Optimizing resource allocation in a portfolio of projects related to technology infusion using heuristic and meta-heuristic methods

by

Maximiliano S. Zuloaga

B.S. in Industrial Engineering
National University of Rosario, 2010

M.S. in Finance
National University of Rosario, 2012

SUBMITTED TO THE SYSTEM DESIGN AND MANAGEMENT PROGRAM IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN ENGINEERING AND MANAGEMENT AT THE MASSACHUSETTS INSTITUTE OF TECHNOLOGY

FEBRUARY 2017

©2017 MAXIMILIANO S. ZULOAGA. All rights reserved.

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part in any medium now known or hereafter created.

Signature redacted

Signature of Author: ________________________________

System Design and Management Program
October 20, 2016

Certified by: ________________________________

Bryan R. Moser, Ph.D.
MIT System Design and Management, Lecturer
U. of Tokyo, Associate Professor
Thesis Supervisor

Accepted by: ________________________________

Warren Seering
Weber-Shaughness Professor of Mechanical Engineering
Optimizing resource allocation in a portfolio of projects related to technology infusion using heuristic and meta-heuristic methods

by

Maximiliano S. Zuloaga

Submitted to the System Design and Management program on October 20, 2016 in partial fulfillment of the requirements for the Degree of Master of Science in Engineering and Management

Abstract

In today’s competitive environment, manufacturing companies are under constant pressure to improve previous products or release new ones. Nonetheless, most products are not designed and built from scratch, but rather, are based on previous versions of the product with the addition of incremental improvements given by the infusion of new technologies. The objective of this research is to focus on continuous improvements where the level of required change is small to medium, which is the most common manner that companies use to achieve advancements in their products or systems.

Most of the available literature related to project scheduling assumes that projects are non-iterative and do not consider rework in the analyzes. On the other hand, studies that analyze cyclical projects focus on product design and development, which usually requires a level of experimentation that makes them inherently different from advancements due to incremental improvements.

At the same time, the literature on technology innovation is abundant and there are frameworks to assess the impact of transferring various technologies into existing products. However, there has not been proposed a method that specifically addresses the planning and scheduling process required to infuse technologies. Furthermore, the definitive selection for infusion cannot be applied without taking into account available resources, time required to mature technologies and the interaction among them. Portfolio selection and the scheduling process have usually been treated separately although they are interdependent in this particular case. Different plans can make quite different demands on system resources and its availability will impact the portfolio of selected technologies.

This thesis intents to bridge the gap between the portfolio scheduling as well as processes for technology selection and insertion by taking a holistic approach, while the iterative nature of activities, due to rework, is included into the model. Therefore, methods for effectively allocating resources in a portfolio of projects related to technology infusion are recommended. Initially, a heuristic method is proposed based on priority rules. However, as the assumptions of the model are loosened a novel method is suggested that combines Genetic Algorithm (GA) and Artificial Bee Colony (ABC). Numerical results indicate that the hybrid meta-heuristic method based on GA-ABC is effective in finding good resource allocations while considering rework; which is shown, can affect the projects that comprise the portfolio and therefore is worthwhile planning for.

Thesis Supervisor: Bryan R. Moser, Ph.D.
Title: MIT System Design and Management, Lecturer. U. of Tokyo, Associate Professor
Table of Contents

Abstract ................................................................................................................................. 3

Table of Figures .................................................................................................................... 7

Table of Tables ..................................................................................................................... 9

Acknowledgements ............................................................................................................. 11

Nomenclature ..................................................................................................................... 13

1. Introduction .................................................................................................................... 15
2. Technology Infusion ...................................................................................................... 17
3. RCMPSP Characteristics .............................................................................................. 21
4. Literature Review .......................................................................................................... 25
   4.1 Exact Methods ........................................................................................................... 25
   4.2 Heuristic Methods .................................................................................................... 26
       4.2.1 Priority Rule Based ......................................................................................... 26
       4.2.2 Other Heuristics ............................................................................................. 28
       4.2.3 Classical Meta-Heuristics .............................................................................. 28
       4.2.4 Non-Standard Meta-Heuristics ..................................................................... 32
       4.2.5 Other Tools and Methods .............................................................................. 32
5. This Research ................................................................................................................ 37
   5.1 Purpose ..................................................................................................................... 37
   5.2 Base Model ............................................................................................................... 38
   5.3 Improved Model ........................................................................................................ 39
6. Base Model: Priority Rule-Based Multi-Project Algorithm ............................................. 43
   6.1 General ..................................................................................................................... 43
   6.2 Assumptions ............................................................................................................. 47
   6.3 Priority Rules ........................................................................................................... 48
   6.4 Tie-Breakers .......................................................................................................... 52
   6.5 Inputs ....................................................................................................................... 52
   6.6 Outputs .................................................................................................................... 53
7. Improved Model: Hybrid Meta-Heuristic Algorithm Based on GA and ABC ............... 55
   7.1 Assumptions ............................................................................................................. 55
   7.2 One Priority Rule Per Project ............................................................................... 55
   7.3 Activity Relationships Within Projects .................................................................... 56
   7.4 Project Cancellation Is Allowed ............................................................................ 57
   7.5 Multi-Mode ............................................................................................................. 59
   7.6 Considering Differences in Performance ............................................................... 61
   7.7 Representation ......................................................................................................... 62
   7.8 Validation Generator ............................................................................................... 64
   7.9 Schedule Generation Scheme .................................................................................. 65
7.10 Genetic Algorithm................................................................................................................. 66
7.11 Artificial Bee Colony.............................................................................................................. 70
7.12 Hybrid Algorithm (GA-ABC) ................................................................................................ 72
7.13 Reason for Choosing The Proposed Methods .................................................................... 74
7.14 Considering Rework .............................................................................................................. 77
7.15 Task Duration Depending On Resource Allocation.......................................................... 78
7.16 Including Rework in the GA-ABC Algorithm ........................................................................79
8. Results.........................................................................................................................................83
  8.1 Introduction .......................................................................................................................... 83
  8.2 Base Model .......................................................................................................................... 84
  8.3 Improved Model .................................................................................................................... 90
9. Conclusion...................................................................................................................................99
10. Limitations and future work .................................................................................................101
11. Appendix ..................................................................................................................................103
  11.1 Appendix I: Standard Data Sets for Multiple Projects .........................................................103
  11.2 Appendix II: Standard Data Sets for the MRCMPSP ..........................................................105
  11.3 Appendix III: Selected Instances for Studying the MRCSP .................................................107
  11.4 Appendix IV: Example of Project Portfolio for the MRCMPSP ..........................................109
  11.5 Appendix V: Example of Adding Rework to Project Portfolio ............................................111
12. References ................................................................................................................................115
Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Risk vs. Return Among Technologies</td>
<td>18</td>
</tr>
<tr>
<td>2</td>
<td>Technology Evolution</td>
<td>19</td>
</tr>
<tr>
<td>3</td>
<td>Evolution of Sales with an Infused Technology</td>
<td>21</td>
</tr>
<tr>
<td>4</td>
<td>RCPSP Hierarchy</td>
<td>22</td>
</tr>
<tr>
<td>5</td>
<td>Multi-Project vs. Single “Big” Project Approach</td>
<td>41</td>
</tr>
<tr>
<td>6</td>
<td>Types of Heuristics Selected</td>
<td>43</td>
</tr>
<tr>
<td>7</td>
<td>Explanation of Algorithm</td>
<td>46</td>
</tr>
<tr>
<td>8</td>
<td>Priority Rules</td>
<td>47</td>
</tr>
<tr>
<td>9</td>
<td>Diverse Utility Functions</td>
<td>47</td>
</tr>
<tr>
<td>10</td>
<td>Comparison of Different Utility Functions</td>
<td>47</td>
</tr>
<tr>
<td>11</td>
<td>Curve Type 1</td>
<td>47</td>
</tr>
<tr>
<td>12</td>
<td>Curve Type 2</td>
<td>47</td>
</tr>
<tr>
<td>13</td>
<td>Curve Type 3</td>
<td>47</td>
</tr>
<tr>
<td>14</td>
<td>One PR per Project</td>
<td>56</td>
</tr>
<tr>
<td>15</td>
<td>Different Activity Relationships</td>
<td>56</td>
</tr>
<tr>
<td>16</td>
<td>DSM of Activity Relationships</td>
<td>57</td>
</tr>
<tr>
<td>17</td>
<td>Utility if Projects are Cancelled</td>
<td>58</td>
</tr>
<tr>
<td>18</td>
<td>Project Cancellation</td>
<td>58</td>
</tr>
<tr>
<td>19</td>
<td>AL Representation</td>
<td>63</td>
</tr>
<tr>
<td>20</td>
<td>RK Representation</td>
<td>64</td>
</tr>
<tr>
<td>21</td>
<td>PR Representation</td>
<td>64</td>
</tr>
<tr>
<td>22</td>
<td>Types of Schedules</td>
<td>65</td>
</tr>
<tr>
<td>23</td>
<td>Selection Operator</td>
<td>67</td>
</tr>
<tr>
<td>24</td>
<td>Crossover of Different Groups</td>
<td>68</td>
</tr>
<tr>
<td>25</td>
<td>Crossover Within the Same Group</td>
<td>68</td>
</tr>
<tr>
<td>26</td>
<td>Exploring the Search Space</td>
<td>75</td>
</tr>
<tr>
<td>27</td>
<td>Exploiting a Region</td>
<td>75</td>
</tr>
<tr>
<td>28</td>
<td>DSM of Rework</td>
<td>77</td>
</tr>
<tr>
<td>29</td>
<td>Actions to Control Project Deadline</td>
<td>79</td>
</tr>
<tr>
<td>30</td>
<td>Available Resources vs. Utilized Resources</td>
<td>80</td>
</tr>
<tr>
<td>31</td>
<td>Solutions in the Pareto Front and ( \epsilon )-dominance</td>
<td>80</td>
</tr>
<tr>
<td>32</td>
<td>Variation of Deviation with the Number of Activities</td>
<td>87</td>
</tr>
<tr>
<td>33</td>
<td>Variation of Deviation with the Complexity</td>
<td>87</td>
</tr>
<tr>
<td>34</td>
<td>Variation of Deviation with the Utilization Factor</td>
<td>88</td>
</tr>
<tr>
<td>35</td>
<td>Scenarios Applied to Portfolio 1</td>
<td>89</td>
</tr>
<tr>
<td>36</td>
<td>Comparison of Methods</td>
<td>91</td>
</tr>
<tr>
<td>37</td>
<td>Comparison Of Performance Using MNR Procedure vs. Without It</td>
<td>93</td>
</tr>
<tr>
<td>38</td>
<td>Comparison of Performance Simple GA vs. Hybrid GA-ABC</td>
<td>94</td>
</tr>
<tr>
<td>39</td>
<td>Example of Effect of Rework</td>
<td>95</td>
</tr>
<tr>
<td>40</td>
<td>Trade-Space for the Example Portfolio and Pareto Frontier</td>
<td>96</td>
</tr>
</tbody>
</table>
Figure 41 – Comparison of the Best Solution Achieved Without Consider Rework and the Best Solutions When it is Included .................................................................97
Figure 42 – Rework Matrix for Instance J20_62_8 .................................................................................................................111
Figure 43 - Rework Matrix for Instance J20_17_8 ....................................................................................................................112
Figure 44 - Rework Matrix for Instance J20_63_1 ....................................................................................................................112
Figure 45 - Rework Matrix for Instance J20_15_2 ....................................................................................................................113
Figure 46 - Rework Matrix for Instance J20_11_8 ....................................................................................................................113
Table of Tables

Table 1 – Decisions of the base model ................................................................................................................................. 53
Table 2 – Organizational Matrix Indicating Efficiency ........................................................................................................ 61
Table 3 – Decisions About GA ............................................................................................................................................. 66
Table 4 – Decisions of Improved Model ............................................................................................................................ 74
Table 5 – Characteristics of Three of the Selected Instances ............................................................................................ 86
Table 6 – Found Solutions Applying the Method Based on PR ............................................................................................ 86
Table 7 – Example of a Project Portfolio ............................................................................................................................ 88
Table 8 – Found Solutions Applying the Meta-heuristic Method ......................................................................................... 90
Table 9 – Numerical Comparison of Methods .................................................................................................................... 91
Table 10 – Performance with Multiple Modes ................................................................................................................... 92
Table 11 – Comparison with Other Methods After 5000 Generations ............................................................................... 93
Table 12 – Data Sets for Testing I ......................................................................................................................................... 103
Table 13 – Data Set for Testing II .......................................................................................................................................... 105
Table 14 – Instances with Multiple Modes ........................................................................................................................... 107
Table 15 – Characteristics of the Selected Project Portfolio ............................................................................................. 109
Table 16 – Characteristics of the Selected Project Portfolio ............................................................................................. 109
This Page Was Intentionally Left Blank
Acknowledgements

I would like to acknowledge the support received from my thesis advisor, Professor Bryan R. Moser, who had an active involvement throughout the whole thesis. The time dedicated and the guidance provided by him were invaluable to finish this document. Also, Professor James Lyneis helped me to achieve a view of the problem that I was trying to address from a system dynamic perspective. Finally, I want to thank Professor Olivier de Weck for his input on technology infusion taken from his experience on the subject as well as his contribution in the development of the meta-heuristic method that I proposed.
This Page Was Intentionally Left Blank
Nomenclature

\( J \): Number of projects in a multi-project portfolio
\( j \): Project index \((1 \leq j \leq J)\)
\( N_j \): Number of activities in project \( j \)
\( j \): Project index \((1 \leq j \leq J)\)

\( A \): Number of precedence relationships in a project network
\( m_{ij} \): Mode in which activity \( i \) belonging to project \( j \) is executed
\( d_{ij} \): Duration of activity \( i \) belonging to project \( j \)
\( d_{ijm} \): Duration of activity \( i \) belonging to project \( j \) in mode \( m \)
\( K_j \): Number of type of renewable resources used by project \( j \)
\( k \): Renewable resource index \((1 \leq k \leq K)\)
\( K_{ij} \): Number of types of renewables resources used by activity \( i \) in project \( j \)
\( r_{ijk} \): Amount of renewable resource type \( k \) used by task \( i \) in project \( j \)
\( r_{ijkm} \): Amount of renewable resource type \( k \) used by task \( i \) in project \( j \) in mode \( m \)
\( w_{ijkm} \): Amount of non-renewable resource type \( k \) used by task \( i \) in project \( j \) in mode \( m \)
\( R_k \): Renewable amount of resource \( k \) available in each time period
\( W_k \): Total amount of non-renewable resource \( k \) available for the whole portfolio
\( t \): Time period index
\( P_{ij} \): Set of all predecessors of activity \( i \) in project \( j \)
\( i \): Element of \( P_{ij} \)
\( CP_j \): Critical path duration assuming resource unconstrained for project \( j \)
\( C \): Measure of network complexity

\( n^* \): Total number of activities in the whole portfolio of projects
\( N_{mij} \): Number of modes of activity \( i \) in project \( j \)
\( MDur \): Maximum task duration of the activities within the portfolio
\( AUf_k \): Average utilization factor for renewable resource \( k \).
\( TPD \): Total project duration
\( TMS \): Total make-span
\( AHP \): Analytic hierarchy process
\( DSM \): Design structure matrix
\( WTM \): Work transformation matrix
\( CCPM \): Critical chain project management

\( PR \): Priority rule
\( TB \): Tie-breaker
TS: Tabu search
LR: Lagrangean relaxation
EA: Evolutionary algorithm
GA: Genetic algorithm
PSO: Particle swarm optimization
ACO: Ant colony optimization
ABC: Artificial bee colony optimization
FBI: Forward-backward improvement
SGS: Schedule generation scheme
MDP: Markov decision process
ABS: Agent-based simulation
MAS: Multi-agent simulation

NPV: Net present value
E[ΔNPV]: Expected marginal net present value

RS: Resource sharing (policy)
RD: Resource dedication (policy)
RIP: Resource investment problem
RIPSP: Resource investment project scheduling problem
RCPSP: Resource-constrained project scheduling problem
RCMPSP: Resource-constrained multi-project scheduling problem
MCMPSP: Multi-mode resource-constrained multi-project scheduling problem
RCMPSPPTT: Resource-constrained multi-project scheduling problem with transfer times
DRCMPSP: Distributed resource-constrained multi-project scheduling problem
PSPLIB: Project scheduling problem library (http://www.om-db.wi.tum.de/psplib/main.html)
MPSPLIB: Multi-project scheduling problem library (http://www.mpslib.com)
1. Introduction

There are different ways in which the overall value available to customers could be improved. One path (1) is to reduce manufacturing cost and pass those savings to the customer by reducing prices. Another way (2) is to innovate, modifying the current architecture of the product and therefore, increasing the value that the customer receives. This would provide the potential flexibility to expand firm margins as well as customer value (if the raise in customer value exceeds the cost due to the modification). This thesis will focus on the second path (2) to improve customer value.

A great deal of attention has been given to so called “disruptive technologies”, which might eliminate entire families of related products and render entire industries obsolete while, at the same time, it is allowing new products, enterprises and industries to emerge. These “innovation jumps” are one way in which innovation occurs. However, another way is by means of continuous smaller improvements. This method is far more common in industry, especially in tech companies.

The manner in which products progressively evolve is mainly by means of technology infusion, which is at the core of new products. Considering the fast-paced world in which companies are now submerged and the increasing level of competition, a method is required that allows companies to effectively evaluate and infuse technologies into their line of products.

At the same time, pressure from competition is leading firms to place an ever increasing emphasis in adding value to their products, delivering those products fast and by a promised date. It was indicated that innovation through the infusion of technology into a host system is one of the paths to follow in adding value. However, the time-to-market is also key to the success of a new version of a particular product and that is shown in a study that indicated that a product, which is late to market by 6 months in a product life of 5 years, could lose one third of its total recoverable profit [Nichols, 1990]. Because, the emergence of new technologies creates the necessity of incorporating them into current products to take advantage of greater functionality, greater reliability or product differentiation, it is important to handle projects effectively so that target dates are achieved.

Nowadays, the ability of firms to manage multiple projects effectively is of utmost importance. Projects have become an increasingly common structure for organizing work and dealing with many projects at the same time is common in most manufacturing and service companies, particularly for high-tech firms. Single-project settings have become rare in business today. A study was mentioned by [Lova, Maroto and Tormos, 2000], where a survey was performed in which one of the main results indicated that 84% of the companies emphasized that they usually deal with multiple projects. Other research indicate that managers usually deal with up
to four projects at the same time [Maroto et al., 1999]. Besides, a research by [Payne, 1995] revealed that 90% of projects take place in a multi-project context. Therefore, a relatively small improvement in their management could result in enormous benefit for the company.

Now that I have shown the importance of multi-project management, the question that arises is whether there are difficulties that result consistent in every multi-project environment. [Lazlo & Goldberg, 2008] addressed that issue in their research. They determined that independently of the nature of the portfolio of projects or particularities of the companies, a common vital quandary develops in every situation relating to the proper allocation of resources among simultaneous projects. Two main causes are proposed to explain the afore-mentioned situation, where one assumes the problem is a result of inadequate project scheduling. By appropriate multi-project scheduling, resources are supposed to move from one project to another efficiently. The other proposed reason, is that the resource allocation problem is a consequence of over commitment, where too many projects are accepted for the existing level of resources.

This thesis proposes to develop a methodology for effectively allocating resources and scheduling activities in a portfolio of projects related to technology infusion by combining different heuristic and meta-heuristics using computer modeling and Monte Carlo simulation. Initially, the resource allocation of the portfolio will be proposed based on different priority rules. Next, the initial model is expanded and many assumptions are relaxed in order to attain a flexible model that could fit a variety of situations. The initial method will be tested with the loosen assumptions and compared with a meta-heuristic method that combines evolutionary algorithm and swarm intelligence. In today’s technology-driven world, given the scale and speed in which new versions of products are being launched, having a comprehensive model that is suited for the fast changes that are necessary to manufacture different products, is critical for tech companies.

This thesis is structured as follows. Section 2 provides a brief overview of the importance of technology infusion as an essential mean that tech firms rely on to add value to their products. Section 3 discusses the basic characteristics of the static resource-constrained multi-project scheduling problem (RCMPSP). Section 4 presents a comprehensive literature review of the methods and tools that have been used in the past years to target the RCMPSP. Section 5 provides my personal stance about the suitability of the methods and tools, mentioned in the previous section, as applied in solving the current problem of technology infusion. Section 6 provides the explanation of the basic model that is proposed to solve the most basic RCMPSP; whereas in section 7, many assumptions of the basic RCMPSP are relaxed and the possibility of rework is introduced into the model. In section 8 the results of the basic and improved methods are provided as well as a comparison between both. Finally, section 9 and 10 correspond to the conclusion and limitations of the models, respectively.
2. Technology Infusion

Nowadays, the ever-increasing nature of competition is pressing hard on most manufacturing companies to either improve previous products or release new ones. However, most products are not designed and built from scratch, they are based on previous versions of the product where new technologies are infused in order to add, or improve, features that allow companies to attain a competitive advantage.

Technology improvements may range from minor improvements of existing components of a particular system, new subsystem advancements or disruptive innovations that drastically modify the architecture of the host system or product. Those radical innovations are not the focus of this thesis. The objective is to focus on continuous improvements where the level of invasiveness (related to the amount of design change) is small or medium, which are the most common advancements that companies have to deal with.

Sometimes, companies gradually develop new technologies in their corporate research labs. After, a certain level of maturity is attained, the technology is candidate for infusion. Another approach, is to acquire technologies from universities, suppliers or by the acquirement of smaller companies. This last approach will require to closely work with the developers of the new technology in order to assure that the readiness level necessary for making a decision on infusion is reached [Crawley et al., 2016].

Therefore, one key aspect that must be consider is the readiness of the technology. One common model used to explain the lifecycle of a particular technology is the “S-Curve” model, which attempts to explain the lifecycle of a particular technology in terms of different phases, such as: early development, fast adoption and stagnation. Another usual method is the Technology Readiness Level (TRL), in which the technology is positioned into one of nine different stages. These methodologies allow to compare the level of maturity of a particular technology, however that knowledge is not enough to decide which particular one should be selected. There are three basic pieces of information that are missing [Smaling and de Weck, 2007]:

- Level of difficulty in transitioning a specific technology from a laboratory environment to operational use.
- Effect that the technology might have on the current attributes of the product and associated manufacturing cost.
- Capture the expected value impact over time that the product with the infused technology might provide to the company.

