A FUZZY INFERENCE SYSTEMS APPROACH
FOR RESOURCE CONSTRAINED SCHEDULING
AND CLOSELY RELATED PROBLEMS

A Thesis
Submitted to the Faculty of Graduate Studies and Research
In Partial Fulfillment of the Requirements
for the Degree of
Master of Applied Science
in Industrial Systems Engineering
University of Regina

by
Imran Nathan Khan
Regina, Saskatchewan
August 2013

Copyright 2013: I.N. Khan
Imran Nathan Khan, candidate for the degree of Master of Applied Science in Industrial Systems Engineering, has presented a thesis titled, *A Fuzzy inference Systems Approach for Resource Constrained Scheduling and Closely Related Problems*, in an oral examination held on August 2, 2013. The following committee members have found the thesis acceptable in form and content, and that the candidate demonstrated satisfactory knowledge of the subject material.

External Examiner: Dr. Jing Tao Yao, Department of Computer Science

Supervisor: Dr. Rene Mayorga, Industrial Systems Engineering

Committee Member: Dr. Mohamed Ismail, Industrial Systems Engineering

Committee Member: Dr. Amr Henni, Industrial Systems Engineering

Chair of Defense: Dr. Mohamed El-Darieby, Software Systems Engineering
ABSTRACT

Today, the need to use some kind of Intelligent Systems (IS) techniques from the Soft Computing / Computational Intelligence fields is rapidly growing. This is because intelligent processes have the capability to deal with problems relating user-experience-knowledge and also they can consider data being uncertain or vague. This research focuses on applying IS techniques to solve the Resource Constrained Scheduling Problem (RSCP) and some related problems.

Traditional methods to solve the RSCP have proven to be useful; however, they are somewhat limited, time consuming, and not always effective. Among the traditional heuristic methods that have been used for years, the Serial method, the Parallel method, the Branch and Bound algorithm, and the Utility Index calculation, have played an important role in solving the RSCP. These methods focus on the scheduling of activities in a project and aim to do it in an effective way considering several factors such as, priority, availability of resources, and time frames.

In this Thesis in order to improve computational effort, efficiency, and the option to consider different scenarios with each input being uncertain/vague; three new methods: the Intelligent Serial Scheduling, the Intelligent Parallel Scheduling, and the Intelligent Branching, are proposed to solve the RSCP. Additionally, two new methods: the Intelligent Utility Index, and the Intelligent Schedule Performance Indicator; are proposed to solve closely related problems to the RSCP.
The proposed methodology for the solution of the RSCP and closely related problems is based on IS techniques, such as Fuzzy Inference Systems (FIS) from the Soft Computing / Computational Intelligence fields. These IS techniques lead to an effective scheduling process that allows to easily manipulate different and diverse parameters considered in the scheduling process. Furthermore, these techniques permit to consider each input/output to have diverse values/qualifiers (representing linguistic terms); and simultaneously, allow the user to include their knowledge and to use uncertain or vague data.

In this Thesis the proposed methodology is described in detail, and different scenarios are analyzed in order to observe its reach, and to prove its functionality and effectiveness. Moreover, remarks, images, and charts are included to illustrate the approach and provide the reader with a better understanding.
ACKNOWLEDGEMENTS

I wish to express my sincere thanks and appreciation to Dr. Rene V. Mayorga for his outstanding support and guidance throughout the making of this research and the completion of my program. Without his advice and help, the completion of this Thesis would have not been possible.

A special thanks to my wife for her love, her unconditional support, her trust and faith. The Thesis and all the work completed would have not been possible without her effort and patience.

I wish to thank my family for their unconditional support throughout my studies. Special thanks to my brother and my parents who encourage me to always give the best, for guiding me throughout my entire life, and for making my dreams, hopes, and achievements possible. Without their support none of this would have been possible.

Also, thanks to all my relatives and friends who supported and helped me in many ways.

A special thanks to the Faculty of Graduate Studies and Research for appointing me as a Graduate Teaching Assistant (GTA) in the winter semester of 2012. This provided me with educational experience as well as valuable funding for my education.

I would like to thank the University of Regina in general for making this possible and for letting me be a part of such an amazing experience.
POST DEFENSE ACKNOWLEDGEMENTS

I wish to express my sincere thanks and appreciation to my examination committee members: Dr. Amr Henri, Dr. Mohamed Ismail, and Dr. Jing Tao Yao. Sincere thanks for all the advice and guidance in the making of this Thesis.

I would like to thank Dr. Rene V. Mayorga, for his outstanding support and guidance. Thanks a lot for all the hard work, recommendations, and patience.
# TABLE OF CONTENTS

ABSTRACT .......................................................................................................................... I

ACKNOWLEDGEMENTS ................................................................................................. III

POST DEFENSE ACKNOWLEDGEMENTS ..................................................................... IV

TABLE OF CONTENTS ...................................................................................................... V

LIST OF ABBREVIATIONS AND SYMBOLS ................................................................... VIII

LIST OF FIGURES .............................................................................................................. IX

LIST OF TABLES ............................................................................................................... X

1. INTRODUCTION ........................................................................................................... 1
   1.1 RESEARCH BACKGROUND .................................................................................. 1
   1.2 OBJECTIVES ......................................................................................................... 3
   1.3 THESIS ORGANIZATION ..................................................................................... 4

2. RESOURCE CONSTRAINED SCHEDULING ............................................................... 5
   2.1 CURRENT RSCP METHODS .............................................................................. 5
   2.2 PRIORITY RULES ................................................................................................. 7
   2.3 THE SERIAL METHOD .......................................................................................... 8
   2.4 THE PARALLEL METHOD .................................................................................. 10
   2.5 RESOURCE MOVING TIME .............................................................................. 18
   2.6 THE BRANCH AND BOUND ALGORITHM ......................................................... 19
      2.6.1 Definition ....................................................................................................... 19
      2.6.2 The Algorithm .............................................................................................. 19
   2.7 UTILITY INDEX AND RESOURCE DIVERSION IN THE SCHEDULING
      PROCESS ................................................................................................................... 23
   2.8 SCHEDULING PERFORMANCE ......................................................................... 26
   2.9 NEW PROPOSED METHODOLOGY VS. EXISTING APPROACHES .................... 27
   2.10 SUMMARY .......................................................................................................... 27

3 LITERATURE REVIEW .................................................................................................. 29
   3.1 INTELLIGENT SYSTEMS (IS) TECHNIQUES ...................................................... 29
   3.2 FUZZY LOGIC ....................................................................................................... 29
      3.2.1 Fuzzy Sets ..................................................................................................... 30
      3.2.2 Fuzzy Rules and Fuzzy Reasoning .............................................................. 32
      3.2.3 Fuzzy Inference Systems (FIS) .................................................................. 33
      3.2.4 Fuzzy Modeling ........................................................................................... 36
# LIST OF ABBREVIATIONS AND SYMBOLS

<table>
<thead>
<tr>
<th>Abbreviation and Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IS</td>
<td>Intelligent Systems</td>
</tr>
<tr>
<td>FIS</td>
<td>Fuzzy Inference System</td>
</tr>
<tr>
<td>FL</td>
<td>Fuzzy Logic</td>
</tr>
<tr>
<td>MF</td>
<td>Membership Function</td>
</tr>
<tr>
<td>I</td>
<td>Input Value</td>
</tr>
<tr>
<td>O</td>
<td>Output Value</td>
</tr>
<tr>
<td>RCSP</td>
<td>Resource Constrained Scheduling Problem</td>
</tr>
<tr>
<td>RCS</td>
<td>Resource Constrained Scheduling</td>
</tr>
<tr>
<td>AI</td>
<td>Artificial Intelligence/Intelligent</td>
</tr>
<tr>
<td>UI</td>
<td>Utility Index</td>
</tr>
</tbody>
</table>
**LIST OF FIGURES**

Figure 2 - 1 Decision - in process set relation ................................................................. 8
Figure 2 - 2 Serial method algorithm .................................................................................. 10
Figure 2 - 3 Parallel method activity distribution ................................................................. 12
Figure 2 - 4 Parallel Method Algorithm ................................................................................ 12
Figure 2 - 5 Activity network [5] ......................................................................................... 13
Figure 2 - 6 a) Resource conflict before applying the parallel method [5] .............................. 16
Figure 2 - 6 b) Resource conflict after applying the parallel method [5] ................................. 16
Figure 2 - 7 Different resource links ..................................................................................... 19
Figure 2 - 8 Branching of activities ...................................................................................... 21
Figure 2 - 9 Scheduling based on Utility Index and Resource diversion [18] ......................... 25
Figure 3 - 1 Membership function of cheap ........................................................................ 32
Figure 3 - 2 Fuzzy inference system process ....................................................................... 34
Figure 3 - 3 Defuzzification schemes [9] .............................................................................. 35
Figure 3 - 4 Mamdani system using product and max for t-norm and t-conorm operators respectively [9].......................................................... 36
Figure 4 - 1 Intelligent Serial Scheduling process flow .......................................................... 46
Figure 4 - 2 Intelligent Parallel Scheduling process flow ....................................................... 51
Figure 4 - 3 Intelligent Branching Scheduling process flow ................................................ 54
Figure 4 - 4 Intelligent Utility Index process flow ................................................................ 58
Figure 4 - 5 Intelligent Schedule Performance Indicator process flow ................................ 62
Figure 5 - 1 Intelligent Serial Scheduling Mamdani model 1 ............................................. 65
Figure 5 - 2 Intelligent Serial Scheduling Mamdani model 2 ............................................. 66
Figure 5 - 3 Intelligent Parallel Scheduling Mamdani model 1 ........................................... 66
Figure 5 - 4 Intelligent Parallel Scheduling Mamdani model 2 ........................................... 67
Figure 5 - 5 Intelligent Branching Scheduling Mamdani model ......................................... 68
Figure 5 - 6 Intelligent Utility Index Mamdani model ........................................................ 68
Figure 5 - 7 Intelligent Schedule Performance Indicator Mamdani model .......................... 69
Figure 5 - 8 Intelligent Serial Scheduling Step 1 Scenario #1 GUI ...................................... 73
Figure 5 - 9 Intelligent Serial Scheduling Step 2 Scenario #1 GUI ...................................... 73
Figure 5 - 10 Intelligent Serial Scheduling Step 1 Scenario #2 GUI .................................... 75
Figure 5 - 11 Intelligent Serial Scheduling Step 2 Scenario #2 GUI .................................... 76
Figure 5 - 12 Intelligent Serial Scheduling Step 1 Scenario #3 GUI .................................... 77
Figure 5 - 13 Intelligent Parallel Scheduling Step 1 Scenario #1 GUI .................................. 81
Figure 5 - 14 Intelligent Parallel Scheduling Step 2 Scenario #1 GUI .................................. 81
Figure 5 - 15 Intelligent Parallel Scheduling Step 1 Scenario #2 GUI .................................. 83
Figure 5 - 16 Intelligent Parallel Scheduling Step 2 Scenario #2 GUI .................................. 84
Figure 5 - 17 Intelligent Parallel Scheduling Step 1 Scenario #3 GUI .................................. 86
Figure 5 - 18 Intelligent Branching Scheduling Scenario #1 GUI ....................................... 88
Figure 5 - 19 Intelligent Branching Scheduling Scenario #2 GUI ....................................... 89
Figure 5 - 20 Intelligent Branching Scheduling Scenario #3 GUI ....................................... 91
Figure 5 - 21 Intelligent Utility Index Scenario #1 GUI ...................................................... 92
Figure 5 - 22 Intelligent Utility Index Scenario #2 GUI ................................................................. 94
Figure 5 - 23 Intelligent Utility Index Scenario #3 GUI ................................................................. 95
Figure 5 - 24 Intelligent Schedule Performance Indicator Scenario #1 GUI ............................... 97
Figure 5 - 25 Intelligent Schedule Performance Indicator Scenario #2 GUI ............................... 98
Figure 5 - 26 Intelligent Schedule Performance Indicator Scenario #3 GUI ............................... 100

LIST OF TABLES

Table 2 - 1 Most common priority rules [5] ................................................................................... 8
Table 2 - 2 Maximum available resource amount [5] ................................................................. 14
Table 2 - 3 Activity flow throughout process [5] ........................................................................ 15
Table 2 - 4 Predecessor activity throughout process [5] .......................................................... 15
Table 2 - 5 Utility index parameters [18] ..................................................................................... 24
Table 5 - 1 Intelligent Serial Scheduling 1\textsuperscript{st} Scenario Step 1 results ......................... 71
Table 5 - 2 Intelligent Serial Scheduling 1\textsuperscript{st} Scenario Step 2 results .............................. 72
Table 5 - 3 Intelligent Serial Scheduling 2\textsuperscript{nd} Scenario Step 1 results ............................. 74
Table 5 - 4 Intelligent Serial Scheduling 2\textsuperscript{nd} Scenario Step 2 results ............................. 75
Table 5 - 5 Intelligent Serial Scheduling 3\textsuperscript{rd} Scenario Step 1 results ............................. 77
Table 5 - 6 Intelligent Parallel Scheduling 1\textsuperscript{st} Scenario Step 1 results ............................ 79
Table 5 - 7 Intelligent Parallel Scheduling 1\textsuperscript{st} Scenario Step 2 results ............................ 80
Table 5 - 8 Intelligent Parallel Scheduling 2\textsuperscript{nd} Scenario Step 1 results.......................... 82
Table 5 - 9 Intelligent Parallel Scheduling 2\textsuperscript{nd} Scenario Step 2 results .......................... 83
Table 5 - 10 Intelligent Parallel Scheduling 3\textsuperscript{rd} Scenario Step 1 results .......................... 85
Table 5 - 11 Intelligent Branching Scheduling 1\textsuperscript{st} Scenario results ................................. 87
Table 5 - 12 Intelligent Branching Scheduling 2\textsuperscript{nd} Scenario results ............................... 89
Table 5 - 13 Intelligent Branching Scheduling 3\textsuperscript{rd} Scenario results ............................... 90
Table 5 - 14 Intelligent Utility Index 1\textsuperscript{st} Scenario results .................................................. 92
Table 5 - 15 Intelligent Utility Index 2\textsuperscript{nd} Scenario results ............................................... 93
Table 5 - 16 Intelligent Utility Index 3\textsuperscript{rd} Scenario results ............................................... 95
Table 5 - 17 Intelligent Scheduling Performance Indicator 1\textsuperscript{st} Scenario results .................. 96
Table 5 - 18 Intelligent Scheduling Performance Indicator 2\textsuperscript{nd} Scenario results ............... 98
Table 5 - 19 Intelligent Scheduling Performance Indicator 3\textsuperscript{rd} Scenario results ................... 99
CHAPTER 1 – INTRODUCTION

1.1. Research Background

The need of project managers, constructors, and contractors to perform simultaneously multiple projects under limited resources has grown rapidly and considerably over the years. The scheduling of multiple activities under resource constraints normally leads to cumbersome computation. In addition to these constraints, precedence and priority constraints are also found between each activity due to the scarce sharing of resources, [11], [20].

One of the main objectives of recent research studies has been to yield and outstanding solution to this problem; this means, to find a way to schedule resources to maximize efficiency, lower the total times, and keep overall flow of work.

Resource Constrained Scheduling Problems (RCSP) arise when in a project with activities, a resource or set of resources are found with limited capacity or availability, [11]. These projects due to limited availability are constrained in their capacity to meet the predefined objectives, and generally lead to delaying the completion of necessary tasks in the minimum total completion time. Since the parameters defining a RSCP could vary in nature and quantity, this leads to a problem which increases substantially with the project’s network size.

Nowadays, conventional Intelligent Systems (IS) techniques from the Soft Computing / Computational Intelligence fields have proven to be extremely useful in the treatment of engineering problems. By combining these non-
conventional techniques with some of the current traditional methods for scheduling; a new advanced approach can be developed for the efficient solution of the RSCP and closely related problems. Also, knowing that users have a wide variety of situations and problems to solve, and, in many cases, relying on uncertain, vague, and inaccurate data, the RSCP scenarios can be processed properly using these non-conventional techniques.

Some non-conventional Intelligent Systems techniques that are well known include: Artificial Neural Networks (ANNs), Fuzzy Inference Systems (FIS), and Adaptive Neuro-Fuzzy Inference Systems (ANFIS), [12]. It is worth to mention, that many decision making methodologies can be based on Fuzzy Logic (FL). This is due to the fact that Fuzzy Inference Systems (FIS) have the capability to deal with real life problems that are based on user knowledge and experience and that they can also deal with uncertain and vague data, [12].

In order to appreciate the reach of these techniques and the advantage they might present when dealing with Resource Constrained (RC) problems, the new methodology proposed here should be explained and evaluated carefully. This research intends to fully understand the factors that should be considered in resource scheduling and on the implementation and application of Intelligent Systems (IS) techniques as a new approach to solve the RSCP and closely related problems. To properly consider these criteria, three new FIS methods, are proposed for the solution of the RCSP. Additionally, two other new FIS methods are also proposed for the solution of closely related problems to the RSCP.
By applying these useful FIS tools, several advantages can be foreseen:

(a) The proposed FIS techniques should demonstrate their flexibility and have the ability to deal with intuitive user knowledge.

(b) By creating an intelligent environment, a convenient user interface is created and the amount of total work is reduced given that in the case of the proposed FIS, the information is translated into common language input/output via antecedent-consequent relations.

1.2. Objectives

Given all the previously mentioned requirements and issues, it is desirable to develop an efficient approach to solve the RSCP and related problems. In doing so, this research has focused on achieving two main objectives:

1) **Reach of non-conventional techniques in resource constrained scheduling:** To perform a detailed study and discuss different methods for the Resource Constrained Scheduling Problem (RSCP) and implement various nonconventional IS techniques to find effective solutions that are able to perform scheduling in the best possible way. In addition, properly analyse the results obtained to make a fair conclusion on the proposed IS nonconventional methodology in terms of performance and efficiency.

2) **Treatment of uncertain information:** To make use of the advantages that arise when working with IS techniques from the Soft Computing / Computational Intelligence fields, in order to develop a procedure that
allows the users to manipulate the information required for the scheduling process in a more convenient and common language manner.

1.3. Thesis Organization

In the next Chapter, the Resource Constrained Scheduling Problem (RSCP) is defined; this is done in order to provide a better understanding of the problem and the importance of its treatment. The Chapter breaks down in explaining two current traditional heuristic methods for the solution of the RCSP including: the Serial method and the Parallel method; as well as other traditional techniques such as the Branch and Bound algorithm, and the Utility Index calculation. The Chapter 3 contains a literature review, focusing on some of the nonconventional Intelligent Systems (IS) techniques from the Soft Computing / Computational Intelligence fields that are used throughout the Thesis, i.e., Fuzzy Inference Systems (FIS). Afterwards, Chapter 4 introduces in detail the nonconventional Intelligent Systems methodology that is proposed here to solve the RCSPs and closely related problems. The Chapter 5 presents the implementation of the proposed methodology in a Matlab computer language as a platform for proof of principle. In Chapter 5, different scenarios are considered in order to demonstrate the performance, functionality, and effectiveness of the new proposed FIS methodology. This Chapter mainly focuses on analysing the results obtained from the proposed FIS methods. The final Chapter includes a summary and conclusions of the research accomplished as well as recommendations and future work.
CHAPTER 2 – RESOURCE CONSTRAINED SCHEDULING

The Resource Constrained Scheduling Problem (RCSP) had its beginnings in the 1950’s, when the extensive research on resource allocation arose, [11].

One of the main parameters for a project to be a success is without a doubt the ability to complete it within a time limit, regardless of its size. Delays result in every day’s lost in the total time of completion results, and probably revenue that is not likely to be recovered at a later time. Various techniques have been widely used as an administration tool to improve scheduling and project management tasks, not to mention, supporting project managers to attempt project completion on time and within financial availability, [15].

2.1. Current RSCP Methods

As it has been discussed, the Resource-Constrained Scheduling Problem arises from the struggling to schedule a project considering limited resource capacity and to minimize its total duration when having precedence relations and constant changes in the amount of resources, [11].

In this case each activity may be characterized by a unique duration and a singular collection of resource requirements that have to be available each time period the activity is being executed, [20].

Traditional methods have been a very useful tool for the RSCP when dealing with resource, time, and precedence constraints for the past decade. Current methods present advantages in reducing computational times, are
robust, and obtain acceptable results. However, the quality of these results and
the overall solution is often not satisfying.

"Heuristics based on priority rule have been one of the most important
solution techniques for resource-constrained scheduling with renewable
resources"; furthermore, they are contained in commercial packages. However,
since modern packages provide the possibility of including own procedures
comfortably by incorporated programming languages, more efficient, easy to
implement priority rule-based procedures are of great interest for project
management software developers and users", [16].

