A THREE-PHASE APPROACH FOR ROBUST PROJECT SCHEDULING : AN APPLICATION FOR R&D PROJECT SCHEDULING

by

CANAN ÇAPA

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of the requirements for the degree of
Master of Science

Sabanci University

August, 2013

A THREE-PHASE APPROACH FOR ROBUST PROJECT SCHEDULING :
AN APPLICATION FOR R&D PROJECT SCHEDULING

Approved by:

Prof. Dr. Gündüz Ulusoy ………………………..
(Thesis Supervisor)

Assoc. Prof. Dr. Güvenç Şahin ………………………..

Assoc. Prof. Dr. Selçuk Çolak ………………………..

Assoc. Prof. Dr. Cem Güneri ………………………..

Assist. Prof. Dr. Kemal Kılıç ………………………..

Date of Approval: …13/08/2013...

ACKNOWLEDGEMENTS

I would like to thank all people who have helped and inspired me during my thesis study. I especially want to thank my thesis advisor Prof. Dr. Gündüz Ulusoy for his guidance and support from the initial level to the end. Without his support and assistance this research would have never existed. Assist. Prof. Dr. Kemal Kılıç deserve special thanks for his support and assistance during my thesis. I am especially grateful to my colleagues and dear friends Umut Beşikci and Fikri Küçüksayacıgil for their helpful advices, endless support and friendship. They were always there for me whenever I needed their help. I would also thank to my dear friends, Merve Keskin, Ekin Köker, Utku Olgun and Ali Çetin Suyabatmaz for their valuable friendship and moral support through my thesis study. I would also like to thank TUBITAK for providing me the financial support. I am also thankful to the employees of the R&D Department for providing me support in obtaining the real R&D project data during the implementation of the developed approach. I offer my regards and blessings to all of those who supported me in any respect during the completion of my thesis. Finally, my deepest gratitude goes to my family for their love and support throughout my life.

GÜRBÜZ PROJE ÇİZELGELEME İÇİN ÜÇ-AŞAMALI YAKLAŞIM :
AR-GE PROJE ÇİZELGELEME İÇİN BİR UYGULAMA

Canan ÇAPA

Endüstri Mühendisliği, Yüksek Lisans Tezi, 2013
Tez Danışmanı: Prof. Dr. Gündüz Ulusoy

Anahtar Kelimeler: Veri Madenciliği, Araştırma & Geliştirme, Proaktif Proje Çizelgeleme, Reaktif Proje Çizelgeleme, Çok-Amaçlı Genetik Algoritma

ÖZET

Projelerin yürütülmesi süresince, özellikle çoklu proje ortamında, proje planlarının ve bütçelerinin sapmasına, dolayısı ile de teslim tarihlerinin gecikmesine, kaynakların boş beklemesine, iç envanterin ve sistem oynaklığının artmasına sebep olan beklenmeyen durumlarla karşılaşılır. Bu tezde, Türkiye'de lider bir ev eşyası şirketinin Ar-Ge Merkezi'ndeki rastsal ve dinamik proje çizelgeleme ortamı kaynak kısıtlı çoklu proje çizelgeleme problemi olarak ele alınmış ve Ar-Ge projelerinin çizelgelenmesi için veri madenciliği ile proje çizelgeleme tekniklerini harmanlayan üç-aşamalı bir yaklaşım önerilmiştir. Proje aktiviteleri bölünebilir olup birbirleri arasında genel öncelik ilişkileri vardır. Birinci aşamada, projelerin kaynak kullanımlarındaki sapmalara göre sınıflandırılması ve aktivitelerin kaynak kullanım sapmalarının tahmini için modeller geliştirilmiştir. İkinci aşamada, iki-amaçlı genetik algoritma kullanan iki adet proje çizelgeleme yaklaşımı önerilmiştir. Önerilen iki-amaçlı genetik algoritmanın amaçları; genel proje tamamlanma süresini minimize etmek ve aktivitelerin olası gerçekleşme durumlarında başlangıç zamanlarından sapmalarının toplamını minimize etmektir. İkinci aşama birinci aşamanın çıktısını kullanarak baskın alternatif çözümler türetmektedir. Reaktif aşama olarak adlandırılan üçüncü aşama ise, proje planlarını etkileyen beklenmeyen bir durum olduğunda, proje ana çizelgelerini revize eder ve proje yöneticilerinin "eğer-ise analizleri" yapmalarına olanak sağlayarak gerçekleşebilecek riskler karşısında daha iyi önlem planları hazırlamalarına olanak sağlar.