There are many proposed ways to measure different technologies. Usually, one metric of return is used that reflects the potential value that each technology may bring to the firm, and one
metric of risk that should estimate the difficulty of integrating the new technology into the parent system.

Whenever there is a portfolio of many competing technologies in which a company may invest but resources necessary to infuse them are scarce, then it becomes a necessity defining a method to measure and prioritize those technologies. As an example in Figure 1, technology 3 provides the maximum return compared with the others but happens to be the most difficult to implement. Technology 1, on the other hand, would present less difficulty yet it provides the lowest return. Technologies 4 and 5 are less appealing options, based on this analysis, because there is another technology that delivers higher return with less associated risk (technology 2 is better than 5 in both dimensions and technology 1 is a better option than 4).

![Figure 1 - Risk vs. Return Among Technologies](image)

For the analysis mentioned above, it is possible to define both the metrics of risk and return, and then perform a qualitative analysis based on experience or intuition. However, doing such analysis qualitatively and expecting to be accurate might be a difficult task. In this respect, [Smaling, 2005] and [Smaling and de Weck, 2007] developed a quantitative framework for assessing the impact of technology infusion into current as well as future products and it was applied to the analysis of a hydrogen-enhanced internal combustion engine. Furthermore, [Suh et al, 2009] based on the previous framework worked to modify it, grounded on different critics and suggestions of the previously mentioned framework. The new methodology was applied to a complex printing system.

Technology selection is a crucial phase when planning a new version of a product or system. Once one or more technologies were selected from a pool of available ones, it is imperative to take into account the interaction among the selected technologies. [Patel & Mavris, 2006] analyzed the interaction among technologies. This issue will be addressed in section 7. Choosing one or more technologies from a group of candidates is one part of the problem that companies have to face. Once a few technologies are selected for their expected return and potential associated risk, it becomes necessary to plan for their infusion. They should be assessed for the interaction between different technologies, estimate the required resources
and verify if each technology has the appropriate maturity level. Some technologies may require more maturation. Lastly, it is necessary to schedule the activities that will lead to the effective infusion into the host system.

However, all the mentioned steps should be performed within a time frame. If the adoption of new technologies is modeled as a diffusion process shaped as an “S-curve”, before the current product or system enters the stagnation phase; ideally, the novel technologies should have been incorporated into the host system so as to create a new version of the product (Figure 2).

![Figure 2 – Technology Evolution](image)

In other words, it is expected that if the level of sales from a particular product are shown in a graph; normally, in an initial stage, the amount of sales would gradually increase until reaching a plateau and end up with a decrease in sales during a final stage. Ideally, the new version of the product should be launched before the sales of the old version enter the final phase. If a product is launched too early, it could negatively affect the level of sales of the previous version; whereas, launching too late could potentially cause competitive products to seize a greater market share. Figure 3 shows an example of this.

![Figure 3 – Evolution of Sales with an Infused Technology](image)

Finally, it is important to take into account that different technologies might have diverse maturity levels and therefore require dissimilar target dates. That is why, the process is better modeled as a multi-project scheduling problem, where the most common situations of technology infusion are:

- One new technology infused in many products
- Many new technologies infused in one product
- Many new technologies infused in various products
This Page Was Intentionally Left Blank
3. RCMPSP Characteristics

In general, the word “project” is associated with the materialization of something that was previously non-existent and it is done for one time. Projects are common structures for organizing work in today’s enterprises.

Diverse resources in varying amounts are required depending on the nature of a project. However, companies have a limited resource supply and therefore, it becomes a constraint. The resource-constraint project scheduling problem (RCPSP), is the problem of finding a way to allocate limited resources in order to complete a particular project while optimizing some predefined objective function.

The basic RCPSP deals with only one project, but there are many generalizations that can be performed in order to make a more comprehensive problem. The multi-mode resource-constrained multi-project scheduling problem (MRCPSP) is a generalization of the more basic RCPSP in two dimensions. In addition, RCMPSP is when multiple projects have to be scheduled simultaneously. Finally, tasks can be executed in multiple modes in (MRCPSP) [Wauters et al., 2014].

A lot of research has been done on RCPSP over the last fifty years. Nevertheless, the literature available on the multi-project problem has only increased during the last two decades. When more than one project is considered, there are two general approaches:

- Combining all different projects into one “big” project by the addition of two dummy activities at the beginning and at the end.
- Treating all projects separately, as a multi-project where each one is independent from the others.

If the second approach is considered, then it is a RCMPSP (resource constraint multi-project scheduling problem) where the objective is to maximize the performance, represented by some objective function, of a set of projects that take resources from a limited source. Each time concurrent activities of one or more projects of the portfolio exceed the amount of available resources, then a decision should be made about what activities to perform and which should be delayed as well as define how to best allocate the resources at hand. While aggregating many projects into one single “big” project has a considerable amount of research devoted to solve it, there are certain disadvantages to consider. To begin with, it implicitly considers that the returns, or penalties, associated with each project are the same; whereas in the RCMPSP it
is possible to account for different utility functions and target dates associated with each project, which is more realistic. Second, there are cases in which different projects might be assigned to several project managers who are only interested in the success of those projects under their responsibility. Finally, considering projects separately allows for project termination if one in particular is restricting the overall utility of the portfolio.

Optimization methods impractical for large scale RCMPSP because they are NP-hard [Lenstra and Kan, 1978]. There are two basic ways of facing this issue:

- Using heuristic and meta-heuristic methods aiming for a broad application in diverse situations.
- Using exact methods with the objective of optimizing, but focusing on small scale problems or addressing a particular situation that allows simplification of the model.

The basic RCMPSP can be characterized by a set of \( j = 2,3, \ldots, J \) projects where each project contains \( i = 1,2, \ldots, N \) activities. Task duration \( d_{ij} \) is deterministic. Tasks might be sequential or independent between each other. All projects and activities within the portfolio are known before scheduling them. The different projects consume resources from a common source, which means that resources might be shared among projects of the same portfolio. Each activity \( i \) requires exactly \( r_{ik} \) units of resource \( k \) during the whole period that the task is being done. Resources are always renewable so that the pool of available resources is \( R_k \) at every time period. \( P_{ij} \) is the set of predecessors of task \( i \) in project \( j \). Let \( O_t \) be the set of on-going activities at time \( t \). The objective is to optimize a function of the finish dates \((F_{1j},\ldots,F_{Nj})\) and target dates \((T_1,\ldots,T_j)\). The RCMPSP consists of the following points:

\begin{align*}
\text{Optimize:} & \quad f(F_{1j},\ldots,F_{Nj}; T_1,\ldots,T_j) \\
\text{Subject to:} & \quad F_{ij} - d_{ij} \geq F_{ij} \\
& \quad \forall i \in N_j, \quad j \in J \\
& \quad \forall i \in N_j, \quad j \in J, \quad i \in P_{ij} \\
& \quad \sum_{i,j \in O_t} r_{ijk} \leq R_k \\
& \quad \forall i \in O_t, \quad k \in K, \quad t \geq 0 \\
& \quad F_{ij} \geq 0 \\
& \quad \forall i \in N_j, \quad j \in J \\
\end{align*}

1. The objective is to optimize a pre-specified performance measure.
2. Activities cannot start until all the preceding tasks are finished.
3. The amount of resource $k$ utilized at every time period should always be less than or equal to the total availability $R_k$.
4. The starting time of each activity should be positive.

The afore mentioned characteristics comprise the basic RCMPSP, but there are many paths to follow if it were necessary to expand it:

- Dynamic vs static problem: it could be allowed that projects arrive at any moment and not necessarily prior to scheduling. A dynamic environment considers an open project portfolio, where a new project might arrive and be integrated into the portfolio.
- Project interdependencies: it is possible that some projects might be entangled so that the performance in one project might affect others.
- Activity relationship: activities apart from being independent or sequential, might be interdependent.
- Resources dedication (RD) policy: resources might not be shared across different projects in the same portfolio.
- Stochastic duration: activity duration might not be deterministic.
- Activity preemption: tasks might be preempted.
- Project preemption: projects might be cancelled.
- Non-renewable resources: resources might deplete and therefore stop being available.
- Transfer time: resources require time and money to switch from one project to another or even one task to another of the same project.
- Activity mode: tasks could be performed in different modes, requiring diverse numbers or types of resources as well as having different durations.

In section 6, a method to solve the problem with the above-mentioned assumptions will be proposed. Later, in section 7, many suppositions will be loosened so as to achieve a more realistic model.
4. Literature Review

As was addressed in the previous section, this thesis will focus on the RCMPSP, taking a multi-project approach by treating different projects of the same portfolio separately. Two general approaches that may be applied are exact methods and heuristics.

4.1 Exact Methods

Some exact algorithms used to tackle the RCMPSP are the ones of [Brucker et al, 1998], [Demeulemeester and Herroelen, 1997], [Decko et al, 1991] or [Vercellis, 1994] to mention a few. Since it was demonstrated that the RSPSP were a generalization of the job-shop scheduling problem and therefore is non-deterministic polynomial-time hard (NP-hard) [Blazewicz et al., 1983], most of the efforts shifted to the development of heuristic procedures, which allow attainment of “good” solution rather than optimal ones. As mentioned before, RCPSP was proven to be NP-hard and because RCMPSP is a generalization of the more basic RCPSP, it is NP-hard as well. The reason is that exact algorithms lead to impractical execution time when the number of activities increase. Hence, optimal approaches are generally used for generating benchmark solutions, dealing with simplified problems or combined with heuristic methods.

The following studies are examples of methods that combine some form of optimization with heuristics:

A solution methodology that combines deterministic dynamic programming, Lagrangian relaxation and heuristics was developed by [Ni, Luh and Moser, 2008] in order to design project schedules with special attention to varied task dependencies and communication activities. Because concurrent engineering has been widely used in design projects and often the progress of a particular task depends on the evolution of other activities, communication activities were modeled as regular ones. The relationship among activities apart from sequential and independent (which are the most common), could be concurrent (if activities should be performed at the same time). The function that was optimized was the weighted sum of penalties on task tardiness. The authors state that even though the method was applied to a single project, it could be extended to manage multiple projects that share a common pool of resources.

A stochastic dynamic programming approach combined with Lagrangian relaxation and heuristics was used in [Luh, Liu and Moser, 1999] to study the scheduling of design projects with uncertain number of iterations, seeking to minimize weighted tardiness, earliness and risk penalties. Although the method was applied to handle uncertain number of iterations, it could be extended to include other kind of uncertainties.
Because a lot of research performed in project scheduling assumes information to be invariable even though uncertainties during project execution exist, [Wang et al., 2014] presented a method for the stochastic RCMPSP based on a Markov decision process (MDP) with an analysis of events that might arise during the execution of projects. In order to deal with the problems associated with dimensionality in stochastic optimization, the authors use existing priority rules and predefined probability thresholds so that the action space and state space is limited, achieving an approximation. The study includes an analysis of their proposed method against certain priority rules such as: FCFS, EDD, COVERT, WSPT and ATC. The objective is to minimize the expected total tardiness. Finally, the authors mentioned that a possible expansion of their work is the potential application to problems that contain multiple project networks where tasks might be executed in multiple modes.

4.2 Heuristic Methods

Heuristic procedures, on the other hand, are focused on finding a “good” solution rather than an optimal one. These methods could be sub-divided into four groups: priority rule (PR) based heuristics, other heuristics, classical meta-heuristics and non-standard meta-heuristics.

4.2.1 Priority Rule Based

The most common method to solve RCMPSP are priority rule based; because they are intuitive, easy to implement, usually easy to understand and fast in term of computational effort [Brucker et al., 1999]. A multitude of priority rules have been tested so far. They can be classified by the information they require in: (1) project-based, (2) activity-based, (3) resource-based and (4) composite [Kolisch, 1996]. Priority rule based heuristics are made up of two components: A schedule generation scheme (SGS) and a priority rule. There are two different schemes: serial and parallel. The former focuses on activity-increments, while the latter makes time-increments. In each stage, the generation scheme creates a decision set for the tasks that remain to be scheduled. Next, a priority rule is used to choose from the decision set. Finally, a tie-breaker (TB) might be used if ties appear in the previous step.

Directly addressing the RCMPSP, [Browning and Yassine, 2010a] compared the performance of 20 different priority rules, using a parallel generation scheme (P-SGS) based on the characteristics of the portfolio such as: complexity, resource contention and resource distribution. In order to compare the performance of those 20 priority rules, a study was performed where the constant factors were: 3 projects per portfolio, 20 activities per project and 4 types of resources per task. To account for the different characteristics of each portfolio, 7 different levels of resource distribution, 11 levels of resource contention, 4 levels of...
complexity and 2 levels of variability in resource contention were considered. The authors performed a full factorial experiment, using a generator by [Browning and Yassine, 2010b], measuring and analyzing the performance of the diverse PR standardizing tie-breakers (TB) to increase comparability. Two objective functions were considered, average percent delay and maximum percent delay, and the experiment was performed for both. The results for each objective function were summarized in tables where the practical use of them require managers to do a qualitative characterization of the portfolio in terms of complexity, resource contention and resource distribution. According to the authors, this is particularly practical because most project managers do not (or cannot) build a formal and accurate activity network model, making other heuristic models, that depend on having an activity network, obsolete.

Studies have shown that the most suitable priority rules for iterative project portfolios differ significantly from those portfolios in which projects are acyclical [Browning and Yassine, 2015]. In this research, the performance of the 20 different priority rules used for acyclical projects are compared in the case of iterative-activity projects and 11 new rules were added that account for some characteristics of iterative projects. The results were also summarized in two decision tables. The study indicated that, if managers employ the best PRs for acyclical projects, they probably will not accomplish the best results in cyclical ones. Those 11 new PRs based on features of iterative projects, usually fail to outperform the 20 previously considered PRs.

[Vazquez et al., 2013] indicates that there are endless instances for the RCMPSP, but no single PR is best suited for every instance. Therefore, a learning process is proposed which will determine the most appropriate PR with the most fitting tie-breaker (TR).

Considering the RCMPSP [Singh, 2014] proposed a hybrid algorithm based on priority rules and analytic hierarchy process (AHP). The performance of the multi-project schedule was measured in terms of the make-span as well as in terms of the total cost deviation when comparing the actual duration of the projects with the resource unconstrained critical path duration. In order to define the sequence of activities, only priority rules were used. Nevertheless, project priority was determined by the AHP, which is a procedure to assign weights to many different projects based on some pre-defined dimensions using a scheme of pairwise comparisons.

[Kruger and Scholl, 2008] studied the RCMPSC considering transfer time of resources. In every other research mentioned in this literature review, the authors assumed that resources can be transferred from one project to another without any expenses in time or cost. However, [Kruger and Scholl, 2008] included sequence-dependent and resource-dependent transfer times. This kind of multi-project problem with transfer times is identified as RCMPSP-TT. In this study the objective function is to minimize the total average delay of the portfolio and the resource allocation is performed applying different priority rules. The results indicate that the addition of transfer time increases the average delay considerably. The authors also emphasized that only time oriented objective functions were used, but cost aspects should be
considered. In addition, they concluded that transfer scheduling rules usually had an effect on the objective function considered.

4.2.2 Other Heuristics

Other heuristic methods include those that can neither be classified as PR based heuristics nor as meta-heuristics. This category includes forward-backward improvement (FBI), sampling methods and others.

A multi-criteria heuristic method was developed by [Lova, Maroto and Tormos, 2000] that aims to improve resource allocation in multi-project scheduling. The method is divided into two phases. The first phase seeks to minimize a time-related variable such as average portfolio delay or maximum portfolio delay by iterative forward-backward passes. The second phase seeks to minimize a non-time related variable like project splitting, in-process inventory or idle resources; or maximize another non-time related variable - resource levelling. After both phases are finished, the final multi-project feasible schedule is presented. Through computational study, the authors show that the mentioned method reach better results than with heuristic methods based on priority rules such as maximum total work content and minimum latest finish time. The authors showed that the proposed method improves the feasibility of multi-project schedule obtained from heuristic methods based on priority rules such as MINLFT and MAXTWK. The method was compared against common project management software and obtained better results as well.

A hybrid heuristic that combine multi-pass random sampling and backward-forward improvement method was formulated in [Lova and Tormos, 2002] directly addressing the RCMPSP. The best possible configuration of the parameters was proposed. This configuration was obtained though a computational study, which targeted minimizing average project portfolio delay as well as maximum project duration.

4.2.3 Classical Meta-Heuristics

Many studies solve the RCMPSP by applying a classical meta-heuristic approach. These approaches follow well-known paradigms that have been applied to solve many different problems in various fields, such as: genetic algorithms, simulated annealing and tabu search among others. Some of these methods are briefly described below with research related to the RCMPSP.

Genetic algorithm (GA) is a problem solving technique based on the evolutionary ideas of natural selection which have been successfully applied to a considerable number of project
scheduling problems and extended to the multi-project case as well. GA belongs to a larger class of evolutionary algorithms (EA). Solution information is codified in a string called chromosome. The algorithm tries to improve the chromosome's potential “fitness function” by some operators such as mutation, crossover and selection. The more fitted an individual is the more probable it is to be selected for the new generation.

In [Mitsuyuki et al., 2014] a method is proposed to plan the production strategy that will cut peak demand of electricity in shipyard. This methodology is suited for the RCPSP. It is based on an organization model that indicates what task each resource is suitable for and the costs associated with them, a production model which shows the precedence relation among activities and some constraints about electricity consumption as well as available work size area. Additionally, it uses discrete event simulation, a combination of priority rules and random key-based genetic algorithm to calculate the production strategy. The objective function is to minimize the constant and variable cost associated with doing all activities with the proposed allocation of resources.

A special case of the RCMPSP was studied in [Besikci, Bilge and Ulusoy, 2014]. This research considered a multi-project situation where activities have different usage modes. A particular budget is assigned for the portfolio and instead of a resource sharing (RS) policy among the diverse projects of the portfolio from a pool of resources (that is more common in this type of problems), it considers a resource dedication (RD) policy in which each project can only spend its assigned resources without the possibility of accessing resources from other projects of the portfolio. This situation may occur in some R&D projects where the development process is technology intensive and when different projects are distributed across space that resource sharing is expensive or difficult. There are cases in which resource characteristics may prevent RS, such as the possession of heavy equipment dedicated to the projects or projects that require expertise in some particular area and cannot count on personnel from other projects because they do not possess the required knowledge. Considering the RD policy, the multi-project scheduling problem reduces to a multi-mode resource constrained project scheduling problem (MRCPS) for every particular project in the portfolio. In order to solve this problem, the authors propose two approaches: a two-phase and a monolithic genetic algorithm.

A paper from [Gonçalves et al., 2006] presents a GA suitable for the RCMPSP where the scheduling is based on three things: priorities of the activities, delay times and release dates. The algorithm was tested on a set of 10, 20, 30, 40 and 50 projects composed of 1200, 2400, 3600, 4800 and 6000 tasks, respectively.

Simulated annealing algorithm (SA) is a stochastic method for combinatorial optimization problem. This algorithm tries to minimize the thermal energy of the system by cooling down a temperature parameter. When the thermal energy of the system is minimized, the solution is in a stable state and consequently is a good solution. Any particular solution that has a lower
thermal energy, will always be accepted. However, if the thermal energy is higher (worse than the current solution), there is a probability of accepting the new solution and that probability decreases over time. SA uses the mentioned mechanism to avoid being trapped on a local optimum. There are some papers about SA application to solve project scheduling problems [Boctor, 1996] [Bouleimen and Lecocq, 2003].

In a research performed by [Chen and Shahandashti, 2009], a hybrid GA and SA is proposed for tackling the multi-project resource constraint scheduling problem. The GA, as it is a population based algorithm, provides a comprehensive exploration of the search space by recombining solutions to obtain new ones, whilst the SA algorithm focuses on the localized examination. The hybrid GA-SA algorithm was compared with other meta-heuristic methods like SA, GA and modified simulated annealing (MSA). The comparison was made using three test and three real portfolios of projects. The results were that the hybrid algorithm performed better in most cases, especially when the complexity of the multi-project scheduling increased.

A backward-forward hybrid GA with SA was applied to the RCMPSP by [Sonmez and Uysal, 2015]. This research tested with a set of 26 different portfolios formed by some projects contained in the PSPLIB. The effectiveness of the hybrid algorithm was verified by comparing the result with three popular priority rules and the outcome attained using Microsoft Project.

A multi-project scheduling method suitable for resource constraints was proposed by [Dalfard and Ranjbar, 2012], presenting a combined method of simulated annealing and a number of the best priority rules for multi-projects with the goal of minimizing maximum portfolio delay. For this research, twenty of the most common priority rules were used together with a simulated annealing algorithm where the parameters were tested as follows:

- Initial temperature: four levels (1, 0.9, 0.8 and 0.75)
- Temperature decrease rate: four levels (0.02, 0.04, 0.05 and 0.08)
- Final temperature: three levels (0.001, 0.002 and 0.005)
- Number of iterations in each temperature: one level (1000)

After performing Taguchi analysis for the diverse values of the parameters, the best result was attained with IT=0.9, TDR=0.04, FT=0.002 and NIT=1000.

Another meta-heuristic that has been applied for solving scheduling problems belong to the so called “swarm intelligence” group, which are nature-inspired meta-heuristics. Three common methods that have been used to solve the RCPSP as well as the RCMPSP are particle swarm optimization (PSO), ant colony optimization (ACO) and artificial bee colony (ABC). In PSO, a population of possible solutions are treated as “particles” that move through the solution space, and are evaluated according to some fitness criterion in each time period. Over time, particles are accelerated towards those which have better fitness values. Ant colony optimization (ACO) emulates how ants direct each other to resources while exploring their
environment by laying down pheromones. Those pheromones correlate with the probability of ants revisiting certain place and linger some time depending on the fitness value of the food source. Finally, ABC is an optimization algorithm, initially proposed by [Karaboga & Basturk, 2008] to solve multidimensional optimization problems, inspired by royal honey-bee foraging behavior. An example of the usage of ACO for solving the MRCPSP was proposed by [Li and Zhang, 2013], utilizing two types of pheromones regarding the solution in terms of sequence and mode selection of activities. The method was tested using projects from the project scheduling problem library (PSPLIB) and compared with other common meta-heuristics such as: GA, SA and PSO. The results for the ACO were usually better than those obtained with the other methods.