"On the other hand and although priority rule based methods do not
always give the best results, they are indispensable when solving large problem
instances quickly. Therefore, good priority rule-based methods are needed to
determine the initial solutions for heuristic procedures", [16].

Two different components when using priority rules for the scheduling of
resources can be distinguished, a schedule generation and a priority rule. In the
scheduling generation, two different methods can be defined, the serial method
and the parallel method.

It is important to note that the content on current traditional heuristic
methods in this Chapter is based on Hanh Quang Le, 2008 [20], for information
and demonstrative purposes
2.2. Priority Rules

The term “Priority Rule” simply arises from the need of setting levels of importance between activities. Heuristic techniques are well known for taking advantage of this labeling given that its methods are based on priority.

“Priority rule sorts the set of decision activities in the scheduling process. The sort of activities obtained with priority rules depends on the approach used and often obtains different schedules. Priority rules can be set on the basis of the information: activity-based, project-based, and resource-based. Activity based rules assign high priority to an activity based on a parameter or characteristic of the activity itself, for instance activity’s duration or float”, [20].

“Project-based rules assign priorities to activities based on the project they belong to, or characteristics of that project, such as the shortest activity from the shortest project (SASP) rule. Resource-based rules assign priority in terms of resource demands in each project or activity”, [20].

Table 2 - 1 presents the most widely used heuristic priority rules available in literature, [20]. It is important to note that there are more existing priority rules that a user could choose from depending on the type of project. As mentioned before, the Table below presents just a few only for demonstrative purposes.
In summary, priority rules are used to sort importance. By doing this, the user and/or program will automatically choose the activity with the highest priority based on a set rule, and this will result in saving time and an easier scheduling approach.

### 2.3. The Serial Method

“The Serial method is a scheme based on activities. It consists of \( j \) stages, where \( j \) represents the number of activities to be scheduled. At every stage, an activity is selected and scheduled as early as possible (earliest precedence) and taking into consideration resource availability”, [20]. In this scheme, two different sets of activities are found: the first one contains the activities that are eligible for scheduling and is called the “decision set”. The second one contains the set of activities already scheduled and is called the “in process set”.

<table>
<thead>
<tr>
<th>Priority Rule (Primary)</th>
<th>Commentary</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOF</td>
<td>Shortest Operation First</td>
</tr>
<tr>
<td>MINSLK</td>
<td>Minimum Slack First</td>
</tr>
<tr>
<td>SASP</td>
<td>Shortest Activity Shortest Project</td>
</tr>
<tr>
<td>LALP</td>
<td>Longest Activity Longest Project</td>
</tr>
<tr>
<td>MOF</td>
<td>Maximum Operation First</td>
</tr>
<tr>
<td>MAXSLK</td>
<td>Maximum Slack First</td>
</tr>
<tr>
<td>MINTWK</td>
<td>Minimum Total Work Content</td>
</tr>
<tr>
<td>MAXTWK</td>
<td>Maximum Total Work Content</td>
</tr>
<tr>
<td>FCFS</td>
<td>First Come First Served</td>
</tr>
</tbody>
</table>

Table 2 – 1 Most common priority rules, [20].
Hence, in each stage, an activity is selected from the set of decision activities in the order established by a priority rule and scheduled as soon as possible, taking into account the precedence relationships of the activities and the availability of the resources (Figure 2 - 1).

![Diagram of Decision - In Process Set Relation]

**Figure 2 – 1 Decision – In Process Set Relation**

The activity selected is removed from the decision set and added to the scheduled set. The serial scheme finishes when all activities are in the scheduled set.

It is important to note that once an activity is scheduled, the resources will not be available until the activity is completed. Unlike the parallel method, which will be explained later, where priority is updated and might change, releasing resources even if the activity is interrupted.

To better understand the serial method procedure, figure 2 – 2 will display the step-by-step algorithm used to follow the method, [20].
2.4. The Parallel Method

The Parallel method is again a time oriented scheme and $n$ stages can be found. In each stage, a set of activities is selected for scheduling and each stage $n$, is associated with a schedule time and three sets of activities: a decision set, an active set, and a complete set, [20].
Because of the schedule time $tn$, the set of scheduled activities is divided into subsets: scheduled activities, which are completed up to the schedule time, are in the complete set; and those which are not completed are in the active set. Additionally, in the decision set activities that are being considered for scheduling can be found. Resource availability and precedence relations are taken into consideration so the scheduling can take place.

“Each stage is made up of two steps:

Step 1: The new schedule time is determined and activities with a finish time equal to the new schedule time are removed from the active set and put into the complete set. This may place a number of activities into the eligible set.

- Step 2: One activity from the decision set is selected with a priority rule and scheduled to start at the current schedule time taking into account the availability of resources and precedence relations. Afterwards, this activity is removed from the eligible set and put into the in-process set”, [20].

Notice that “Step 2 is repeated until the eligible set is empty, i.e., activities were scheduled or are no longer available for scheduling with respect to resource constraints. The parallel scheme finishes when all activities are in the complete or in-process sets”. [20].

Figure 2 - 3 shows a scheme if how activities flow throughout the parallel method. To better understand the parallel method procedure, as in the serial method, an algorithm is presented in Figure 2 – 4 to explain it step by step.
Figure 2 – 3 Parallel method activity distribution

Figure 2 – 4 Parallel method algorithm, [20].
Example

To demonstrate the proposed scheduling procedure, an example network consisting of two small projects with a total of seven actual activities is used. Figure 2 - 5 presents the detailed data of the example through an activity network.

![Activity Network Diagram]

Figure 2 – 5 Activity network, [20].

“The time to move the resources from project 1 to project 2 is assumed to be one day. Note that available resource amounts are enough to schedule individual projects according to the unconstrained resource conditions”. [20]

The maximum available resource amount is presented in Table 2 - 2.

“The resource conflicts among projects occur when the individual projects are scheduled simultaneously as shown in in Figure 2 - 6 (a). As it can
be seen the overload of resources RA, RB and RC concentrate on the beginning phase of the two-project network. To solve these problems, resource moving time is applied to minimize project make-spans under the resource constrained and resource transfer time”, [20].

<table>
<thead>
<tr>
<th>Resource Name</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 2 – 2 Maximum available resource amounts, [20].

To schedule the projects in this case, the heuristic parallel method and the priority rules are applied. The scheduling process and scheduling results are presented in Table 2 - 3 and Table 2 - 4 respectively.

The resources utilization profile is represented in Figure 2 - 6(b), note that the results shown are after the methods are applied and the resource conflicts are eliminated. Hence, all of the required resources after scheduling are within the available amount. The resource transfer time is also included into the scheduling process.

For example as shown in the Table 2 - 3, activity P2-2 has a resource predecessor - activity P1-1 and the starting date of activity P2-2 is delayed one day as a result of the movement of the resources from activity P1-5 to P2-2, [20].
Table 2 – 3 Activity flow throughout process, [20].

<table>
<thead>
<tr>
<th>Scheduling time step</th>
<th>Activity</th>
<th>R-P</th>
<th>D</th>
<th>Start</th>
<th>Finish</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
</tr>
</thead>
<tbody>
<tr>
<td>T=0</td>
<td>P1-2</td>
<td></td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>P2-1</td>
<td></td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>T=4</td>
<td>P1-1</td>
<td>P1-2</td>
<td>5</td>
<td>4</td>
<td>9</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>T=5</td>
<td>P2-3</td>
<td>P1-2; P2-1</td>
<td>5</td>
<td>6</td>
<td>11</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>T=9</td>
<td>P1-3</td>
<td>P1-1</td>
<td>5</td>
<td>9</td>
<td>14</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>T=11</td>
<td>P2-2</td>
<td>P1-1; P2-3</td>
<td>4</td>
<td>12</td>
<td>16</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>T=14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T=16</td>
<td>P2-4</td>
<td>P2-2</td>
<td>4</td>
<td>16</td>
<td>20</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2 – 4 Predecessor activity throughout process, [20].

<table>
<thead>
<tr>
<th>P</th>
<th>Activity</th>
<th>D</th>
<th>S</th>
<th>F</th>
<th>A - P</th>
<th>R - P</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>1-1</td>
<td>5</td>
<td>4</td>
<td>9</td>
<td>-</td>
<td>P1-2</td>
</tr>
<tr>
<td></td>
<td>1-2</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1-3</td>
<td>5</td>
<td>9</td>
<td>1</td>
<td>4</td>
<td>P1-1; P1-2</td>
</tr>
<tr>
<td>P2</td>
<td>2-1</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>2-2</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>P2-1</td>
<td>P1-1; P2-2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2-3</td>
<td>5</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>P2-1</td>
</tr>
<tr>
<td></td>
<td>2-4</td>
<td>4</td>
<td>1</td>
<td>6</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Where:

A-P: Activity Predecessor

R-P: Resource-Predecessor

D: Activity Duration
The figure 2 - 6 shows the final resource conflicts (presented on the left (a)) and the total resource utilization after applying the parallel method (presented on the right (b)).

**Figure 2 – 6**

- **a)** Resource conflict before applying the parallel method
- **b)** Resource utilization profile after applying the parallel method, [20].

**Review**

After analyzing both methods, several observations and differences can be found. First, as previously mentioned, in the Serial method, priority updates are not considered, this is, once an activity is scheduled it will remain active until total completion. This will lead to the resources being freed until the activity is finished.
On the other hand, the Parallel method allows priority updates as the start of a given time frame (e.g. one day). This releases the resources from an activity even if it has not been completed causing interruptions from time to time.

“Considering the apparent advantages of maintaining resource work continuity, such as, maximized learning curve effect, minimized costs for transportation, and mobilization/demobilization of resources, the serial method is often preferred over the parallel method. However, many analyses have been performed on these types of heuristics in the multi-project context concluding that the parallel method is superior to the serial scheme when the time criteria, which means, project delay and multi-project duration are minimized”, [24].

Also, some advantages could be observed after analyzing both techniques:

a) Quite easy to understand and to implement, given that the only factors considered are the priority of activities and availability of resources.

b) Acceptable computational times.

c) Algorithms obtain acceptable and fast results even for big projects.

In addition the disadvantages observed were:

a) Resource moving time from one activity/project to another badly affects the project.

b) Allocating a resource from one project/activity to another always involves extra costs and time losses.

c) The resource moving time noticeably influences the total project duration in the scheduling processes.
2.5. Resource Moving Time

As previously discussed, in the activity scheduling process, an activity is scheduled depending on resource availability, priority, and technical relations with precedence activities. However, if there are not enough required resources for scheduling, this activity should be delayed until resources become available.

After some activities have been completed, or after a change in priority (in the case of the Parallel method) takes place, all or a percentage of the resources previously being consumed are freed. At this time, the required resources are available throughout its whole duration.

“Like this, one or more resource links are created between scheduling activity and completed activities and the start date of the scheduling activity will be the delay time plus the transfer time to move the resource from the finished activity to the scheduling activity”, [20].

“Generally, there are three models of the resource transfer among activities: a) the resources are transferred from many activities to one activity; b) the resources are transferred from one activity to many activities; c) the resources are transferred from one activity to one activity”, [20].

Figure 2 - 7 shows a clear representation of the different links created, [20].
2.6. The Branch and Bound Algorithm

2.6.1. Definition

The main component of the branch-and-bound algorithm is a time-oriented, branching scheme. It will be shown that this branching scheme generates at least all active schedules, so that traversing the search tree will result in an optimal solution. Inversely, the branching scheme tries to avoid constructing non-active schedules, which cuts down the search space considerably.

2.6.2. The Algorithm

There are several variables that are to be considered for the completion of this algorithm.

- Checkable constraints (an optimal solution should remove the constraint with no further complication).
- Soluble constraints (fast optimal solution).
- Defined constraints (constraints in consideration)
• Set of activities.

**Procedure**

The branch and bound algorithm is mainly focused on following several steps.

(1) The activities are selected and the branching process starts.

(2) If activity can be completed according to constraints, perform scheduling and branch to the next activity.

(3) If activity cannot be completed according to constraints, then perform extension of activity and branch to the same activity.

(4) Continue with the activities until all activities are scheduled.

(5) The search tree ends when A has 0 activities.

**Example**

To provide a better understanding to the functionality of the algorithm, 9 activities were considered to be analyzed through constrains and a timeline in days is set to start making the branching. Figure 2 - 8 displays the activity flow, the ones with no highlighting represent the activities with no constraints, therefore ready to be scheduled; the activities highlighted in gray represent the ones with constraints, meaning that their scheduling has to be delayed until the constrain is deleted.
Now, considering all 9 activities, the branching process will be explained.

First, activity 1 is considered, supposing the activity has no constraints (precedence constraints, resource constraints, etc.) because it is the very first activity in the process, the activity in day one can be scheduled, furthermore, it opens up activities 2 and 3.

Next, on day number 2, since activity 1 has already been scheduled, activity 2 and activity 3 are now considered. Here, let’s suppose activity number 3 has a soluble constraint, therefore, it can be solved quickly and the activity is ready for scheduling, consecutively, it will give opening to activities 5 and 6 since activity 4 has a precedence relationship with activity 2.
In contrast, supposing activity 2 has a resource constrain, it will have to let other activities finish to release resources and reallocate them to activity 2 itself. Hence, activity 2 is not scheduled and moved to the next day.

Now, considering day number 3, activities 2, 5 and 6 are now to be analyzed. Activity 2 still has resource constrains; therefore it cannot be scheduled just yet. Consecutively, activity 5 is attached to time constraints as well. On the other hand, activity 6 has no significant constraints and will lead to the opening of activity 7.

Moving on the timeline, day 4 is now considered. On this day three activities will be analyzed; activity 2, activity 5, and activity 7. Since activity 3 and 6 have finished, the resources have been reallocated and therefore activity 2 has now enough resources to be scheduled. Also, a feasible solution has been given to activity 5 which can now be considered for scheduling. Activity 7 now faces a precedence relation and has to wait for activity 4 to finish so that then it can be scheduled.

On day number 5, since activity 2 has been scheduled, activity 4 is now opened. Now, activity 7 only has to wait for activity 4 to be scheduled to be scheduled as well.

On day number 6, activities 7 and 8 now have no constraints, therefore both can be scheduled as early as possible and they both give opening to the last activity.

The branching process ends once all activities have been scheduled taking into consideration its constraints.
By studying this algorithm and by making an example, a few advantages and disadvantages are identified.

**Review**

After analyzing the previous method, several results can be observed. First of all, it has proven to solve constraints in a more logic way given that the parameters considered to schedule activities are precedence relations, other general constraints that might come up while working with the process, and resources. This will allow to solve large scale problems, and give the user the feeling that the process is treated in a more logic way.

Unlike the previous methods (serial and parallel), this process does not consider priority rules, which in the end, might result in the final schedule being not be the best scheme for project delivery and time manners.

**2.7. Utility Index and Resource Diversion in the Scheduling Process**

This section presents a resource allocation tool called the Utility Index (UI) for the resource constrained scheduling, [1].

The goal of this tool is to help a planner (user) prioritize the work at hand, optimize the allocation of resources available, and minimize activity and project duration. The method makes use of the perceived value earned by the user upon completion of an activity in terms of resource utilization.
“Within this context, an activity's utility index may be defined as the ratio of the number of required resources to accomplish the specific activity to the total number of resources required by all competing activities at the time of resource allocation”, [1].

This index indicates how many units of resources can be diverted from each activity in order to minimize the longest activity’s total time and project’s total duration. Therefore, the higher the utility index for each activity, the less resources should be taken from it for diversion.

To have a better understanding, an example of a small network consisting of four activities (A, B, C, and D) is taken, [1].

Table 2 – 5 will show the activities, the resource requirement for each activity in order to be completed, the total resource requirement for the project, and finally the calculation of the utility index.

<table>
<thead>
<tr>
<th>Activity Index (i)</th>
<th>Activity Title</th>
<th>Activity Duration</th>
<th>Resource Requirements</th>
<th>Utility Index (UI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>8</td>
<td>4</td>
<td>4/18</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>3</td>
<td>3</td>
<td>3/18</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>10</td>
<td>5</td>
<td>5/18</td>
</tr>
<tr>
<td>4</td>
<td>D</td>
<td>3</td>
<td>6</td>
<td>6/18</td>
</tr>
</tbody>
</table>

Table 2 – 5 Utility index parameters, [1].
The utility index for each activity, as previously defined, is obtained by dividing each activity’s resource requirement by the total number of resources required to complete all activities. Thus, the utility index (UI) for activity A is UI (A) = 4/18, for activity B is UI (B) = 3/18, for activity C is UI (C) = 5/18, and for activity D would then be UI (D) = 6/18. It can be concluded after observing the results, that activity B being the one with less duration and resource demand is the one that can have the most resources diverted to the activity with the highest duration and resource demand (D), followed by activity A, and consequently activity C, in order to shorten the project’s total duration, [1].

The following figure, displays a scheduling scenario where the Utility Index (UI) has been calculated and resources have been diverted to the activity with longest duration and highest resource requirements (D).

Figure 2 – 9 Scheduling based on utility index and resource diversion, [1].
In general, it can be said that the aim of the utility index is to determine which activity should be crashed and to what extent. Without the Utility Index calculation, crashing activities for resource diversion might have a negative impact on some activities and therefore the project. Thus, the utility index is an extremely useful tool to determine the best possible resource reallocation path.

2.8. Scheduling Performance

Another interesting subject when it comes to the scheduling of activities and resources for many users has been and will always be the performance of current schedules and the uncertainty if a project will be completed or not based on current work plans. Questions such as, is the way I scheduled my activities the best possible way to do it? How close am I to the project's deadline, and so, what should I do in the case activities are delayed? The need to somehow determine if the current schedule being used is yielding effective results and if the project will be able to be completed according to plan, normally leads to a big concern. At the same time, it is necessary to develop an automatic intelligent tool to indicate to the user the current schedule performance and what to do in case that the circumstances change while having activities and resources already scheduled, considering time issues and completion dates. In Chapter 4, a new IS method it is proposed as a tool to give indication to the issues/problems mentioned above. The new IS method considers parameters that might affect schedules already in process. This is done in order to ease and give planners and users a general and quick
response in case that unexpected and unforeseen things happen. It is important to mention here that currently there is no traditional heuristic method, nor other intelligent methods for this purpose.

2.9. New Proposed Methodology vs. existing approaches

In the current literature, several methods have been developed with the aid of fuzzy logic and other different Intelligent Systems and close techniques such as artificial neural networks, optimization and genetic algorithms, to enhance the traditional Serial and Parallel Scheduling, [2], [13], [19], [23], [27], [28], [31], [33]; as well as the Branch and Bound Algorithm, [29], [33]. However, the proposed methodology, presented in Chapter 4, is new, different, and distinct, from actual implementations.

2.10. Summary

In this Chapter several traditional methods for the Resource Constrained Scheduling Problem as well as a tool for alternative resource reallocation have been explained. As it can be seen from the preceding sections, all the methods present advantages when it comes to minimizing times, cost, and computational effort. However, the results are not always as quick as the user would expect or like. In addition, in many cases uncertain information might become a noticeable problem. For instance, considering the Branch and Bound algorithm and the constraints it includes in solving the problem; the constraints might not always be clear and/or might be so many that the user can find the approach too
complicated and long. These simply are some of the reasons why Intelligent Systems (IS) techniques are proposed here as a different approach but more efficient to traditional heuristic methods. Also, a new and original IS methodology is proposed in Chapter 4 as an easy and quick method to measure the performance of schedules already developed. The following Chapter 3 presents a brief literature review on the Fuzzy Modeling, explaining its functionality and its advantages.
CHAPTER 3 –LITERATURE REVIEW

This Chapter focuses on explaining the related techniques used in this research; mainly on, fuzzy modeling. It is important to mention most of the information in this Chapter is subject from the book Neuro-Fuzzy and Soft Computing, [12], for convenience of presenting concepts, diagrams, and notation.

3.1. Intelligent Systems (IS) Techniques

Intelligent systems and its applications play a very important role to researchers worldwide, as they have proven to be very useful and effective in a vast variety of projects and problems. Among the different techniques that can be found today, i.e., fuzzy logic, artificial neural networks, inverse programming, are widely used in a variety of situations.

One of the main concepts of fuzzy logic deals with data that can be represented as membership functions, this is, fuzzify the crisp inputs to assign a range of continuous data instead of working with discrete values, [12]. Also, fuzzy logic gives the advantage of dealing with a level of uncertainty by assigning different qualifiers for each measurable parameter, [12].