A THREE-PHASE APPROACH FOR ROBUST PROJECT SCHEDULING: AN APPLICATION FOR R&D PROJECT SCHEDULING

Canan ÇAPA

Industrial Engineering, MS Thesis, 2013
Thesis Supervisor: Prof. Dr. Gündüz Ulusoy

Keywords: Data Mining, Research & Development, Proactive Project Scheduling, Reactive Project Scheduling, Multi-Objective Genetic Algorithm

ABSTRACT

During project execution, especially in a multi-project environment unforeseen events arise that disrupt the project process resulting in deviations of project plans and budgets due to missed due dates and deadlines, resource idleness, higher work-in-process inventory and increased system nervousness. In this thesis, we consider the preemptive resource constrained multi-project scheduling problem with generalized precedence relations in a stochastic and dynamic environment and develop a three-phase model incorporating data mining and project scheduling techniques to schedule the R&D projects of a leading home appliances company in Turkey. In Phase I, models classifying the projects with respect to their resource usage deviation levels and an activity deviation assignment procedure are developed using data mining techniques. Phase II, proactive project scheduling phase, proposes two scheduling approaches using a bi-objective genetic algorithm (GA). The objectives of the bi-objective GA are the minimization of the overall completion time of projects and the minimization of the total sum of absolute deviations for starting times for possible realizations leading to solution robust baseline schedules. Phase II uses the output of the first phase to generate a set of non-dominated solutions. Phase III, called the reactive phase, revises the baseline schedule when a disruptive event occurs and enables the project managers to make “what-if analysis” and thus to generate a set of contingency plans for better preparation.

TABLE OF CONTENTS

ÖZET vi

ABSTRACT vii

LIST OF FIGURES x

LIST OF TABLES xi

INTRODUCTION 1

1.1. PROJECT MANAGEMENT AND SCHEDULING 1
1.2. RESOURCE CONSTRAINED PROJECT SCHEDULING 3
1.3. MULTI-PROJECT SCHEDULING WITH MULTI-SKILLED RESOURCES 3
1.4. MULTI-OBJECTIVE PROJECT SCHEDULING 4
1.5. UNCERTAINTY IN PROJECT SCHEDULING 4
1.6. ROBUST PROJECT SCHEDULING 5
1.7. SUBJECT AND THE ORGANIZATION OF THE THESIS 6

PROJECT SCHEDULING LITERATURE REVIEW 8

2.1. PROJECT SCHEDULING PROBLEM 9
2.2. SOLUTION PROCEDURES FOR THE SINGLE OBJECTIVE RCPSP AND ITS EXTENSIONS 12
2.3. SOLUTION PROCEDURES FOR THE MULTI-OBJECTIVE RCPSP 33
2.4. DEALING WITH UNCERTAINTY IN RCPSP DOMAIN 45
2.5. RISK ANALYSIS 52

PROBLEM DEFINITION AND ENVIRONMENT 54

3.1. R&D PROJECT SCHEDULING 54
3.2. PROJECT MANAGEMENT ENVIRONMENT OF THE R&D DEPARTMENT 56
3.3. PROBLEM DEFINITION AND FORMULATION 64

A THREE-PHASE APPROACH FOR ROBUST PROJECT SCHEDULING 69

4.1. PHASE I: DEVIATION ANALYSIS PHASE 70
4.2. PHASE II: PROACTIVE PROJECT SCHEDULING WITH A BI-OBJECTIVE GA 79
4.3. PHASE III: REACTIVE PROJECT SCHEDULING PHASE 92

DEVIATION ANALYSIS WITH REAL DATA 97

5.1. DATA 97
5.2. STEP I: DEVIATION ANALYSIS OF PROJECTS 105
5.3. STEP II: ACTIVITY DEVIATION ASSIGNMENT PROCEDURE 113

PROACTIVE PROJECT SCHEDULING WITH REAL DATA 122

6.1. FINE-TUNING OF THE BI-OBJECTIVE GA PARAMETERS 123
6.2. DATA 131
6.3. RESULTS OBTAINED WITH THE SINGLE PROJECT SCHEDULING APPROACH 135
6.4. RESULTS OBTAINED WITH MULTI-PROJECT SCHEDULING APPROACH 144
6.5. COMPARISON OF THE RESULTS OBTAINED WITH SINGLE PROJECT AND MULTI-PROJECT SCHEDULING APPROACHES 159