[Chen and Ju, 2015] performed a comparative analysis of swarm intelligence and heuristic priority rules for solving a multi-project problem, specifically the MRCPSP where activities have multiple execution scenarios. The authors adopted the single-project approach instead of the multiple-project approach. The objective function was to minimize the make-span of the project. Simulation experiments were performed in order to test the different capabilities for various scale scheduling problems comparing the performance of popular priority rules and swarm intelligence algorithms. Specifically, the authors used for swarm meta-heuristics: artificial bee colony (ABC) algorithm, ACO and PSO. For priority rules, they used both parallel generation scheduling (P-SGS) and serial generation scheduling (S-SGS). In addition, they used different priority rules for each scheme. The results of the simulation showed that swarm intelligence is superior to priority rule based heuristic for large scale problems, but present almost the same performance for small scale ones.

Tabu search (TS) is an approach that explores the vicinity of a particular solution, where the algorithm possess memory. It records the solutions which have been obtained, which prevents the search from going back to previously visited solution, thereby avoids being sunk into a local optimum [Glover, 1990] [Thomas and Salhi, 1998].

Differential evolution algorithms (DE) is a powerful technique that combine simple arithmetic operators with the classical crossover, mutation and acceptance operators. The basic scheme in DE is generating trial parameter vectors. Mutation and crossover are used to generate new vectors (trial vectors). Selection then determines which of the vectors will survive the next generation. Researches have been conducted using DE to solve project scheduling problems. Damak et al. (2009) solved the MRCPSP with a DE algorithm. In their approach a solution is represented by a mode assignment vector and a position vector. Neighbor solutions are generated using two mutation and crossover operators. Selection operator uses the values of the objective function which is penalized for infeasible solutions. The performance of this algorithm is evaluated on the benchmark instances. The obtained results were compared with
the results obtained by two other approaches: simulated annealing by [Bouleimen and Lecocq, 2003] and particle swarm optimization by [Jarboui et al., 2008].

4.2.4 Non-Standard Meta-Heuristics

Finally, non-standard meta-heuristics correspond to those methods that can be viewed as meta-heuristics, but do not follow one of the classic schemes. These methods include truncated branch and bound, disjunctive arc based and other methods. This kind of meta-heuristics have been used for solving the RCPSP, but are not common for the case with multi-projects.

A proposed methodology that approaches the RCMPSP was made by [Speranza and Vercellis, 1993] using a class of branch and bound procedure. In this method, two performance criteria were considered: NPV and service level (expressed as the difference between the completion of the projects and the corresponding target dates). The method is based upon a hierarchy of integer programming optimization models. A branch and bound procedure was introduced to solve those models.

4.2.5 Other Tools and Methods

4.2.5.1 Design Structure Matrix

The design structure matrix (DSM) is a useful tool that has been used, among many other applications, for project scheduling to analyze the relationship among tasks. In a DSM, the number of columns and rows are equivalent to the number of tasks in the project. On each row, the DSM will indicate which tasks precede the one corresponding to that particular row. It will show precedence relations as well as those tasks that are independent from each other. The DSM can be used to find alternative sequences of tasks and therefore has been widely applied in decomposition and clustering of large-scale projects. Furthermore, DSM has been used to indicate which activities have a probability of rework [Eppinger et al., 1994] and [Smith & Eppinger., 1997a]. DSM has been widely applied for modeling product design and development projects because of its capability for describing more complex relationships among tasks, such as interdependence or conditional relations. These complicated iterations among activities lead to iterations in PDD projects. Two different iteration models using DSM were presented by Smith and Eppinger. One is a sequential iteration model [Smith & Eppinger., 1997a] and the other is the parallel iteration model [Smith & Eppinger., 1997b]. The former assumed that coupled tasks were executed sequentially and rework was governed by a probabilistic rule, where repetition probabilities and activity durations were considered constant in time. The main limitation of this model is its difficulty in defining rework probabilities. The latter model supposed that the coupled design tasks were executed in parallel and iteration was governed
by a sequential rework rule. This model used the extended DSM named work transformation matrix (WTM).

The shortcomings of the WTM model are discussed in [Chen & Xiao, 2014]. A new tearing approach and inner iteration method was proposed as a complement for the WTM. Additionally, an ABC algorithm was suggested to search for optimal decoupling schemes. The study outlined by [Chen & Ju, 2010] adopted the parallel iteration model based on WTM, but included some modifications. These modifications incorporated the effect of incomplete information transfer during successive iterations and considered different reasons that could affect the time of each successive iteration, such as: learning effect and propagation change. Finally, a GA was proposed in order to find the optimum scheme as well as efforts of coupled task sets.

4.2.5.2 System Dynamics

System Dynamics can be devised as a versatile method that allows to model complex socio-technical, political and business related processes using systems thinking [Sternman, 2000]. The method is grounded in the theory of nonlinear dynamics and the fact that many causal loops exist that are interrelated and together affect the outcome of a system.

Project management is one of the most successful applications of system dynamics. [Lyneis and Ford, 2007] provides a history of the different underlying structures over time and gives comments over the benefits and limitations of the model. Traditional models focus on implementing an effective work-plan by making decisions regarding task scheduling, resource allocation and cost-time trade-offs. System dynamic models provide insight about strategic decision in projects along with the evaluation of managerial policies. These characteristics have been utilized in post mortem diagnosis that provides evaluation of the impacts of various decision undertaken during the project. Current models need to be improved by combining operational models with strategic models [Rodrigues, 1994].

[Lyneis and Ford, 2007] indicated that project managers usually employ three common actions to avoid missing a target date: hire additional personnel, work overtime and increase work intensity. However, these actions taken to close the gap between actual project performance and targets lead to ripple effects. For instance, hiring may dilute overall worker experience and prevent fluid communication leading to increasing errors and rework. Overtime eventually leads to fatigue which also increases mistakes and later rework. Finally, working faster directly generates an increase in errors. The afore-mentioned ripple effects go on to produce secondary and even tertiary feedbacks. The authors mention is that another possible way to reduce the difference between real performance and objectives is to extend the target date, but they state that not much research has been done on the consequences of this policy from a system dynamic perspective.
Delivering products and services on time and within certain budget determines success or failure of many companies. However, most complex development projects suffer from serious schedule and cost overruns [Lyneis et al., 2001]. The best known application of system dynamics has been arguably in the resolution of cost-overruns and schedule disputes, where consulting firm had applied system dynamics in different contract disputes. However, [Lyneis et al., 2001] showed a more “proactive” facet of system dynamics, which could be applied for strategic/tactical project management in the following ways: pre-project (competitors’ analysis, risk analysis, reward system and mitigation of risks), during the project (risk management and change management) and after-project (evaluation of best practices and training). The study concluded that system dynamics significantly improve the performance and quality of results obtained in complex projects.

[Zohar and Goldberg, 2008] studied the resource allocation conflicts that may arise in high-tech companies with matrix structures. In this research, the authors recognize that traditionally projects are considered as independent from each other. However, in a multi-project environment projects compete for those limited resources among each other. Three sources of conflict are recognized among managers of projects with high priority, managers of projects with low priority and functional managers. The conflict occurs by the interaction of the aforementioned three groups. Coalitions are possible between these groups in order to affect the resource allocation. A system dynamic model is proposed to emulate the flow of resources under uncertainty, in multi-project settings, and with the mentioned characteristics.

Capacity erosion in team projects is studied in [Rahmandad and Repenning, 2015]. This research analyzed two different teams of the same software company that work on projects in which the product was similar, but the outcome for both was completely different. The proposed reason for the difference was that there is a stock of features under development, if those features increase there is pressure to produce more which increase the flow of finished features. A stable equilibrium exists, in which each disturbance makes the system naturally stabilize itself. If the pressure is too high, however, there is a possibility that the tipping point is reached. This tipping point happens to be an unstable equilibrium and any perturbation will precipitate the amount of features released downwards. This occurs due to the accumulation of errors caused by the increased work pressure that forms a reinforcing loop. If a shift in the feature demand takes place, the line in which the tipping point is located would shift upwards, making the gap between the stable equilibrium and the tipping point smaller.

4.2.5.3 Critical Chain Project Management

Critical chain project management (CCPM) is a project management approach developed by [Goldratt, 1997] and it results from applying theory of constraints (TOC) specifically in project environments. CCPM seeks to eliminate safety time embedded in the activities to protect the
This safety time is caused by bad multitasking, student syndrome and Parkinson’s Law. At the same time, CCPM aggregates the large amounts of safety time added to tasks, by traditional methods, into diverse buffers. These buffers are monitored to assess the performance of a project. Defenders of this approach claim that it greatly improves schedule, cost and scope [Leach, 1999]. Because of these results, many researches on the critical chain method applied to project management have been conducted. Yet, when comparing the literature available against RCPSP and its extensions, the researches are scarce. CCPM could be used to manage multi-projects by following certain steps [Wang, 2011]: 1) Reduce activity duration by eliminating safety margins, 2) identifying the critical chain, 3) creating the project buffer, 4) creating feeding buffers, 5) organizing projects according to resource constraints, 6) creating capacity constraint buffers, 7) introducing drum buffers and 8) providing feedback.

In order to tackle the multi-project resource-constraint scheduling problem in automobile industry, specifically for R&D processes, [Yang and Fu, 2014] proposed a multi-project method based on CCPM and evidence reasoning (ER). To prove the applicability of the method, a case study of a Chinese automobile enterprise was conducted, concluding that the method can provide several benefit when managing multiple projects.

[Peng and Huang, 2013] proposed an optimization method for the critical chain project scheduling problem (CCPSP). Since CCPSP is NP-hard, a heuristic based on DE is presented where the authors design a mutation strategy that improves the global search ability as well as convergence speed, and another mutation is used to enhance local search ability. The method was tested using a modification of some standard data-sets contained in the PSPLIB.

### 4.2.5.4 Multi-Agent Systems

Multi-agent system (MAS) is composed of multiple interacting agents within an environment. While those agents might have fairly simple behavior when considered individually. Their interactions lead to the emergence of a collective intelligent behavior. The paradigm of MAS can help to find solutions especially in cases where some social conduct takes place. Usually, the RCMPSP is solved with the assumption of centralized decision making in which the resource allocation and scheduling decisions are executed centrally in an integrated manner. In practice, however, the resource allocation and scheduling functions are largely performed in a decentralized manner. Normally, a project manager competes with other project managers for the necessary global resources. In those situations, the global resources are first allocated to the competing project managers who in turn schedule the activities of their projects. The problem of resource allocation and scheduling assuming a decentralized decision making environment presents challenging combinatorial difficulties and is designated as distributed resource constrained multi-project scheduling problem (DRCMPSP) in literature. In these type of problems, projects and resources are usually modelled as agents. Projects demand resources
for fulfilling their scheduled planned work, whereas resources offer their capabilities and workforce.

An agent based approach is proposed by [Araúzo et al., 2010] for the dynamic scheduling and control in multi-project environments. The proposed MAS operates with three types of agents: resource managers, project managers and a MAC agent that basically acts as an auctioneer. The method allocates resources to projects dynamically, and it decides about project the possible acceptance or rejection taking into account its impact on the existing portfolio in terms of value, profitability, schedule and operational information. This approach uses an auction based mechanism so as to allocate global resources where price adjustment is based on resources capacity conflicts. Other methods make use of a similar market mechanism, however, price adjustment is based on task precedence conflicts rather than resource conflicts. The case proposed by [Lee et al., 2003], considers a precedence conflict-free schedule is searched though a tâtonnement type procedure. Five agent classes are introduced: project manager, resource manager, task agent, resource agent and coordinator. Following similar research line [Confessore et al, 2007] as well as [Homberger, 2007] proposed other iterative combinatorial auction mechanisms.

[Adhau et al., 2012] proposes another distributed multi-agent system using auctions based negotiation (DMAS/ABN) approach for resolving the resource conflicts and allocating multiple different types of shared resources amongst multiple competing projects. The difference between DMAS/ABN and other methods is that all MAS approaches covered in previous paragraphs suffer from the shortcoming of considering only one type of shared resource, when in practice there is hardly any project that can actually fulfill that assumption. The proposed method also allows random project release times for the projects that arrive at any time over a planning horizon. Furthermore, the same authors in [Adhau et al., 2013] studied the situation of firms dealing with projects that are distributed at diverse locations. Thus, the authors incorporated the feature of resource transfer time as well as associated cost for execution and control. The results showed significant increment in multi-project duration when transfer times are considered. This delay could be minimized by project managers if transfer times are incorporated in the planning phase.
5. This Research

5.1 Purpose

In section 2, there is an overview about the importance of technology infusion as a significant manner for adding value to customers. Likewise, in that same section, there are references to studies that have developed specific frameworks to assess the impact of new technologies if they were to be infused into a host system [Suh et al., 2010] [Smaling & de Weck, 2007]. However, the result after applying those mentioned methodologies indicates which technologies are more suitable for infusion based on the expected return and a measure of how risky the infusion process might be. Nonetheless, once technologies are selected, there is no appropriate model that covers the following step, which is the project execution.

Three main questions should be addressed before deciding on infusing certain technology into a system [Crawley et al., 2016]: (1) Will the technology be ready for infusion? (2) Will the technology create additional value for the firm, customers and other stakeholders? (3) How will the technology be effectively transferred into the product?

The available frameworks related to technology infusion, only partially respond to the second and third interrogations. To answer the first as well as part of the third question requires accounting for the impact of each project on the existing portfolio, not only in terms of value and profitability, but also considering schedule and operational information. The current approach acts as a necessary complement of previous frameworks by contributing to fill the gap between the literature in portfolio project management; usually focused on project selection based on financial information, with multi-project management; mainly concerned with operational matters, activity scheduling and resource allocation.

A novel methodology is proposed that acts as a necessary complement for current technology infusion frameworks and provides an approach for combining portfolio project management with project planning. Thereby, the decision of the projects that should contain the portfolio of technologies to infuse is also based on the actual feasibility of executing the activities necessary to release the new version of the product or system in the exact moment according to plan.

More than one technology might be infused into a parent product or one technology may be incorporated into many products. Either way, the best way to model these situations is with a portfolio of projects that will have different target dates according to the technology to infuse, market characteristics and resources available. Therefore, I will focus on a multi-project case with a shared pool of resources, which is a generalization of the more basic RCPSP. The previous section provided a summary of various studies that have been performed to address
the RCMPSP. At the same time, it gave a brief outline of the main characteristic of the optimization methods available in a situation with multiple projects.

The purpose of the current section is, based on the literature researched, to propose a methodology to tackle the basic RCMPSP, which will be developed in detail in section 6. Likewise, a method will be suggested to apply in case the assumptions of the RCMPSP (mentioned in section 3) are relaxed. The details of the improved model will be covered in section 7. The results attained when both base and improved methods are applied will be shown in section 8. Lastly, the concluding remarks and possible future improvements will be covered in section 9 and 10, respectively.

5.2 Base Model

In order to solve the RCMPSP, PRs remain a popular method mainly for their speed and simplicity [Vazquez et al., 2013] and because of the following reasons stated in [Browning and Yassine, 2015]:

- PRs have a low computational effort that makes them suitable for large scale problems common in RCMPSP.
- PRs are utilized together with other heuristic or exact methods.
- PRs are used as initial solutions for other, more complex, heuristic methods.
- PRs are common in many project scheduling softwares.

Besides, in [Browning and Yassine, 2010a] it is indicated that most project managers do not (or cannot) create the formal activity network that is required to apply any other model. Therefore, the authors summarized the results of their research in tables that require managers to do a rough qualitative characterization of a portfolio in term of complexity, resource contention and resource distribution. Based on those tables, managers can identify which are the most appropriate PRs.

I will argue that in order to have a good estimate of the level of complexity, resource contention or resource distribution, it is necessary to have an estimate of the activity network in the first place. Any miss-categorization can make the result vary widely, which is even more noticeable in [Browning and Yassine, 2015] where PRs are studied for a portfolio of iterative projects. Moreover, project characteristics have an important impact on the performance of different PRs when diverse performance objectives are considered [Patterson, 1976].

On the other hand, in [Browning and Yassine, 2010a] and [Browning and Yassine, 2015] many PRs are analyzed and explained. For the basic and improved methods, I am going to use many of the PRs covered in the mentioned researches while complementing them with others included in [Mitsuyuki et al., 2014] and [Wang et al., 2015].
In their research, [Chen and Ju, 2015] showed that diverse meta-heuristic models applying swarm intelligence tend to yield similar results as PR-based methods for small scale problems. However, as the scale increases (either by adding more projects to the portfolio or having bigger projects) then swarm methods become more fitting. The main reason appears to be that due to its hybrid searching mechanism which combines local and global searching operations, swarm intelligence gets closer to the optimal solution. Therefore, for the basic model I will use priority rules, but for the improved model which contains relaxed assumptions, and therefore as the space of possible solutions becomes bigger, I will provide a meta-heuristic algorithm that could potentially feed from PR-based solutions as initial input.

A study performed by [Vazquez et al., 2013] proposed a learning process to determine the most appropriate PR plus tie-breaker (TR) pair for each instance in a multi-project problem. The result obtained could be used directly as a solution or incorporated into another heuristic or exact method as an entrance to a later optimization process.

The method that I propose in order to tackle the basic RCMPSP will be based on the work of [Vazquez et al., 2013], but instead of using a serial SGS I will use a parallel one because a study carried out by [Lova and Tormos, 2001] showed the superior performance of parallel SGS in a multi-project context. I believe this is a good fit because the results could be either used directly or as starting solutions for the posterior method that I will propose to target the situation where assumption are relaxed. However, the objective functions that are used in [Vazquez et al., 2013] correspond to minimizing the average percent delay and overall completion time; whereas, I intend to use a utility function that relates the overall return of each project depending on the actual finish date and a target date. Depending on the characteristics of the market, product involved, technologies to infuse and nature of competitors, there might be projects in which there is no benefit to finishing early, while for other projects there might be a premium for finishing early. I have not seen such kind of utility function applied in any of the works researched.

### 5.3 Improved Model

The genetic algorithm (GA) presented by [Gonçalves et al., 2006] to solve the multi-project problem as well as the method proposed by [Lova et al., 2009] for projects with multiple execution modes, will become the base for the improved model. The initial solutions used will be obtained from the procedure utilized to solve the base model. Also, the objective function proposed by the authors corresponds to a combination of tardiness, earliness and flow time. However, the objective function that I propose to use is the same as in the base model. The relation among tasks for the base model will correspond to precedence and independence, whether the activities are sequential or not. Nonetheless, for the improved model it will be
considered another type of relation, which is concurrent. The dependencies among tasks will follow the one proposed by [Ni, Luh and Moser, 2008].

The base model will assume a shared pool of renewable resources that is available for all of the projects within the portfolio. Yet, the improved model will also feature nonrenewable resources, and the variation of processing time so that the allocation of a nonrenewable resource will modify an activity duration. A larger allocation to an activity reduces its processing time. Resources could be either local (assigned to a particular project) or global (shared among all projects).

Therefore, the planner could aim at either minimizing the project make-span subject to a fixed upper limit on the nonrenewable resources, called “the budget problem”; or at minimizing the total allocation of resources subject to a determined bound on the make-span, called “the deadline problem” [Brucker et al., 1999].

The improved model will consider that a certain budget is allocated to all of the projects and the intention would be to allocate the resources in order to maximize the total value of the portfolio subject to a budget limit (budget problem).

Previous research has already considered that a variation in resource allocation can alter its processing time. One way to model this is by using multi-modes per activity, where one task may have different resources allocated, associated cost and duration, depending on the mode. This allows diverse but discrete time-resource, time-cost and resource-resource trade-offs. This problem has been addressed in [Chen and Ju, 2015] or [Besikci et al., 2014] among others.

On a separate note, system dynamics considers policies – such as overtime, work intensity and hiring personnel - that may affect the duration of different tasks and the total project [Lyneis and Ford, 2007]. These policies lead to different ripple effects and knock-off effects that affect the amount of rework and ultimately project duration. However, this is viewed from a strategic policy point of view rather than an operational one. I will consider these three effects mentioned, in a discrete way, as causes that may affect task duration as well as modifiers of rework. There is no other work that considered the most common policies that project managers take to avoid missing a target date in an operational application such as project scheduling.

Additionally, it is important to take into consideration that not all resources might be equally fitting to perform every task. This comment could be considered as a special case of multi-mode activity where the duration might be affected by the type of resource that is used. [Mitsuyuki et al., 2014] considered what the authors call “organization model”, where each resource is related to its variable and constant cost, as well as the tasks that each resource is suitable for. I plan to consider a similar approach but expand it to allow resources to be partially fitting for some tasks.
Finally, the possibility of rework will be taken into account in the improved model. Rework will turn the project into a cyclical one. In the literature review there are some works that consider iterative activities in the case of RCPSP [Meier et al., 2016] [Chen and Ju, 2010] [Luh et al., 1999] as well as the RCMPSP [Browning and Yassine, 2015], even though the literature is mostly devoid of studies that consider the multi-project problem with iterative tasks. New PRs will be considered based on the research made by [Browning and Yassine, 2015] where they studied 11 new priority rules that account for some basic characteristics of iterative projects. Finally, system dynamic models show how work intensity and overtime affect the amount of rework [Lyneis and Ford, 2007]. However, that association is missing in project scheduling models. I will relate amount the of rework with overtime and work intensity, which in turn will also modify task duration.
In Figure 6, a decision tree is displayed that represents sequential and connected decisions. Each node represents decisions that can be controlled by the decision maker who has a finite number of possible assignments represented by the branches. Finally, the endpoints represent a complete assignment of all decisions. This diagram is a summary of the methods mentioned in the literature review and those that were selected for the base as well as the improved model. Therefore, a heuristic multi-pass method is proposed based on priority rules for the base model; whereas, the improved model might take the priority rule approach just as an initial solution but the proposed approach is a hybrid algorithm combining GA and swarm intelligence.
6. Base Model: *Priority Rule-Based Multi-Project Algorithm*

Heuristic algorithms usually take one activity at a time, taking into account precedence relation among tasks as well as resource constraints. In order to schedule the activities, it is required some criteria or priority rule, based on some metric, so that the order of activities is established.