3.2. Fuzzy Logic

Since fuzzy logic’s first implementations, it has been widely used to manipulate information in modeling and a variety of problems in engineering and
other areas, [32], [12]. It is mainly used to transfer user knowledge and language into variables that can take qualifiers or numerical values. It can also be described as a convenient methodology that can be implemented from small projects, i.e., simple classifications, differentiating from different parameters, etc. to multi-channel PC’s, software, and control systems, amongst other things, [32]. Fuzzy logic is able to provide an easy way to give satisfying results based on imprecise, missing, or vague information. It is as close as human processing given that it is based on if-then judgement instead of attempting a solution with mathematical procedures, [32].

3.2.1. Fuzzy Sets

The concept of fuzzy set was originally introduced by Zadeh in 1965, [34]; it is basically defined as a class of objects with assigned membership function grades. Such sets are defined by membership functions which can assign each set a membership grade from zero to one.

In other words, a fuzzy set (class) A in X is characterized by a membership (characteristic function) \( f_A(x) \) which associates with each point in X a real number in the interval [0, 1].

“A fuzzy set can be mathematically expressed as: If X is a collection of objects denoted generically by x, then a fuzzy set A in X is defined as a set of ordered pairs: \( A = \{(x, \mu A(x)) \mid x \in X \} \), where \( \mu A(x) \) is called the membership
function (or MF) for the fuzzy set A. The MF maps each element of X to a membership grade (or membership value) between 0 and 1”, [34].

**Example**

Assume a student wants to buy a new laptop. As usual, the student is looking out on the market for cheap prices. Cheap can be represented as a fuzzy set on a universe of prices, and depends on the student’s budget. Then the following can be considered:

- Below $1000 dollars, laptops are considered cheap, and prices make no real difference to the student.
- Between $1000 and $2000 dollars, a variation in price will make the student to be inclined a bit more on the cheaper laptop.
- Between $2000 and $3000 dollars, another variation in price will make a clear difference between the cheapest and the more expensive laptop.
- More than $3000 dollars, the cost is way too high leaving this option out of the student’s consideration.

Figure 3 – 1 displays the membership function of cheap, considering the above explanations and example.
3.2.2. Fuzzy Rules and Fuzzy Reasoning

In order to understand the relation between fuzzy rules and fuzzy reasoning, many examples can be used; most of these will have a structure quite similar as the following example:

- **Rule**: if there is snow on the ground, the roads will be slippery
- **Information**: there is snow on the ground
- **Inference**: the roads will be slippery

From this example, a clear idea of the process of fuzzy reasoning can be observed, basically, the rule will act as the if-then condition, the information will be the inputs (set of information), and the inference system will be the output. In fuzzy set theory, fuzzy if-then rules and fuzzy reasoning are the engine of fuzzy reasoning.

In summary, the process can be reduced to the following steps, [34]:

![Figure 3-1 Membership function of “cheap”](image)
1. Comparison: To observe the information available to then compare with the fuzzy rules established. This allows finding the degree of compatibility with respect to each membership function.

2. Combination: This step summarizes the creation of fuzzy rules, this means, relate compatibility to membership functions in a single rule using AND or OR operators to form the defining conditions.

3. Output creation: To combine different rules taking into consideration inputs (available information), to then come up with a consequent action (output).

3.2.3. Fuzzy Inference System (FIS)

In a few words, fuzzy inference systems (FIS) can be described as simple and efficient strategies to update, handle, input, and analyze information by the use of antecedent - consequent parameters.

Before continuing to solve the methods described with fuzzy logic, an explanation will be made to clarify the process by which the techniques will go through.

Think of a fuzzy inference system as a set of three boxes. The first box (fuzzyfier) will be given an input, which will be given by the user or a controller, which will consequently be converted into fuzzy values, this is, transform the discrete inputs into continuous values. The second box (inference engine) will process the data from the fuzzyfier (box 1), taking into consideration if – then rules (antecedent – consequent parameters) and their proper weighting by the
use of appropriate T-norm and T-conorm operators, [12]. Actions are executed if the information supplied satisfies the established conditions. Another way to look at the inference engine is the following:

![Fuzzy Inference System process](image)

**Figure 3 - 2 Fuzzy Inference System process**

(1) Read parameters which values are considered to be inputs
(2) Evaluate conditions
(3) Fire rules whose conditions are satisfied

Finally, the resulting data goes through the last box (defuzzyfier) where it will be converted back to crisp values and evaluate the results obtained. Figure 3 – 2 displays the idea of the previous explanation.

In the current literature (Nero-Fuzzy and Soft Computing), [12]; mainly three fuzzy inference systems to work with are:

- Mamdani Fuzzy Inference Systems
- Sugeno Fuzzy Inference Systems
- Tsukamoto Fuzzy Inference Systems

For the Mamdani FIS, a defuzzification method is also needed in order to obtain a crisp value that best represents a fuzzy set. The five most commonly
used defuzzification methods are: smallest of max, largest of max, mean of max, centroid of area, and bisector of area. Figure 3-3, from [12], shows the different kind of defuzzification methods. However, on this research, the one used will be the centroid of area, which is the reminiscent of the calculation of expected values of probability distributions and can be expressed as:

- Centroid of area COA:

\[
COA = \frac{\int z\mu_A(z) z \, dz}{\int \mu_A(z) dz},
\]

where \( \mu_A(z) \) is the aggregated output MF.

Figure 3-3 Deffuzification Schemes, [12].
Figure 3 – 4, also from [12], shows a Mamdani FIS using product and max for T-norm and T-conorm operators. The result is that the consequent will be expressed by a function in the weighted average form.

![Figure 3 - 4 Mamdani System using product and max for T-norm and T-conorm operators respectively, [12].](image)

3.2.4. Fuzzy Modeling

As a clear picture of the structures and operations of several types of fuzzy inference systems has been described, the following can be concluded involving fuzzy modeling:

- A fuzzy inference system is designed according to previous knowledge, known behaviours, or actions, of a target system.
The fuzzy system is then able to understand and reproduce this behaviour. For instance, if the system is a human operator in charge of designing an airplane, then the fuzzy inference system becomes a fuzzy logic controller that can make changes and regulate the process. Moreover, if the target system is a school teacher, then the fuzzy inference becomes an expert system for academic processes and grading.

Furthermore, fuzzy modeling can have several features. It easily provides the user to incorporate human expertise/knowledge directly into the modeling process. This will help in a way that it will be easier to manipulate, easier to understand, and a lot easier to work with. Also, as fuzzy modeling presents the advantage of antecedent-consequent relations, the use of numerical data plays an important role in this process. In addition, the following steps are also a part of fuzzy modeling; these steps are the following:

- Select relevant antecedent-consequent variables
- Choose a type of fuzzy inference system according with the information available and the type of target system desired.
- Establish if-then rules, to assign a result (output) for every input.

3.3. Summary

The previous techniques are explained with the main purpose of giving a clear idea of how the new methodology proposed here is implemented to solve
the Resource Constrained Scheduling Problem. In other words, these FIS techniques are applied to new scheduling methods to prove that Intelligent Systems (IS) techniques have an excellent reach for problem solving. By doing this, the advantages previously mentioned are shown.

The next chapter presents the new proposed methodology for solving the RCSP and related problems.
This Chapter presents the five proposed new methods based on IS techniques: three of them focus on solving the scheduling of activities; one provides a different approach to reallocating resources based on the Utility Index; and another one focus on creating an Intelligent tool to measure Schedule Performance. The IS methodology proposed here somehow extends some concepts of the traditional heuristic scheduling methods described in Chapter 2. However, here the use of Intelligent Systems (IS) techniques is fully applied leading to totally new and different approaches. One of the important differences between the proposed methodology and the traditional methods is that the current traditional methods treat inputs and outputs in a binary manner (i.e. as “yes” or “no” values), and having only two options for a response; while the proposed new methods consider different qualifiers for inputs and outputs and allow to include uncertainty in the different measured parameters. This Chapter presents the proposed IS methodology as follows:

(1) Proposed new Intelligent Serial Scheduling method.
(2) Proposed new Intelligent Parallel Scheduling method.
(3) Proposed new Intelligent Branching Scheduling method.
(5) Proposed new intelligent Schedule Performance Indicator method.
4.1. Approach with Fuzzy Inference Systems (FIS)

As it has been previously mentioned, the five new IS methods are based on Fuzzy Logic, [12]. In order to consider the parameters needed and to simulate the different situations, a proper computer language and a platform/interface are required. In this case, the Matlab Fuzzy Logic Toolbox has been selected. This platform/interface facilitates the inputs and outputs to have different qualifiers. Although here a particular computer language (Matlab), [12], and its Fuzzy Logic Toolbox are used as a platform; it is important to notice that the proposed FISs can be implemented using any other computer languages/platforms.

To follow the FIS process, [12], the following steps have been set to describe the process that Fuzzy Logic uses in order to give the best possible results:

(a) Define Objectives: The user can define certain goals and objectives to be obtained after the process is complete, including, minimizing scheduling times, scheduling all desired activities, completing the project considering external parameters (priority, constraints, completion times), etc.

(b) Observe Environment: Observe current conditions and parameters. This is setting measurable rules and parameters such as availability of resources, number of activities to schedule, priority considerations, and completion times and convert them into membership functions.
(c) Set If – Then rules: Set the rules inputs and outputs representing the real system that each FIS intends to model.

(d) Select a proper Fuzzy Inference Model.

(e) Performing actions: As inputs, outputs, and rules are defined; obtain results based on antecedent – consequent relations to meet established goals.

4.1.1. Fuzzy Inference Model

The Fuzzy model that is selected for developing each one of the proposed IS methods is the Mamdani Fuzzy Inference Model, [12]. The construction of this model and specific details (on Rules, Membership Functions, T-Norms, T-CoNorms, Defuzzification method) are provided in the Appendix A.

4.2. Intelligent Serial Scheduling

The Intelligent Serial Scheduling is the first method that is presented. This method is based on the traditional Serial Method, [20], given that it considers the same parameters to complete the scheduling of activities. However, as mentioned before, here as a Fuzzy Inference Systems (FIS); the resultant method allows assigning different qualifiers to the parameters. Once again, the interface that is used for the simulation of the FIS system is the Matlab Mamdani Toolbox.

In order to complete the Intelligent Serial Scheduling, the parameters from the traditional method are considered and divided into two different FIS. The first one called Step 1 – Main Scheduling Phase is executed considering the
decision set; this means, if there are activities to be scheduled, the priority each activity has for the user, the availability of resources, and the precedence relations. Hence, this first system has inputs and outputs that are divided as follows:

**Step 1 – Main Scheduling Phase Inputs:**

1) Decision Set [Empty ; Some Activities ; Many Activities]
2) Activity Priority [Low Priority ; Mid Priority ; Highest Priority]
3) Resources [None ; Not Enough ; Enough]
4) Relation [No Precedence ; No Significant Precedence ; Precedence]

**Step 1 – Main Scheduling Phase Outputs:**

1) Action [Do not Schedule ; Schedule]
2) Process [Stop, Go to Step 2, Pause]
3) Activity List [Update ; Do not Update]

Input 1: Decision Set: The decision set holds the activities that are being considered for scheduling. The options that are considered for the analyses are that the decision set has either *some activities* or *many activities* remaining (not scheduled yet) or *no activities* at all. In the case where there are no activities found, it might be because either all activities have been scheduled, or no activities have been assigned to the decision set.
Input 2: Activity Priority: This input looks for the highest priority. This means that all activities are assigned a level of priority (minimum slack, maximum slack, maximum total work content, etc.) based on priority rules (explained earlier in Chapter 2), and a highest priority is given. In other words, after assigning each activity its level, a highest priority is given and the activity that matches it, is the first one considered for scheduling.

Input 3: Resources: Resources play a key part in the scheduling process, considering that without resources no activity can be executed. In the analysis, the way of taking advantage of the Mamdani toolbox is to give a membership value for enough resources, not enough resources, or none found for the scheduling.

Input 4: Relation: Precedence relations appear when one activity has a predecessor which is holding it from being scheduled. This can be due to many reasons including that an activity needs to release resources in order for the next one to be scheduled, or simply because the activity gives a result and depending on it the next activity is then opened or not.

Output 1: Action: Based on all the inputs given, the system simply gathers all the information and determines if the activity can be scheduled or not.

Output 2: Process: As long as the decision set has activities remaining, the obvious thing to do is to keep the process running until all activities have been scheduled, this output considers precisely this input and evaluates if the
process should continue (go to Step 2), or if the case is presented where the decision set is empty, stop the process.

Output 3: Activity List: As an activity with the highest priority has been scheduled, all other activities with an initial non highest priority value should be updated. Once again as an activity is scheduled, the second activity with the highest priority is now be considered as the highest priority and should be the next one to be scheduled. Output 3 updates the priority in the activity list.

The previous data provides the necessary information for the system to take actions in order to reach its goal. The next step is to determine the antecedent-consequent rules, also known as the if-then rules. The rules for this first FIS are presented on Appendix B – Section 1.

Once the if – then rules have been established for the first system (Step 1), the next step is to define the second system (Step 2) which is called the 

Finishing Phase. In contrast to the first system (Step 1), this system considers only two inputs and two outputs. In is important to note that one of the outputs in system 1 (process), has the option to direct the system to go to Step 2 (system 2). The inputs considered in this system are not exactly fed from the outputs from system 1, however, as different parameters are affected depending on the outputs of system 1, the inputs of this system are somewhat related to the previous outputs.

Once the above has been explained, the inputs and outputs for the second system (Step 2) are be shown next:
**Step 2 – Finishing Phase Inputs:**

1) In Process Set [No Activities ; Some Activities ; Many Activities]

2) Activity Progress [Not Completed ; Partially Completed ; Completed]

**Step 2 – Finishing Phase Outputs:**

1) Resources [None Released ; Some Released ; All Released]

2) Action [Stop, Go to Step 1]

Input 1: In Process Set: As the Process output from system 1 puts activities in the in process set, this input considers this set and it is determined if it has some activities, many activities, or no activities scheduled.

Input 2: Activity Progress: This checks if the activity scheduled has already been completed, has been partially completed, or is not completed and has not had any progress. If it is still in process it means it is still consuming some or all of the resources it requires and other activities may not be able to be scheduled until the activity finishes and/or some resources might be released.

Output 1: Resources: As resources are used while activities are in the In Process Set, the activity progress indicates if whether an activity is in progress, or has been completed. The freedom of resources depends on this parameter; as an activity finishes resources are freed and become available for other activities to be scheduled.
Output 2: Action: Output 2 indicates the system to go back to Step 1, as Step 1 is the system that determines when to stop the scheduling process depending on the existence of activities.

Moreover, the if – then rules for system 2 (Step 2) are presented in Appendix B – Section 2:

The following diagram illustrates the process in which both systems interact.

![Diagram of Intelligent Serial Scheduling Process Flow](image)

Figure 4 – 1 Intelligent Serial Scheduling Process Flow

As previously mentioned, the proposed Intelligent Serial Scheduling is a new and different approach for the RCSP. Some of the current techniques that have been recently developed which are somewhat related to; but are clearly
different and distinct to the technique proposed here are: the creation of an Adaptive Multi-Objective Hybrid Genetic Algorithm, [31], which uses the Serial Method to evaluate process fitness; a proposed heuristic model to deal with fuzzy RCSP problems, [27], where the duration of each activity is considered as uncertain whereas resource availability and demand are crisp; and a new Heuristic Model for Fully Fuzzy Project Scheduling, [33], which deals with RCSP in context of uncertain data. However, the method in [33] focuses on selecting activities with the shortest duration to choose as the highest priority, as well as considers other parameters such as the demand for each activity. Lastly, fuzzy dynamic programming has also been used to model Fuzzy RCSP problems, [19].

4.3. Intelligent Parallel Scheduling

The Intelligent Parallel Scheduling is the second proposed FIS method that is presented. As in the first proposed method, this one also uses Fuzzy Logic to solve the RSCP. This method of scheduling it is somewhat similar to the Intelligent Serial Scheduling, proposed in Section 4.2; it is so, because two Fuzzy Inference Systems are considered. The first FIS is also called Step 1 – Main Scheduling Phase and is executed considering the same parameters as the previous method. Thus, the first system (Step 1) has inputs and outputs that are divided as follows. Note that since the same parameters are considered as in the Intelligent Serial Scheduling for the first system; no further explanations are given again here for each parameter.
**Step 1 – Main Scheduling Phase Inputs:**

1) Decision Set [Empty ; Some Activities ; Many Activities]
2) Activity Priority [Low Priority ; Mid Priority ; Highest Priority]
3) Resources [None ; Not Enough ; Enough]
4) Relation [No Precedence ; No Significant Precedence ; Precedence]

**Step 1 – Main Scheduling Phase Outputs:**

1) Action [Do not Schedule ; Schedule]
2) Process [Stop, Go to Step 2, Pause]
3) Activity List [Update ; Do not Update]

The corresponding rules for the first system are presented on Appendix B – Section 3.

Once the If – Then rules have been established for the first system (Step 1), the next step is to define the second FIS system (Step 2) which is called the *Prioritizing Phase*. It is important to remember from Chapter 2 that the parallel method, [20], might change priority while activities are being completed; this leads to stopping current scheduled activities and assign resources and time to the newest highest priority activity. The proposed FIS system considers three inputs and four outputs. It is important to note that as in the Intelligent Serial Scheduling presented in Section 4.2; one of the outputs in system 1 (process), has the option to direct the system to go to Step 2 (system 2). Once again the inputs considered in this system are not exactly fed from the outputs from system 1; however, as different parameters are affected depending on the
outputs of system 1, the inputs of this system are somewhat related to the previous outputs.

Once the above has been explained, the inputs and outputs for the second system (Step 2) are shown next:

**Step 2 – Prioritizing Phase Inputs:**

1) Active Set [Empty ; Some Activities ; Many Activities]
2) Priority Status[Change Priority ; Maintain Priority]
3) Activity Progress [Not Completed ; Partially Completed ; Completed]

**Step 2 – Prioritizing Phase Outputs**

1) Resources [None Released ; Some Released ; All Released]
2) Activity Status [Move to Decision Set ; Move to Complete Set ; Maintain in Active Set]
3) Process [Stop Process ; Go to Step 1]
4) Priority Action [Same Priority ; Re-Prioritize]

Input 1: Active Set: The active set holds the activities that have been scheduled. The options that are considered for the analysis are that the active set has *some activities, many activities, or no activities* at all. In the case where there are no activities found, it might be because either all activities have been scheduled, or no activities have been scheduled.

Input 2: Priority Status: This input characterizes the parallel scheduling as it is the parameter that indicates if the priority to schedule activities has
changed. As previously mentioned, and according to Chapter 2, the parallel scheme stops all activities currently scheduled if priority changes.

Input 3: Activity Progress: This checks if the activity scheduled has already been completed, has been partially completed or not completed and has not had any progress. The difference with the Intelligent Serial Scheduling is that if the priority changes, activities that have not been completed stop and go back to the decision set.

Output 1: Resources: As discussed earlier, resources are consumed while activities are in the active set, in contrast to the Intelligent Serial Scheduling, as the parallel scheme updates priority on given time frame; resources are released for re-allocation even if activities have not been completed.

Output 2: Activity Status: This output moves activities to different sets depending on completion and/or re-prioritizing of the process. Activities that have not been completed remain in the active set until completion unless re-prioritizing happens; in this case they are moved back to the decision set. When activities are completed they are moved to the complete set as explained earlier in Chapter 2.

Output 3: Process: Output 2 indicates the system to go back to Step 1, as Step 1 is the system that determines when to stop the scheduling process depending on the existence of activities.
Output 4: Priority Action: As the priority status input indicates if whether or not to change priority, the priority action then keeps the same priority or re-prioritizes activities in the decision set.

In addition, the antecedent – consequent rules for the second system (Step 2) are presented on Appendix B – Section 4. The following diagram represents the process in which both systems interact.

![Intelligent Parallel Scheduling Process Flow](image)

**Figure 4 – 2 Intelligent Parallel Scheduling Process Flow**

Similarly to the Serial Scheduling, some approaches and techniques have been recently developed related to the Intelligent Parallel Scheduling, [2],
However, the novel method proposed here is clearly different and distinct to those approaches.

In 2011, Savas Balin, [2], used fuzzy processing times for parallel machine scheduling problems, where a robust genetic algorithm is used to minimize the maximum completion time.

Likely, a scheduling problem with fuzzy due-dates on unrelated parallel processes was developed, [13]. In the work presented in [13], the membership functions are considered and related to job processors to minimize the makespan.

Additionally, the work in [28], presents a Neuro Fuzzy Network in conjunction with hybrid intelligent optimization techniques in Parallel Job Scheduling; where the performance tuning of a fuzzy controller is used for the scheduling of parallel jobs.

One other related approach is the Project Scheduling under Uncertainty using Fuzzy Modeling, [23]; where two genetic algorithms are presented with the idea of improving operational levels of planning.