REACTIVE PROJECT SCHEDULING WITH REAL DATA 161
7.1. BASIC SCHEME OF THE IMPLEMENTATION ROUTINE 162
7.2. DISRUPTIONS 163
7.3. IMPLEMENTATION 169
CONCLUSION AND FUTURE WORK 186
BIBLIOGRAPHY 189
APPENDIX A. Project-Based Non-Dominated GA Parameters 204
APPENDIX B. Best GA Parameters in Terms of Single Performance Measures 205
APPENDIX C. Activity-Resource Schedule for Project 08-040 Obtained with Single Project Scheduling Approach and Fitness Calculation Procedure1 206
APPENDIX D. Resource Work Schedules for Project 08-040 Obtained with Single Project Scheduling Approach and Fitness Calculation Procedure1 208

LIST OF FIGURES

Figure 3. 1. Organizational Chart of R&D Department 57

Figure 4. 1. Pseudocode for Population Management of Proposed Bi-objective GA 83
Figure 4. 2. Pseudocode for Fitness Calculation Procedure 85
Figure 4. 3. Crowding Distance of the i^{th} Solution (Deb et.al., 2002)... 88
Figure 4. 4. Pseudocode for Crowding Distance Calculation (Deb et.al., 2002) 89
Figure 4. 5. Gantt Chart of an Example Robust Baseline Schedule 94
Figure 4. 6. Gantt Chart of an Example Repaired Schedule1: Case 1 94
Figure 4. 7. Gantt Chart of an Example Repaired Schedule2: Case 1 95
Figure 4. 8. Gantt Chart of an Example Repaired Schedule1: Case 2 95
Figure 4. 9. Gantt Chart of an Example Repaired Schedule2: Case 2 96

Figure 5. 1. Number of Projects in Each Project Deviation Class 102
Figure 5. 2. Number of Activities in Each Project Deviation Class 102
Figure 5. 3. Distribution the NHD - Test, Measurement and Analysis Combination 114
Figure 5. 4. Frequency Distribution for Activity Class “Meeting and Reporting” 116
Figure 5. 5. Frequency Distribution for “Test-Measurement” activity class- 50% PHD, 50% PLD Project Deviation Class Combination 117
Figure 5. 6. Probability Distribution for “Test-Measurement” Activity Class- 50% PHD, 50% PLD Project Deviation Class Combination 117

Figure 6. 1. Hypervolume Values in Two Different Non-domination Fronts 125
Figure 6. 2. Illustration of the Performance Metric Denoted as HER 126
Figure 6. 3. Pseudocode for Fine-Tuning of the GA Parameters 128
Figure 6. 4. Project Network of Project 08-040 136

Figure 7. 1. Pseudocode of the Implementation Routine Developed 162
Figure 7. 2. Project Network of the Test Project 08-024 170

LIST OF TABLES

Table 5. 1. Input Features Used in Feature Subset Selection 99
Table 5. 2. Results of Feature Subset Selection Analysis 100
Table 5. 3. Activity Statistics 103
Table 5. 4. Project Deviation Class Based Activity Statistics 104
Table 5. 5. Activities' Project Deviation Class Statistics Based on Their Deviation Type 105
Table 5. 6. Classification Results for the Numeric Output 107
Table 5. 7. Classification Results for the Nominal Output ..... **Error! Bookmark not defined.**
Table 5. 8. Probabilistic Classification Results for Numeric Output 109
Table 5. 9. Performances of Probabilistic Results for the Deviation Level Prediction 110
Table 5. 10. Average Variability Results of the Classification Approaches 112
Table 5. 11. Frequency and Probability Information for the NHD-Test, Measurement and Analysis Class Combination 114
Table 5. 12. Frequency Information for the Activity Class "Meeting and Reporting" 115
Table 5. 13. Activity Deviation Assignment Results for the Actual Project Deviation Classes 118
Table 5. 14. Activity Deviation Assignment Results with Deterministic Project Deviation Class Prediction 119
Table 5. 15. Activity Deviation Assignment Results with Probabilistic Project Deviation Class Prediction 119
Table 5. 16. The Probabilities of Activities' Tendency of Having Negative and Positive Deviations for Two Different Projects 120