6.1 General

There are two main schedule generation schemes (SGS) generally used to build feasible schedules: serial SGS which perform activity increments and parallel SGS that use time increments. The first group correspond to algorithms that sort all activities according to some priority rule and schedule tasks to begins as soon as precedence restrictions and resource constraints allow. On the other hand, parallel SGS analyze period by period if resources are enough to schedule all permitted activities (those that have no precedence restrictions). If resources are not sufficient, then activities are chosen following certain priority rule and tasks that are left to program in the future are evaluated in another period of time. At the same time, scheduling with both SGS could be done forward or backward.

![Figure 7 - Explanation of Algorithm](image)

![Figure 8 - Priority Rules](image)

The current model is an algorithm based on time with forward scheduling where time is broken down into unitary time steps. As time progresses, tasks evolve being included in one of four different groups (R, S, O and C). All the activities of the diverse projects of the portfolio start in the R (remaining activity) group. When precedence relations allow activities go from R to S
(selected activity) group. The afore-mentioned group is composed of all still un-started activities that could potentially be selected if there were infinite resources. However, as resources are limited it is necessary to choose from S activities based on some priority rule. Therefore, activities that belong to S are ordered based on PR and they are sequentially selected until resources are not enough to perform any more tasks. That way, for those activities where resources are sufficient, they are moved to O (on-going activity) group. There is an increment in time equal to the tasks with the minimum duration activity in the O group. Then, all tasks that were completed are moved to C (complete activity) group. Finally, as some activities were finished, it is possible that some tasks from R group might fulfill every precedence constraint and be changed to S group. This process is performed until all activities are part of the C group. Figure 7 illustrates the process that was mentioned, while Figure 8 shows the PR that could be selected for ordering tasks when resources do not suffice.

In this document it will denoted the finishing time of each resource constrained project belonging to a particular portfolio as \( C_i \), the associated cost by \( f_i(C_i) \) and the associated profit by \( g_i(C_i) \). There are mainly two types of total cost functions as well as total profit.

\[
f_{\text{max}}(C) = \max \{ f_i(C_i) \mid i = 1, \ldots, n \}
\]
\[
\Sigma f_i(C)
\]

The scheduling problem in these cases is to find a feasible schedule which minimizes any of the two mentioned total cost function or that maximizes the total profit functions.

The most common objective functions are the makespan \( \max \{ C_i \mid i = 1, \ldots, n \} \), total flow time \( C_i \), and weighted (total) flow time \( \Sigma w_i C_i \).

Other objective functions depend on due dates \( d_i \) or resource unconstraint project finishing time \( U_i \) and are defined as follows:

\[
L_i = C_i - d_i \quad \text{(lateness)}
\]
\[
D_i = C_i - U_i \quad \text{(delay)}
\]
\[
E_i = \max \{ 0, d_i - C_i \} \quad \text{(earliness)}
\]
\[
T_i = \max \{ 0, C_i - d_i \} \quad \text{(tardiness)}
\]
\[
AD_i = |C_i - d_i| \quad \text{(absolute deviation)}
\]
\[
SD_i = (C_i - d_i)^2 \quad \text{(square deviation)}
\]
\[
U_i = 0 \text{ if } C_i \leq d_i / 1 \text{ otherwise} \quad \text{(unit penalty)}
\]

With each of these aforementioned functions \( G \) there are seven different possible objectives. I will give the cases just for delay, but the procedure is the same for other cases:
1. Total Delay = $\sum D_i$
2. Average Delay = $\frac{\sum D_i}{\text{max}(i)}$
3. Weighted Delay = $\sum (w_i \times D_i)$
4. Average Percent Delay = $\frac{\sum (D_i/U_i)}{\text{max}(i)}$
5. Maximum Delay = $\max(D_i) - \max(U_i)$
6. Maximum Percent Delay = $\frac{\max(D_i) - \max(U_i)}{\max(U_i)}$
7. Maximum Weighted Delay = $\max(w_i \times D_i) - \max(w_i \times U_i)$
8. Average delay = $\Sigma$

The most common objectives are maximum lateness and total lateness. However, other objective functions which are widely used are total tardiness, weighted tardiness, total unit penalty, weighted unit penalty, average percent delay, maximum percent delay, total absolute deviation, weighted absolute deviation and weighted earliness. Linear combinations of these objective functions are also considered. For the multi-project case, the two most common objective functions are total make-span (TMS) and total project duration (TPD). The former corresponds to the time elapsed between the beginning of the earliest project within the portfolio to the end of the latest one; while, the latter relates to the sum of the differences between the critical path duration and the actual duration of the project.

Finally, another common way to proceed is by defining different penalties for ending a project before or after a target date and doing the same procedure for certain important activities. Therefore, the objective function is to minimize the overall penalties incurred when scheduling all tasks.

For the purpose of this thesis and considering that previous research in technology infusion have used a probabilistic NPV analysis to assess the economic return due to the new technology [Smaling, 2005] and [Luh et al, 2009]. A probabilistic simulation could be performed using Mot Carlo simulation (for example) in order to estimate the NPV values that might result as a consequence of infusing each technology. The diverse results of the simulation account for the uncertainties of the performance that the new technology may carry as well as the response of the market to it. Therefore, each technology can be assessed in terms of $E[\Delta\text{NPV}]$ and $\sigma[\Delta\text{NPV}]$. Taking into account the $E[\Delta\text{NPV}]$, it is possible to assign that value to certain target date. If the target date is missed, then it is expected that the market penalizes the expected return by lowering it. Also competitors’ behavior is another factor that could potentially affect it.

In Figure 10, basic utility functions are shown and explained. This utility functions reflect the expected return depending on the time. Utility functions do not have to follow the same pattern as the three cases shown below, these are simple examples that were included in the model but if another function is chosen the model could work all the same. The red square
shown in the figures correspond to an infeasible finishing date. The line where the red figure ends marks the resource unconstrained finish date that cannot be improved if resources are limited. Therefore, with resource constraints the actual finishing date should be equal to or after the red figure ends.

Utility Functions

1. The ultimate utility will be $E[\Delta NPV]$ if the project is finished at the target date or before it. As time goes beyond the target date, the utility will decrease until eventually it will equal the NPV if the current technology is maintained. At that moment, when the utility curve crosses the x axis, it will be the same weather the new technology is infused or not.

2. The utility is equal to $E[\Delta NPV]$ if the project is finished at the target date and is sequentially lower as the finish date goes beyond the target date. However, this function is different from the previous one in that there is a premium for finishing early. The earlier the project finishes the higher the utility.

3. The utility is again equal to $E[\Delta NPV]$ if the project finishes at the target date or during certain range that includes that date. Now, if the finishing date goes above or beyond the afore-mentioned range, then the utility is lower.

For instance, if the utility function follows a curve like Figure 10(a), finishing early would provide the same utility than finishing at the target date. On the other hand, in a curve like Figure 10(b) finishing before the target date would yield a bigger utility. Therefore, it would be
important to try shift resources to project with utility curve 10b because finishing early would provide benefit compared with a project with utility curve 10a.

At a high level, it is possible to say that curve type 1 penalizes tardiness while earliness is not rewarded. Curve type 2 castigates tardiness but earliness is compensated. Finally, curve type 3 penalized tardiness as well as earliness.

**Parameters required:**

Curve 1:
- $E[\Delta NPV]$
- Target Date
- Maximum Late Date (MLD)

Curve 2:
- $E[\Delta NPV]$
- Target Date
- Maximum Late Date (MLD)
- Maximum $E[\Delta NPV]$

Curve 3:
- $E[\Delta NPV]$
- Target Date
- Maximum Late Date (MLD)
- Minimum Early Date (MED)

### 6.2 Assumptions

**General**

- Tasks are of fixed duration
- Expected durations are known or can be estimated
- Resources necessary for a given task are known and fixed
- Maximum amount of resource types is known and fixed
- $E(\Delta NPV)$ for a target date are known and target dates are given
• Activities are done once (there is no rework)
• Project ends when all tasks are completed
• Portfolio of projects ends when all projects are finished
• Once an activity has started, resources assigned to that task cannot be reallocated until the activity is finished
• Activity preemption is not allowed
• Project preemption is not permitted

Control Variables
• Best Priority Rule for the portfolio
• Activity scheduling

6.3 Priority Rules

Parameters:
- $i =$ activity index $1 \leq i \leq N$
- $j =$ project index $1 \leq j \leq J$
- $ES_{ij} =$ early start of activity $i$ and project $j$
- $LS_{ij} =$ late start of activity $i$ and project $j$
- $EF_{ij} =$ early finish of activity $i$ and project $j$
- $LF_{ij} =$ late finish of activity $i$ and project $j$
- $d_{ij} =$ duration of activity $i$ and project $j$
- $CP_j =$ critical path duration of project $j$ without resource constraints
- $AS_j =$ set of activities already scheduled in project $j$
- $TS_{ij} =$ number of activities that follow activity $i$ in project $j$
- $CS_{ij} =$ number of activities that follow activity $i$ in project $j$ and belong to the critical path considering the case of unconstrained resources
- $ACTIM_{ij} =$ corresponds to the path of maximum duration starting from activity $i$ of project $j$ until the end of the project

List of PR

1. **FCFS:** first come first served

   *Formula: $Min(ES_{ij})$
   *Basis: Activity
   *Tie-Breaker: RAN

2. **SOF:** shortest operation first

   *Formula: $Min(d_{ij})$
   *Basis: Activity
   *Tie-Breaker: FCFS
3. **MOF**: maximum operation first

   *Formula*: $\text{Max}(d_{ij})$
   *Basis*: Activity
   *Tie-Breaker*: GRES

4. **MINSLK**: minimum slack

   *Formula*: $\text{Min}(\text{SLK}_{ij})$ where $\text{SLK}_{ij} = \text{LS}_{ij} - \text{Max}(\text{ES}_{ij}, t)$
   *Basis*: Activity
   *Tie-Breaker*: FCFS

5. **MAXSLK**: maximum slack

   *Formula*: $\text{Max}(\text{SLK}_{ij})$
   *Secondary Formulas*: $\text{SLK}_{ij} = \text{LS}_{ij} - \text{Max}(\text{ES}_{ij}, t)$
   *Basis*: Activity
   *Tie-Breaker*: GRES

6. **SASP**: shortest activity from shortest project

   *Formula*: $\text{Min}(f_{ij})$
   *Secondary Formulas*: $f_{ij} = \text{CP}_i + d_{ij}$
   *Basis*: Activity/Project
   *Tie-Breaker*: FCFS

7. **LALP**: longest activity from longest project

   *Formula*: $\text{Max}(f_{ij})$
   *Secondary Formulas*: $f_{ij} = \text{CP}_i + d_{ij}$
   *Basis*: Activity/Project
   *Tie-Breaker*: GRES

8. **MINTWK**: minimum total work content

   *Formula*: $\text{Min}(\sum_{k=1}^{K} \sum_{i \in AS_j} (d_{ijk} \cdot r_{ijk}) + d_{ij} \sum_{k=1}^{K} r_{ijk})$
   *Basis*: Activity/Resource
   *Tie-Breaker*: FCFS

9. **MAXTWK**: maximum total work content

   *Formula*: $\text{Max}(\sum_{k=1}^{K} \sum_{i \in AS_j} (d_{ijk} \cdot r_{ijk}) + d_{ij} \sum_{k=1}^{K} r_{ijk})$
Basis: Activity/Resource
Tie-Breaker: FCFS

10. **GRES**: greatest resource requirement

   *Formula*: $\text{Max}(\sum_{k=1}^{K} r_{ijk})$
   *Basis*: Resource
   *Tie-Breaker*: RAN

11. **LRES**: least resource requirement

   *Formula*: $\text{Min}(\sum_{k=1}^{K} r_{ijk})$
   *Basis*: Resource
   *Tie-Breaker*: RAN

12. **SST**: sum of early and late start

   *Formula*: $E_{Sij} + L_{Sij}$
   *Basis*: Activity
   *Tie-Breaker*: FCFS

13. **EDDF**: earliest due date first

   *Formula*: $\text{Min}(L_{Sij})$
   *Basis*: Activity
   *Tie-Breaker*: FCFS

14. **LCFS**: last come first served

   *Formula*: $\text{Max}(E_{Sij})$
   *Basis*: Activity
   *Tie-Breaker*: RAN

15. **MAXSP**: maximum schedule pressure

   *Formula*: $\text{Max}(\frac{t-LF_{ij}}{d_{ij}W_{ij}})$
   *Basis*: Activity
   *Tie-Breaker*: FCFS

16. **MINLFT**: maximum schedule pressure

   *Formula*: $\text{Min}(LF_{ij})$
   *Basis*: Activity
Tie-Breaker: FCFS

17. **WACRU:** weighted activity critically & resource utilization

Formula: \( \text{Max}(w \sum_{q=1}^{N_i} (1 + SLK_{iq})^{-\alpha} + (1 - w) \sum_{k=1}^{R_{Max,k}} \frac{r_{ik}}{R_{Max,k}}) \)

Secondary Formulas: \( N_i \) is the number of immediate successors of activity \( i \), \( w \) is the weight associated with \( N_i \), \( SLK_{iq} \) is the slack in the \( q \) successor of activity \( i \) and \( \alpha \) is a weight parameter

Basis: Activity/Resource

Tie-Breaker: FCFS

18. **TWK-EST:** MAXTWK followed by EDDF

Formula: \( \text{Max}(\sum_{k=1}^{K} \sum_{l \in AS_J} (d_{ij} r_{ijk}) + d_{ij} \sum_{k=1}^{K} r_{ijk}) \)

Secondary Formula: EDDF

Basis: Activity/Resource

Tie-Breaker: FCFS

19. **TWK-LST:** MAXTWK followed by LCFS

Formula: \( \text{Max}(\sum_{k=1}^{K} \sum_{l \in AS_J} (d_{ij} r_{ijk}) + d_{ij} \sum_{k=1}^{K} r_{ijk}) \)

Secondary Formula: LCFS

Basis: Activity/Resource

Tie-Breaker: FCFS

20. **MTS:** Maximum total successors

Formula: \( \text{Max}(TS_{ij}) \)

Basis: Activity

Tie-Breaker: FCFS

21. **MCS:** Maximum critical successors

Formula: \( \text{Max}(CS_{ij}) \)

Basis: Activity

Tie-Breaker: FCFS

22. **ACTIM:** Active time

Formula: \( \text{Max}(ACTIM_{ij}) \)

Basis: Activity

Tie-Breaker: FCFS
23. **GRD**: greatest resource demand  
   \[ \text{Formula: } \max(d_{ij} \times \sum_{k=1}^{K} r_{ijk}) \]  
   \[ \text{Basis: Activity/Resource} \]  
   \[ \text{Tie-Breaker: RAN} \]

24. **LRD**: least resource demand  
   \[ \text{Formula: } \min(d_{ij} \times \sum_{k=1}^{K} r_{ijk}) \]  
   \[ \text{Basis: Activity/Resource} \]  
   \[ \text{Tie-Breaker: RAN} \]

25. **RAN**: Random  
   \[ \text{Basis: Activity} \]  
   \[ \text{Tie-Breaker: FCFS} \]

### 6.4 Tie-Breakers

When priority rules assign the same value to different activities, it is necessary to break the tie. TB are priority rules used in a second step, once the actual PR has been applied. The list of TB is shorter than the list of PR. It is possible that more than one TB should be needed in order break all ties. The initial TB is mentioned with each PR in the previous list.

**List of TB**
- RAN: random
- FCFS: first come first served
- GRES: greatest resource requirement
- SOF: shorted operation first
- GRD: greatest resource demand

### 6.5 Inputs

**Portfolio Related:**
- Number of projects in the portfolio
- Maximum amount available per resource type

**Project Related:**
- Number of activities per project
- DSM representation of each project showing predecessors
- Duration per activity per project
- Amount of each resource type required per activity per project
- Utility function parameters (already explained)

### 6.6 Outputs

- Associated overall utility using each pair PR-TB
- List of start dates for the PR-TB that yields the best result

In the morphological matrix shown in Table 1, the different decisions are listed with its associated alternatives. The alternatives chosen for each decision are indicated in red font and summarizes the previous analysis of the current section. The parallel SGS was chosen for scheduling with forward direction. It was selected a multi-project approach, where the proposed method to solve the scheduling problem was a heuristic based on priority rules (PR) and tie-breakers (TB).

<table>
<thead>
<tr>
<th>Decision</th>
<th>Alternative 1</th>
<th>Alternative 2</th>
<th>Alternative 3</th>
<th>Alternative 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGS for Scheduling</td>
<td>Serial</td>
<td>Parallel</td>
<td>Both</td>
<td></td>
</tr>
<tr>
<td>Scheduling Direction</td>
<td>Forward</td>
<td>Backward</td>
<td>Both</td>
<td></td>
</tr>
<tr>
<td>Portfolio Approach</td>
<td>Single &quot;Big&quot; Project</td>
<td>Multiple Projects</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Method Applied</td>
<td>Exact</td>
<td>Heuristic</td>
<td>Combined</td>
<td></td>
</tr>
<tr>
<td>Heuristic Applied</td>
<td>Priority Rule Based</td>
<td>Classical Meta-Heuristics</td>
<td>Non-Classical Meta-Heuristics</td>
<td>Miscellaneous Heuristic</td>
</tr>
</tbody>
</table>

*Table 1 - Decisions of the base model*
7. Improved Model: *Hybrid Meta-Heuristic Algorithm Based on GA and ABC*

7.1 Assumptions

Below are listed the assumptions for the improved model, where those that are in bold correspond to the ones that differ from the base model.

- Task duration depends on the resource allocation
- Resources assigned to a particular task might vary
- Maximum amount of resource types is known and fixed
- \( E(\Delta NPV) \) for a target date are known and target dates are given
- Activities might be done more than once (rework)
- Project ends when all tasks are completed
- Portfolio of projects ends when all projects are finished
- Resources might be reassigned once an activity has started
- Activity preemption is not allowed
- A particular project might be eliminated of the portfolio

Control Variables

- Priority rules are evaluated per project
- Activity scheduling
- Portfolio project selection
- Resource allocation

7.2 One Priority Rule Per Project

Several objective functions have been used for the RCMPSP and many studies have shown that the performance of PR is tied to the objective function that was chosen [Kurtulus, 1985] or [Browning & Yassine, 2010]. Since the objective function proposed in this thesis, depends on numerous factors such as: target date, strategic value of the project and the effect of time-to-market on project return. It is possible to encounter a situation in which one project might benefit from finishing as early as possible, whereas other projects might have target dates that do not require to do so. Therefore, one way of expanding the possible solutions of the base model, developed in the previous section, is to propose one PR per project instead of one that is applied to the whole portfolio.

In order to apply one PR per project, it is necessary to implement \( J+1 \), where \( J \) is equal to the number of projects within the portfolio, PR that do not need to be different from each other. The reason behind being \( J+1 \) is that, there is one PR per project \( P_j \) \((j = 1, 2, \ldots, J)\) and another for
the portfolio to regulate the order in which the diverse PR should be combined. An example of this procedure is displayed in Figure 14; where project 1 is ordered with the maximum operation first (MOF), project 2 follows the PR of shortest operation first (SOF) and project 3 is ordered according to earliest due date first (EDDF). These PR define in which manner the tasks within each project are prioritized. Nevertheless, it should be added one more PR to state how the activities among different projects are going to be prioritized. In the example it follows a first come first served (FCFS) order.

![Figure 14 – One PR per Project](image)

7.3 Activity Relationships Within Projects

![Figure 15 – Different Activity Relationships](image)

**Precedence:** this relationship takes place for any two general tasks $i$ and $j$, if some task $i$ may only start after activity $j$ is finished. See Figure 15 (a) where activity $j$ can only start after the progress on activity $i$ is 100%.

$$\text{Completion Date}(i) + 1 \leq \text{Start Date}(j) \quad \forall i \text{ and } j \text{, } i \neq j \text{, } i = 1,2,\ldots,n \text{, } j = 1,2,\ldots,n$$
**Pace:** this relationship happens when at every time $t$, the progress of task $i$ could only be less than or equal to that of activity $j$. This would require to verify that the constraint is fulfilled at any time. In order to simplify the constraints [Ni et al., 2008] proposed that activity $i$ should not start before task $j$ is started, and cannot be finished before activity $j$ is done. Refer to Figure 15 (b) where the progress of activity $i$ and $j$ should be almost similar at every moment.

$$\text{Completion Date}(j) \leq \text{Completion Date}(i) \quad \& \quad \text{Start Date}(j) \leq \text{Start Date}(i)$$

\[\forall i, j, \ i \neq j, \ i = 1,2,\ldots,n \quad j = 1,2,\ldots,n\]

**Independence:** in this cases there is no constraint between the different activities. See Figure 15 (c).

**No Constraints**

Although each of the mentioned relationships could be represented using flowcharts, especially the ones corresponding to precedence and independent iterations, often the interdependent activities cannot be easily shown. An example is indicated in Figure 16. DSM provide an explicit way to display the required relationships among tasks and thereby is going to be the method used in the current document.

---

**7.4 Project Cancellation Is Allowed**

There are situations in which the firm does not possess the necessary resources to manage the entire portfolio of projects. In such cases, one solution could be to focus available resources in those projects that will yield important benefit or the ones that will provide a strategic advantage, postponing the others. However, the associated outcome of a particular project varies though time. This effect originates because competitors might react, customer expectations might change or new technologies may arrive. A late time to market might produce that continuing with the current product ends up being better than infusing a technology in an unsuitable moment. By cancelling projects, it would leave resources available
for the remaining ones. Therefore, it is important to incorporate, in the model, the possibility of calling off projects if that yields a greater overall benefit for the company.

As it is displayed in Figure 17, when priority rule 12 is used (SST) it is more convenient to cancel the fourth project rather than doing it because it produces a better result; whereas when priority rule 24 is applied (LRD), all 5 projects should be done to maximize the outcome.