4.4. Intelligent Branching Scheduling

This proposed method can deal with a large variation of constraints. As previously seen it is based on the traditional branch and bound method, [6], presented in Chapter 2. Not only does it consider resources, activities, and precedence relations; but also, other possible constraints such as time,
management issues, simple constraints, as well as significant constraints that might even stop the whole process. By approaching this method from a Fuzzy Inference Systems perspective; the method becomes extremely useful given that membership grades are given to the different parameters that are considered. The proposed FIS inputs and outputs are explained next; however, it is important to note that all parameters are placed together in one single system. This is because, unlike the proposed methods presented in Sections 4.2, and 4.3, this scheme considers only one activity set, and does not consider different groups (decision set, active set, in process set). Therefore, the inputs and outputs used for this method are:

**System Inputs:**

1) Activity Set [Empty ; Some Activities ; Many Activities]
2) Activity Status [No Action ; Activities Selected ; No Activities Selected]
3) Relation [No Precedence ; Ni Significant Precedence ; Precedence]
4) Constraints [Soluble ; Defined ; Not Found]

**System Outputs:**

1) Action [Schedule ; Branch Activity ; Next Activity ; Stop Process]

Input 1: Activity Set: Once again, the system considers the existence of activities in the active set that are considered for scheduling.

Input 2: Activity Status: This indicates if activities have been selected for the scheduling process, the different grades include *no action*, which refers to
not having any activities for scheduling, activities selected, and no activities selected, which refers to having activities for scheduling but haven’t been selected as which will be scheduled and in what order.

Input 3: Relation: As before, precedence relations appear when one activity has a predecessor which is holding it from being scheduled.

Input 4: Constraints: The Intelligent Branching Scheduling considers all the constraints in particular groups, as in the traditional Branch and Bound Algorithm, [6]. The constrains can be soluble (which as previously explained are constraints that have a quick solution and do not prevent the activity from being scheduled); defined constraints (which are constraints that have significant value and does not make the scheduling possible); and lastly no constraints (which will allow the activity to be scheduled with no further complication).

Output 1: Action: The output considers the action to be made, including if whether or not to schedule or branch an activity, the selection of activities to be scheduled, and the termination of the process if no activities are found.

In addition, the antecedent – consequent rules are presented on Appendix B – Section 5 to observe how the different parameters complete the scheduling process. The following diagram represents the process in which the system interacts.
Some other methods have been recently developed considering the Branch and Bound Algorithm and the use of Intelligent Systems, [29], [5]. Again, the novel method proposed here is clearly different and distinct to those approaches.

A method which modifies the Branch and Bound algorithm considering three machine flow shop problems using fuzzy processing times and fuzzy arithmetic is developed in [29]. This method is used to determine the minimum completion times by getting a scheduling result with a membership function.

Another technique was developed, [5] where the possibility of obtaining an optimal solution as early as possible in the branch and bound search is pursued. Here, membership functions are utilized to accomplish this.

4.5. Intelligent Utility Index

As previously explained in Chapter 2, the Utility Index (UI), [1], helps measure which of the selected activities for scheduling should be chosen for resource diversion in order to minimize total times. However, as useful as the traditional utility index method is; the utility index is calculated for each activity
regardless of activity duration and resource demand, and then a decision is made to whether not to divert resources. The new method proposed here focuses on the creation of an intelligent environment that directly indicates to the user the value the Utility Index (low, mid, high); and therefore which corresponding action should be executed, depending again on activity resource demand and duration. The proposed new method saves time and it automatically indicates which action should be performed for each activity.

The inputs and outputs of the proposed FIS explained next.

**System Inputs:**

1) Activity Set [Empty ; Some Activities ; Many Activities]
2) Activity Status [Not Selected ; Selected]
3) Resource Requirement [Low ; Mid ; High]
4) Activity Duration [Short ; Medium ; Long]

**System Outputs:**

1) Utility Index [Low ; Mid ; High]
2) Resources [Do not Divert ; Consider Diverting ; Divert]
3) Process [Select New Activity ; Stop Process]

Input 1: Activity Set: The activity set holds the activities that are being considered for scheduling. The options that are considered for the analysis are that the set has *some activities*, *many activities* or *no activities* at all. In the case
where there are no activities found, it might be because either all activities have been checked, or no activities have been considered for scheduling.

Input 2: Activity Status: Indicates if an activity has been selected to start the calculation, the different grades include *no action*, which refers to not having an activity selected, and *activity selected*, which refers to having an activity selected for calculation.

Input 3: Resource Requirement: This input indicates what the resource requirement for the activity currently selected is. As previously seen (Chapter 2), activities with *low* resource requirement have a low utility index, therefore, being the ones that are to be considered for resource diversion. Activities with *mid* resource requirement, have a mid-high utility index which might be considered for resource diversion. Activities with a *high* resource requirement which are the ones that cannot be considered for crashing. Once again, the advantage of using the proposed FIS method is that different qualifiers for inputs and outputs make the process easier and allow the process to make a decision based on uncertain data.

Input 4: Activity Duration: The activity duration is the other parameter that determines if the activity needs more resources to be completed on time, and at the same time, give indication if it can be considered for resource diversion.

Output 1: Utility Index: Based on the above parameters, the process considers if the Utility Index (UI) to be *high*, *mid*, or *low*, and therefore give an
indication of what should be done with resources. Once again, depending on the value of the UI, the action varies.

Output 2: Resources: As the UI is estimated (low, med, high), based on this value an action is to be considered. If activity has a low UI, then resources can be diverted, if activity has a mid UI value, then the resources might be considered for diversion, but if the activity has a high UI, resources are not to be considered for diversion, asking the process to select a new activity.

Output 3: Process: This is the output that controls the process, where there are activities still remaining to choose for the calculation, the process continues. In the case where no activities are there, or all activities have been taken for calculation, then the process stops.

The corresponding if – then rules for this FIS are shown in Appendix B – Section 6. The following diagram illustrates the process in which the FIS interacts.

![Intelligent Utility Index Process Flow](image)

**Figure 4 – 4 Intelligent Utility Index Process Flow**

It is important to note that there are no similar methods in the current literature as the one proposed here. The proposed method takes the principle
and parameters used in the traditional Utility Index tool, [1], along with fuzzy logic, to construct and develop a new resource allocation method.

4.6. Intelligent Schedule Performance Indicator

The Intelligent Schedule Performance Indicator is a new tool, as there are no similar methods in the current literature. Current methods allow the user to somewhat understand time estimates and completion dates based on the established schedule, [26]. However, these techniques are based on mathematical operations, unlike the one proposed here, which is based on fuzzyfied data. Additionally, the new proposed method indicates the user what to do for any given performance the schedule might have.

This intelligent method focuses on giving the user an indicator of the performance of a current schedule. This is, the system considers time frames, if activities are going according to initial planned completion dates, the arising of new precedents (unexpected constraints) that might delay the completion times, and if more resources than it was originally forecasted are needed. For example, consider a simple schedule where the painting of a house is needed. Six activities are considered and scheduled. The six activities are scheduled according to precedence constraints, resource availability, and time. The estimated completion time for the project is two weeks. Suddenly, after the second day, somebody accidentally drops a bottle of paint creating the need to buy more (more resources), and possibly creating a precedence constraint given that without paint, a given last activity such as drying, cannot be completed until
the house is fully painted. Consequently, the original schedule is affected because as more resources are needed, precedence relations are created, and activities are delayed, hence, the due date to complete the project grows closer. The Intelligent Performance Indicator proposed indicates precisely this; the performance of the current schedule based on measurable parameters (which might or might not change) and if a re-scheduling is needed in order to complete the project as early as possible or at least the most important activities.

Having explained what the new FIS method is after, the system will be explained below.

The inputs and outputs considered are:

**System Inputs:**

1) Due Date [Now ; Near ; Far]
2) Activities Status [On time ; Delayed ; Very Delayed]
3) New Constraints [None ; Not Significant ; Significant]
4) Resources [None Extra Needed ; Some Extra Needed ; Many Extra Needed]

**System Outputs:**

1) Performance [Very Low ; Low ; Good ; Very Good ; Excellent]
2) Action [No Action ; Consider Rescheduling ; Immediate Rescheduling]

Input 1: Due date: The due date of a project is perhaps the most important factor to be considered when scheduling any activity. The importance
of delivering on time has for long been one of the biggest concerns for users. For this input, the due date qualifiers considered will be *now* (which means the due date of the project is already here); *near* (which means the due date of a project will come soon and all activities should be close to finishing); and *far* (which means the due date is still far ahead and activities still have time to be completed).

**Input 2: Activity Status:** This input indicates if activities are *on time* (according to original plan), have been *delayed* by one or many unexpected events or situations, and *very delayed*, which means activities have been affected considerably by unexpected events and will not be able to finish on time.

**Input 3: New Constraints:** As in the example of the paint mentioned above, where the situation prevents other activities from starting, one or more constraints might arise during the execution of the activities. For this input the qualifiers are *none*, where these are not present; *not significant*, where constraints do not represent a risk for activities to be delayed; and *significant*, which is, if some have arose and affect the completion times of activities.

**Input 4: Resources:** As unexpected situations happen, once again, like in the example of the person dropping the bottle of paint, more resources are needed that were originally not considered. The qualifiers in this case are *none extra needed*, *some extra needed*, and *many extra needed*. 
Output 1: Performance: This is the indicator of schedule performance that indicates to the user if whether or not good results are being achieved by the way activities are scheduled and if other external unexpected situations have affected the initial scheduling. The qualifiers vary from very low, low, good, very good, and excellent. It is perhaps the most important of the outputs as the user can find out if the overall final results would be what it was initially expected.

Output 2: Action: Depending on the performance the schedule has, action indicates what the user should do. Qualifiers for this output include no action (which will be the case that the schedule is running as expected and no further action is required as the project will be delivered on time); consider rescheduling (which tells the user to consider the option of rescheduling activities and resources given the way the process is going); immediate rescheduling (which indicates that there is a high possibility that completion time will not take effect as expected and a rescheduling is extremely necessary).

The corresponding If – Then rules for this FIS are given on Appendix B – Section 7. The following diagram illustrates the FIS interaction.

![Figure 4 – 5 Intelligent Schedule Performance Indicator Process Flow](image)
4.7. Summary

In this Chapter five new methods using Intelligent Systems (IS) techniques have been proposed for solving the RCSP and closely related problems. Fuzzy Inference Systems (FIS) along with a computer platform have made the completion of the proposed methodology possible. Although here a particular computer language (Matlab) and its Fuzzy Logic Toolbox are used as a platform; it is important to notice that the proposed FIS methodology is computer and language platform independent; that is, it can be implemented using any other computer and languages platforms.

The following Chapter presents the results obtained on simulating different scenarios for each proposed Fuzzy Inference System (FIS). As previously mentioned, three different scenarios are presented for each new method. The Tables presenting the results are also included and a summary outlining differences and advantages are also presented.
CHAPTER 5 – RESULTS AND ANALYSIS

This Chapter presents the results obtained from the implementation and application of the proposed Fuzzy Inference Systems (FIS) presented in Chapter 4. Different scenarios are considered for each one of the proposed FIS models and results are analyzed.

The Fuzzy modeling results presented here correspond to seven different Mamdani FIS models. Two of these models being for Step 1 and Step 2 for the Intelligent Serial Scheduling method; two for the Step 1 and Step 2 for the Intelligent Parallel Scheduling method; and one for the Intelligent Branching Scheduling method. Additionally, one FIS model being for the Intelligent Utility Index; and the last one being for the Intelligent Schedule Performance Indicator.

This Chapter is organized as follows:

(1) Experimental results and observations when applying the proposed methodology in Chapter 4 to different scenarios.

(2) Observations

(3) Summary

5.1 Fuzzy Inference models

In order to implement the proposed Mamdani FIS models, the Matlab software package, [12], and its Fuzzy Logic Toolbox were selected as a platform. The details on the implementation and operation of the proposed Mamdani FIS models in the Fuzzy Logic Toolbox are provided in Appendix A-2.
The Intelligent Serial Scheduling consisting of two Mamdani models, as previously explained, is analyzed next. The first Mamdani model to be considered is the Step 1 (Main Scheduling Phase) which is illustrated in the following figure:

![Figure 5 – 1 Intelligent Serial Scheduling Mamdani model 1](image)

As previously explained, the inputs and outputs defined for Step 1 are considered in the Mamdani model. As the system takes the If – Then rules to execute an action, surface plots resulting from different input combinations are obtained and are shown in Appendix C – Section 1.

The values obtained from the plots can be also obtained from the antecedent – consequent rules.

As mentioned earlier in this Chapter, several scenarios are considered in order to demonstrate the functionality of all of the proposed methods.
The second Mamdani model to be considered is Step 2 (Finishing Phase) which is illustrated in the following figure:

![Figure 5 – 2 Intelligent Serial Scheduling Mamdani model 2](image1)

Moreover, the surface plots resulting from different input combinations for Step 2 for the Intelligent Serial Scheduling are presented in Appendix C – Section 2.

Next, the Mamdani model for Step 1 in the Intelligent Parallel Scheduling method is presented below:

![Figure 5 – 3 Intelligent Parallel Scheduling Mamdani model 1](image2)
The Mamdani Model for Step 2 of the Intelligent Parallel Scheduling is also illustrated next:

![Intelligent Parallel Scheduling Mamdani model](image)

**Figure 5 – 4 Intelligent Parallel Scheduling Mamdani model 2**

The corresponding surface plots resulting from different input combinations for Step 1 and 2 for the Intelligent Parallel Scheduling Mamdani Models are presented in Appendix C – Section 3 and 4 respectively.

As the Mamdani models for all the systems corresponding to the Intelligent Serial Scheduling and the Intelligent Parallel Scheduling have been presented; next, the Intelligent Branching Scheduling is shown below.

The Mamdani model for the Intelligent Branching Scheduling is:
In this case, the surface plots resulting from different input combinations for the previous model are presented in Appendix C – Section 5.

The Intelligent Utility Index, the fourth proposed new method has the following model. Again, the input combination plots are given in Appendix C – Section 6. The Mamdani system for the Intelligent Utility Index is presented below.
Lastly, the Intelligent Schedule Performance Indicator Mamdani model is shown. The corresponding surface plots resulting from different input combinations are provided in Appendix C – Section 7.

![Diagram of Intelligent Schedule Performance Indicator Mamdani model]

**Figure 5 – 7 Intelligent Schedule Performance Indicator Mamdani model**

Once the above figures are presented, the experimental results are given for each FIS method next. The experimental results provided are from three different scenarios for each particular method; that is, different values of inputs have been considered and the observed results demonstrate the proposed methodology effectiveness and functionality.

### 5.2. Fuzzy Inference Systems (FIS) Experimental Results

As previously mentioned, three different scenarios are given for each method. For the different scenarios, values for the inputs are given considering
some of the most common situations used today when trying to schedule activities in a given project. Each scenario includes a brief description of the situation, along with the values for each case, and the resulting outputs. This is done in order to demonstrate that the proposed methodology is able to schedule the activities considering all the parameters that also play a role in the traditional methods, as explained in Chapter 4. Also, a Table displaying the values of inputs and outputs is provided to clearly illustrate the results. Also, a Graphical User Interface (GUI), [12], has been developed to show the results instantaneously for the considered and any other scenarios and make the process highly interactive. Note that this allows for the user for any scenario to vary the inputs in some sliders and to obtain a corresponding output. The developed GUI’s also serve to demonstrate and confirm the presented Tables results for any scenario.

5.3. Intelligent Serial Scheduling 1st Scenario

Scenario #1:

Consider some activities have been selected and prioritized for scheduling and put into the decision set. The resource requirement for each activity has also been defined and supposing the process has just started, there are no precedence relations. Therefore the following is assumed:

- The decision set has one or more activities considered for scheduling.
- There is an activity with the highest priority.
- There are enough resources for the activity to start.
- As the process just started, there are no precedence relations that prevent the activity from being scheduled.

Hence, Table 5 – 1 shows the results when the above parameters are inputs to the Intelligent Serial Scheduling FIS model. As previously mentioned, Step 1 and Step 2 are connected as one of the outputs from Step 1 (Mamdani model 1) indicates to go to Step 2. Consequently, as activities are scheduled, the *in process* set starts having activities, and finally one of the outputs in Step 2 indicates to go back to Step 1.

The outputs give the system a loop process. However, the outputs of the first system (Step 1) are not to be considered directly as inputs, but only as variables for the inputs in the second system (Step 2).

<table>
<thead>
<tr>
<th>Input Parameter</th>
<th>Input Parameter</th>
<th>Output Parameter</th>
<th>Deffuzified Value</th>
<th>Resulting Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision Set</td>
<td>Some Activities</td>
<td>Action</td>
<td>5.06</td>
<td>Schedule</td>
</tr>
<tr>
<td>Priority</td>
<td>Highest Priority</td>
<td>Process</td>
<td>5</td>
<td>Go to Step 2</td>
</tr>
<tr>
<td>Resources</td>
<td>Enough</td>
<td>Activity List</td>
<td>3.97</td>
<td>Update Activity List</td>
</tr>
<tr>
<td>Relation</td>
<td>No Precedence</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 5 – 1 Intelligent Serial Scheduling 1st Scenario Step 1 results**
From the above results, the outputs indicate the following: schedule the activity; update the activity list (meaning that another activity now has the highest priority), go to Step 2. Hence the following is obtained:

<table>
<thead>
<tr>
<th>Input Parameter</th>
<th>Input</th>
<th>Output Parameter</th>
<th>Deffuzified Output Value</th>
<th>Resulting Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>In Process Set</td>
<td>Some Activities</td>
<td>Resources</td>
<td>-4.04</td>
<td>None Released</td>
</tr>
<tr>
<td>Activity Progress</td>
<td>Not Completed</td>
<td>Activity Progress</td>
<td>7</td>
<td>Go to Step 1</td>
</tr>
</tbody>
</table>

Table 5 – 2 Intelligent Serial Scheduling 1st Scenario Step 2 results

From the above Table, the results from Step 2 are obtained. As inputs, the FIS system considers the in process set, which is where activities scheduled are located (from Step 1), and the progress of activities in there. As for this 1st scenario, Step 1 scheduled an activity therefore resulting in the in process set having some activities. Also, for the first scenario, the activity progress of the activity previously scheduled is set to be not completed. Thus, no resources from that scheduled activity are released and the output indicates to go to Step 1.

The GUI's for both of the previous systems for this scenario, which show and confirm the Tables results, are the following:
Intelligent Serial Scheduling

System 1

Decision Set

- Empty
- Some Activities
- Many Activities

Action

- Schedule

Output 1

Activity Priority

- Low Priority
- Mid Priority
- Highest Priority

Process

- Go to Step 2

Output 2

Resources

- None
- Not Enough
- Enough

Activity List

- Update List

Output 3

Relation

- No Precedence
- Significant

Figure 5 – 8 Intelligent Serial Scheduling Step 1 Scenario #1 GUI

Intelligent Serial Scheduling

System 2

In Progress Set

- Empty
- Some Activities
- Many Activities

Action

- Go to Step 1

Output 3

Activity Progress

- Not Completed
- Partially Completed
- Completed

Resources

- None Released

Output 1

Figure 5 – 9 Intelligent Serial Scheduling Step 2 Scenario #1 GUI
5.4. Intelligent Serial Scheduling 2nd Scenario

Scenario #2:

For Scenario #2 consider that more activities have been added and again prioritized into the decision set. The resource requirement for each new activity has also been given. For this example, also consider that some activities have already been scheduled; the resources to schedule the next activity with the highest priority are not enough creating a precedence relation until an activity currently scheduled finishes and releases some resources. Therefore the input and output parameters for this Scenario are shown in the following Table. In addition, the GUI in the 2nd Scenario Step 1 is given in Figure 5 – 10.

<table>
<thead>
<tr>
<th>Input Parameter</th>
<th>Input Parameter</th>
<th>Output Parameter</th>
<th>Deffuzified Output Value</th>
<th>Resulting Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision Set</td>
<td>Many Activities</td>
<td>Action</td>
<td>4.34</td>
<td>Do not Schedule</td>
</tr>
<tr>
<td>Priority</td>
<td>Highest Priority</td>
<td>Process</td>
<td>5</td>
<td>Go to Step 2</td>
</tr>
<tr>
<td>Resources</td>
<td>Not Enough</td>
<td>Activity List</td>
<td>6.32</td>
<td>Do not Update Activity List</td>
</tr>
<tr>
<td>Relation</td>
<td>Precedence</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5 – 3 Intelligent Serial Scheduling 2nd Scenario Step 1 results
As no activities have been scheduled, but having one or more previously scheduled and considering the activity previously scheduled is still in progress but almost completed, the values taken for Step 2 in this case are:

<table>
<thead>
<tr>
<th>Input Parameter</th>
<th>Input Parameter</th>
<th>Output Parameter</th>
<th>Deffuzified Output Value</th>
<th>Resulting Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>In Process Set</td>
<td>Many Activities</td>
<td>Resources</td>
<td>1.85</td>
<td>Some Released</td>
</tr>
<tr>
<td>Activity Progress</td>
<td>Partially Completed</td>
<td>Activity Progress</td>
<td>7</td>
<td>Go to Step 1</td>
</tr>
</tbody>
</table>

Table 5 – 4 Intelligent Serial Scheduling 2nd Scenario Step 2 results
Again, in order to show the above results from the previous Tables, the GUI for the second system (Step 2) is presented next.

