFT_i (Early Finish Time), LST_i (Latest Start Time) and LFT_i (Latest Finish Time) are obtained by the critical path method (CPM).