To incorporate project cancellation into the model, it is necessary to compare the results attained when each project is abandoned; leaving the combination that yields the best result. Nevertheless, it is not required to analyze all possible combinations; this is easy to understand with an example. In Figure 18, project 1 have an $E(\Delta NPV)$ of $2,000 compared with continuing with the current product if the target date is attained; while projects 2 and 3, have $1,500 and $400 respectively. Assuming that the base model provides the best combination of PR-TB, which yields $3,200, the only combination that is worth to check is if cancelling project 3 might liberate resources for projects 1 and 2, that could potentially reach a maximum combined $E(\Delta NPV)$ of $3,500. Combination of projects 2 and 3 cannot obtain more than $1,900, whilst projects 1 and 3 could achieve a maximum of $2,400.

Therefore, the base model starts considering all projects within the portfolio. However, in those situations where cancelling one or more projects could potentially generate a better result; then, the combinations are analyzed and the best portfolio will remain regardless of the initial number of projects.
Another consideration that should be taken into account are the interactions among the different projects of the portfolio. Each project might correspond to the infusion of a particular technology into a product. It is also possible that one technology is being infused into many products. Finally, it could be that many technologies are infused into many products. In any case, there are various types of interactions that could be possible among projects. Therefore, if two projects A and B are considered, the interactions could be as follows:

- Independent: project A is independent of project B.
- Enabling: project A must be done in order to do project B.
- Inclusive: project A must be done in order to do project B and vice-versa.
- Exclusive: if project A is done, then project B cannot be executed, and vice-versa.

These interactions can be included into a matrix. There are many different ways in which those interactions could be coded. With this information, the combinations of projects could be reduced eliminating those that are infeasible. If $a_{ij}$ is the element of a matrix in row $i$ and column $j$, then:

- If $a_{ij}=a_{ji}=1$, it corresponds to an independent interaction.
- If $a_{ij}=a_{ji}=0$, it relates to inclusive interaction.
- If $a_{ij}=a_{ji}=-1$, it corresponds to exclusive interaction.
- If $a_{ij}=1$ & $a_{ji}=0$, it relates to an enabling interaction.

### 7.5 Multi-Mode

The multi-mode resource-constrained multi-project scheduling problem (MRCMPSP), similar to the previous RCMPSP that have been covered so far, also consists of a portfolio of $i$ projects ($i=1, ..., N$), where each one contains a set of $j$ activities ($j = 1, ..., J$) that have to be scheduled under precedence and resource constraints. Nonetheless, when multiple execution modes are considered, each activity $j$ ($j = 1, ..., J$) can be executed in one of diverse $M_{ij}$ modes, which represents a combination between its resource requirements and its duration. When multiple modes are considered it is possible to include non-renewable resources apart from the renewable ones. The resource constraints mean that the available amount $R_k$ for every renewable resource $k$ ($k = 1, ..., K$) is limited per period of time for every resource and the amount available of each nonrenewable $W_k$ resource $w$ ($w = 1, ..., W$) is limited for the entire project duration. Each activity $j$ ($j = 1, ..., N$) executed in mode $m$ ($m = 1, ..., M_j$) has duration $d_{ijm}$ and requires $r_{ijmk}$ units of renewable resource $k$ ($k = 1, ..., K$) and $w_{ijmw}$ units of nonrenewable resource $w$ ($w = 1, ..., W$). The goal of solving the MRCMPSP is to find sequence and mode selection for each activity as well as the resultant schedule, including start times and resource allocation policies, that leads to a maximization of the return of the project portfolio.
Adding multiple modes to the previous scheduling problem, allow to incorporate many tradeoffs such us:

- Duration-Resource: changing the amount of resource affect the duration of the activities.
- Resource-Resource: modifying one type of resource might affect the required amount of another class of resource.

Now that the multi-mode is studied, it is possible to define different types of resources that where not possible to consider before. In this thesis, the following definitions for each kind of resource is considered:

**Renewable resources**: are required to perform the different tasks, where each activity has diverse resource requirements. However, once the task is finished, the resources become available again. Typical examples are tools, equipment and space.

**Non-renewable resources**: this resources are consumable, having a limited consumption availability for the entire portfolio. If a particular activity requires some amount of this kind of resources, then they do not become available again once the task is finished. Classic examples are money or raw material, where the availability of these resources within the entire project is limited.

**Doubly constraint resources**: similar to renewable resources, this resources are constrained on a periodic basis; and similar to non-renewable resources, are also constraint for the total project duration. A possible example is a total budget with an extra restriction of a maximum limit per period.

Before the hybrid meta-heuristic method is started, [Sprecher et al., 1997] proposed a pre-processing procedure to reduce the search space whenever there are activities with multiple modes involved as well as non-renewable resources. This pre-procedure consists on eliminating inefficient modes, non-executable modes and redundant non-renewable resources. It should be considered that deleting a mode that is non-executable might cause that another mode becomes non-executable.

**Inefficient mode**: a mode is called inefficient if there is another mode, belonging to the same activity, in which the duration is lower and the resource request of both renewable and non-renewable resources are not less.
**Redundant non-renewable resource:** when the sum of the maximum request of a particular non-renewable resource is less than its availability. For the case with multiple projects, it should be considered the activities of all the projects in the portfolio and not only of a particular project.

**Non-executable mode:** when the execution of a particular mode might violate resource constraint in any given schedule.

### 7.6 Considering Differences in Performance

So far, it has been considered that every kind of renewable resource is equally qualified to address each task. However, when analyzing a resource such as manpower, it is noticeable that not all employees display similar performance for a particular activity. An experienced worker might perform twice as better as an inexperienced one. Thereby, even though both kind of workers can be assigned to a particular task, the performance will be different and the overall duration of the task will also vary. In addition, the cost per hour associated with each type of employee might be different as well, originating a trade-off between performance and cost.

<table>
<thead>
<tr>
<th>Type</th>
<th>Resource</th>
<th>Quantity</th>
<th>Cost (USD/h)</th>
<th>h/week</th>
<th>Activity 1</th>
<th>Activity 2</th>
<th>Activity 3</th>
<th>Activity 4</th>
<th>Activity 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expect</td>
<td>1</td>
<td>2</td>
<td>45</td>
<td>40</td>
<td>1.25</td>
<td>1.25</td>
<td>1.25</td>
<td>1.25</td>
<td>1.25</td>
</tr>
<tr>
<td>Experienced</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>35</td>
<td>40</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Part Time</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>25</td>
<td>30</td>
<td>1</td>
<td>0.75</td>
<td>0.75</td>
<td>0</td>
</tr>
<tr>
<td>Intern</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>20</td>
<td>30</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2 – Organizational Matrix Indicating Efficiency

As it can appreciated in Table 2, where it is shown one type of renewable resource (manpower) and one class of non-renewable resource (money). Resources 3 and 4 that are part-time workers and interns respectively, cannot take part in activities 4 and 5. Resource 2, that correspond to experienced workers can perform any activity with normal performance, whilst experts can equally take part in all activities, with the addition that they perform better than experienced workers. The different modes of a particular activity can be derived from an organizational table as the one above once the constraints are taken into account. For instance, if it is assumed that activity 3 requires at least one experienced worker or expert, it takes 400 normal hours and the total amount of workers should be 2. The diverse modes are listed below:

- **Mode 1:** 2 experienced, $14,000 and 5 weeks of duration
- **Mode 2:** 2 experts, $14,400 and 4 weeks of duration
- **Mode 3:** 1 experienced and 1 part-time, $13,760 and 6.4 weeks of duration
7.7 Representation

In order to apply a meta-heuristic method (such as GA, TS or PSO) to the scheduling problem, first it is necessary to start defining a representation for the solution space. In [Kolisch and Hartmann, 2000] are mentioned five different representations.

- Activity List (AL)
- Random Key (RK)
- Priority Rule (PR)
- Shift Vector (SV)
- Schedule Scheme (SS)

Also, in order to convert the representation into a schedule, a decoding procedure is required, which is related with the selected representation. Finally, operators should be determined to produce new solutions based on previous ones. Two groups of operators can be established:

- Unary Operators: create a new solution based on an existing one. Usually used for local search procedures. In GA it is used for mutation.
- Binary Operators: define a new solution based on two different solutions already in existence. Used in GA for crossover.

The first three representations that were listed are the most widespread and will be developed in the following paragraphs.

**Activity List Representation:** the position of a particular activity in the AL defines the relative priority of that activity compared with the others. This type of representation is easily applicable with a serial SGS as a decoding procedure because it is directly applied to any AL representation to transform it into a schedule. Parallel SGS could also be applied with AL representation, but the task to be schedule is selected according to the position of the activity in the list. The procedure is best illustrated with an example.

\[
AL = [2,4,5,1,3,6]
\]

The representation means that activity 1 will have the second position, the following activity will take the fourth place and so on. A feasible activity list is one where each activity has a higher index than its predecessors.
There are many unary operators that can be used for this representation. A popular one could be swapping two of the indexes, verifying that no precedence restriction is violated. In the case of binary operators, the difficulties are more notorious especially for the parallel SGS, for instance:

\[ \text{AL}^f = [2, 4, 5, 1, 3, 6] \]
\[ \text{AL}^m = [1, 2, 4, 6, 5, 3] \]

The “Child” could be determined selecting an integer \( i \) between 0 and 6 (in this case). Therefore, all the positions from 0 to \( i \) would be taken from one parent and the rest from the other. Considering \( i = 3 \) the child would be as follows:

\[ \text{AL}^c = [2, 4, 5, 1, 6, 3] \]

In the example showed in Figure 19, the first three positions from the AL are taken from the father and the last three from the mother. However, it can be seen that activity five is repeated and task one is missing. Therefore, it is necessary not to consider those activities already taken from the father.

**Random Key Representation:** the position of an activity is based on a priority value, between 0 and 1, attributed to each task. With this representation both the parallel and the serial SGS could be used to decode the schedule, assuring that feasible solutions are found.

\[ \text{RK} = [0.47, 0.59, 0.20, 0.97, 0.23, 0.04] \]

Numerous unary operators could be utilized, where a common one was already explained for the previous representation by swapping two priority value. Other possibility would be to modify just one priority value. Considering binary operators, it is possible to apply uniform crossover that consist on generating random numbers and if the priority value of certain activity is above that random number it would copy the value from the father otherwise the mother would be selected.
\[ RK^F = [0.47, 0.59, 0.20, 0.97, 0.23, 0.04] \]
\[ RK^M = [0.56, 0.39, 0.30, 0.47, 0.16, 0.36] \]

In the following example, displayed in Figure 20, the cut off value is 0.5 and if the random number were above that, the information would replicate that of the mother. While, the father would be selected if the random number is below the cut off value.

\[
\begin{array}{cccccccc}
0.47 & 0.59 & 0.22 & 0.97 & 0.23 & 0.04 & \text{Father} \\
0.56 & 0.39 & 0.31 & 0.47 & 0.36 & 0.36 & \text{Mother} \\
0.21 & 0.45 & 0.65 & 0.13 & 0.76 & 0.98 & \text{Random Number} \\
>0.5 & >0.5 & >0.5 \\
0.47 & 0.59 & 0.31 & 0.97 & 0.16 & 0.36 & \text{Child} \\
\end{array}
\]

*Figure 20 – RK Representation*

**Priority Rule Representation:** this representation is grounded on priority rules. Either the parallel or the serial SGS could be used to schedule the different activities.

\[ PR = [\text{MOF, EDDF, SOF, MAXSLK, MINSLK, MCS}] \]

Regarding unary operators, one priority rule could be randomly selected and change for another. In case of binary operators, it could be used the same path taken for the AL representation. A random number \( i \) is selected and the first \( i \) priority rules are taken from one parent, while the remaining would come from the other (see Figure 21).

*Figure 21 – PR Representation*

### 7.8 Validation Generator

Many meta-heuristic methods are suitable for a continuous multi-dimensional search space. However, for any project scheduling problem, the search space is discrete. Therefore, a type of representation and decoding procedure should be used in order to convert a continuous value vector obtained through the use of the meta-heuristic, into a discrete value vector. This procedure was covered in the previous section. Nevertheless, since the MRCMPSP is a
precedence constraint optimization problem, it is possible that a new vector turns out to be invalid because it is violating precedence constraints. Furthermore, it is conceivable that a set of mode selections end up being infeasible because the non-renewable resource constraint might be infringed. Whenever a situation like the one mentioned takes place, there are five paths to follow:

- Keep the infeasible solution: it will affect negatively the objective function making less probable that the infeasible solution survives.
- Iterate for a predetermined number of cycles: iterate for a limited number of cycles trying to turn the solution into a feasible one, the final result is kept whatever the outcome.
- Iterate until the solution becomes feasible: iterate until it becomes feasible or until all options were explored.
- Discard infeasible solution: directly the solution is eliminated.
- Guide the search: so that no infeasible solution is chosen in the first place.

7.9 Schedule Generation Scheme

The core difference between serial and parallel SGS is that the former perform activity-increment; whereas the latter performs time-increment.

The serial SGS generates feasible and, as [Kolisch, 1996] showed, active schedules. These are schedules where none of the activities may start earlier without delaying some other activity. The optimal solution will always be part of the active schedules.

The parallel SGS always generates feasible schedules, as the serial SGS does. However, it has been shown by [Kolisch, 1996] that the parallel SGS constructs non-delay schedules, which is a schedule in which no resource is kept idle while an activity is waiting to be executed. Non-delay schedules are included in the group of active schedules. Therefore, the parallel SGS searches in a smaller solution space that the serial SGS. Notwithstanding, there is an important shortcoming with parallel SGS because [Kolisch, 1996] exposed that often non-delay schedules
do not contain optimal ones, while optimal schedules are always in the active set. Thus, if the parallel SGS is applied, it is necessary to perform some procedure, like adding delay between activities, so as to explore the solution space that belongs to the active set but is outside the non-delay schedules. To reduce the solution space, parameterized active schedules, introduced by [Gonçalves et al., 2005] are used. The idea behind parameterized active schedules is to increase or reduce the search space by controlling the maximum delay allowed.

In the morphological matrix displayed in Table 3, the different decisions are listed with its associated alternatives. All this information was covered in previous paragraphs, but in this table the information is summarized and the alternatives chosen for each decision are indicated in red font.

<table>
<thead>
<tr>
<th>Decision</th>
<th>Alternative 1</th>
<th>Alternative 2</th>
<th>Alternative 3</th>
<th>Alternative 4</th>
<th>Alternative 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Representation</td>
<td>Activity List</td>
<td>Random Key</td>
<td>Priority Rule</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SGS for Scheduling</td>
<td>Serial</td>
<td>Parallel</td>
<td>Both</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unary Operator</td>
<td>Swap Moves</td>
<td>Random Changes</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Operator</td>
<td>One-Point Crossover</td>
<td>Two-Point Crossover</td>
<td>Uniform Crossover</td>
<td>Shift</td>
<td>None</td>
</tr>
</tbody>
</table>

*Table 3 – Decisions About GA*

7.10 Genetic Algorithm

Genetic algorithm is a meta-heuristic optimization technique developed by Holland, based on natural evolution and the survival of the fittest. It is considered a population-based method, because it improves a set of solutions from one generation to the next. There are three common genetic operators: selection, crossover and mutation. Many variation of Gas can be achieved by varying the mentioned three operators.

**Selection:** this operator is responsible for finding those individuals that are more “fitted for survival”. In nature, it is usually seen that individuals from a particular species compete for scarce resources and survival, ultimately being those individuals that are more fitted the ones that prevail over the less fitted ones. A fitter individual has more chances of producing higher number of offspring and consequently the genetic material is carried on through multiple
generations. There are several selection operators, two of the most common are: ranking-based and proportional.

In this thesis, selection operator is implemented, initially, by ranking the individuals of a generation according to their fitness value. The most fitted at the top, while the less fitted are located at the bottom. Then, the entire generation is divided into two groups, "the good" representing the top 40% and "the fair" that correspond to the 60% of the remaining individuals. Within the former group, a small percentage at the "top" that account for the best individuals are copied from the current generation into the following one. This strategy is known as "elitist" and guarantees that the genetic material of the most fitted individuals survives through generations. In Figure 23 it is displayed the process between two generations.

![Figure 23 - Selection Operator](image)

The remaining entities of the current generation are selected for crossover with a probability that goes associated with the fitness function, where the probability value is calculated as:

$$ p_v = \frac{fit_v}{\sum_{l=1}^{\text{Pop}} fit_l} $$

*Pop = Total population in each generation.*

*fit_v = Fitness value of the v member of a particular generation.*

Therefore, the better fitter the individual is, higher are the chances of being selected for crossover. Finally, a minor percentage at the end of the new generation correspond to totally new individuals that are randomly generated, adding variety to the genetic material present in each generation and prevents a rapid convergence to a local minimum.
Crossover: this a binary operator because it combines the information of two parent chromosomes to spawn one or more new individuals that inherit characteristics of both parents. There are several ways of implementing this mechanism, being the most common: one-point crossover, two-point crossover and uniform crossover.

The crossover operator selected depends if the two individuals chosen were part of the same or different groups. If both belong to “the good” (G) or “the fair” (F), then a two-point crossover with \( P_{\text{cross}} = 0.9 \) is applied. Two non-negative integers are generated, \( n_1 \) and \( n_2 \), where \( n_1 > n_2 \). Given a pair of selected individual, “mother” and “father”, two offspring are generated, a “son” (S) and a “daughter” (D). The S receives the first \( n_1 \) spots in the chromosome as well as those above \( n_2 + 1 \) from the “father”; whilst from the “mother”, the S inherits the sequence between \( n_1 + 1 \) and \( n_2 \). The D is generated in the same manner, but in those genes where the S receives from the “father”, the D inherits from the “mother”, and vice versa. An example of this type of crossover operator is shown in Figure 24.

![Figure 24 - Crossover of Different Groups](image)

On the other hand, when the selected individuals belong to different groups, a uniform crossover is proposed. For each gene, a random number between 0 and 1 is randomly generated and if the value is below a certain threshold that in this case is \( P_{\text{cross}} = 0.7 \), the gene corresponding to G is selected; otherwise the genetic material from F is inherited. With this procedure, the probability of receive genes from the most fitted individual is increased. In Figure 25, it can be seen an example of the procedure mentioned.

![Figure 25 - Crossover Within the Same Group](image)
Mutation: After the crossover operator has been applied, the offspring might still resemble the parents. Mutation modifies the chromosomes adding some extra variability that allows to increase the variation between one generation to the next and helps to prevent rapid convergence produced when a generation is trapped in a local optimum. There are different manners to implement this operator as well.

Mutation could be considered as a unary operator that creates a new chromosome based on an existing one. After crossover has been applied, then mutation is considered. The approach taken to implement this operator was that each position in the chromosome has a probability $P_{\text{mut}}$ of being selected for mutation. The probability is zero for the “top” individuals of the new generation and increases as it goes down, until it reaches the individuals at the “bottom” for whom the probability $P_{\text{mut}}=1$ and thereby are randomly generated individuals. Whenever an entity is selected for mutation, it randomly reselects a value $n_{\text{new}}$ within the interval $[n_{\text{old}} - \beta n_{\text{old}}, n_{\text{old}} + \beta n_{\text{old}}]$ from a uniform distribution, where $\beta$ is a value that decreases through the generations.

Immigration: these correspond to totally new individuals that are randomly generated and might not have similar genetic material as the other members of the new generation. These individuals can be thought as immigrant or new settlers that bring their own genetic characteristics to the population. This process is performed to prevent premature convergence.

Chromosome codification: as the representation used for the proposed method corresponds to random key representation, every gene that composes the chromosome is a number between 0 and 1. A chromosome represents a particular solution of the scheduling problem that is encoded as a vector of random keys and the purpose of the meta-heuristic method proposed is to attain the chromosome that yields the best possible value of the objective function. In an indirect representation, such as the one used, it is required a special procedure to decode the solution contained in the chromosome, which is called phenotype. Each chromosome is made of $3n^* + m$ genes, where $n^*$ represents the number of activities in the whole portfolio and $m$ signifies the number of projects in the portfolio.

$$\text{Chromosome} = (gene_1, ..., gene_{n^*}, gene_{n^*+1}, ..., gene_{2n^*}, gene_{2n^*+1}, ..., gene_{3n^*}, gene_{3n^*+1}, ..., gene_{3n^*+m})$$

The initial $n^*$ genes define the priorities. Genes between $n^*+1$ and $2n^*$ are used to determine the mode in which the activity is executed. The following $n^*$ genes represent the delay before starting each activity and the last $m$ genes define if the project is completed or cancelled.
**Decoding**: in order to obtain the solution (phenotype) from the chromosome it is necessary to decode it. Therefore, it is necessary to decode priorities, modes, delays and cancelled projects.

**Priority**: the initial $n^*$ genes of the chromosome represent the priority of the activities $i$ ($i=1, ..., n^*$) that belong to the portfolio of projects.

$$Priority_i = Gene_i$$

**Mode**: the genes between $n^*+1$ and $2n^*$ correspond to the mode. For each activity $i$ there are certain modes in which the activity might be performed. Then, the space between 0 and 1 is divided in as many segments as modes are for a particular activity. The mode will depend on the segment in which the gene is included. In the equation below $Nm_i$ corresponds to the number of modes of activity $i$.

$$Mode_i = \text{RoundUp} \left( \frac{Gene_{n^*+i}}{1/Nm_i} \right)$$

**Delay**: genes between $2n^*+1$ and $3n^*$ are used to define the delay times. In the equation below $MDur$ correspond to the maximum duration amongst all task durations. The factor 1.2 is proposed after trying with values between 1 and 2.

$$Delay_i = Gene_{2n^*+i} \times MDur \times 1.2$$

**Cancellation**: the last $m$ genes indicate which projects of the portfolio are going to be performed and which ones are going to be cancelled. If the value of the gene is closer to 1 then the project $j$ ($j=1, ..., m$) is performed, otherwise project $j$ is cancelled. It should be mentioned that if the result is infeasible when taking into account the interaction among technologies, the genes corresponding to the cancellation are modified until a feasible outcome is attained.

$$Cancellation_j = \text{Round}(Gene_{3n^*+j})$$

**Population Size**: the size could be proportional to the number of activities within the portfolio multiplied by a number between 2 and 20. This policy is not adequate because the number of activities might differ depending on the portfolio. Another possibility is when the size is proportional to the number of genes in the chromosome multiplied by a number between 2 and 8.

$$Pop\ Size = 4 \times \#Genes$$

7.11 **Artificial Bee Colony**

Artificial bee colony (ABC) belongs to the group of “swarm intelligence”, which refer to the collective behavior of decentralized and self-organized systems; usually composed by agents
that follow simple rules, where their interactions lead to the emergence of intelligent behavior. The most common algorithms of this group that were applied to the project scheduling and planning problem are: ABC, ACO and PSO [Chen & Ju, 2015].