![Intelligent Serial Scheduling System 2 GUI](image)

**Figure 5 – 11 Intelligent Serial Scheduling Step 2 Scenario #2 GUI**

### 5.5. Intelligent Serial Scheduling 3rd Scenario

Scenario #3:

For the third scenario, consider all activities in the decision set have been scheduled, in other words, the decision set is empty. For this particular case, the values are shown in Table 5 – 3. Note that since the decision set is empty, the system output does not indicate to go to Step 2; and this Step is omitted, (as specified in the If – Then rules). Note that there is no need for resource release, and activities do not affect precedence relations.
Table 5 – 5 Intelligent Serial Scheduling 3rd Scenario Step 1 results

Once again, in order to show the results from the previous Table, the GUI for the system (Step 1) in the third Scenario is presented next.

![Intelligent Serial Scheduling System 1](image-url)
5.6. Intelligent Parallel Scheduling 1st Scenario

Scenario #1:

For the first Scenario in the Intelligent Parallel Scheduling, the same case is considered as in the Serial Intelligent Scheduling for Step 1. Consider some activities have been selected and prioritized for scheduling and put into the decision set. The resource requirement for each activity has also been defined and supposing the process has just started, there are no precedence relations. Therefore the following is assumed:

- The decision set has one or more activities considered for scheduling.
- There is an activity with the highest priority.
- As the process just started, there are enough resources for the activity to start.
- As the process just started, there are no precedence relations that prevent the activity from being scheduled.

Hence, Table 5 – 6 shows the results after inputting the above parameters into the Intelligent Parallel Scheduling. Note that as mentioned before, the same rule applies to Step 1 and Step 2. Step 1 and Step 2 are connected as one of the outputs from Step 1 (Mamdani model 1) directs the system to go to Step 2 and as activities are scheduled, the active set starts having activities, also, one of the outputs in Step 2 directs the system to go back to Step 1, giving the system a loop process, however, as in the Intelligent Serial Scheduling, the outputs of the system 1 (Step 1) are not to be considered
directly as inputs, but only as variables for the inputs in the second system (Step 2).

<table>
<thead>
<tr>
<th>Input Parameter</th>
<th>Input Parameter</th>
<th>Output Parameter</th>
<th>Deffuzified Value</th>
<th>Resulting Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision Set</td>
<td>Some Activities</td>
<td>Action</td>
<td>5.85</td>
<td>Schedule</td>
</tr>
<tr>
<td>Priority</td>
<td>Highest Priority</td>
<td>Process</td>
<td>5</td>
<td>Go to Step 2</td>
</tr>
<tr>
<td>Resources</td>
<td>Enough</td>
<td>Activity List</td>
<td>4.54</td>
<td>Update Activity</td>
</tr>
<tr>
<td>Relation</td>
<td>No Precedence</td>
<td></td>
<td></td>
<td>List</td>
</tr>
</tbody>
</table>

Table 5 – 6 Intelligent Parallel Scheduling 1st Scenario Step 1 results

As previously explained, the Intelligent Parallel Scheduling method might update priority at any time during the scheduling process (which is the case of this Scenario). Therefore, Step 2 checks the activities in the active set, their progress, and if the priority changes. For this case, the following values are given and obtained:
<table>
<thead>
<tr>
<th>Input Parameter</th>
<th>Input Parameter</th>
<th>Deffuzified Output Value</th>
<th>Resulting Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Set</td>
<td>Some Activities</td>
<td>4.9</td>
<td>All Released</td>
</tr>
<tr>
<td>Priority Status</td>
<td>Change Priority</td>
<td>-3.48</td>
<td>Move to Decision Set</td>
</tr>
<tr>
<td>Activity Progress</td>
<td>Partially Completed</td>
<td>7</td>
<td>Go to Step 1</td>
</tr>
<tr>
<td></td>
<td>Priority Action</td>
<td>6.85</td>
<td>Reprioritize</td>
</tr>
</tbody>
</table>

**Table 5 – 7 Intelligent Parallel Scheduling 1st Scenario Step 2 results**

As priority changed, even if the priority had not been completed and was still in the active set, as it is the case in this Scenario, the resulting output indicates to release all resources being used. Then, the activities not completed go back to the decision set, and the process reprioritizes all activities.

Lastly, the system goes back to Step 1 to check for a new activity with the highest priority.

Moreover, the antecedent – consequent rules are presented next. Figure 5 – 13 shows the GUI for the first system (Step 1), and figure 5 – 14 shows the GUI for the second system (Step 2) respectively.
Figure 5 – 13 Intelligent Parallel Scheduling Step 1 Scenario #1 GUI

Figure 5 – 14 Intelligent Parallel Scheduling Step 2 Scenario #1 GUI
5.7. Intelligent Parallel Scheduling 2\textsuperscript{nd} Scenario

Scenario #2:

Consider for this Scenario that Step 2 has indicated to reprioritize as priority changed. Also, more activities have been added to the decision set, all resources have been released in consequence of the change of priority, and there are no precedence relations. Then:

\begin{tabular}{|c|c|c|c|c|}
\hline
Input Parameter & Input Parameter & Output Parameter & Deffuzified Output Value & Resulting Output \\
\hline
Decision Set & Many Activities & Action & 5.85 & Schedule \\
\hline
Priority & Highest Priority & Process & 5 & Go to Step 2 \\
\hline
Resources & Enough & Activity List & 3.04 & Update Activity List \\
\hline
Relation & No Precedence & & & \\
\hline
\end{tabular}

\textbf{Table 5 – 8 Intelligent Parallel Scheduling 2\textsuperscript{nd} Scenario Step 1 results}

Once again, an activity has been scheduled and system 1 has indicated the system to go to Step 2. The active set once again has activities and in this Scenario, priority does not change. Also, the activities are considered as not totally completed yet. Thus, the values given and obtained for Step 2 for this Scenario are:
<table>
<thead>
<tr>
<th>Input Parameter</th>
<th>Input</th>
<th>Output Parameter</th>
<th>Deffuzified Output Value</th>
<th>Resulting Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Set</td>
<td>Some Activities</td>
<td>Resources</td>
<td>-3.58</td>
<td>None Released</td>
</tr>
<tr>
<td>Priority Status</td>
<td>Maintain Priority</td>
<td>Activity Status</td>
<td>-0.00438</td>
<td>Maintain in Active Set</td>
</tr>
<tr>
<td>Activity Progress</td>
<td>Not Completed</td>
<td>Process</td>
<td>7</td>
<td>Go to Step 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Priority Action</td>
<td>3.32</td>
<td>Maintain Priority</td>
</tr>
</tbody>
</table>

Table 5 – 9 Intelligent Parallel Scheduling 2nd Scenario Step 2 results

Again, the GUI’s for the systems are presented next.

![Intelligent Parallel Scheduling System 1](image)

Figure 5 – 15 Intelligent Parallel Scheduling Step 1 Scenario #2 GUI
Activities and resources remain the same until either priority changes or activity (ies) is (are) partially or fully completed. In this case, the system output indicates to go back to Step 1 and continue the scheduling.

5.8. Intelligent Parallel Scheduling 3rd Scenario

Scenario #3:

For the third scenario, consider all activities in the decision set have been scheduled, in other words, the decision set is empty and all activities are either in the active or the complete set. Note that as the decision set is empty, the system output indicates not go to Step 2 (as specified in the If – Then rules), as
there is no need for resource release, and activities do not affect precedence relations. Also note that once all activities have left the decision set, priority does not change as all activities would be scheduled.

The following Table displays the given and obtained values:

<table>
<thead>
<tr>
<th>Input Parameter</th>
<th>Input Parameter</th>
<th>Output Parameter</th>
<th>Deffuzified Output Value</th>
<th>Resulting Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision Set</td>
<td>Empty</td>
<td>Action</td>
<td>2.59</td>
<td>Do Not Schedule</td>
</tr>
<tr>
<td>Priority</td>
<td>None</td>
<td>Process</td>
<td>2</td>
<td>Stop Process</td>
</tr>
<tr>
<td>Resources</td>
<td>Not Enough</td>
<td>Activity List</td>
<td>6.82</td>
<td>Do Not Update Activity List</td>
</tr>
<tr>
<td>Relation</td>
<td>Precedence</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5 – 10 Intelligent Serial Scheduling 3rd Scenario Step 1 results

Once again, the GUI for Step 2 in the third Scenario is presented below:
5.9. Intelligent Branching Scheduling 1st Scenario

Scenario #1:

The first Scenario for the Intelligent Branching Scheduling considers that several activities have been placed in the activity set; however, activities have not been selected for scheduling (order and importance). There are no precedence relations yet and the constraints found are soluble. The following Table shows the inputs and outputs for this Scenario:
As no activities were selected, even considering all other parameters, the system requests to select the next activity for scheduling. Once this is done, a different Scenario is considered (Scenario 2).

Also, the Graphical User Interface (GUI) in the next figure is provided to confirm the results from the previous Table.

Once again the GUI serves to confirm that the output delivers the expected results.
5.10. Intelligent Branching Scheduling 2\textsuperscript{nd} Scenario

Scenario #2:

The second Scenario considers the activities in the activity set have been selected. There are no precedence relations and constraints are still soluble. Since there is nothing preventing the activity selected from being scheduled, the system should execute the scheduling membership function. Table 5 – 12 displays the results.
### Table 5 – 12 Intelligent Branching Scheduling 2nd Scenario results

Moreover, the figure confirming the above results is shown below:

![FIS Intelligent Branching](image)

**Figure 5 – 19 Intelligent Branching Scheduling Scenario #2 GUI**
5.11. Intelligent Branching Scheduling 3rd Scenario

Scenario #3:

The third and last Scenario considers once again existing activities in the activity set, also, activities have been selected; however, in this case, there is a precedence relation and the constraints found are defined, thus the results are shown in the Table below:

<table>
<thead>
<tr>
<th>Input Parameter</th>
<th>Input</th>
<th>Output Parameter</th>
<th>Deffuzified Output Value</th>
<th>Resulting Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity Set</td>
<td>Some Activities</td>
<td>Action</td>
<td>-0.794</td>
<td>Branch Activity</td>
</tr>
<tr>
<td>Activity Status</td>
<td>Activities Selected</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relation</td>
<td>Precedence</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Constraints</td>
<td>Defined</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5 – 13 Intelligent Branching Scheduling 3rd Scenario results

The next figure confirms once again the values from the Table above. With this figure, the last scenario for the Intelligent Branching Scheduling is given and therefore, the experimental results for the Fuzzy models have all been presented.
5.12. Intelligent Utility Index 1st Scenario

The first Scenario for the Intelligent Utility Index considers some activities are in the activity set, an activity has been selected to perform the utility index calculation and to find out if resources should be diverted from it or not. The resource requirement for the activity is high and the activity duration is long. According to what was explained earlier, since the resource requirement is high and the activity has a long duration, the system should say that the Utility Index (UI) is high and that resources should not be diverted from this activity. Consequently, a new activity should be selected. To prove this, the inputs and outputs given and obtained are displayed in the following Table.
<table>
<thead>
<tr>
<th>Input Parameter</th>
<th>Input Parameter</th>
<th>Output Parameter</th>
<th>Defuzzified Output Value</th>
<th>Resulting Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity Set</td>
<td>Some Activities</td>
<td>Utility Index</td>
<td>7.5</td>
<td>High</td>
</tr>
<tr>
<td>Activity Status</td>
<td>Activity Selected</td>
<td>Resources</td>
<td>2.44</td>
<td>Do not Divert</td>
</tr>
<tr>
<td>Resource Requirement</td>
<td>High</td>
<td>Process</td>
<td>7.02</td>
<td>Select New Activity</td>
</tr>
<tr>
<td>Activity Duration</td>
<td>Long</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 5 – 14 Intelligent Utility Index 1st Scenario results**

**Intelligent Utility Index**

**Activity Set**
- Empty
- Some Activities
- Many Activities

**Activity Status**
- Not Selected
- Selected

**Resource**
- Low
- Md
- High

**Activity Duration**
- Short
- Medium
- Long

**Utility Index**
- High

**Resources**
- Do not divert

**Process**
- Select New Activity

**Figure 5 – 21 Intelligent Utility Index Scenario #1 GUI**
The figure above serves to confirm the values from Table 5 – 14.

5.13. Intelligent Utility Index 2\textsuperscript{nd} Scenario

The second Scenario for the Intelligent Utility Index considers once again some activities are in the activity set and that an activity has been selected to perform the utility index calculation. This time the resource requirement for the activity is medium and the activity duration is also medium. The system for this case should say that the Utility Index (UI) is medium and that resources should be considered for diverting. Consequently, a new activity should be selected. To prove this, the inputs and outputs given and obtained are displayed in the following Table.

<table>
<thead>
<tr>
<th>Input Parameter</th>
<th>Input</th>
<th>Output Parameter</th>
<th>Deffuzified Output Value</th>
<th>Resulting Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity Set</td>
<td>Some Activities</td>
<td>Utility Index</td>
<td>5.5</td>
<td>Med</td>
</tr>
<tr>
<td>Activity Status</td>
<td>Activity Selected</td>
<td>Resources</td>
<td>5.5</td>
<td>Consider Diverting</td>
</tr>
<tr>
<td>Resource Requirement</td>
<td>Med</td>
<td>Process</td>
<td>7.02</td>
<td>Select New Activity</td>
</tr>
<tr>
<td>Activity Duration</td>
<td>Med</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5 – 15 Intelligent Utility Index 2\textsuperscript{nd} Scenario results
Moreover, the figure below confirms the results from the above Table.

![Intelligent Utility Index Diagram]

**Figure 5 – 22 Intelligent Utility Index Scenario #2 GUI**

**5.14. Intelligent Utility Index 3rd Scenario**

The third and last Scenario for the Intelligent Utility Index considers many activities are in the activity set and that an activity has been selected to perform the utility index calculation. In this instance, the resource requirement for the activity is low and the activity duration is short. The system for this case should say that the Utility Index (UI) is low and that resources should be diverted. Consequently, a new activity should be selected. To prove this, the inputs and outputs given and obtained are displayed in the Table below.
<table>
<thead>
<tr>
<th>Input Parameter</th>
<th>Input Parameter</th>
<th>Deffuzified Output Value</th>
<th>Resulting Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity Set</td>
<td>Many Activities</td>
<td>3.37</td>
<td>Low</td>
</tr>
<tr>
<td>Activity Status</td>
<td>Activity Selected</td>
<td>7.18</td>
<td>Divert</td>
</tr>
<tr>
<td>Resource Requirement</td>
<td>Low</td>
<td>7.11</td>
<td>Select New Activity</td>
</tr>
<tr>
<td>Activity Duration</td>
<td>Short</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5 – 16 Intelligent Utility Index 3rd Scenario results

![Intelligent Utility Index](image)

Figure 5 – 23 Intelligent Utility Index Scenario #3 GUI
Once again the figure above confirms the results.

5.15. Intelligent Schedule Performance Indicator 1st Scenario

The first scenario for the Intelligent Schedule Performance Indicator considers that the project has a due date that is near and activities are delayed, new constraints have arose but are not significant, and no extra resources are needed. According to what was explained earlier, the performance of the current schedule should be bad and rescheduling should be considered. The following Table shows the given inputs and the outputs obtained.

<table>
<thead>
<tr>
<th>Input Parameter</th>
<th>Input</th>
<th>Output Parameter</th>
<th>Deffuzified Output Value</th>
<th>Resulting Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Due Date</td>
<td>Near</td>
<td>Performance</td>
<td>3.49</td>
<td>Bad</td>
</tr>
<tr>
<td>Activity Status</td>
<td>Delayed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>New Constraints</td>
<td>Not Significant</td>
<td>Action</td>
<td>5.84</td>
<td>Consider Rescheduling</td>
</tr>
<tr>
<td>Resources</td>
<td>No Extra Needed</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5 – 17 Intelligent Schedule Performance Indicator 1st Scenario results

Moreover the figure below confirms the above values:
5.16. Intelligent Schedule Performance Indicator 2\textsuperscript{nd} Scenario

The second scenario for the Intelligent Schedule Performance Indicator considers that the project has a due date that is now and activities are on time, no new constraints have arose, and no extra resources are needed. According to what was explained earlier, the performance of the current schedule should be excellent and no further action is required. The following Table shows the given inputs and the outputs obtained, and the figure below confirms the Table values.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Input</th>
<th>Output Parameter</th>
<th>Deffuzified Output Value</th>
<th>Resulting Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Due Date</td>
<td>Now</td>
<td>Performance</td>
<td>8.35</td>
<td>Excellent</td>
</tr>
<tr>
<td>Activity Status</td>
<td>On Time</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>New Constraints</td>
<td>None</td>
<td>Action</td>
<td>3.06</td>
<td>No Action</td>
</tr>
<tr>
<td>Resources</td>
<td>No Extra Needed</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 5 – 18 Intelligent Schedule Performance Indicator 2nd Scenario results**

**Intelligent Schedule Performance Indicator**

**Figure 5 – 25 Intelligent Schedule Performance Indicator Scenario #2 GUI**
5.17. Intelligent Schedule Performance Indicator 3rd Scenario

The third scenario for the Intelligent Schedule Performance Indicator considers that the project has a due date that is now but this time activities are very delayed, significant new constraints have arose, and some extra resources are needed. According to what was explained earlier, the performance of the current schedule should be very bad and immediate rescheduling is required. The following Table shows the given inputs and the outputs obtained, and the figure below confirms the Table values.

<table>
<thead>
<tr>
<th>Input Parameter</th>
<th>Input</th>
<th>Output Parameter</th>
<th>Defuzzified Output Value</th>
<th>Resulting Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Due Date</td>
<td>Now</td>
<td>Performance</td>
<td>1.86</td>
<td>Very Bad</td>
</tr>
<tr>
<td>Activity Status</td>
<td>Very</td>
<td>Delayed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>New Constraints</td>
<td>Significant</td>
<td>Action</td>
<td>7.93</td>
<td>Immediate Rescheduling</td>
</tr>
<tr>
<td>Resources</td>
<td>Some Extra Needed</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5 – 19 Intelligent Schedule Performance Indicator 3rd Scenario results

Once again, the next figure confirms the results obtained.
5.18. Results Analysis

After the proposed novel methodology and the results obtained, several observations can be made.

It has been illustrated how the Fuzzy Inference Systems (FIS) are used to determine the best possible solution for the scheduling of activities in the Resource Constrained Scheduling Problem and closely related problems.

Moreover, the advantages of all systems when it comes to giving a solution when data is uncertain or conditions change rapidly, makes each process easier and possible to implement in any situation for any project in
general. It has been shown that qualitative analysis based on linguistic indicators used during the assessment of constraints during the scheduling process is internally converted to crisp numbers by each one of the FIS models. Moreover, the results of the outputs in all models have proven to be effective, useful, and reliable, as they match the expected outcome when scheduling activities.

In traditional methods, total computational times have always been a struggle (Hang Quang Le, [20], Fest, et al, [8], Hans, et al, [10]); normally, the methods involving multiple activities and constraints generally translate into large waiting times for computing and reliability. By approaching the scheduling problem with the proposed IS methodology, computational time is always kept at minimum having values (on the computer language/platform used) between 0.008 seconds to 0.03 seconds when determining an output as inputs are given. Therefore, Fuzzy Inference Systems (FIS) once more have demonstrated to be extremely reliable and fast in terms of computational effort.

The greatest difference between traditional and the new proposed methods is that in traditional methods the inputs/outputs and parameters are considered in a binary fixed manner such as a “yes” or “no” qualifiers. Whereas, Fuzzy Inference Systems (FIS) allow to have different/diverse qualifiers for each parameter and have an ability to deal with a degree of uncertainty.

There is actually no proper way to directly compare the new proposed methodology to traditional methods; however, an indirect comparison can be made based on:
1. Computational time: Maximum computational time may grow in traditional methods as the number of activities in a project gets larger. However, in the new proposed methods, even if the number of activities is large as well as the number of antecedent – consequent rules, the computer platform to implement the methods provides the user with a result in microseconds.