Table 6. 1. Parameter Values Tested in Selection of the Best Parameter Combinations 124
Table 6. 2. Non-dominated GA Parameters Obtained with the Fine-Tuning Procedure 129
Table 6. 3. Normalized Performance Measure Values for the Non-dominated GA Parameters 129
Table 6. 4. Overall Weighted Scores for each Non-dominated Parameter Combination 130
Table 6. 5. GA Parameters Used in the Implementation 130
Table 6. 6. The Number of Projects by Their Initiation Year 131
Table 6. 7. Total Number of Activities for Each Project the Project Set 132
Table 6. 8. Performance Values of Non-Dominated Baseline Schedules Obtained with Single Project Scheduling Approach and Fitness Calculation Procedure1 136
Table 6. 9. Robust Activity Schedule of Project 08-040 Obtained with Single Project Scheduling Approach and Fitness Calculation Procedure1 137

Table 6. 10. Robust Baseline Schedules of Projects Obtained with Single Project Scheduling Approach and Fitness Calculation Procedure1 138
Table 6. 11. Performance Values of Non-Dominated Baseline Schedules Obtained with Single Project Scheduling Approach and Fitness Calculation Procedure2 139
Table 6. 12. Robust Activity Schedule of Project 08-040 Obtained with Single Project Scheduling Approach and Fitness Calculation Procedure2 140
Table 6. 13. Robust Baseline Schedules of Projects Obtained with Single Project Scheduling Approach and Fitness Calculation Procedure2 141
Table 6. 14. Performance Comparison of Results obtained with Single Project Scheduling Approaches 143
Table 6. 15. Expected Completion Times of Active Projects Scheduled with Multi-Project Scheduling Approach and Fitness Calculation Procedure1 when Project 08-035 Initiated 145
Table 6. 16. Activity Schedule of Project 08-035 Obtained with Multi-Project Scheduling Approach and Fitness Calculation Procedure1 146
Table 6. 17. Schedule Change of Existing Activities Affected by the Initiation of Project 08-040 Scheduled with Multi-Project Scheduling Approach and Fitness Calculation Procedure1 147
Table 6. 18. Effect of Initiation of Project 08-040 to the Completion Times of Existing Projects Scheduled with Multi-Project Scheduling Approach and Fitness Calculation Procedure1 148
Table 6. 19. Final Robust Baseline Schedules of Projects Obtained with Multi-Project Scheduling Approach and Fitness Calculation Procedure1 148
Table 6. 20. Robust Baseline Schedules of Projects Obtained with Multi-Project Scheduling Approach and Fitness Calculation Procedure1 When Each Project is Initiated 149
Table 6. 21. Performance Values of Projects Obtained with Multi-Project Scheduling Approach and Fitness Calculation Procedure1 149
Table 6. 22. Final Robust Baseline Schedules of Projects Obtained Using the Multi-Project Scheduling Approach with Fitness Calculation Procedure1 and Divided TSAD Strategy 150
Table 6. 23. Robust Baseline Schedules of Projects Obtained Using Multi-Project Scheduling Approach with Fitness Calculation Procedure1 and Divided TSAD Strategy When Each Project is Initiated 151
Table 6. 24. Performance Values of Projects Obtained with Multi-Project Scheduling Approach with Fitness Calculation procedure1 and Divided TSAD Strategy 151
Table 6. 25. Expected Completion Times of Active Projects Scheduled with Multi-Project Scheduling Approach and Fitness Calculation Procedure2 When Project 08-035 Initiated 152
Table 6. 26. Performance Values Obtained with Multi-Project Scheduling Approach when Project 08-035 Initiated 153
Table 6. 27. Activity Schedule of Project 08-035 Obtained with Multi-Project Scheduling Approach and Fitness Calculation Procedure2 153
Table 6. 28. Schedule Change of Existing Activities Affected by the Initiation of Project 08-040 Scheduled with Multi-Project Scheduling Approach and Fitness Calculation Procedure2 154
Table 6. 29. Effect of Initiation of Project 08-040 to the Completion Times of Existing Projects Scheduled with Multi-Project Scheduling Approach and Fitness Calculation Procedure2 154

Table 6. 30. Final Robust Baseline Schedules of Projects Obtained with Multi-Project Scheduling Approach and Fitness Calculation Procedure2 155
Table 6. 31. Robust Baseline Schedules of Projects Obtained with Multi-Project Scheduling Approach and Fitness Calculation Procedure2 When Each Project is Initiated 156
Table 6. 32. Performance Values of Projects Obtained with Multi-Project Scheduling Approach and Fitness Calculation Procedure2.. 156
Table 6. 33. Final Robust Baseline Schedules of Projects Obtained with Multi-Project Scheduling Approach and Fitness Calculation Procedure2 and Divided TSAD Strategy 157
Table 6. 34. Robust Baseline Schedules of Projects Obtained with Multi-Project Scheduling Approach and Fitness Calculation Procedure2 with Divided TSAD Strategy When Each Project is Initiated 157
Table 6. 35. Performance Values of Projects Obtained with Multi-Project Scheduling Approach with Fitness Calculation Procedure2 and Divided TSAD Strategy 158
Table 6. 36. Final Completion Times of Projects Obtained with Suggested Approaches 159