The ABC is a population-based algorithm inspired by the intelligent foraging behavior observed in real honey bees. ABC considers three different kind of honeybees: employed, onlookers and scouts. Each solution in the search space is treated as a food source, where the fitness value of the solution is represented by the amount of nectar contained in each food source. The number of employed bees equals the food sources around the hive. Employed bees share their information with onlookers, while onlookers select one of the food sources according to this information. Finally, a scout is a bee performing random search around the hive in order to find new food sources.

The initial population: correspond to SN number of randomly generated food sources that is create applying the following formula:

\[ x_i^j = x_{\min}^j + \alpha(x_{\max}^j - x_{\min}^j) \]

where \( \alpha \) is a random number within the interval (0,1). \( i=1, 2, \ldots, \text{SN} \) and \( j=1, 2, \ldots, D \) are the lower and upper bounds for dimension \( j \), respectively. Each food source is assigned to SN number of employed bees.

The search phase: each employed bee explores the neighborhood of every food source \( x_i^j \) by modifying one parameter obtaining \( v_i^j \).

\[ v_i^j = x_i^j + \gamma_i^j(x_i^j - x_k^j) \]

Where \( \gamma_i^j \) is a random number in the range [-1, 1], \( k=1, 2, \ldots, \text{SN} \) and \( j=1, 2, \ldots, D \). Afterwards, \( x_i^j \) is compared with \( v_i^j \) and greedy selection is applied depending on the amount of nectar in each location. If \( v_i^j \geq x_i^j \), then \( v_i^j \) replace \( x_i^j \), otherwise \( x_i^j \) remains as the food source.

Selection phase: every employed bee shares the information about the diverse food sources with the onlooker bees which depending on the amount of nectar in each source execute some kind of selection scheme like ranking based, tournament selection, proportional selection, among others. In the original ABC algorithm, roulette selection is applied:

\[ p_i = \frac{f_{iti}}{\sum_{i=1}^{SN} f_{iti}} \]
where $\text{fit}_i$ is the fitness value of each food source and the higher the value, the more likely it is that the food source is selected by the onlooker bees. Then, onlooker bees will continue to search the neighboring area by using the equation previously indicated for the search phase.

$$v'_i = x'_i + \gamma'_i (x'_i - x'_k)$$

**Scout bee phase**: this is the last phase and it takes place if any food source cannot be improved further after searching the neighborhood for a pre-established number of cycles. In that case, the food source is abandoned and scout bees are sent to explore the region around the hive for brand new food sources to exploit. The formula corresponds to the one utilized for the initial population.

### 7.12 Hybrid Algorithm (GA-ABC)

This algorithm takes the strategy that was indicated in the subsection: Genetic Algorithm. However, it adds components of ABC so as to attain a nice balance between exploration and exploitation of the search space as well as reaching a suitable convergence speed.

The initial population is set with a number of chromosomes that depends on the genes contained in each one of them as it was explained in previous sections. The population of solution is created using the priority rules available from the base model and then it is populated with randomly created priorities so as to guarantee a varied population. This procedure is explained in detail in [Lova et al., 2009]. The initial mode assignment is performed by following a procedure called Minimum Normalized Resources (MNR) that reduces the probability of starting with infeasible solutions with respect to non-renewable resources. The MNR implies selecting the mode that has the minimum resource requirement of non-renewable resources (NW).

$$NW_{ijm} = \sum_{k \in NW} \frac{w_{ijm_k}}{W_k}$$

The modes are initially selected following the MNR criterion, but in order to add diversity to the initial population, the modes of 50% of the total individuals are randomly changed, if the solutions are feasible then solutions are kept. Otherwise, for those assignments that turned out to be infeasible it will change the value randomly to another set of modes as long as the NW value is reduced. The procedure is performed for up to 100 iterations. For the genes that correspond to project cancellation, random key numbers are created and if the results are compared with the infeasible technology interactions so as to maintain a population of feasible
solutions for selection. Afterwards, the individuals of the population are selected as it was mentioned in the section related to GA, and the crossover operator as well as mutation are applied as explained. Then, each solution enters the employee bee phase where one by one, each food source is compared with one neighbor solution that is obtained by modifying one gene. Greedy selection is applied and the best solution remains (if solutions have the same fitness value, the new solution remains). Subsequently, a probability vector is calculated according to the fitness value of each solution in order to decide which ones enter the onlooker bee phase. With the selected solutions, the neighborhood is analyzed by selecting at least three solutions changing one gene, if the solutions are better it increases the probability of continuing with the search up to certain iterations. However, if any of the three solutions is better it goes to the next food source. Finally, the algorithm examines which solutions did not improved for the last 50 iterations and it send scout bees to find new solutions unless it is a member of the “top” group in which case the solution remain for the potential of creating fitted individuals after crossover or mutation.

To maintain diversity in the population, the mechanism of immigration as well as directed mutating is introduced. The differences between solutions is measured with a population affinity index (PAI).

\[
\text{PAI} = \sqrt{\sum_{i=1}^{n} [(\text{priority}_i - \text{priority}_l)^2 + (\text{delay}_i - \text{delay}_l)^2 + (\text{mode}_i - \text{mode}_l)^2 + (\text{cancellation}_i - \text{cancellation}_l)^2]}
\]

Whenever, the PAI is close to 0 then it means that the difference between two solutions is minimal. In addition, it is used as a global indicator of the variation in the entire population. In case that the population is not varied enough, a procedure is applied with certain probability \( P_{\text{rep}} \) that consist on exchanging one individual of the population that does not belong to the “top” group for a randomly generated individual with a probability \( P_{\text{change}} \) using the MNR procedure previously described. This procedure helps to maintain a diverse class of individuals without converging prematurely. The values of \( P_{\text{rep}} \) and \( P_{\text{change}} \) are set to 0.5 and 0.2 respectively.

The morphological matrix displayed in Table 4 indicates the high-level decisions taken when choosing the current method. It is a heuristic method that belongs to the class of classical meta-heuristics. The method combines genetic algorithm (GA), which belongs to the evolutionary algorithms; and artificial bee colony (ABC), that is a type of swarm intelligence method. In addition, in order to improve the local search characteristics some components of tabu search were introduced.
Table 4 – Decisions of Improved Model

7.13 Reason for Choosing The Proposed Methods

This sub-section is devoted to explaining the reason behind choosing the selected methods. Every meta-heuristic should be designed with the objective of exploring the search space as thoroughly as possible. With that in mind, two concepts arise that lead meta-heuristic applications to high performance: exploration and exploitation. Even though the aforementioned terms are common in scientific literature related to meta-heuristics, there have been researchers that refer to them as diversification and intensification, respectively. These expressions might be used as synonyms or terms somewhat related. In any case, the main idea is that the search should be diversified enough so that different regions of the search space are explored, avoiding staying in a particular and limited region alone because other areas might contain better solutions. At the same time, some effort should be spent in searching the neighborhood around good solutions because better points might be located close to them. Many other definitions can be found in literature, but they all seem to share similar principles. “The search performed by a meta-heuristic approach should be “clever” enough to both intensively explore areas of the search space with high quality solutions, and to move to unexplored areas of the search space when necessary”. [Blum & Roli, 2003].

**Exploration:** refers to the broad examination of unvisited regions of the search space. In some literature the term is stated as diversification. Figure 26 displays an example.

**Exploitation:** is the thorough search around good solutions found in the past. This term is also indicated as intensification. Figure 27 shows an example.
In order to achieve a successful search with a meta-heuristic method, both forces must be contemplated and balanced. In other words, it is necessary to perform a good job in exploring and exploiting the search space. "A meta-heuristic will be successful on a given optimization problem if it can provide a balance between the exploitation of the accumulated search experience and the exploration of the search space to identify regions with high quality solutions in a problem specific, near optimal way." [Stützle, 1999].

Two differentiated categories of meta-heuristics could be defined: population-based and trajectory-based. The first group correspond to methods that select a group of solutions and achieve new solutions based on the previous ones; whilst the second group perform trajectories in the search space by successive improvements of a given solution. For the latter group simulated annealing and tabu search are typical examples, while for the former group evolutionary algorithms and swarm intelligence are good exponents.

Population-based methods provide a natural way of exploring the search space even though the performance depends on the way the set of solutions is manipulated. On the other hand, trajectory-based methods offer a natural way of exploiting a limited region of the search space. Therefore, in the case of trajectory-based methods it is important to provide mechanisms to explore different regions of the search space so that it avoids being trapped in a local optimum. Conversely, population-based methods require mechanisms to explore in more detail the area surrounding the set of solutions. Hence, even though a particular method could naturally provide a natural way of exploring or exploiting the search space, it is important to include mechanisms so that there is a balance between exploitation and exploration.
Every method, either population-based or trajectory-based, possess both exploration and exploitation components. It could be possible that some methods provide a natural way of exploring or exploiting the solution space, and in those cases they should be balanced with components that provide the dimension in which the selected method shows weakness. All meta-heuristic should have components that allow to explore the whole search space while each region is exploited so that there is a good probability that the best solution was chosen. For instance, in a trajectory-based method such as SA, the decrease of the temperature parameter drives the system from exploration to exploitation. Another example is that, previously in this section, it has been indicated that for selection in GA it is proposed elitism as a mechanism, that means that the top individuals are copied unchanged from one generation to the next. This mechanism guarantees that elite individuals survive through generations, but could lead to rapid convergence and prevents exploring other regions of the search space. However, that procedure is counterbalanced by another mechanism such as immigration, which guarantees that random individuals are being created, thereby favoring the exploration of new regions of the search space. The example provided shows that diverse mechanisms could be chosen to balance the exploration and exploitation of the search space.

In addition, two other dimensions to consider when selecting and designing a meta-heuristic method are: convergence and memory.

**Convergence:** relates to the time that takes to arrive to a situation in which solutions from successive generations are fairly similar. Even though a particular method might do a do exploration and exploitation of the search space, if convergence is fast it does not allow enough time to search the whole space, potentially being trapped in a local optimum. Contrariwise, if convergence is slow it will make the search computationally expensive.

**Memory:** refers to the how the information about previous generations affect the new one. In order words, if a method uses only the information about the previous generation to build the next, then it is a low-memory method. Conversely, if the new generation is built with information from many generations back, then the method makes use of memory.

**Hybridization:** to this point, I have covered the main characteristics that efficient meta-heuristic methods should account for. Now, I will focus on hybridization of meta-heuristics, which is an ongoing research field. The reasons behind this practice is to harness the advantages of each method, while complementing the shortcomings of one method with the strengths of another. One manner of hybridization consists on including components of both population-based and trajectory-based algorithms. The motivation is that population-based methods are better suited for recognizing promising region in the search space, whilst trajectory-based methods are
superior in exploiting those promising areas previously found. The combination of ABC and GA was proposed in this thesis because GA is usually regarded as an algorithm with fast convergence, whereas ABC has components with great exploratory characteristics but suffers from slow convergence. By combining both methods convergence turns out balanced.

### 7.14 Considering Rework

Every innovative and developmental process is inherently cyclical. Usually, iterations represent the rework of a particular activity due to discrepancies between expected and actual results, where the differences are found due to feedback information discovered later in the project. Furthermore, with “concurrent engineering” where design and engineering work in parallel, the increase in iterations became a huge challenge. Even though the process of infusing technology into a product or system is not as iterative as building a product from scratch, it has certain amount of iterations in it. Therefore, it becomes important to incorporate this feature into the model to make it more realistic of a technology infusion process. Of course, the amount of iteration will depend on the nature of the projects, where software projects will probably have a lot of iterations. On the other hand, infusing technology into an aircraft will require most of the work to be done upfront. In the middle, is that high tech consumer products will fall and that is the focus of the current thesis. It is included the possibility of rework with the following assumptions:

- The probability of discovering rework after each activity is previously known.
- It is possible to either encounter rework immediately after the same task is finished or during other activities with a pre-established probability.
- It is not possible to do, at the same time, activities that depend on the activity that requires rework.
- If certain activity need rework at some time, the probability of redoing posterior activities remains unaffected.
- Resources necessary for redoing an activity are previously known.

![Figure 28 - DSM of Rework](image)

To indicate the activities that necessitate rework, the DSM was used. Cycles in the network can be produced by feedback arcs, which represent the potential yet not the certainty of having to return to an activity previously done. Feedback arcs have a probability associated with them.
Hence, they will be indicated with numbers between 0 and 1 located in the matrix supradiagonal, which denote the probability that the activity of the corresponding row i requires rework while task j, indicated by the column, is being performed. For instance in Figure 28 Activity 2 has 10% chances of rework after its completion. At the same time, there is a 10% chances of redoing activity 1 while doing activity 3 and 20% chances of redoing activity 1 while activity 5 is being performed.

Regarding the time required for each iteration once rework is included, two elements are considered: learning effect and degree of connectivity. These elements determine the duration of a particular activity when it is not being done for the first time. The learning effect considers the fact that people who complete an activity gain experience through successive iterations. Learning reduces the time required to complete certain task or increases the performance of workers. Degree of connectivity is related with the level of dependency among successive tasks. A higher dependency means that more time is necessary to propagate the change thought all dependent activities. The parameters $\varphi_{ij}$ ($0 \leq \varphi_{ij} \leq 1$) and $\gamma_{ij}$ ($\gamma_{ij} \geq 0$) capture both the degree of connectivity and learning effect for task i from project j, respectively. The resources necessary after successive iterations can be calculated with the following formula:

$$r_{ijm}^* = \left[ t_{ijm}^0 + \left( t_{ijm} - t_{ijm}^0 \right) \times e^{-\gamma_{ij}(k-1)} \right] \times \varphi_{ij}$$

where $t_{ijm}$ is the time required to finish task i from project j in mode m when is done for the first time. $t_{ijm}^0$ is the minimum time possible and k represents the number of iterations. The parameter that represents the degree of connectivity is calculated as follows:

$$\gamma_{ij} = \sum_{i=1}^{N} l_{ij}$$

where $l_{ij}$ is the dependency intensity associated with task i from project j. $r_{ijm}^*$ are the resources required for activity i from project j in mode m. As the number of iterations increase, the factor $e^{-\gamma_{ij}(k-1)}$ becomes closer to zero and the necessary resources decrease until reaching the minimum value ($t_{ijm}^0$).

7.15 Task Duration Depending On Resource Allocation

To finish a project before the stipulated deadline, it is common in project management practice and has been a particular focus of many project dynamic models, to take action so as to reduce the gap that could potentially exist. Three common actions can be taken to correct a situation in which project managers forecast that they will miss a deadline [Lyneais & Ford, 2007]: (1) use additional workers, (2) do overtime and (3) work faster. Thus, as it is indicated in Figure 29, any
of the mentioned actions could be considered in order to reduce the overall project duration. However, those proposed actions have its associated ripple effects because doing overtime would directly increase the cost and in time could lead to fatigue, that increase the errors and consequently affects rework. Working faster would directly increase the error rate, adding rework. Finally, using more available resources would leave less of them to use in other tasks.

By recombining different renewable resources at disposal, it is possible to reach combinations that surpass the possible mode variations. Therefore, it is required to divide the class of renewable resources into two separate groups. On the one hand, there are resources that follow the discrete mode division. For instance, this category is composed by those resources that may not be used by two tasks that are active. On the other hand, resources like manpower could be further divided, as long as they fulfill certain constraints, and reassigned to two or more tasks at the same time. Nonetheless, there are some assumptions that should be considered:

- It is necessary to provide either the normal resource-hours for each task or the duration (not both).
- If the required resource-hours are provided for a particular task it is assumed to be fixed. Therefore, until the required resource-hours is not met, the activity won’t be finished.
- The resources allocated to certain activity are based on previous experiences, but they might vary even while the activity is being done. Thus, the duration of the task is variable.

![Figure 29 - Actions to Control Project Deadline](image)

7.16 Including Rework in the GA-ABC Algorithm

Once rework is included in the model and considering the fact that activities might modify its duration depending on the resources that were assigned to them, the optimal scheduling depends on what happens in reality. Probably, it will not be possible to optimize for every
probabilistic scenario. The optimal solution will be different in each case and reality will collapse into one of them, but it is not possible to know in which particular one upfront. However, it is possible to find resource allocations that are robust, leading to consistently better results than other resource allocations.

It is used simulation for evaluating how the fitness function is impacted by the different scenarios that depend on how rework activities turn out to be distributed as well as the resource allocation. Stochastic simulation provides a varying outcome because the fitness value may change with identical chromosomes due to the differences in rework as well as in duration. Different fitness values will be related with one chromosome where it is possible to calculate the expected value and the variation from that expected value leaving the outcome in a location within the graph that relates expected return with variation.
The characteristics of trade-spaces were already covered in section 2, but the main feature is that it is possible to identify the best tradeoffs, which consist of elements that are superior to other in at least one dimension while inferior with regard to others. The concept of \( \varepsilon \)-dominance consist on dividing the objective space into discrete hyper-boxes, as it is shown in Figure 31. This procedure allows to reduce cases where insignificant differences in the fitness value lead to situations in which elements near the Pareto-frontier are categorized as dominated solutions. Those elements that are located in a narrow swatch near the frontier are guaranteed to be incorporated into the Pareto-front with this procedure.

It should be noted that this approach has a major downside that resides in the fact that evaluating the fitness value of each chromosome though simulation could result in a time consuming process. However, the \( \varepsilon \)-dominance criterion leads to a reduced number of different outcomes which limits the varying results of the simulation.
8. Results

8.1 Introduction

This section is devoted to showing the results obtained while running the proposed methods for the base and improved models. The methods are compared in the following manners:

- **Using standard benchmark instances**: the PSPLIB contains diverse data sets for several types of resource constrained project scheduling problems, together with the best found solutions for each instance.
- **Comparing the methods proposed for the base model against the improved one**: as the assumptions of the base model are loosen the necessity of a new method will become patent.
- **Comparing with results obtained using other methods from literature**: when there are results available regarding the performance of different methods using similar models.

Once defined how the methods are going to be compared, it is necessary to determine what are the dimensions for comparison and how are going to be measured. The goal will be to establish the usefulness of a particular method when a particular schedule of activities, resource allocation and portfolio of projects is proposed.

The usefulness of the method will depend on two factors:

- **Credibility**: according to the definition adopted in this document, a method is credible if it can achieve a feasible solution, that is “good” and within a reasonable time. Therefore, credibility depends upon:
  - Feasibility: no constraint should be broken
  - Computational Time: reasonable when compared to other methods
  - Effectivity: in searching the space of solutions for those with “good” fitness values

- **Inclusiveness**: it depends on the capabilities of absorbing the relaxed assumptions of the models. Those points were covered in the previous section and they have to do with the characteristics that the proposed method should be capable of incorporating, such us:
  - Multiple target dates: different projects may have diverse target dates and associated return
  - Technology Interaction: consider the relationship among projects
  - Portfolio selection: capacity of selecting those projects that will maximize return
  - Multiple resource types: consider both renewable and non renewable resources that might be either local or global
  - Multiple modes: duration of activities can vary depending on resource allocation
- Differences in performance: resources may possess different suitability for doing a particular task
- Rework: certain activities may have to be redone and resource allocation may affect that probability

8.2 Base Model

The base model shares the assumptions of the static RCMPSP, which were explained in section 3. The proposed method to deal with it is based on parallel SGS combined with different PRs and TBs. The details of the method were covered in section 6. The current sub-section will show the results obtained when the afore-mentioned method was applied. Initially, the method was utilized to solving the RCPSP, which is similar to the multi-project case. However, the Project Scheduling Problem Library (PSPLIB) contains numerous instances for the case with single projects, providing a standardized benchmark for comparing the credibility of the method. The instances of the PSPLIB are available on a public web site http://www.om-db.wi.tum.de/psplib/ (last verification of address: 10-10-2016) which contains different problem sets for various types of resource constrained project scheduling problems. All the instances were generated using the project generator ProGen. In case of the RCMPSP, the instances available in the MPSPLIB will be used. These test problems were generated by [Homberger, 2007] and can be used as a benchmark for problems with multiple projects. All the test problems consist of 2, 5, 10 or 20 project instances, composed of 30, 90 or 120 activities each, obtained from the PSPLIB that were combined into diverse portfolios which are publicly available, together with the best found solutions, from the web site http://mpsplib.com (last corroboration of address: 10-10-2016). More details of the problem sets are available in the appendix (section 11.1).

Several data sets were selected with diverse number of activities. The study included instances of 30, 60, 90 and 120 activities. Nevertheless, in the following paragraphs it will be shown the cases of instances j30_2_2 and J30_45_8 (composed of 30 tasks), and J120_32_4 (composed of 120 activities). For the rest of the instances, an overall summary will be presented categorizing the result according to three important characteristics: number of activities, network complexity and utilization factor.

**Number of activities:** refer to the amount of non-dummy activities that compose the project. In the case of a portfolio of projects corresponds to the sum of all non-dummy tasks for the whole portfolio (n*).

**Network complexity:** there are many form of measuring the complexity of a network. In this document, it will be used the measure proposed in [Browning & Yassine, 2010]. If A’ is the
number of non-redundant arcs which indicate precedence relationships and \( N \) is the number of tasks, then the complexity \( C \) is calculated as follows:

\[
C = \frac{4A' - 4N - 4}{(N - 2)^2}
\]

This type of complexity measure only considers precedence relationships and number of activities without taking into account any topological characteristics of the network. However, the advantage is that it is intuitive and normalizes the result that could be between 0 and 1. High-complexity networks will be more precedence-constraint, having values of \( C \) closer to 1. If the value of \( C \) is in the segment: (1) \([0; 0.2)\) it will be categorized as “low”, (2) \([0.2; 0.5)\) it will be considered as “medium” and (3) \([0.5; 1]\) is “high”.