2. Flexibility: The new proposed methods are more flexible given that they allow the use of uncertain/vague data and the possibility of implementing different values for inputs and outputs.

5.19. Summary

This Chapter presented the entire research results and included different scenarios and examples to demonstrate the effectiveness and reliability of the proposed novel FIS methods. It has been shown the advantages of using Intelligent Systems (IS) techniques when solving the Resource Constrained Scheduling Problem and closely related problems. Taking into account all the methods that are considered for this work, the strength of Fuzzy modeling is confirmed, as well as the stability and reliability of the proposed FIS models. Consequently, it can be stated that the objectives of this research have been reached, as the proposed FIS methods have proven to be accurate, consistent, and a good solution for the complex scheduling problem.
CHAPTER 6 – CONCLUSIONS

The main purpose of this Thesis is to present a novel methodology that leads to original and new approaches to schedule activities and resources under constraints, limited resource capacity, and time frames, as well as other related known problems. To reach this objective, Intelligent Systems (IS) techniques, in particular Fuzzy Inference Systems, from the Soft Computing / Computational Intelligence fields are used and fully evaluated leading to the conclusion that the objectives have been reached. This Thesis can be considered organized in four stages.

The first stage of the Thesis focuses on studying and identifying the Resource Constrained Scheduling Problem (RCSP), as well as the current traditional methods for its solution and their reach. The understanding of current limitations and scheduling components are essential to develop new techniques.

The second stage of the Thesis consists on reviewing Intelligent Systems (IS) techniques (e.g. Fuzzy Inference Systems, FIS) and discuss the advantages that can present for the solution of the RCSP and closely related problems.

The third stage of the Thesis concentrates on the implementation of FIS techniques to develop five original and new approaches to solve the Resource Constrained Scheduling Problem (RCSP) and closely related problems.

Finally, the last stage of the Thesis presents different experimental scenarios to demonstrate the effectiveness and functionality of each developed
method. This leads to favorable results and the confirmation of the proposed methods efficiency.

In general, the advantages of using Fuzzy Inference Systems (FIS) techniques when solving the problems previously discussed are summarized next.

First and most importantly, Fuzzy Inference Systems allow having as many qualifiers for each input and output as desired, while traditional methods are based on exact binary (i.e. yes or no) values, and not leaving any room to consider data uncertainty. For this reason no direct comparison can be made between the proposed methods and their corresponding traditional methods. Still, an indirect comparison could be made in terms of computational complexity and execution time, effectiveness, and flexibility. Although it may appear that the proposed methods increase the computational complexity and execution time; the number of rules for each method is not excessive, and still leads to very fast execution times. Furthermore, in terms of effectiveness and flexibility the proposed methods are superior to their corresponding traditional methods.

In the case of all five methods, the ability to deal with uncertainty and to treat the problems with different grades of inputs and outputs (qualifiers) allows the facilitation of the scheduling process. Dealing with inaccurate data and the ability to set values for the different parameters considered based on the user’s expertise is extremely helpful when solving the scheduling and closely related problems.
Moreover, using Fuzzy Logic and the interface chosen for the elaboration of the methods (Matlab – Mamdani toolbox), creates a user friendly environment and allows the model complexity to be reduced.

Lastly, it is important to note that proposed new methodology is quite easy to understand and to implement in a user friendly environment, and that it is computer/language platform independent.

6.1. Summary of Results

In summary, from the research developed the following can be listed:

- Five new Fuzzy Inference Systems (FIS) methods have been proposed to solve effectively and efficiently the Resource Constrained Scheduling Problem (RCSP) and closely related problems such as Utility Index measurement and Schedule Performance evaluation.
- Seven different FIS models have been properly presented, developed, and tested.
- Analysis of different common Scenarios are given for each and every one of the proposed FIS to observe the stability and reliability of each of the proposed FIS systems.

6.2. Conclusions

Fuzzy Modeling leads to an important and relevant improvement to the solution of the RSCP and closely related problems. In this research, it has been demonstrated that Intelligent Systems (IS) tools, in particular Fuzzy Inference
Systems, have allowed approaching the Resource Constrained Scheduling Problem in a different manner. The strength of Fuzzy Modeling based on human knowledge, dealing with large quantities of information and/or imprecise information, and the ability to include uncertainty through different qualifiers for the different parameters, presents an advantage to improve existing traditional scheduling techniques. Furthermore, it also confirms the importance of integrating user experience and the reach of Fuzzy Inference Systems.

It is important to mention that the results obtained in this research, lead to a new manner for solving the scheduling of resources and activities and for the development of new user friendly tools that enhance current processes and needs.

6.3. Recommendations and Future Work

This research focused on proposing five new IS/FIS methods from the Soft Computing / Computational Intelligence fields. However, some improvements and possible future studies can consider the following issues:

1) The proposed methodology could be improved by including other qualifiers/uncertainty parameters such as uncertain linguistic and/or user judgements.

2) Using the proposed models in some other real-world cases.

3) Other traditional methods currently used for the solution of the RSCP could be also treated with IS/FIS techniques. Among the several methods that refer to resource scheduling, the analytical
techniques, including the implicit enumeration technique and the iterative forward and backward scheduling algorithm can be considered.

4) Examining the effect and results of using other (but appropriate) different or custom membership function shapes.

5) Implementing the proposed methodology in other computer language/platforms that allow its inclusion into mobile computing devices for practical implementations. This implementation would allow expert knowledge to be available at any location.
REFERENCES


APPENDIX A

A. Mamdani Fuzzy Inference Models

The Fuzzy model selected for developing each one of the proposed IS methods is the Mamdani Fuzzy Inference Model. The reason is that it is a general platform that allows considering and defining data from the user's perspective in order to properly identify the processes with non-measurable parameters. Furthermore, it also assumes the real user's control situation, which uses fuzzy judgements. In general, this type of model allows vague linguistic expressions fully functional by integrating it with the system.

Section 1. Types of Membership Functions

Different types of membership functions were considered for the building the proposed FIS systems. Among the different types that can be used the triangular, trapezoidal, generalized bell, generalized gauss, and the sigmoidal functions are selected. In order to have a better idea on why these types of functions are chosen, a brief explanation is made. Moreover, the functions are used for fine-tuning a Fuzzy Inference System (FIS) to achieve desired results. It is also important to note that more types of membership functions are available in the Matlab Fuzzy Logic Toolbox; however, for simplicity and convenience, only the above mentioned functions are used.

- Triangular Membership function: This type of function is nothing more than a collection of three different points forming a triangle.

- Trapezoidal Membership function: Has almost the same characteristics as the triangular function with the difference that this one has a flat top which acts as a truncated curve.
Both of these membership functions have the advantage of and computational efficiency, and they have been widely used, especially in real-time implementations. However, since they only deal with straight lines, they lack smoothness at the corner points specified by the parameters.

![Graphs of membership functions]

a) Triangular membership function, [22], b) Trapezoidal membership function, [22].

- Generalized Bell and Generalized Gaussian Membership functions: These functions are specified by three parameters. They can approach a non-fuzzy set if the free parameter is tuned. Because of its smoothness and concise notation, they are quite popular for specifying fuzzy sets. Both present the advantage of being non-zero at all points. The next figure shows an example of both functions.

![Graphs of membership functions]

c) Gaussian membership function, [22], d) G. Bell membership function, [22].

- Sigmoidal Membership function: This type of function is heavily open to the right or the left; therefore, it is extremely useful when representing concepts such as “very bad” or “excellent”. The next figure shows an example of both functions.
Section 2. Defuzzification Method

The defuzzification method chosen for the obtaining a crisp value from the output of the Fuzzy systems is the Centroid of Area method (commonly named Center of Area). This method returns the center of area under the curve. It is the most commonly used defuzzification method as it effectively calculates the best value between multiple output linguistic terms. It is also fast and does not require high computational effort.

The following figure displays the idea previously mentioned.

Now that the FIS model has been specified, the membership functions to be used have been analysed, and the defuzzification method has been set; then, the proposed new IS methods are presented next.
A-2. FIS Mamdani Models Implementation and Operation in the Matlab Toolbox

Section 1. Building a FIS Model in the Matlab Toolbox

Before presenting the different Mamdani models for each method, it is important to explain how the models were created in the Fuzzy Toolbox in the interface chosen (e.g. Matlab).

The toolbox makes the process interactive at all times, since it does not require code programming or any other scripts, but only requires the user to define parameters by interacting and manipulating the tools. Moreover, two different tools can be found; the editable tools and the view-only tools.

The editable tools include the following:

1) FIS Editor: The FIS editor is where the user defines the names and number of inputs and outputs the system will have as well as the type of defuzzification method. It is important to note that the toolbox does not limit the number of inputs and outputs. However, they may be limited by the available memory of the computer used.
2) Membership Function Editor: It is used to define the shapes and ranges of all the membership functions associated with each variable. The figure below displays the Membership Function Editor.

![Membership Function Editor](image)

3) Rule Editor: This tool allows the user to define the antecedent – consequent rules taken into consideration the output desired. This will define the behaviour of the system. The next figure displays the rule editor.
Once the editable tools have been explained, the view-only tools will be explained below.

Note that two types of view-only tools are found; the rule viewer and the surface viewer.

1) Rule Viewer: The rule viewer allows viewing the fuzzy inference diagram. It is used to see which rules are active and which membership functions influence the results.

2) Surface Viewer: The surface viewer is used to view the result of the different combinations between inputs and outputs. It generates and plots an output for the system. The following figure displays the surface viewer.
Surface Viewer

In order to show to the reader the different inputs and outputs for each one of the different FIS models proposed several figures are provided next. Also, the surface plots resulting from different input combinations for each FIS method are shown in Appendix B.
B. Mamdani Models Fuzzy (if – then) Rules

It is important to remember that the antecedent – consequent rules are what define which output can be obtained depending on the combination of inputs. The different combinations allow finding a solution to every possible situation. Having noted this, the if-then rules for System 1 (Step 1) of the Intelligent Serial Scheduling are presented below.

Section 1.

Intelligent Serial Scheduling (System 1) – Step 1

1. If the decision set is empty, priority is low, there are none resources, and there is no precedence relation, then, stop the process.
2. If the decision set is empty, priority is low, there are none resources, and there is no significant precedence relation, then, stop the process.
3. If the decision set is empty, priority is low, there are none resources, and there is precedence relation, then, stop the process.
4. If the decision set is empty, priority is low, there are not enough resources, and there is no precedence relation, then, stop the process.
5. If the decision set is empty, priority is low, there are not enough resources, and there is no significant precedence relation, then, stop the process.
6. If the decision set is empty, priority is low, there are not enough resources, and there is precedence relation, then, stop the process.
7. If the decision set is empty, priority is low, there are enough resources, and there is no precedence relation, then, stop the process.
8. If the decision set is empty, priority is low, there are enough resources, and there is no significant precedence relation, then, stop the process.
9. If the decision set is empty, priority is low, there are enough resources, and there is precedence relation, then, stop the process.
10. If the decision set is empty, priority is mid, there are none resources, and there is no precedence relation, then, stop the process.
11. If the decision set is empty, priority is mid, there are none resources, and there is no significant precedence relation, then, stop the process.

12. If the decision set is empty, priority is mid, there are none resources, and there is precedence relation, then, stop the process.

13. If the decision set is empty, priority is mid, there are not enough resources, and there is no precedence relation, then, stop the process.

14. If the decision set is empty, priority is mid, there are not enough resources, and there is no significant precedence relation, then, stop the process.

15. If the decision set is empty, priority is mid, there are not enough resources, and there is precedence relation, then, stop the process.

16. If the decision set is empty, priority is mid, there are enough resources, and there is no precedence relation, then, stop the process.

17. If the decision set is empty, priority is mid, there are enough resources, and there is no significant precedence relation, then, stop the process.

18. If the decision set is empty, priority is mid, there are enough resources, and there is precedence relation, then, stop the process.

19. If the decision set is empty, priority is highest, there are none resources, and there is no precedence relation, then, stop the process.

20. If the decision set is empty, priority is highest, there are none resources, and there is no significant precedence relation, then, stop the process.

21. If the decision set is empty, priority is highest, there are none resources, and there is precedence relation, then, stop the process.

22. If the decision set is empty, priority is highest, there are not enough resources, and there is no precedence relation, then, stop the process.

23. If the decision set is empty, priority is highest, there are not enough resources, and there is no significant precedence relation, then, stop the process.

24. If the decision set is empty, priority is highest, there are not enough resources, and there is precedence relation, then, stop the process.

25. If the decision set is empty, priority is highest, there are enough resources, and there is no precedence relation, then, stop the process.
26. If the decision set is empty, priority is highest, there are enough resources, and there is no significant precedence relation, then, stop the process.

27. If the decision set is empty, priority is highest, there are enough resources, and there is precedence relation, then, stop the process.

28. If the decision set has some activities, priority is low, there are none resources, and there is no precedence relation, then, do not schedule, go to step 2, and update activity list.

29. If the decision set has some activities, priority is low, there are none resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and update activity list.

30. If the decision set has some activities, priority is low, there are none resources, and there is precedence relation, then, do not schedule, go to step 2, and update activity list.

31. If the decision set has some activities, priority is low, there are not enough resources, and there is no precedence relation, then, do not schedule, go to step 2, and update activity list.

32. If the decision set has some activities, priority is low, there are not enough resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and update activity list.

33. If the decision set has some activities, priority is low, there are not enough resources, and there is precedence relation, then, do not schedule, go to step 2, and update activity list.

34. If the decision set has some activities, priority is low, there are enough resources, and there is no precedence relation, then, do not schedule, go to step 2, and update activity list.

35. If the decision set has some activities, priority is low, there are enough resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and update activity list.

36. If the decision set has some activities, priority is low, there are enough resources, and there is precedence relation, then, do not schedule, go to step 2, and update activity list.

37. If the decision set has some activities, priority is mid, there are none resources, and there is no precedence relation, then, do not schedule, go to step 2, and update activity list.

38. If the decision set has some activities, priority is mid, there are none resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and update activity list.

39. If the decision set has some activities, priority is mid, there are none resources, and there is precedence relation, then, do not schedule, go to step 2, and update activity list.

40. If the decision set has some activities, priority is mid, there are not enough resources, and there is no precedence relation, then, do not schedule, go to step 2, and update activity list.
41. If the decision set has some activities, priority is mid, there are not enough resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and update activity list.

42. If the decision set has some activities, priority is mid, there are not enough resources, and there is precedence relation, then, do not schedule, go to step 2, and update activity list.

43. If the decision set has some activities, priority is mid, there are enough resources, and there is no precedence relation, then, do not schedule, go to step 2, and update activity list.

44. If the decision set has some activities, priority is mid, there are enough resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and update activity list.

45. If the decision set has some activities, priority is mid, there are enough resources, and there is precedence relation, then, do not schedule, go to step 2, and update activity list.

46. If the decision set has some activities, priority is high, there are none resources, and there is no precedence relation, then, do not schedule, go to step 2, and do not update activity list.

47. If the decision set has some activities, priority is high, there are none resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and do not update activity list.

48. If the decision set has some activities, priority is high, there are none resources, and there is precedence relation, then, do not schedule, go to step 2, and do not update activity list.

49. If the decision set has some activities, priority is high, there are not enough resources, and there is no precedence relation, then, do not schedule, go to step 2, and do not update activity list.

50. If the decision set has some activities, priority is high, there are not enough resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and do not update activity list.

51. If the decision set has some activities, priority is high, there are not enough resources, and there is precedence relation, then, do not schedule, go to step 2, and do not update activity list.

52. If the decision set has some activities, priority is high, there are enough resources, and there is no precedence relation, then, schedule, go to step 2, and update activity list.

53. If the decision set has some activities, priority is high, there are enough resources, and there is no significant precedence relation, then, schedule, go to step 2, and update activity list.

54. If the decision set has some activities, priority is high, there are enough resources, and there is precedence relation, then, do not schedule, go to step 2, and do not update activity list.

55. If the decision set has many activities, priority is low, there are none resources, and there is no precedence relation, then, do not schedule, go to step 2, and update activity list.
56. If the decision set has many activities, priority is low, there are none resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and update activity list.

57. If the decision set has many activities, priority is low, there are none resources, and there is precedence relation, then, do not schedule, go to step 2, and update activity list.

58. If the decision set has many activities, priority is low, there are not enough resources, and there is no precedence relation, then, do not schedule, go to step 2, and update activity list.

59. If the decision set has many activities, priority is low, there are not enough resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and update activity list.

60. If the decision set has many activities, priority is low, there are not enough resources, and there is precedence relation, then, do not schedule, go to step 2, and update activity list.

61. If the decision set has many activities, priority is low, there are enough resources, and there is no precedence relation, then, do not schedule, go to step 2, and update activity list.

62. If the decision set has many activities, priority is low, there are enough resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and update activity list.

63. If the decision set has many activities, priority is low, there are enough resources, and there is precedence relation, then, do not schedule, go to step 2, and update activity list.

64. If the decision set has many activities, priority is mid, there are none resources, and there is no precedence relation, then, do not schedule, go to step 2, and update activity list.

65. If the decision set has many activities, priority is mid, there are none resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and update activity list.

66. If the decision set has many activities, priority is mid, there are none resources, and there is precedence relation, then, do not schedule, go to step 2, and update activity list.

67. If the decision set has many activities, priority is mid, there are not enough resources, and there is no precedence relation, then, do not schedule, go to step 2, and update activity list.

68. If the decision set has many activities, priority is mid, there are not enough resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and update activity list.

69. If the decision set has many activities, priority is mid, there are not enough resources, and there is precedence relation, then, do not schedule, go to step 2, and update activity list.

70. If the decision set has many activities, priority is mid, there are enough resources, and there is no precedence relation, then, do not schedule, go to step 2, and update activity list.
71. If the decision set has many activities, priority is mid, there are enough resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and update activity list.

72. If the decision set has many activities, priority is mid, there are enough resources, and there is precedence relation, then, do not schedule, go to step 2, and update activity list.

73. If the decision set has many activities, priority is high, there are none resources, and there is no precedence relation, then, do not schedule, go to step 2, and do not update activity list.

74. If the decision set has many activities, priority is high, there are none resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and do not update activity list.

75. If the decision set has many activities, priority is high, there are none resources, and there is precedence relation, then, do not schedule, go to step 2, and do not update activity list.

76. If the decision set has many activities, priority is high, there are not enough resources, and there is no precedence relation, then, do not schedule, go to step 2, and do not update activity list.

77. If the decision set has many activities, priority is high, there are not enough resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and do not update activity list.

78. If the decision set has many activities, priority is high, there are not enough resources, and there is precedence relation, then, do not schedule, go to step 2, and do not update activity list.

79. If the decision set has many activities, priority is high, there are enough resources, and there is no precedence relation, then, schedule, go to step 2, and update activity list.

80. If the decision set has many activities, priority is high, there are enough resources, and there is no significant precedence relation, then, schedule, go to step 2, and update activity list.

81. If the decision set has many activities, priority is high, there are enough resources, and there is precedence relation, then, do not schedule, go to step 2, and do not update activity list.

Now that the rules for the first system have been shown, the if-then rules for System 2 (Step 2) of the Intelligent Serial Scheduling are presented. It is also important to remember that the Intelligent Serial Scheduling was divided into two systems creating a loop process between them.
Section 2.

Intelligent Serial Scheduling (System 2) – Step 2

The antecedent – consequent rules for the Intelligent Serial Scheduling is presented next. It is important to remember that this method is divided into two different systems.

1. If the In Process Set has some activities, and activity progress is not completed, then, none resources are released and go to step 1.

2. If the In Process Set has some activities, and activity progress is partially completed, then, some resources are released and go to step 1.

3. If the In Process Set has some activities, and activity progress is completed, then, all resources are released and go to step 1.

4. If the In Process Set has many activities, and activity progress is not completed, then, none resources are released and go to step 1.

5. If the In Process Set has many activities, and activity progress is partially completed, then, some resources are released and go to step 1.

6. If the In Process Set has many activities, and activity progress is completed, then, all resources are released and go to step 1.

7. If the In Process Set has no activities, then, all resources are released and go to step 1.

As mentioned before, System 2 (Step 2) indicates the process to go back to System 1 (Step 1), hence a loop is created. Once the antecedent rules are given for both of the systems corresponding to the Intelligent Serial Scheduling. The if-then rules for System 1 (Step 1) of the Intelligent Parallel Scheduling are presented in Section 3.
Section 3.

As mentioned earlier, the Intelligent Parallel Method was divided into two systems. The first one called the Main Scheduling Phase has the following antecedent – consequent rules.