Table 7. 1. Robust Baseline Activity Schedules for Project 08-024 170
Table 7. 2. Robust Baseline Activity-Resource Schedules for Project 08-024 171
Table 7. 3. Affected Activity Schedules Obtained in Repaired Schedule1 after Type 1 Disruption 172
Table 7. 4. Affected Activity Schedules Obtained in Repaired Schedule2 after Type 1 Disruption 173
Table 7. 5. Schedule Change Statistics after Type 1 Disruption Occurs... 173
Table 7. 6. Affected Activity Schedules Obtained in Repaired Schedule1 after Type 2 Disruption 174
Table 7. 7. Affected Activity Schedules Obtained in Repaired Schedule2 after Type 2 Disruption 175
Table 7. 8. Schedule Change Statistics after Type 2 Disruption Occurs... 175
Table 7. 9. Affected Activity Schedules Obtained in Repaired Schedule1 after Type 3 Disruption 176
Table 7. 10. Affected Activity Schedules Obtained in Repaired Schedule2 after Type 3 Disruption 177
Table 7. 11. Schedule Change Statistics after Type 3 Disruption Occurs... 177
Table 7. 12. Affected Activity Schedules Obtained in Repaired Schedule1 after Type 4 Disruption 178
Table 7. 13. Affected Activity Schedules Obtained in Repaired Schedule2 after Type 4 Disruption 179
Table 7. 14. Schedule Change Statistics after Type 4 Disruption Occurs... 179
Table 7. 15. Affected Activity Schedules Obtained in Repaired Schedule1 after Type 5 Disruption 180
Table 7. 16. Affected Activity Schedules Obtained in Repaired Schedule2 after Type 5 Disruption 180
Table 7. 17. Schedule Change Statistics after Type 5 Disruption Occurs... 181
Table 7. 18. Affected Activity Schedules Obtained in Repaired Schedule1 after Type 6 Disruption 182
Table 7. 19. Affected Activity Schedules Obtained in Repaired Schedule2 after Type 6 Disruption 182
Table 7. 20. Schedule Change Statistics after Type 6 Disruption Occurs... 182
Table 7. 21. A Possible Project Execution Scenario Obtained with Implementation Routine................. 184
Table 7. 22. Event Explanations of the Possible Project Execution Scenario... 185

CHAPTER 1

INTRODUCTION

The research field of proactive-reactive project scheduling has been recently emerging. Before giving the related literature (Chapter 2) and a rigorous definition of the proactive-reactive project scheduling problem (Chapter 3), this chapter gives a short introduction to the concepts of project scheduling. The first section is devoted to the concepts of project management and project scheduling. Afterwards, the standard problem in project scheduling is shortly revisited. Following the multi-project scheduling and multi-skilled resources concepts, multi-objective nature of the scheduling problem is explained. Then, a short introduction of uncertainty in project scheduling and robust project scheduling, the sub-domain of project management that is the subject of this thesis, is given. We conclude this chapter with the explanation of the subject and the organization of this thesis.

1.1. PROJECT MANAGEMENT AND SCHEDULING

In all sectors of the economy, an appreciable amount of work is accomplished through managing projects. Among other factors, as a result of the global expansion of the information technology (IT) sector and the increase in research and development

(R&D) and engineering services, project management finds more use in practice as a management paradigm. Industrial projects have started to face tight time and resource constraints due to shortening lifetime and increasing competition as well as globalization of markets. Project management is a complex decision making process involving the unrelenting pressures of time and cost. The growing interest in the field of project management has resulted in many new theories, techniques and computer applications to support project managers in achieving their objectives. However, most projects are obviously still far from being perfectly managed.