**Utilization factor**: is essentially a ratio of the amount of required (renewable) resources to total available resources. In this document, it is calculated as the average utilization factor (AUF) proposed by [Kurtulus & Davis, 1982] where the project, or the portfolio of projects, is divided into \( S \) intervals. The amount of resources \( k \) necessary in any given interval \( S \) is calculated as follows:

\[
W_{sk} = \sum_{t=T_1}^{T_2} \sum_{j=1}^{J} \sum_{i=1}^{N} r_{ijk} X_{ijt}
\]

where \( T_1 \) and \( T_2 \) are the extremes of each interval \( S \). The amount of resource \( k \) required by activity \( i \) of project \( j \) (in case of a portfolio) is \( r_{ijk} \). \( X_{ijt} \) is 1 if activity \( i \) of project \( j \) is being performed at time \( t \), and is 0 otherwise. The AUF for each resource \( k \) is calculated as follows:

\[
AUF_k = \frac{1}{S} \sum_{s=1}^{S} \frac{W_{sk}}{SR_k}
\]

The overall AUF when there are more than 1 resource involved is equal to the maximum \( AUF_k \). Whenever the \( AUF>1 \) it indicates that on average there is at least one resource that is constrained over the course of the project (or portfolio of projects). Conversely, if \( AUF<1 \) it means that on average the total available quantity of each resources type is more than sufficient.

In the paragraphs below, it is showed the application of the method based on PR and TB. The method is shown for solving three instances (J30_2_2, J30_45_8 and J120_32_4) of the PSPLIB.
The critical path (CP) is calculated with resource unconstrained. The “best solution” indicates the best solution achieved considering limited resources, supposing that the objective is the minimization of the total make-span and complying with the assumptions of the RCPSP.

<table>
<thead>
<tr>
<th>J30_2_2</th>
<th>J30_45_8</th>
<th>J120_32_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Activities</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Complexity</td>
<td>0.03 (low)</td>
<td>0.2 (med)</td>
</tr>
<tr>
<td>Utilization Factor</td>
<td>0.6</td>
<td>1.1</td>
</tr>
<tr>
<td>CP</td>
<td>46</td>
<td>65</td>
</tr>
<tr>
<td>Best Solution</td>
<td>51</td>
<td>94</td>
</tr>
</tbody>
</table>

Table 5 – Characteristics of Three of the Selected Instances

By applying the method based on PR and TB, the results indicate that for instance J30_2_2, it reaches the best possible solution using many different PR which makes unnecessary to use TB. On the other hand, for instance J120_32_4 the best PR is notoriously deviated from the best possible solution. The deviation is measured with the ensuing formula:

\[
\text{Deviation} = \frac{(\text{Solution}_{\text{Found}} - \text{Solution}_{\text{BestPossible}})}{\text{Solution}_{\text{BestPossible}}}
\]

Table 6 shows, for the three selected instances, the deviation from the best possible solution. In case of the instance J120_32_4, the best PR turned out to be MAXSP. For the instance J30_45_8 there are two PR that produced the same result: EDDF and TWK-LST. Finally, J30_2_2 had 13 PR that reached the best possible solution, such as: FCFS, MINSLK, EDDF, MINLFT, MAXSP, TWK-LST, MCS and others.

<table>
<thead>
<tr>
<th>J30_2_2</th>
<th>J30_45_8</th>
<th>J120_32_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best Solution</td>
<td>51</td>
<td>94</td>
</tr>
<tr>
<td>Best Found Solution</td>
<td>51</td>
<td>100</td>
</tr>
<tr>
<td>Deviation</td>
<td>0%</td>
<td>4%</td>
</tr>
<tr>
<td>Best PR</td>
<td>FCFS / MINSLK / etc.</td>
<td>EDDF / TWK-LST</td>
</tr>
</tbody>
</table>

Table 6 – Found Solutions Applying the Method Based on PR

One question that is important to respond is how credible the results are. From the small sample provided in Table 6, it is noticeable that deviation may range extensively. Therefore, it will be analyzed if the deviation is associated with some project characteristics. The different instances that were studied had been categorized according to the number of activities, degree of complexity and utilization factor. The procedure was similar to the one explained for the
previous three cases. For each one of them the deviation was calculated and the results are shown in the following paragraphs.

When the number of activities is relatively small (30), the average deviation from the best possible solution is slightly more than 7%. However, as the number of activities increases, the deviation augments as well, reaching an average value of 24% when the number of activities is 120 (refer to Figure 32). The reason is that as the number of activities rises, the deviation propagates and exacerbates.

In case of complexity, when it is low (C<0.2) the average deviation is about 14%; whereas if the complexity is high (C≥0.5) the average deviation descends to 7%. Figure 33 displays this result which might seem counterintuitive because it appears that a higher complexity should be correlated with more difficulties in reaching “good” solutions. However, the measure of complexity C as it is defined in the current sub-section only takes into account precedence
relationship. Therefore, high complexity is associated with elevated precedence relationships, which means that the possible options of tasks that could potentially be selected in each period is limited. Conversely, as the precedence relationships are low; there are, on average, more activities than could potentially be selected per time period producing more differences in the total make-span of the project depending on the order in which tasks are effectively selected.

Finally, the average utilization factor is a measure of the average tightness of the constraints on each of the required resources. When the ratio is equal to one, it means that on average the resources required are similar as the available ones. As it can be seen in Figure 34, when the resources available are more than necessary (AUF<1) the deviation is low and increases as the resource constraints become more apparent because some more activities should be delayed because resources become scarcer. The increment is continuous until it reaches a peak between \([1.3,1.5]\) where the deviation is approximately 22%. If the utilization factor keeps increasing, the deviation descends because when resource constraints are too tight there are fewer activities that could be performed at the same time.

![Figure 34 - Variation of Deviation with the Utilization Factor](image)

Up to this point, it has been shown the potential of the method applied to the case with one project where it has been indicated the characteristics of projects and the associated performance in each case. When the method is applied to the multi-project case, results follow similar pattern depending on the three characteristics covered: number of activities, network complexity and utilization factor. It will be considered the case of the following portfolio of projects:

<table>
<thead>
<tr>
<th>Portfolio Instances</th>
<th>Projects</th>
<th>Activities</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
</tr>
</thead>
<tbody>
<tr>
<td>J30, 45</td>
<td>2</td>
<td>60</td>
<td>13</td>
<td>11</td>
<td>13</td>
<td>16</td>
</tr>
</tbody>
</table>

*Table 7 - Example of a Project Portfolio*
The characteristics of both instances that compose the portfolio are shown in Table 5. If it is considered that the objective is to reduce the total make-span of the portfolio, which is measured from the earliest activity to the end of the last task (independently of the project that belongs to). In that case, the PR that yields the best result is maximum total successors (MTS). However, in [Sonmez & Uysal, 2015] it is shown that applying backward-forward hybrid genetic algorithm for solving the same portfolio, the result is better than any achievable with the proposed priority rules.

Nonetheless, it is important to notice that having minimization of total portfolio delay as an objective function has important limitations in the case of technology infusion projects because time-to-market as well as the maturity level of each technology and their interactions, makes necessary to consider separate target dates for each project. It has been emphasized in section 2 the importance of considering different target dates for each project. Consequently, it will be used the proposed objective function that was covered in Section 6.1. Thus, the table below displays diverse variations of the parameters of the objective function and it will be shown how it affects the results of the proposed method.

<table>
<thead>
<tr>
<th>Scenario 1:</th>
<th>Scenario 2:</th>
<th>Scenario 3:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project J30_2_2:</strong></td>
<td><strong>Project J30_2_2:</strong></td>
<td><strong>Project J30_2_2:</strong></td>
</tr>
<tr>
<td>Target Date: 120 days</td>
<td>Target Date: 90 days</td>
<td>Target Date: 140 days</td>
</tr>
<tr>
<td>MDL: 200 days</td>
<td>MDL: 200 days</td>
<td>MDL: 200 days</td>
</tr>
<tr>
<td>Curve Type: 1</td>
<td>Curve Type: 1</td>
<td>Curve Type: 3</td>
</tr>
<tr>
<td>E[(\text{ANPV}): 1.000</td>
<td>E[(\text{NPV}): 1.500</td>
<td>E[(\text{NPV}): 1.500</td>
</tr>
<tr>
<td><strong>Project J30_45_8:</strong></td>
<td><strong>Project J30_45_8:</strong></td>
<td><strong>Project J30_45_8:</strong></td>
</tr>
<tr>
<td>Target Date: 120 days</td>
<td>Target Date: 140 days</td>
<td>Target Date: 140 days</td>
</tr>
<tr>
<td>MDL: 200 days</td>
<td>MDL: 200 days</td>
<td>MDL: 200 days</td>
</tr>
<tr>
<td>Curve Type: 1</td>
<td>Curve Type: 1</td>
<td>Curve Type: 3</td>
</tr>
<tr>
<td>E[(\text{ANPV}): 1.500</td>
<td>E[(\text{NPV}): 1.500</td>
<td>E[(\text{NPV}): 1.500</td>
</tr>
</tbody>
</table>

**Figure 35 – Scenarios Applied to Portfolio 1**

Considering scenario 1, the maximum return possible is $2,500 if both projects end within 120 days. The result will decrease if any of the projects finish beyond the target date. So far, project cancellation is not allowed. After running the method, the result indicate that the best PR turns out to be MTS, giving a total return of $2,388. Project J30_2_2 finishes within the 120 days, but projects J30_45_8 ends after 129 days. Regarding scenario 2, the maximum possible return is $3,000 and after applying the method 5 different PRs reach that sum, none of them is MTS though. The 5 PRs are: MAXTWK, EDDF, MINLFT, TWK-LST and TWK-EST. Finally, for scenario 3 the maximum return is again $3,000. The main difference in this scenario compared to the preceding ones is that scenario 1 and 2 only penalized tardiness, whereas scenario 3 castigate
both tardiness and earliness. The PR that attains the best result is LCFS yielding $2.765, where project J30_2_2 has an estimated end date in 111 days and project J30_45_8 finishes exactly in 140 days.

In general, the results for the base method reveal that PR performance depends on the chosen objective function, or in this case the parameters selected for the objective function. In the paragraph above it is shown the example of three scenarios in which by modifying parameters of the objective function slightly, the PR that provides the best outcome vary in each case. In addition, as the parallel SGS that is used to create the project plan generates only non-delay schedules (for an explanation of the characteristics of different SGS, refer to section 7.9) and there are cases in which earliness is penalized, it could be beneficial delaying some activities in order to finish the project as close as possible from the target date. Therefore, the method should include the possibility of delaying tasks. Finally, it has been shown that solutions obtained though PRs deviate from the best possible schedules as: (1) number of activities increase, (2) complexity decreases and (3) utilization factor gets closer to the interval [1.3-1.5]. The fact that in this document the focus is a project portfolio makes probable that the number of activities will be elevated. Hence, when the objective is minimization of make-span, best possible solutions are usually not attained with PRs.

8.3 Improved Model

From the previous sub-section, it became patent that it was necessary to create a more effective method for finding solutions to the RCMPSP. The details of the method were covered in section 7. Basically consists of a hybrid meta-heuristic method combining GA and ABC, and from now on it will referred to as “improved method”. Initially, it was compared the improved method for the same 40 instances that were considered for the method based on PR. The results indicate that after 5000 iterations the improved method is able to find the best possible solutions in each case. The computational time never went over 2 seconds. In Table 8 it is possible to appreciate the outcome obtained for the instances shown in the previous sub-section. As it is noted, the deviation from the best possible solution result in any case 0%.

<table>
<thead>
<tr>
<th></th>
<th>J30_2_2</th>
<th>J30_45_8</th>
<th>J120_32_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best Solution</td>
<td>51</td>
<td>94</td>
<td>136</td>
</tr>
<tr>
<td>Best Found Solution</td>
<td>51</td>
<td>94</td>
<td>136</td>
</tr>
<tr>
<td>Deviation</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Iterations</td>
<td>5000</td>
<td>5000</td>
<td>5000</td>
</tr>
<tr>
<td>Computational Time (sec)</td>
<td>0.6</td>
<td>0.9</td>
<td>1.6</td>
</tr>
</tbody>
</table>

Table 8 – Found Solutions Applying the Meta-heuristic Method
Regarding multi-projects, in a study performed by [Chen & Shahandashti, 2009] there are two multi-project instances, where one corresponds to a test portfolio and the other to a real portfolio. In both portfolios it has been compared the performance of different meta-heuristic methods:

- A genetic algorithm
- Simulated annealing
- Hybrid GA – SA algorithm
- Modified simulated annealing (arithmetically improved)
- Modified simulated annealing (logarithmically improved)

In the current document it is compared the performance obtained with the improved method against the mentioned meta-heuristics only for the test portfolio, which is composed of 74 activities and 2 renewable resources.

<table>
<thead>
<tr>
<th>Method</th>
<th>Average</th>
<th>Runs</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA</td>
<td>135.5</td>
<td>10</td>
</tr>
<tr>
<td>SA</td>
<td>135.4</td>
<td>10</td>
</tr>
<tr>
<td>GA+SA</td>
<td>134.5</td>
<td>10</td>
</tr>
<tr>
<td>MSA(1)</td>
<td>134.2</td>
<td>10</td>
</tr>
<tr>
<td>MSA(2)</td>
<td>133</td>
<td>10</td>
</tr>
<tr>
<td>Base Model</td>
<td>154.7</td>
<td>10</td>
</tr>
<tr>
<td>Improved Model</td>
<td>126.1</td>
<td>10</td>
</tr>
</tbody>
</table>

*Table 9 - Numerical Comparison of Methods*

In Table 9 are shown the numerical result of applying the improved method covered in section 7 to the test portfolio. As it can be appreciated, the results indicate that after 10 runs outperforms on average the other five methods. The method was stopped in each run after 5,000 iterations, which took an average of 16 seconds each. The Figure 36 illustrates what the numerical results.

*Figure 36 - Comparison of Methods*
Up to now, it has been proven the credibility of the improved method in solving single project problems as well as multi-projects. It has been defined that credibility depends on three factors: feasibility, efficiency and computational time. The previous results show the capacity of the improved method to find the best possible solutions (feasibility and efficiency) within a good computational time. Now, it will be shown how inclusive the improved method is. Section 7.5 indicates the characteristics of incorporating different modes per activity, where duration of tasks vary depending on the resources allocated and at the same time adds a new type of resource (non-renewable). In order to test the performance of the algorithm, it have been used the multi-mode problems contained in the PSPLIB (already mentioned in section 8.2). Specifically, it was used instances belonging to the data sets J10, J14, J16 and J20 containing 10, 14, 16 and 20 non-dummy activities respectively. The details of the specific instances considered from each set are indicated in section 8.2. The results show that, the optimal solution is known with the MRCPSP assumptions and supposing that the objective is the reduction of the make-span. Every instance is solved with a computational effort of 1000, 3000 and 5000 iterations.

<table>
<thead>
<tr>
<th>Instance</th>
<th>Iterations</th>
<th>Avg. Deviation (%)</th>
<th>Avg. Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>J10</td>
<td>1000</td>
<td>0.36</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>3000</td>
<td>0.15</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>5000</td>
<td>0.09</td>
<td>0.7</td>
</tr>
<tr>
<td>J14</td>
<td>1000</td>
<td>1.23</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>3000</td>
<td>0.64</td>
<td>0.6</td>
</tr>
<tr>
<td></td>
<td>5000</td>
<td>0.42</td>
<td>0.9</td>
</tr>
<tr>
<td>J16</td>
<td>1000</td>
<td>1.42</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>3000</td>
<td>0.79</td>
<td>0.6</td>
</tr>
<tr>
<td></td>
<td>5000</td>
<td>0.60</td>
<td>0.9</td>
</tr>
<tr>
<td>J20</td>
<td>1000</td>
<td>2.52</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>3000</td>
<td>1.34</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td>5000</td>
<td>1.04</td>
<td>1.1</td>
</tr>
</tbody>
</table>

*Table 10 – Performance with Multiple Modes*

It can be appreciated in Table 10, the average deviation from the best possible solutions as well as the average computational time required depending on the number of generations (1000, 3000 or 5000). A detailed analysis has been performed comparing the results of the improved method against other algorithms available in literature, where data corresponding to the other methods was taken from a research made by [Lova et al., 2009]. The results show that,
although it does not outperform every algorithm, it attains a very good performance in every case.

<table>
<thead>
<tr>
<th>Improved Method</th>
<th>J10</th>
<th>J14</th>
<th>J16</th>
<th>J20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lova et al. 2009</td>
<td>0.09</td>
<td>0.42</td>
<td>0.60</td>
<td>1.04</td>
</tr>
<tr>
<td>Alcaraz et al. 2003</td>
<td>0.06</td>
<td>0.32</td>
<td>0.44</td>
<td>0.87</td>
</tr>
<tr>
<td>Jozeďowska et al. 2001</td>
<td>1.16</td>
<td>2.60</td>
<td>4.07</td>
<td>6.74</td>
</tr>
</tbody>
</table>

Table 11 – Comparison with Other Methods After 5000 Generations

The instances mentioned in Table 11 could be considered as small-sized projects. However, once different instances are considered together into a single portfolio it becomes a challenging MRCMPSP. In the appendix, specifically section 11.2, there is a list with many instances that are openly available and could be used as a benchmark for the MRCMPSP. It will be considered the example of instance A-5 for the remaining of the current sub-section, a summary of its characteristics as well as the parameters used for the objective function can be found in section 11.3.

It has been indicated in section 7.12 a procedure named MNR that leads to the creation of feasible solutions, in terms of non-renewable resources, for the initial population. At first glance, it can be considered a time consuming process. Therefore, it has been studied the influence of applying the mentioned procedure against leaving the unfeasible solutions at the beginning so that the objective function can penalize them if they are not feasible.

![Figure 37 - Comparison Of Performance Using MNR Procedure vs. Without It](image)

It is expected that applying the MNR procedure will consume time upfront, since it has to search for feasible solutions. However, it might seed up the process of finding good solutions
after many generations. The experiment performed consisted on taking different portfolios and compare the results obtained employing the MNR procedure versus not applying it. For each portfolio, 10 runs were performed and the best solution was recorded after 1, 3 and 5 seconds. The results for the portfolio explained in section 11.4 is shown in Figure 37, where it can be appreciated that applying the MNR procedure leads consistently to better solutions (blue line) that without it (red line). Besides, it is noticeable that after one second the difference, although consistent, it is not very distinguishable. Nevertheless, as the time increases (3 and 5 seconds) the difference becomes more apparent. The time that takes finding feasible solutions, in terms of non-renewable resources, is outweighed with the better quality feasible solutions that are found after crossover and mutation operators are applied in successive generations. The reason of the notorious advantage of the MNR procedure is that, in practice, a feasible non-renewable mode assignment was obtained in approximately 92% of individuals within four attempts for the portfolios that were tested.

In addition, section 7.12 covered the concept of hybridization between evolutionary algorithm and swarm intelligence. As was indicated in section 7.13, simple GA usually suffer from the problem of rapid convergence, a fact that signify that without mechanisms that assure a varied individuals in each generation, it could lead to exploring only a sub-section of the whole solution space potentially leading to a local optimum instead of the global one. The mechanism that was proposed, in this document, to address the suggested issue was the hybridization with ABC, which provides very good exploratory characteristics. In order to test the suitability of the hybridization, an experiment was performed in which the results for a simple GA and the GA-ABC algorithm are compared for different portfolios. In each case, 10 runs were executed for each portfolio and the best feasible solution was documented.

![Figure 38 – Comparison of Performance Simple GA vs. Hybrid GA-ABC](image-url)
The results, for the same portfolio utilized in the previous paragraph, are shown in Figure 38, where the hybrid GA-ABC (blue line) proved to be slightly better than the simple GA (red line) after 1000, 3000 and 5000 generations. Nonetheless, there is another conclusion that could be drawn from the graph which corresponds to the variability of the results during different runs. It is notorious that the areas corresponding to the red ellipses is greater than blue ones. This means that simple GA might yield quite different results from one experiment to the other, where sometimes it could lead to solutions that are far from optimal. Furthermore, in some cases it stops making improvements displaying premature convergence. On the other hand, the hybrid algorithm has mechanisms than ensure the variability of the population which guarantees that the results are better and do not notoriously change from one run to another. In addition, the hybrid approach shows continuous improvements indicating that the algorithm does not converge prematurely.

Until now, the improved method has been tested for situations without rework. However, it has been indicated that one of the factors which affects the measure of inclusiveness is including the possibility of redoing activities as well as relating performance with the amount rework that could be done. The portfolio detailed in section 11.4 (appendix), was tested assuming the feedback arcs indicated in section 11.5 (appendix).

Figure 39 displays the expected return when rework is not considered (first column) and adding the average effect of it (third column). The best schedule and resource allocation achieved by applying the improved model to the example developed in section 11.4 was simulated 5000 times with the rework conditions established in section 11.5. It is noticeable, for the example considered, the impact that rework has on the duration of each project and consequently in the expected return, which was 61% lower. Furthermore, if one of the projects is cancelled (j20_11_8) the expected return increases a 9%. Even though it is completely lost the return from the project that was cancelled, it liberates global resources that could be allocated in the remaining projects achieving a greater overall return. The effect of rework was tested with different portfolios, but it should be mentioned that the probabilities of redoing activities as well as the number of feedback arcs can dramatically affect the duration of projects and the expected return.
For the portfolio explained in section 11.4, when iterations due to rework are included (section 11.5), it is interesting to verify if the best solution when rework is zero still stands. One caveat, is that feedback arcs represent the possibility but not the certainty of redoing certain task. Therefore, experimenting with the same solution might yield diverse results over time. Thus, the results will be analyzed in a trade-space where the x-axis represents the average value obtained though many runs and the y-axis depicts the variation over the expected value. Non-dominated solutions are part of the Pareto Frontier, which means that there might be more than one solution that is equally suitable and the final decision depends on a trade-off between expected return and its variation. The goal is to find robust activity schedules and resource allocations that yield a great expected value while keeping the variation at a minimum. The best case scenario (utopia point), would be to achieve maximum expected return with no variation. The question that arises is if the best solution when rework is not considered belongs to the Pareto Frontier when it is incorporated. Figure 40 shows the trade-space with the feasible solutions of the last generation attained when the improved method was executed. Figure 40 (a) shows a blue curve that was drawn separating those solutions that were worse in both dimensions from the ones that provided better outcomes. The result was that 85% of the feasible solutions belonging to the last generation are outperformed by the best solution found. Nevertheless, it does not belong to the Pareto Frontier, as it can be appreciated in Figure 41 where the best solution with zero rework is highlighted with a green circle and it can be found blue dots that yield better results in both dimensions.