**Intelligent Parallel Scheduling (System 1) – Step 1**

1. If the decision set is empty, priority is low, there are none resources, and there is no precedence relation, then, stop the process.
2. If the decision set is empty, priority is low, there are none resources, and there is no significant precedence relation, then, stop the process.
3. If the decision set is empty, priority is low, there are none resources, and there is precedence relation, then, stop the process.
4. If the decision set is empty, priority is low, there are not enough resources, and there is no precedence relation, then, stop the process.
5. If the decision set is empty, priority is low, there are not enough resources, and there is no significant precedence relation, then, stop the process.
6. If the decision set is empty, priority is low, there are not enough resources, and there is precedence relation, then, stop the process.
7. If the decision set is empty, priority is low, there are enough resources, and there is no precedence relation, then, stop the process.
8. If the decision set is empty, priority is low, there are enough resources, and there is no significant precedence relation, then, stop the process.
9. If the decision set is empty, priority is low, there are enough resources, and there is precedence relation, then, stop the process.
10. If the decision set is empty, priority is mid, there are none resources, and there is no precedence relation, then, stop the process.
11. If the decision set is empty, priority is mid, there are none resources, and there is no significant precedence relation, then, stop the process.
12. If the decision set is empty, priority is mid, there are none resources, and there is precedence relation, then, stop the process.
13. If the decision set is empty, priority is mid, there are not enough resources, and there is no precedence relation, then, stop the process.
14. If the decision set is empty, priority is mid, there are not enough resources, and there is no significant precedence relation, then, stop the process.

15. If the decision set is empty, priority is mid, there are not enough resources, and there is precedence relation, then, stop the process.

16. If the decision set is empty, priority is mid, there are enough resources, and there is no precedence relation, then, stop the process.

17. If the decision set is empty, priority is mid, there are enough resources, and there is no significant precedence relation, then, stop the process.

18. If the decision set is empty, priority is mid, there are enough resources, and there is precedence relation, then, stop the process.

19. If the decision set is empty, priority is highest, there are none resources, and there is no precedence relation, then, stop the process.

20. If the decision set is empty, priority is highest, there are none resources, and there is no significant precedence relation, then, stop the process.

21. If the decision set is empty, priority is highest, there are none resources, and there is precedence relation, then, stop the process.

22. If the decision set is empty, priority is highest, there are not enough resources, and there is no precedence relation, then, stop the process.

23. If the decision set is empty, priority is highest, there are not enough resources, and there is no significant precedence relation, then, stop the process.

24. If the decision set is empty, priority is highest, there are not enough resources, and there is precedence relation, then, stop the process.

25. If the decision set is empty, priority is highest, there are enough resources, and there is no precedence relation, then, stop the process.

26. If the decision set is empty, priority is highest, there are enough resources, and there is no significant precedence relation, then, stop the process.

27. If the decision set is empty, priority is highest, there are enough resources, and there is precedence relation, then, stop the process.

28. If the decision set has some activities, priority is low, there are none resources, and there is no precedence relation, then, do not schedule, go to step 2, and update activity list.
29. If the decision set has some activities, priority is low, there are none resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and update activity list.

30. If the decision set has some activities, priority is low, there are none resources, and there is precedence relation, then, do not schedule, go to step 2, and update activity list.

31. If the decision set has some activities, priority is low, there are not enough resources, and there is no precedence relation, then, do not schedule, go to step 2, and update activity list.

32. If the decision set has some activities, priority is low, there are not enough resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and update activity list.

33. If the decision set has some activities, priority is low, there are not enough resources, and there is precedence relation, then, do not schedule, go to step 2, and update activity list.

34. If the decision set has some activities, priority is low, there are enough resources, and there is no precedence relation, then, do not schedule, go to step 2, and update activity list.

35. If the decision set has some activities, priority is low, there are enough resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and update activity list.

36. If the decision set has some activities, priority is low, there are enough resources, and there is precedence relation, then, do not schedule, go to step 2, and update activity list.

37. If the decision set has some activities, priority is mid, there are none resources, and there is no precedence relation, then, do not schedule, go to step 2, and update activity list.

38. If the decision set has some activities, priority is mid, there are none resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and update activity list.

39. If the decision set has some activities, priority is mid, there are none resources, and there is precedence relation, then, do not schedule, go to step 2, and update activity list.

40. If the decision set has some activities, priority is mid, there are not enough resources, and there is no precedence relation, then, do not schedule, go to step 2, and update activity list.

41. If the decision set has some activities, priority is mid, there are not enough resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and update activity list.

42. If the decision set has some activities, priority is mid, there are not enough resources, and there is precedence relation, then, do not schedule, go to step 2, and update activity list.

43. If the decision set has some activities, priority is mid, there are enough resources, and there is no precedence relation, then, do not schedule, go to step 2, and update activity list.

127
44. If the decision set has some activities, priority is mid, there are enough resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and update activity list.

45. If the decision set has some activities, priority is mid, there are enough resources, and there is precedence relation, then, do not schedule, go to step 2, and update activity list.

46. If the decision set has some activities, priority is high, there are none resources, and there is no precedence relation, then, do not schedule, go to step 2, and do not update activity list.

47. If the decision set has some activities, priority is high, there are none resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and do not update activity list.

48. If the decision set has some activities, priority is high, there are none resources, and there is precedence relation, then, do not schedule, go to step 2, and do not update activity list.

49. If the decision set has some activities, priority is high, there are not enough resources, and there is no precedence relation, then, do not schedule, go to step 2, and do not update activity list.

50. If the decision set has some activities, priority is high, there are not enough resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and do not update activity list.

51. If the decision set has some activities, priority is high, there are not enough resources, and there is precedence relation, then, do not schedule, go to step 2, and do not update activity list.

52. If the decision set has some activities, priority is high, there are enough resources, and there is no precedence relation, then, schedule, go to step 2, and update activity list.

53. If the decision set has some activities, priority is high, there are enough resources, and there is no significant precedence relation, then, schedule, go to step 2, and update activity list.

54. If the decision set has some activities, priority is high, there are enough resources, and there is precedence relation, then, do not schedule, go to step 2, and do not update activity list.

55. If the decision set has many activities, priority is low, there are none resources, and there is no precedence relation, then, do not schedule, go to step 2, and update activity list.

56. If the decision set has many activities, priority is low, there are none resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and update activity list.

57. If the decision set has many activities, priority is low, there are none resources, and there is precedence relation, then, do not schedule, go to step 2, and update activity list.

58. If the decision set has many activities, priority is low, there are not enough resources, and there is no precedence relation, then, do not schedule, go to step 2, and update activity list.
59. If the decision set has many activities, priority is low, there are not enough resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and update activity list.

60. If the decision set has many activities, priority is low, there are not enough resources, and there is precedence relation, then, do not schedule, go to step 2, and update activity list.

61. If the decision set has many activities, priority is low, there are enough resources, and there is no precedence relation, then, do not schedule, go to step 2, and update activity list.

62. If the decision set has many activities, priority is low, there are enough resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and update activity list.

63. If the decision set has many activities, priority is low, there are enough resources, and there is precedence relation, then, do not schedule, go to step 2, and update activity list.

64. If the decision set has many activities, priority is mid, there are none resources, and there is no precedence relation, then, do not schedule, go to step 2, and update activity list.

65. If the decision set has many activities, priority is mid, there are none resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and update activity list.

66. If the decision set has many activities, priority is mid, there are none resources, and there is precedence relation, then, do not schedule, go to step 2, and update activity list.

67. If the decision set has many activities, priority is mid, there are not enough resources, and there is no precedence relation, then, do not schedule, go to step 2, and update activity list.

68. If the decision set has many activities, priority is mid, there are not enough resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and update activity list.

69. If the decision set has many activities, priority is mid, there are not enough resources, and there is precedence relation, then, do not schedule, go to step 2, and update activity list.

70. If the decision set has many activities, priority is mid, there are enough resources, and there is no precedence relation, then, do not schedule, go to step 2, and update activity list.

71. If the decision set has many activities, priority is mid, there are enough resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and update activity list.

72. If the decision set has many activities, priority is mid, there are enough resources, and there is precedence relation, then, do not schedule, go to step 2, and update activity list.

73. If the decision set has many activities, priority is high, there are none resources, and there is no precedence relation, then, do not schedule, go to step 2, and do not update activity list.
74. If the decision set has many activities, priority is high, there are none resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and do not update activity list.

75. If the decision set has many activities, priority is high, there are none resources, and there is precedence relation, then, do not schedule, go to step 2, and do not update activity list.

76. If the decision set has many activities, priority is high, there are not enough resources, and there is no precedence relation, then, do not schedule, go to step 2, and do not update activity list.

77. If the decision set has many activities, priority is high, there are not enough resources, and there is no significant precedence relation, then, do not schedule, go to step 2, and do not update activity list.

78. If the decision set has many activities, priority is high, there are not enough resources, and there is precedence relation, then, do not schedule, go to step 2, and do not update activity list.

79. If the decision set has many activities, priority is high, there are enough resources, and there is no precedence relation, then, schedule, go to step 2, and do not update activity list.

80. If the decision set has many activities, priority is high, there are enough resources, and there is no significant precedence relation, then, schedule, go to step 2, and do not update activity list.

81. If the decision set has many activities, priority is high, there are enough resources, and there is precedence relation, then, do not schedule, go to step 2, and do not update activity list.
Section 4.

Now, the if-then rules for System 2 (Step 2) of the Intelligent Parallel Scheduling are presented below. System 2, also called the Prioritizing Stage from chapter 4, indicates the system to go back to System 1 (Step 1). The following combination of rules (presented in Section 4) will make this possible.

Intelligent Parallel Scheduling (System 2) – Step 2

1. If the active set is empty, priority status is change priority, and the activity progress is not completed, then resources are all released, process is go to step 1, and priority action is reprioritize.

2. If the active set is empty, priority status is change priority, and the activity progress is partially completed, then resources are all released, process is go to step 1, and priority action is reprioritize.

3. If the active set is empty, priority status is change priority, and the activity progress is completed, then resources are all released, process is go to step 1, and priority action is reprioritize.

4. If the active set is empty, priority status is maintain priority, and the activity progress is not completed, then resources are all released, process is go to step 1, and priority action is maintain priority.

5. If the active set is empty, priority status is maintain priority, and the activity progress is partially completed, then resources are all released, process is go to step 1, and priority action is maintain priority.

6. If the active set is empty, priority status is maintain priority, and the activity progress is completed, then resources are all released, process is go to step 1, and priority action is maintain priority.

7. If the active set has some activities, priority status is change priority, and the activity progress is not completed, then resources are all released, activity status is move to decision set, process is go to step 1, and priority action is reprioritize.

8. If the active set has some activities, priority status is change priority, and the activity progress is partially completed, then resources are all released, activity status is move to decision set, process is go to step 1, and priority action is reprioritize.
9. If the active set has some activities, priority status is change priority, and the activity progress is completed, then resources are all released, activity status is move to complete set, process is go to step 1, and priority action is reprioritize.

10. If the active set has some activities, priority status is maintain priority, and the activity progress is not completed, then resources are none released, activity status is maintain in active set, process is go to step 1, and priority action is maintain priority.

11. If the active set has some activities, priority status is maintain priority, and the activity progress is partially completed, then resources are some released, activity status is maintain in active set, process is go to step 1, and priority action is maintain priority.

12. If the active set has some activities, priority status is maintain priority, and the activity progress is completed, then resources are all released, activity status is move to complete set, process is go to step 1, and priority action is maintain priority.

13. If the active set has many activities, priority status is change priority, and the activity progress is not completed, then resources are all released, activity status is move to decision set, process is go to step 1, and priority action is reprioritize.

14. If the active set has many activities, priority status is change priority, and the activity progress is partially completed, then resources are all released, activity status is move to decision set, process is go to step 1, and priority action is reprioritize.

15. If the active set has many activities, priority status is change priority, and the activity progress is completed, then resources are all released, activity status is move to complete set, process is go to step 1, and priority action is reprioritize.

16. If the active set has many activities, priority status is maintain priority, and the activity progress is not completed, then resources are none released, activity status is maintain in active set, process is go to step 1, and priority action is maintain priority.

17. If the active set has many activities, priority status is maintain priority, and the activity progress is partially completed, then resources are some released, activity status is maintain in active set, process is go to step 1, and priority action is maintain priority.

18. If the active set has many activities, priority status is maintain priority, and the activity progress is completed, then resources are all released, activity status is move to complete set, process is go to step 1, and priority action is maintain priority.
Once the rules are given for the Intelligent Parallel Scheduling, the if-then rules for the Intelligent Branching Scheduling are presented next. For this method, as explained in chapter 4, only one system was considered and the antecedent – consequent results are the following:

Section 5.

Intelligent Branching Scheduling

1. If activity set is empty, activity status is no action, there is no precedence relation, and constraints are soluble, then, stop process.
2. If activity set is empty, activity status is no action, there is no precedence relation, and constraints are defined, then, stop process.
3. If activity set is empty, activity status is no action, there is no precedence relation, and constraints are not found, then, stop process.
4. If activity set is empty, activity status is no action, there is no significant precedence relation, and constraints are soluble, then, stop process.
5. If activity set is empty, activity status is no action, there is no significant precedence relation, and constraints are defined, then, stop process.
6. If activity set is empty, activity status is no action, there is no significant precedence relation, and constraints are not found, then, stop process.
7. If activity set is empty, activity status is no action, there is a precedence relation, and constraints are soluble, then, stop process.
8. If activity set is empty, activity status is no action, there is a precedence relation, and constraints are defined, then, stop process.
9. If activity set is empty, activity status is no action, there is a precedence relation, and constraints are not found, then, stop process.
10. If activity set is empty, activity status is activities selected, there is no precedence relation, and constraints are soluble, then, stop process.
11. If activity set is empty, activity status is activities selected, there is no precedence relation, and constraints are defined, then, stop process.
12. If activity set is empty, activity status is activities selected, there is no precedence relation, and constraints are not found, then, stop process.

13. If activity set is empty, activity status is activities selected, there is no significant precedence relation, and constraints are soluble, then, stop process.

14. If activity set is empty, activity status is activities selected, there is no significant precedence relation, and constraints are defined, then, stop process.

15. If activity set is empty, activity status is activities selected, there is no significant precedence relation, and constraints are not found, then, stop process.

16. If activity set is empty, activity status is activities selected, there is a precedence relation, and constraints are soluble, then, stop process.

17. If activity set is empty, activity status is activities selected, there is a precedence relation, and constraints are defined, then, stop process.

18. If activity set is empty, activity status is activities selected, there is a precedence relation, and constraints are not found, then, stop process.

19. If activity set is empty, activity status is activities not selected, there is no precedence relation, and constraints are soluble, then, stop process.

20. If activity set is empty, activity status is activities not selected, there is no precedence relation, and constraints are defined, then, stop process.

21. If activity set is empty, activity status is activities not selected, there is no precedence relation, and constraints are not found, then, stop process.

22. If activity set is empty, activity status is activities not selected, there is no significant precedence relation, and constraints are soluble, then, stop process.

23. If activity set is empty, activity status is activities not selected, there is no significant precedence relation, and constraints are defined, then, stop process.

24. If activity set is empty, activity status is activities not selected, there is no significant precedence relation, and constraints are not found, then, stop process.

25. If activity set is empty, activity status is activities not selected, there is a precedence relation, and constraints are soluble, then, stop process.

26. If activity set is empty, activity status is activities not selected, there is a precedence relation, and constraints are defined, then, stop process.
27. If activity set is empty, activity status is activities not selected, there is a precedence relation, and constraints are not found, then, stop process.

28. If activity set has some activities, activity status is no action, there is no precedence relation, and constraints are soluble, then, select activities.

29. If activity set has some activities, activity status is no action, there is no precedence relation, and constraints are defined, then, select activities.

30. If activity set has some activities, activity status is no action, there is no precedence relation, and constraints are not found, then, select activities.

31. If activity set has some activities, activity status is no action, there is no significant precedence relation, and constraints are soluble, then, select activities.

32. If activity set has some activities, activity status is no action, there is no significant precedence relation, and constraints are defined, then, select activities.

33. If activity set has some activities, activity status is no action, there is no significant precedence relation, and constraints are not found, then, select activities.

34. If activity set has some activities, activity status is no action, there is a precedence relation, and constraints are soluble, then, select activities.

35. If activity set has some activities, activity status is no action, there is a precedence relation, and constraints are defined, then, select activities.

36. If activity set has some activities, activity status is no action, there is a precedence relation, and constraints are not found, then, select activities.

37. If activity set has some activities, activity status is activities selected, there is no precedence relation, and constraints are soluble, then, schedule activity.

38. If activity set has some activities, activity status is activities selected, there is no precedence relation, and constraints are defined, then, branch activity.

39. If activity set has some activities, activity status is activities selected, there is no precedence relation, and constraints are not found, then, schedule activity.

40. If activity set has some activities, activity status is activities selected, there is no significant precedence relation, and constraints are soluble, then, schedule activity.

41. If activity set has some activities, activity status is activities selected, there is no significant precedence relation, and constraints are defined, then, branch activity.
42. If activity set has some activities, activity status is activities selected, there is no significant precedence relation, and constraints are not found, then, schedule activity.

43. If activity set has some activities, activity status is activities selected, there is a precedence relation, and constraints are soluble, then, branch activity.

44. If activity set has some activities, activity status is activities selected, there is a precedence relation, and constraints are defined, then, branch activity.

45. If activity set has some activities, activity status is activities selected, there is a precedence relation, and constraints are not found, then, branch activity.

46. If activity set has some activities, activity status is activities not selected, there is no precedence relation, and constraints are soluble, then, select activities.

47. If activity set has some activities, activity status is activities not selected, there is no precedence relation, and constraints are defined, then, select activities.

48. If activity set has some activities, activity status is activities not selected, there is no precedence relation, and constraints are not found, then, select activities.

49. If activity set has some activities, activity status is activities not selected, there is no significant precedence relation, and constraints are soluble, then, select activities.

50. If activity set has some activities, activity status is activities not selected, there is no significant precedence relation, and constraints are defined, then, select activities.

51. If activity set has some activities, activity status is activities not selected, there is no significant precedence relation, and constraints are not found, then, select activities.

52. If activity set has some activities, activity status is activities not selected, there is a precedence relation, and constraints are soluble, then, select activities.

53. If activity set has some activities, activity status is activities not selected, there is a precedence relation, and constraints are defined, then, select activities.

54. If activity set has some activities, activity status is activities not selected, there is a precedence relation, and constraints are not found, then, select activities.

55. If activity set has many activities, activity status is no action, there is no precedence relation, and constraints are soluble, then, select activities.

56. If activity set has many activities, activity status is no action, there is no precedence relation, and constraints are defined, then, select activities.
57. If activity set has many activities, activity status is no action, there is no precedence relation, and constraints are not found, then, select activities.
58. If activity set has many activities, activity status is no action, there is no significant precedence relation, and constraints are soluble, then, select activities.
59. If activity set has many activities, activity status is no action, there is no significant precedence relation, and constraints are defined, then, select activities.
60. If activity set has many activities, activity status is no action, there is no significant precedence relation, and constraints are not found, then, select activities.
61. If activity set has many activities, activity status is no action, there is a precedence relation, and constraints are soluble, then, select activities.
62. If activity set has many activities, activity status is no action, there is a precedence relation, and constraints are defined, then, select activities.
63. If activity set has many activities, activity status is no action, there is a precedence relation, and constraints are not found, then, select activities.
64. If activity set has many activities, activity status is activities selected, there is no precedence relation, and constraints are soluble, then, schedule activity.
65. If activity set has many activities, activity status is activities selected, there is no precedence relation, and constraints are defined, then, branch activity.
66. If activity set has many activities, activity status is activities selected, there is no precedence relation, and constraints are not found, then, schedule activity.
67. If activity set has many activities, activity status is activities selected, there is no significant precedence relation, and constraints are soluble, then, schedule activity.
68. If activity set has many activities, activity status is activities selected, there is no significant precedence relation, and constraints are defined, then, branch activity.
69. If activity set has many activities, activity status is activities selected, there is no significant precedence relation, and constraints are not found, then, schedule activity.
70. If activity set has many activities, activity status is activities selected, there is a precedence relation, and constraints are soluble, then, branch activity.
71. If activity set has many activities, activity status is activities selected, there is a precedence relation, and constraints are defined, then, branch activity.
72. If activity set has many activities, activity status is activities selected, there is a precedence relation, and constraints are not found, then, branch activity.

73. If activity set has many activities, activity status is activities not selected, there is no precedence relation, and constraints are soluble, then, select activities.

74. If activity set has many activities, activity status is activities not selected, there is no precedence relation, and constraints are defined, then, select activities.

75. If activity set has many activities, activity status is activities not selected, there is no precedence relation, and constraints are not found, then, select activities.

76. If activity set has many activities, activity status is activities not selected, there is no significant precedence relation, and constraints are soluble, then, select activities.

77. If activity set has many activities, activity status is activities not selected, there is no significant precedence relation, and constraints are defined, then, select activities.

78. If activity set has many activities, activity status is activities not selected, there is no significant precedence relation, and constraints are not found, then, select activities.

79. If activity set has many activities, activity status is activities not selected, there is a precedence relation, and constraints are soluble, then, select activities.

80. If activity set has many activities, activity status is activities not selected, there is a precedence relation, and constraints are defined, then, select activities.

81. If activity set has many activities, activity status is activities not selected, there is a precedence relation, and constraints are not found, then, select activities.

Now that the if – then rules of the Intelligent Branching Scheduling have been given, the next step is to present the if-then rules for Intelligent Utility Index. This method, like the previous one considers only one system.
Now that the if – then rules of the Intelligent Branching Scheduling have been given, the next step is to present the if-then rules for Intelligent Utility Index. This method, like the previous one considers only one system.

Section 6.

Intelligent Utility Index

1. If activity set is empty, then, stop process.

2. If activity set has some activities and activity status is not selected, then, process is select new activity.

3. If activity set has many activities and activity status is not selected, then, process is select new activity.

4. If activity set has some activities, activity status is selected, resource requirement is low, and activity duration is low, then, utility index is low, resources are diverted, and process is select new activity.

5. If activity set has some activities, activity status is selected, resource requirement is low, and activity duration is medium, then, utility index is low, resources is consider diverting, and process is select new activity.

6. If activity set has some activities, activity status is selected, resource requirement is low, and activity duration is long, then, utility index is low, resources is do not divert, and process is select new activity.

7. If activity set has some activities, activity status is selected, resource requirement is mid, and activity duration is short, then, utility index is mid, resources are diverted, and process is select new activity.

8. If activity set has some activities, activity status is selected, resource requirement is mid, and activity duration is medium, then, utility index is low, resources is consider diverting, and process is select new activity.

9. If activity set has some activities, activity status is selected, resource requirement is mid, and activity duration is long, then, utility index is mid, resources is do not divert, and process is select new activity.
10. If activity set has some activities, activity status is selected, resource requirement is high, and activity duration is short, then, utility index is high, resources is consider diverting, and process is select new activity.

11. If activity set has some activities, activity status is selected, resource requirement is high, and activity duration is medium, then, utility index is high, resources is do not divert, and process is select new activity.

12. If activity set has some activities, activity status is selected, resource requirement is high, and activity duration is long, then, utility index is high, resources is do not divert, and process is select new activity.

13. If activity set has some activities, activity status is selected, resource requirement is low, and activity duration is low, then, utility index is low, resources are diverted, and process is select new activity.

14. If activity set has some activities, activity status is selected, resource requirement is low, and activity duration is medium, then, utility index is low, resources is consider diverting, and process is select new activity.

15. If activity set has some activities, activity status is selected, resource requirement is low, and activity duration is long, then, utility index is low, resources is consider diverting, and process is select new activity.

16. If activity set has some activities, activity status is selected, resource requirement is mid, and activity duration is short, then, utility index is mid, resources are diverted, and process is select new activity.

17. If activity set has some activities, activity status is selected, resource requirement is mid, and activity duration is medium, then, utility index is low, resources is consider diverting, and process is select new activity.

18. If activity set has some activities, activity status is selected, resource requirement is mid, and activity duration is long, then, utility index is mid, resources is do not divert, and process is select new activity.

19. If activity set has some activities, activity status is selected, resource requirement is high, and activity duration is short, then, utility index is high, resources is consider diverting, and process is select new activity.
20. If activity set has some activities, activity status is selected, resource requirement is high, and activity duration is medium, then, utility index is high, resources is do not divert, and process is select new activity.

21. If activity set has some activities, activity status is selected, resource requirement is high, and activity duration is long, then, utility index is high, resources is do not divert, and process is select new activity.

The last system is the Intelligent Schedule Performance indicator and the if-then rules for the system are presented below. Once this is presented, the next step is to present the surface plots representing the different combinations of the given rules.

Section 7.

Intelligent Schedule Performance Indicator

1. If due date is now, activity status is on time, new constraints are none, and resources are none extra needed, then, performance is excellent and action is no action.

2. If due date is now, activity status is on time, new constraints are none, and resources are some extra needed, then, performance is very good and action is no action.

3. If due date is now, activity status is on time, new constraints are none, and resources are many extra needed, then, performance is good and action is consider rescheduling.

4. If due date is now, activity status is on time, new constraints are not significant, and resources are none extra needed, then, performance is very good and action is no action.

5. If due date is now, activity status is on time, new constraints are not significant, and resources are some extra needed, then, performance is very good and action is no action.

6. If due date is now, activity status is on time, new constraints are not significant, and resources are many extra needed, then, performance is good and action is consider rescheduling.

7. If due date is now, activity status is on time, new constraints are significant, and resources are none extra needed, then, performance is good and action is consider rescheduling.

8. If due date is now, activity status is on time, new constraints are significant, and resources are some extra needed, then, performance is good and action is immediate rescheduling.

9. If due date is now, activity status is on time, new constraints are significant, and resources are many extra needed, then, performance is good and action is immediate rescheduling.
10. If due date is now, activity status is delayed, new constraints are none, and resources are none extra needed, then, performance is bad and action is consider rescheduling.

11. If due date is now, activity status is delayed, new constraints are none, and resources are some extra needed, then, performance is bad and action is consider rescheduling.

12. If due date is now, activity status is delayed, new constraints are none, and resources are many extra needed, then, performance is very bad and action is immediate rescheduling.

13. If due date is now, activity status is delayed, new constraints are not significant, and resources are none extra needed, then, performance is bad and action is consider rescheduling.

14. If due date is now, activity status is delayed, new constraints are none, and resources are some extra needed, then, performance is very bad and action is consider rescheduling.

15. If due date is now, activity status is delayed, new constraints are not significant, and resources are many extra needed, then, performance is very bad and action is immediate rescheduling.

16. If due date is now, activity status is delayed, new constraints are significant, and resources are none extra needed, then, performance is very bad and action is immediate rescheduling.

17. If due date is now, activity status is delayed, new constraints are significant, and resources are some extra needed, then, performance is very bad and action is immediate rescheduling.

18. If due date is now, activity status is delayed, new constraints are significant, and resources are many extra needed, then, performance is very bad and action is immediate rescheduling.

19. If due date is now, activity status is very delayed, new constraints are none, and resources are none extra needed, then, performance is very bad and action is immediate rescheduling.

20. If due date is now, activity status is very delayed, new constraints are none, and resources are some extra needed, then, performance is very bad and action is immediate rescheduling.

21. If due date is now, activity status is very delayed, new constraints are none, and resources are many extra needed, then, performance is very bad and action is immediate rescheduling.

22. If due date is now, activity status is very delayed, new constraints are not significant, and resources are none extra needed, then, performance is very bad and action is immediate rescheduling.

23. If due date is now, activity status is very delayed, new constraints are not significant, and resources are some extra needed, then, performance is very bad and action is immediate rescheduling.
24. If due date is now, activity status is very delayed, new constraints are not significant, and resources are many extra needed, then, performance is very bad and action is immediate rescheduling.

25. If due date is now, activity status is very delayed, new constraints are significant, and resources are none extra needed, then, performance is very bad and action is immediate rescheduling.

26. If due date is now, activity status is very delayed, new constraints are significant, and resources are some extra needed, then, performance is very bad and action is immediate rescheduling.

27. If due date is now, activity status is very delayed, new constraints are significant, and resources are many extra needed, then, performance is very bad and action is immediate rescheduling.

28. If due date is near, activity status is on time, new constraints are none, and resources are none extra needed, then, performance is excellent and action is no action.

29. If due date is near, activity status is on time, new constraints are none, and resources are some extra needed, then, performance is excellent and action is no action.

30. If due date is near, activity status is on time, new constraints are none, and resources are many extra needed, then, performance is very good and action is no action.

31. If due date is near, activity status is on time, new constraints are not significant, and resources are none extra needed, then, performance is excellent and action is no action.

32. If due date is near, activity status is on time, new constraints are not significant, and resources are some extra needed, then, performance is very good and action is no action.

33. If due date is near, activity status is on time, new constraints are not significant, and resources are many extra needed, then, performance is very good and action is consider rescheduling.

34. If due date is near, activity status is on time, new constraints are significant, and resources are none extra needed, then, performance is good and action is consider rescheduling.

35. If due date is near, activity status is on time, new constraints are significant, and resources are some extra needed, then, performance is good and action is consider rescheduling.

36. If due date is near, activity status is on time, new constraints are significant, and resources are many extra needed, then, performance is good and action is consider rescheduling.

37. If due date is near, activity status is delayed, new constraints are none, and resources are none extra needed, then, performance is bad and action is consider rescheduling.

38. If due date is near, activity status is delayed, new constraints are none, and resources are some extra needed, then, performance is bad and action is consider rescheduling.
39. If due date is near, activity status is delayed, new constraints are none, and resources are many extra needed, then, performance is very bad and action is immediate rescheduling.

40. If due date is near, activity status is delayed, new constraints are not significant, and resources are none extra needed, then, performance is bad and action is consider rescheduling.

41. If due date is near, activity status is delayed, new constraints are not significant, and resources are some extra needed, then, performance is bad and action is consider rescheduling.

42. If due date is near, activity status is delayed, new constraints are not significant, and resources are many extra needed, then, performance is very bad and action is immediate rescheduling.

43. If due date is near, activity status is delayed, new constraints are significant, and resources are none extra needed, then, performance is very bad and action is immediate rescheduling.

44. If due date is near, activity status is delayed, new constraints are significant, and resources are some extra needed, then, performance is very bad and action is immediate rescheduling.

45. If due date is near, activity status is delayed, new constraints are significant, and resources are many extra needed, then, performance is very bad and action is immediate rescheduling.

46. If due date is near, activity status is very delayed, new constraints are none, and resources are none extra needed, then, performance is very bad and action is immediate rescheduling.

47. If due date is near, activity status is very delayed, new constraints are none, and resources are some extra needed, then, performance is very bad and action is immediate rescheduling.

48. If due date is near, activity status is very delayed, new constraints are none, and resources are many extra needed, then, performance is very bad and action is immediate rescheduling.

49. If due date is near, activity status is very delayed, new constraints are not significant, and resources are none extra needed, then, performance is very bad and action is immediate rescheduling.

50. If due date is near, activity status is very delayed, new constraints are not significant, and resources are some extra needed, then, performance is very bad and action is immediate rescheduling.

51. If due date is near, activity status is very delayed, new constraints are not significant, and resources are many extra needed, then, performance is very bad and action is immediate rescheduling.

52. If due date is near, activity status is very delayed, new constraints are significant, and resources are none extra needed, then, performance is very bad and action is immediate rescheduling.
53. If due date is near, activity status is very delayed, new constraints are significant, and resources are some extra needed, then, performance is very bad and action is immediate rescheduling.

54. If due date is near, activity status is very delayed, new constraints are significant, and resources are many extra needed, then, performance is very bad and action is immediate rescheduling.

55. If due date is far, activity status is on time, new constraints are none, and resources are none extra needed, then, performance is excellent and action is no action.

56. If due date is far, activity status is on time, new constraints are none, and resources are some extra needed, then, performance is excellent and action is no action.

57. If due date is far, activity status is on time, new constraints are none, and resources are many extra needed, then, performance is very good and action is no action.

58. If due date is far, activity status is on time, new constraints are not significant, and resources are none extra needed, then, performance is excellent and action is no action.

59. If due date is far, activity status is on time, new constraints are not significant, and resources are some extra needed, then, performance is excellent and action is no action.

60. If due date is far, activity status is on time, new constraints are not significant, and resources are many extra needed, then, performance is very good and action is consider rescheduling.

61. If due date is far, activity status is on time, new constraints are significant, and resources are none extra needed, then, performance is very good and action is consider rescheduling.

62. If due date is far, activity status is on time, new constraints are significant, and resources are some extra needed, then, performance is good and action is consider rescheduling.

63. If due date is far, activity status is on time, new constraints are significant, and resources are many extra needed, then, performance is good and action is consider rescheduling.

64. If due date is far, activity status is delayed, new constraints are none, and resources are none extra needed, then, performance is bad and action is consider rescheduling.

65. If due date is far, activity status is delayed, new constraints are none, and resources are some extra needed, then, performance is bad and action is consider rescheduling.

66. If due date is far, activity status is delayed, new constraints are none, and resources are many extra needed, then, performance is very bad and action is consider rescheduling.

67. If due date is far, activity status is delayed, new constraints are not significant, and resources are none extra needed, then, performance is bad and action is consider rescheduling.
68. If due date is far, activity status is delayed, new constraints are not significant, and resources are some extra needed, then, performance is bad and action is consider rescheduling.

69. If due date is far, activity status is delayed, new constraints are not significant, and resources are many extra needed, then, performance is bad and action is consider rescheduling.

70. If due date is far, activity status is delayed, new constraints are significant, and resources are none extra needed, then, performance is bad and action is consider rescheduling.

71. If due date is far, activity status is delayed, new constraints are significant, and resources are some extra needed, then, performance is very bad and action is consider rescheduling.

72. If due date is far, activity status is delayed, new constraints are significant, and resources are many extra needed, then, performance is very bad and action is immediate rescheduling.

73. If due date is far, activity status is very delayed, new constraints are none, and resources are none extra needed, then, performance is very bad and action is immediate rescheduling.

74. If due date is far, activity status is very delayed, new constraints are none, and resources are some extra needed, then, performance is very bad and action is immediate rescheduling.

75. If due date is far, activity status is very delayed, new constraints are none, and resources are many extra needed, then, performance is very bad and action is immediate rescheduling.

76. If due date is far, activity status is very delayed, new constraints are not significant, and resources are none extra needed, then, performance is very bad and action is immediate rescheduling.

77. If due date is far, activity status is very delayed, new constraints are not significant, and resources are some extra needed, then, performance is very bad and action is immediate rescheduling.

78. If due date is far, activity status is very delayed, new constraints are not significant, and resources are many extra needed, then, performance is very bad and action is immediate rescheduling.

79. If due date is far, activity status is very delayed, new constraints are significant, and resources are none extra needed, then, performance is very bad and action is immediate rescheduling.

80. If due date is far, activity status is very delayed, new constraints are significant, and resources are some extra needed, then, performance is very bad and action is immediate rescheduling.

81. If due date is far, activity status is very delayed, new constraints are significant, and resources are many extra needed, then, performance is very bad and action is immediate rescheduling.

Once the if-then rules for the FIS systems have been presented, the next step is to show the corresponding surface plots for each system (Appendix B).
C. Mamdani Models Surface Plots

It is important to remember that the images display the surface plots for each method. These plots are the visual representation of the results obtained from the different combinations of inputs and outputs.

Furthermore, the surface plots for System 1 (Step 1) of the Intelligent Serial Scheduling are presented below. Note that the figures have the name of the parameters being considered. The bottom left, and bottom right names represent the inputs while the middle left name represents the output.

Section 1.

Intelligent Serial Scheduling (System 1) – Step 1
Now that the surface plots for the first system have been shown, the plots for System 2 (Step 2) of the Intelligent Serial Scheduling are presented.

Section 2.

Intelligent Serial Scheduling (System 2) – Step 2

Once the Intelligent Serial Scheduling plots are given, the surface plots for System 1 (Step 1) of the Intelligent Parallel Scheduling are presented below.
Section 3.

The same criteria applies for the following figures. The bottom left, and bottom right names represent the inputs while the middle left name represents the output. Once again, keep in mind that the Intelligent Serial Scheduling and the Intelligent Parallel Scheduling were divided into two different systems.

**Intelligent Parallel Scheduling (System 1) – Step 1**
Consequently, the surface plots for System 2 (Step 2) of the Intelligent Parallel Scheduling are presented next.
Section 4.

As mentioned earlier, the Intelligent Parallel Method was divided into two systems. The antecedent – consequent surface plots for the second system are shown below.

Intelligent Parallel Scheduling (System 2) – Step 2
In addition, the surface plots for the Intelligent Branching Scheduling are presented in Section 5.

Section 5.

Once again the plots from the different input and output combinations are presented below for the Intelligent Branching Scheduling. As before, it is important to remember that this method considers only one system.

Intelligent Branching Scheduling
The next step is to present the surface plots for Intelligent Utility Index and they are presented below. Like the previous one, this method considers also one single system.

Section 6.

Intelligent Utility Index
Section 7.

The last system is the Intelligent Schedule Performance Indicator and the surface plots for this system are given in Section 7. As explained before the bottom left, and bottom right names represent the inputs while the middle left name represents the output, and this applies to all methods and images.

**Intelligent Schedule Performance Indicator**
APPENDIX D

D. Experimental Results If-Then Rules

This Appendix presents the if-then rules for each one of the scenarios. The following figures confirm the results obtained in Chapter 5. Note that as the input values are moved, the system delivers an output. This value is the one that the GUI takes into consideration to give a result.

Section 1. Intelligent Serial Scheduling Experimental Results

Section 1 presents the if-then rules confirming the results in Chapter 5 for the Intelligent Serial Scheduling. It is important to remember that the Intelligent Serial Scheduling is divided into two systems; therefore, two figures with if-then rules will be shown for each scenario except for the third scenario where the output indicates the system to stop. The first two images presented next correspond to Scenario #1.

Intelligent Serial Scheduling Step 1 Scenario #1 If – Then rules
Intelligent Serial Scheduling Step 2 Scenario #1 If – Then rules

The next two images presented next correspond to Scenario #2.

Intelligent Serial Scheduling Step 1 Scenario #2 If – Then rules
Intelligent Serial Scheduling Step 2 Scenario #2 If – Then rules

The last image displays the if-then rules for Scenario #3 in the Intelligent Serial Scheduling.

Intelligent Serial Scheduling Step 1 Scenario #3 If – Then rules
Section 2. Intelligent Parallel Scheduling Experimental Results

Section 2 presents the if-then rules confirming the results in Chapter 5 for the Intelligent Parallel Scheduling. It is important to remember that just like the Intelligent Serial Scheduling, the Intelligent Parallel Scheduling is divided into two systems; therefore, two figures with if-then rules will be shown for each scenario except for the third scenario where the output indicates the system to stop. The first two images presented next correspond to Scenario #1.

Intelligent Parallel Scheduling Step 1 Scenario #1 If – Then rules

Intelligent Parallel Scheduling Step 2 Scenario #1 If – Then rules
The next two images presented next correspond to Scenario #2.

Intelligent Parallel Scheduling Step 1 Scenario #2 If – Then rules

Intelligent Parallel Scheduling Step 2 Scenario #2 If – Then rules
The last image displays the if-then rules for Scenario #3 in the Intelligent Parallel Scheduling.

### Intelligent Parallel Scheduling Step 1 Scenario #3 If – Then rules

#### Section 3. Intelligent Branching Scheduling Experimental Results

Section 3 presents the if-then rules confirming the results in Chapter 5 for the Intelligent Branching Scheduling. It is important to remember that this method considers only one system; therefore, one figure with if-then rules will be shown for each scenario. The first figure presented corresponds to Scenario #1, the second one corresponds to Scenario #2.
Intelligent Branching Scheduling Scenario #1 If – Then rules

The next figure corresponds to Scenario #2.

Intelligent Branching Scheduling Scenario #2 If – Then rules
The last image displays the if-then rules for Scenario #3 in the Intelligent Branching Scheduling.

Section 4. Intelligent Utility Index Experimental Results

Section 4 presents the if-then rules confirming the results in Chapter 5 for the Intelligent Utility Index. It is important to remember that this method considers only one system; therefore, one figure with if-then rules will be shown for each scenario. The first figure presented corresponds to Scenario #1, the second one corresponds to Scenario #2.
Intelligent Utility Index Scenario #1 If – Then rules

Intelligent Utility Index Scenario #2 If – Then rules
The last image displays the if-then rules for Scenario #3 in the Intelligent Utility Index.

**Section 5. Intelligent Schedule Performance Indicator Experimental Results**

Section 5 presents the if-then rules confirming the results in Chapter 5 for the Intelligent Schedule Performance Indicator. It is important to remember that this method considers only one system; therefore, one figure with if-then rules will be shown for each scenario. The first figure presented corresponds to Scenario #1; the second one corresponds to Scenario #2, and the third one corresponds to Scenario #3.
Intelligent Schedule Performance Indicator Scenario #1

If – Then rules

Intelligent Schedule Performance Indicator Scenario #2

If – Then rules
Intelligent Schedule Performance Indicator Scenario #3 If – Then rules