Furthermore, once the improved method is executed considering iterations, the Pareto Frontier is shown in red in Figure 40 (b) and Figure 41. The former graph makes clear that the new Pareto Frontier is closer to the utopia point than the previous frontier; whereas, the latter diagram shows the comparison between the best solution without considering rework and the Pareto Frontier (red line) formed by the best solutions when iterations are included. It stands out, from Figure 41, that it is possible to achieve the same expected value with approximately 54% less variation, or the same variation but with an increase in the expected value of about 96.
25%. This result signifies that the best solutions found considering no rework might not be part of the Pareto Frontier when the effect of iterations are included. In addition, the same analysis was performed in 29 other portfolios indicated in section 11.2 and the outcome was that in only 12% of the cases the best solution without rework ended up being part of the Pareto Frontier once iterations were incorporated. Consequently, it is important to plan for rework because the best solution will probably be different.

In conclusion, the current section initiated by specifying the manner it was going to be used for measuring the usefulness of each method, which basically depends on two dimensions: credibility and inclusiveness. Section 8.2 explained the results obtained applying the base method, which was created founded on a pretty rigid model (static RCMPSP). Once the assumptions of the mentioned model were relaxed another method was required to deal with it. Thereby, the improved method was proposed and it was tested in section 8.3. Its outcomes were compared against the base method and results obtained from other algorithms available in literature. It is shown that the new method is able to find feasible, optimal or near optimal solutions in a variety of situations within acceptable computational time. To have a common benchmark for comparison, results were compared using publicly available instances. It was proved the credibility of the improved method for the single project, multi-project and multi-mode cases. Finally, once it was proved it could handle a variety of situations yielding credible results, the method was tested for conditions that are not commonly treated in literature, like
the addition of rework, in order to prove the inclusiveness of the model as well as the ability of
the improved method to deal with it. It is exposed the impact of adding rework, which produce
a notorious increase in multi-project duration, making important to plan for it because it will
probably affect the most effective manners to allocate resources and schedule activities.
9. Conclusion

Most of the time companies do not provide value to customers by means of breakthrough improvements, that result in entirely new products or systems. The most common manner to add value to customers is through continuous improvements, of small to medium impact. New technologies are often at the core of new products. Current technology infusion frameworks assess the impact of introducing new technologies into existing products and results in a selection of one or more options. However, the selection of technologies is not based on operational issues, potential scheduling plan and required resource allocation for an effective infusion. At the same time, the inclusion of a particular project into the portfolio should be decided, not only in terms of value and profitability, but also considering schedule and operational information.

This research gap was the motivation for the current document: to find the most effective resource allocation, activity scheduling and project portfolio selection for a scenario with the characteristics described in the previous paragraph.

Initially, for this purpose, it was proposed a heuristic method that uses parallel SGS and a combination of priority rules and tie-breakers (PR-TB). However, numerical results indicate that the base method, fail to provide flexible schedules due to the non-delay nature of the generated schedules. At the same time, results indicate that the ability to reach optimal solutions is restricted because of limited capabilities for searching the space of solutions, making it not effective enough, and thereby affecting its usefulness. The capacity to achieve optimal solutions decreases as: (1) number of activities increase, (2) complexity decreases and (3) utilization factor gets closer to the interval [1.3-1.5].

Later, it is proposed an original hybrid meta-heuristic method based on swarm intelligence and evolutionary algorithm for effectively schedule activities, allocate resources and manage the project portfolio. Regarding the performance of this new method, numerical results show that high-quality schedules are generated and resources are allocated efficiently.

A series of mechanisms were included with the goal of improving the searching capabilities and therefore improving the quality of solutions. Adding a preprocessing phase to ensure the feasibility of the initial population allows to start with a prime generation of solutions that soundly improves schedule quality as well as resource allocation. Additionally, the inclusion of a mechanism such as the population affinity index (PAI) or the directed exchange of individuals from the population, assures the necessary diversity and numerical results proved that it helps in avoiding rapid convergence as well as improving the final solutions.

Existing meta-heuristic approaches rarely incorporate rework into the model. The improved method accounts for the interaction among the projects that comprise the portfolio as well as
the possibility of rework. To evaluate the different scenarios that arise depending on how feedback arcs are distributed and its associated probability, it is used simulation resulting in a time-consuming process. However, the ε-dominance criterion leads to a reduced number of different outcomes which limits the varying results of the simulation. Although it increases significantly the complexity of the problem, results show it should not be neglected. The proposed method allows to locate the set of best tradeoffs in terms of expected return and risk associated with it. The results demonstrate that adding rework, and relating the performance of resources with some variation in the probability of redoing tasks, produces a notorious increase in multi-project duration that depends on the probability of rework, amount of feedback arcs and how performance affects activity duration.

In addition, results reveal that planning for rework will probably affect the most effective manners to allocate resources and schedule activities. Therefore, project managers should include it into the planning phase. Furthermore, the inclusion of rework might modify the composition of the portfolio that achieves the higher return, as it was shown with a numerical example.
10. Limitations and future work

There are several directions for future work that could be followed. One possibility arises because technologies are rarely ready for infusion. It is probable that they require a certain degree of maturation before incorporating them into the host product or system. However, if the required readiness level (TRL) is not achieved, it might not be beneficial to infuse them. Therefore, reaching different TRLs might be associated with diverse returns that could affect the attractiveness of including a particular technology. The current method might be expanded by investigating how the TRL might affect portfolio selection.

Another possibility related to technology infusion is when the interaction happens among more than two technologies. For instance, if the inclusion of two independent technologies might affect the inclusion of a third one, when the two technologies taken independently do not influence that third one. The current method takes into account relation between pair of technologies but do not consider more complex interactions. Investigating complex technology interaction in the performance space could be an interesting way to extend the current approach.

In terms of the methodology itself, many studies in literature use forward backward improvements (FBI) in a final step, to improve the attained solutions. This simple procedure usually allows to achieve better solutions when the objective is to minimize TMS or TPD. However, it might be interesting to analyze its behavior when the objective function is different. Other possible expansion could be adding a mechanism that considers memory, preventing the exploration of solutions that were visited in the past. The caveat is that as memory increases the computational effort grows. In the improved method, each generation is created based on the information of the previous one. It could be interesting to explore how the results are affected by a mechanism that possibilities the inclusion of information from more than one generation back.

Another topic of interest is to study the dynamic arrival of project to the portfolio in real time. The current model assumes a static scenario, where it is necessary to decide upfront the projects that will conform the portfolio. It has been shown that in certain occasions cancelling a project might benefit the overall return of the portfolio. Therefore, it could be important to establish how each incoming project could potentially affect the schedule, project selection and resource allocation in an existing portfolio.

The improved model considers different types of resources that might be local, if they are used only by one project; or global, if they are shared among every project. The assignation of local resources is assumed to be pre-established. However, local resources might also be distributed based on project characteristics as well as resources characteristics. This initial local resource
allocation could have a great impact in the final allotment of shared resources, making it a relevant topic to consider and potentially incorporate into the model.

Other line of improvement consists in the incorporation of resource transfer time and the related cost for project execution. Transference of resources may take time mainly because when certain resources are physically moved from one location to another, there is time associated with it. Also, when resources need to be adjusted before starting another activity, it takes time. People might require certain training before initiating another activity or machines might need certain set-up time. The literature available on resource transfer time is scarce but studies have shown that it affects both project delay and portfolio overall duration [Adhau et al, 2013] [Krüger and Scholl, 2009]. It could be interest to incorporate this characteristic into the model to verify how solutions are affected.

Finally, it is necessary to mention that the proposed methods were tested with different benchmark instances available from public sources as well as in literature. Numerical results showed that the improved method yielded credible outcomes, while it was able to successfully incorporate: multiple projects with diverse target dates, consider technology interaction, portfolio selection, multiple types of resources, account for differences in performance and rework. However, it has not been proved with different real life scenarios. It could be interesting to apply it in a flexible environment related to technology infusion and validate the proposed method.
11. Appendix

11.1 Appendix I: Standard Data Sets for Multiple Projects

There are 140 problem instances for multiple projects on the public web site http://www.mpslib.com (last verification of address: 10-10-2016). The details of the available instances are summarized in the following table:

<table>
<thead>
<tr>
<th>Subset</th>
<th># Instances</th>
<th># Projects</th>
<th># Activities</th>
<th>Global Resources</th>
<th>Local Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP30_2</td>
<td>5</td>
<td>2</td>
<td>30</td>
<td>1/2/3</td>
<td>3/2/1</td>
</tr>
<tr>
<td>MP90_2</td>
<td>5</td>
<td>2</td>
<td>90</td>
<td>1/2/3</td>
<td>3/2/1</td>
</tr>
<tr>
<td>MP120_2</td>
<td>5</td>
<td>2</td>
<td>120</td>
<td>1/2/3</td>
<td>3/2/1</td>
</tr>
<tr>
<td>MP30_5</td>
<td>5</td>
<td>2</td>
<td>30</td>
<td>1/2/3</td>
<td>3/2/1</td>
</tr>
<tr>
<td>MP90_5</td>
<td>5</td>
<td>2</td>
<td>90</td>
<td>1/2/3</td>
<td>3/2/1</td>
</tr>
<tr>
<td>MP120_5</td>
<td>5</td>
<td>2</td>
<td>120</td>
<td>1/2/3</td>
<td>3/2/1</td>
</tr>
<tr>
<td>MP30_10</td>
<td>5</td>
<td>10</td>
<td>30</td>
<td>1/2/3</td>
<td>3/2/1</td>
</tr>
<tr>
<td>MP90_10</td>
<td>5</td>
<td>10</td>
<td>90</td>
<td>1/2/3</td>
<td>3/2/1</td>
</tr>
<tr>
<td>MP120_10</td>
<td>5</td>
<td>10</td>
<td>120</td>
<td>1/2/3</td>
<td>3/2/1</td>
</tr>
<tr>
<td>MP30_20</td>
<td>5</td>
<td>20</td>
<td>30</td>
<td>1/2/3</td>
<td>3/2/1</td>
</tr>
<tr>
<td>MP90_20</td>
<td>5</td>
<td>20</td>
<td>90</td>
<td>1/2/3</td>
<td>3/2/1</td>
</tr>
<tr>
<td>MP120_20</td>
<td>5</td>
<td>20</td>
<td>120</td>
<td>1/2/3</td>
<td>3/2/1</td>
</tr>
<tr>
<td>MP90_2AC</td>
<td>10</td>
<td>2</td>
<td>90</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>MP120_2AC</td>
<td>10</td>
<td>2</td>
<td>120</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>MP90_5AC</td>
<td>10</td>
<td>5</td>
<td>90</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>MP120_5AC</td>
<td>10</td>
<td>5</td>
<td>120</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>MP90_10AC</td>
<td>10</td>
<td>10</td>
<td>90</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>MP120_10AC</td>
<td>10</td>
<td>10</td>
<td>120</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>MP90_20AC</td>
<td>10</td>
<td>20</td>
<td>90</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>MP120_20AC</td>
<td>10</td>
<td>20</td>
<td>120</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>

|          | 140         | 185        | 1800        |

Table 12 – Data Sets for Testing I

Table 12 indicates the name of the 20 subsets available along with its characteristics. The second column shows the number of instances per subset, totaling 140 instances. Third column indicates the number of projects in the portfolio within each instance. Fourth column specifies the number of activities per project. Finally, the last two columns include the global and local resources necessary in each instance. All resources in each subset are renewable.
This Page Was Intentionally Left Blank
11.2 Appendix II: Standard Data Sets for the MRCMPSP

There are a total of 30 instances for multiple projects with multiple modes publicly available in the web site corresponding to the MISTA challenge 2013, in which different teams compete for scheduling multiple projects taking into account the availability of local and global resources. The web site is [https://gent.cs.kuleuven.be/mista2013challenge/index.html](https://gent.cs.kuleuven.be/mista2013challenge/index.html) (last verification of address: 10-10-2016). The details of the available instances are summarized in the following table:

<table>
<thead>
<tr>
<th>Instance</th>
<th># Projects</th>
<th># Activities</th>
<th>Modes</th>
<th>Non-Renewable</th>
<th>Renewable (Local)</th>
<th>Renewable (Global)</th>
<th>Best Known Solutions</th>
<th>TDP</th>
<th>TMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-1</td>
<td>2</td>
<td>20</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>349</td>
<td>127</td>
</tr>
<tr>
<td>A-2</td>
<td>2</td>
<td>40</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>431</td>
<td>162</td>
</tr>
<tr>
<td>A-3</td>
<td>2</td>
<td>60</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>530</td>
<td>208</td>
</tr>
<tr>
<td>A-4</td>
<td>5</td>
<td>50</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>816</td>
<td>253</td>
</tr>
<tr>
<td>A-5</td>
<td>5</td>
<td>100</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td></td>
<td>380</td>
<td>222</td>
</tr>
<tr>
<td>A-6</td>
<td>5</td>
<td>150</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>792</td>
<td>228</td>
</tr>
<tr>
<td>A-7</td>
<td>10</td>
<td>100</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td></td>
<td>2932</td>
<td>530</td>
</tr>
<tr>
<td>A-8</td>
<td>10</td>
<td>200</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td></td>
<td>349</td>
<td>127</td>
</tr>
<tr>
<td>A-9</td>
<td>10</td>
<td>300</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>4059</td>
<td>743</td>
</tr>
<tr>
<td>A-10</td>
<td>10</td>
<td>100</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>340</td>
<td>163</td>
</tr>
<tr>
<td>B-1</td>
<td>10</td>
<td>100</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>340</td>
<td>163</td>
</tr>
<tr>
<td>B-2</td>
<td>10</td>
<td>200</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td></td>
<td>3030</td>
<td>445</td>
</tr>
<tr>
<td>B-3</td>
<td>10</td>
<td>300</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>390</td>
<td>143</td>
</tr>
<tr>
<td>B-4</td>
<td>15</td>
<td>150</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>4059</td>
<td>743</td>
</tr>
<tr>
<td>B-5</td>
<td>15</td>
<td>300</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>3030</td>
<td>445</td>
</tr>
<tr>
<td>B-6</td>
<td>15</td>
<td>450</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>3030</td>
<td>445</td>
</tr>
<tr>
<td>B-7</td>
<td>20</td>
<td>200</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>3030</td>
<td>445</td>
</tr>
<tr>
<td>B-8</td>
<td>20</td>
<td>400</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>3030</td>
<td>445</td>
</tr>
<tr>
<td>B-9</td>
<td>20</td>
<td>600</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>3030</td>
<td>445</td>
</tr>
<tr>
<td>B-10</td>
<td>20</td>
<td>420</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td></td>
<td>3030</td>
<td>445</td>
</tr>
<tr>
<td>X-1</td>
<td>10</td>
<td>100</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td></td>
<td>3030</td>
<td>445</td>
</tr>
<tr>
<td>X-2</td>
<td>10</td>
<td>200</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>3030</td>
<td>445</td>
</tr>
<tr>
<td>X-3</td>
<td>10</td>
<td>300</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>3030</td>
<td>445</td>
</tr>
<tr>
<td>X-4</td>
<td>15</td>
<td>150</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td></td>
<td>3030</td>
<td>445</td>
</tr>
<tr>
<td>X-5</td>
<td>15</td>
<td>300</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>3030</td>
<td>445</td>
</tr>
<tr>
<td>X-6</td>
<td>15</td>
<td>450</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>3030</td>
<td>445</td>
</tr>
<tr>
<td>X-7</td>
<td>20</td>
<td>200</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>3030</td>
<td>445</td>
</tr>
<tr>
<td>X-8</td>
<td>20</td>
<td>400</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>3030</td>
<td>445</td>
</tr>
<tr>
<td>X-9</td>
<td>20</td>
<td>600</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>3030</td>
<td>445</td>
</tr>
<tr>
<td>X-10</td>
<td>20</td>
<td>410</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>3030</td>
<td>445</td>
</tr>
</tbody>
</table>

Table 13 – Data Set for Testing II

Table 13 displays the name of the diverse instances, which totalize 30. The information presented in each column is the following: (1) instance, (2) number of projects in the instance, (3) number of activities per project, (4) average number of modes per activity, (5) amount of local non-renewable resources, (6) quantity of local renewable resources, (7) number of global renewable resources, (8) total project duration (TPD) of the best known solution and (9) total make-span (TMS) of the best known solution. For an explanation of TPD and TMS refer to section 6.1. The last two columns provide information published up to October 10, 2016.
Appendix III: Selected Instances for Studying the MRCPSP

The following table enumerates all instances that were included from data sets J10, J14, J16 and J20 of PSPLIB available on the public web site http://www.om-db.wi.tum.de/psplib/ (last verification of address: 10-10-2016). The instances were used in order to test the results of the improved method when activities possess multiple modes.

<table>
<thead>
<tr>
<th>J10</th>
<th>J14</th>
<th>J16</th>
<th>J20</th>
</tr>
</thead>
<tbody>
<tr>
<td>J10_2_5</td>
<td>J14_2_8</td>
<td>J16_2_3</td>
<td>J20_9_7</td>
</tr>
<tr>
<td>J10_3_8</td>
<td>J14_3_10</td>
<td>J16_3_5</td>
<td>J20_11_3</td>
</tr>
<tr>
<td>J10_4_9</td>
<td>J14_4_6</td>
<td>J16_4_7</td>
<td>J20_14_6</td>
</tr>
<tr>
<td>J10_6_3</td>
<td>J14_6_3</td>
<td>J16_6_4</td>
<td>J20_18_4</td>
</tr>
<tr>
<td>J10_8_6</td>
<td>J14_8_10</td>
<td>J16_8_3</td>
<td>J20_21_4</td>
</tr>
<tr>
<td>J10_10_7</td>
<td>J14_10_4</td>
<td>J16_10_7</td>
<td>J20_23_9</td>
</tr>
<tr>
<td>J10_12_4</td>
<td>J14_12_6</td>
<td>J16_18_4</td>
<td>J20_25_1</td>
</tr>
<tr>
<td>J10_14_6</td>
<td>J14_14_9</td>
<td>J16_24_6</td>
<td>J20_27_8</td>
</tr>
<tr>
<td>J10_15_9</td>
<td>J14_20_5</td>
<td>J16_27_9</td>
<td>J20_31_5</td>
</tr>
<tr>
<td>J10_31_5</td>
<td>J14_31_5</td>
<td>J16_31_5</td>
<td>J20_34_4</td>
</tr>
<tr>
<td>J10_34_4</td>
<td>J14_34_4</td>
<td>J16_34_4</td>
<td>J20_37_3</td>
</tr>
<tr>
<td>J10_42_7</td>
<td>J14_42_6</td>
<td>J16_42_7</td>
<td>J20_42_7</td>
</tr>
<tr>
<td>J10_44_1</td>
<td>J14_44_1</td>
<td>J16_44_1</td>
<td>J20_44_1</td>
</tr>
<tr>
<td>J10_47_9</td>
<td>J14_47_9</td>
<td>J16_47_9</td>
<td>J20_47_9</td>
</tr>
<tr>
<td>J10_50_6</td>
<td>J14_50_6</td>
<td>J16_50_6</td>
<td>J20_50_6</td>
</tr>
<tr>
<td>J10_53_2</td>
<td>J14_53_2</td>
<td>J16_53_2</td>
<td>J20_53_2</td>
</tr>
<tr>
<td>J10_55_8</td>
<td>J14_55_8</td>
<td>J16_55_8</td>
<td>J20_55_8</td>
</tr>
<tr>
<td>J10_59_7</td>
<td>J14_59_7</td>
<td>J16_59_7</td>
<td>J20_59_7</td>
</tr>
<tr>
<td>J10_60_2</td>
<td>J14_60_2</td>
<td>J16_60_2</td>
<td>J20_60_2</td>
</tr>
<tr>
<td>J10_63_9</td>
<td>J14_63_9</td>
<td>J16_63_9</td>
<td>J20_63_9</td>
</tr>
</tbody>
</table>

Table 14—Instances with Multiple Modes

Table 14 displays those 20 instances that were studied for each indicated problem set. The average results obtained are summarized in section 8.3.
This Page Was Intentionally Left Blank
Appendix IV: Example of Project Portfolio for the MRCMPSP

In the current sub-section, the characteristics of the project portfolio used at the end of section 8.3 will be explained. It corresponds to the instance A-5 already covered in section 11.2. The parameters of the objective function will also be explained:

According to Table 15 there are 4 resources. Two are non-renewable and the others renewable. One of the renewable resources is global, consequently it is shared among the different projects within the portfolio. The parameters corresponding to the objective function are indicated in the following table:

For an explanation of the meaning of each of the parameters mentioned in Table 16, refer to section 6.1.
11.5 Appendix V: Example of Adding Rework to Project Portfolio

This sub-section is a continuation of section 11.4, where it was covered the characteristics of the example project portfolio considering no rework. However, in the following figures it will be indicated the information necessary to include rework in the example used for section 8.3. Each of the following matrixes indicate two things: (1) location of feedback arcs and (2) likelihood of occurrence. In the example provided the number of feedback arcs is either 12 or 16 (depending on the project), whereas the probability may be 0.1, 0.2 or 0.3. For more details on how to interpret the matrixes, refer to section 7.14.

Figure 42 – Rework Matrix for Instance J20_52_8
Figure 43 - Rework Matrix for Instance J20_17_8

Figure 44 - Rework Matrix for Instance J20_63_1
Figure 45 - Rework Matrix for Instance J20_15_2

Figure 46 - Rework Matrix for Instance J20_11_8
12. References