A project management problem typically consists of planning and scheduling decisions. The planning decision is essentially a strategic process wherein planning for requirements of several resource types in every time period of the planning horizon is carried out. Usually, a Gantt chart of projects is developed to generate resource profiles and perform the required leveling of resources by hiring, firing, subcontracting, and allocating overtime resources. On the other hand, project scheduling is the part of project management that involves the construction of a baseline schedule which specifies the precedence and resource feasible start times for each activity. Such a schedule helps to visualize the project and is a starting point for both internal and external planning and communication. Careful project scheduling has been shown to be an important factor to improve the success rate of the project. Scheduling of activities by considering the resource constraints is a major task which is complex and challenging. During the last five decades, project management and scheduling became one of the most important directions in both research and practice of operations management, or, more generally, in operational research. This follows from an extremely large-scale of practical situations in which some structured sets of activities have to be processed using various scarce resources. During these years, the methodology of project scheduling has been developing constantly; trying to model adequately new practical problems arising, and to solve efficiently the resulting optimization problems.

1.2. RESOURCE CONSTRAINED PROJECT SCHEDULING

Most research done in resource-constrained project scheduling concentrates on minimizing the project duration in either deterministic or stochastic environments. The resource-constrained project scheduling problem (RCPSP) aims to minimize the duration of a project subject to finish-start zero-lag precedence constraints and renewable resource constraints in a deterministic environment. As stated among others by Özdamar and Ulusoy (1995), in general, the project scheduling problem is NP complete, meaning that there are no known algorithms for finding optimal solutions in polynomial time. Exhaustive search methods can be used to solve scheduling problems, but requires forbiddingly long execution times as the problem size increases. Blazewicz et al. (1983) have shown that the RCPSP is NP-hard in the strong sense. Many exact and heuristic algorithms have been described in the literature to construct workable baseline schedules that solve the deterministic RCPSP.

1.3. MULTI-PROJECT SCHEDULING WITH MULTI-SKILLED RESOURCES

A group of organizations, called project based organizations perform almost all their work through projects and operate in general on more than one project simultaneously. R&D organizations in particular and large construction companies execute multi-project scheduling procedures regularly. It has been suggested by Payne (1995) that up to 90%, by value, of all projects occurs in a multi-project context. As markets become more competitive, firms' obligation to simultaneously carry out multiple projects by managing scarce resources becomes even more critical. All these projects are interrelated since the same pool of resources are employed to execute them. Moreover, this pool is generally not homogeneous, i.e. each resource has its own specialization, making the resource allocation task for projects more difficult. It is of particular importance for service firms compared to manufacturing firms, where the labor intensity is higher and multi-skilled resources are more common.

1.4. MULTI-OBJECTIVE PROJECT SCHEDULING

Project scheduling problem is clearly a multi-objective problem since managers in real life deal with several objectives at once. There does not just exist a single criterion by which the success of a particular schedule can be measured. Rather, there exist multiple criteria to be satisfied or achieved. Frequently, the objectives to be achieved over these criteria are in conflict with each other. In this case, there does not exist a single ideal solution, which simultaneously satisfies the decision-maker across all criteria. Therefore the need for multi-objective optimization arises. The goal of multi-objective optimization is to find the single solution giving the *best compromise* between multiple objectives. Since usually there is no single solution that optimizes simultaneously all the objectives, selection of the best compromise solution requires taking into account preferences of the decision-maker. The most popular technique for multi-objective optimization is to obtain Pareto optimal solutions. A Pareto optimal set of solutions is a set of solutions that are non-dominated with respect to each other. A solution is called non-dominated if none of the objective functions can be improved in value without impairment in some of the other objective values. The objective for project scheduling may be based on time, such as to minimize the project duration, or on economic aspects, such as to minimize the project cost. However, success relative to time does not imply success in economic terms. Often, time-based objectives are in conflict with cost-based objectives. For that reason, solution procedures making use of the multi-objective optimization techniques represent a significant need especially in a multi-project environment.

1.5. UNCERTAINTY IN PROJECT SCHEDULING

The vast majority of the research efforts in project scheduling assume complete information about the scheduling problem to be solved and a static deterministic environment within which the pre-computed baseline schedule will be executed. In reality the situation is dynamic in the sense that new projects arrive continuously and stochastic in terms of inter-arrival times and work content. Furthermore, the projects are subject to risks. During project execution, unforeseen events, which disrupt the project plans, result in higher costs due to missed due dates and deadlines, resource idleness, higher work-in-process inventory, and increased system nervousness due to frequent rescheduling. As stated in Elmaghraby (2005), the common practice of dealing with uncertainties by taking deterministic averages of the estimated parameters might lead to serious fallacies. These research approaches on project scheduling involving risk do not model risks explicitly, but try to evaluate the risk of schedule and/or budget overruns using stochastic models for activity durations and/or cost.

1.6. ROBUST PROJECT SCHEDULING

In real-life stochastic project settings, not only activity crashing but also delaying the activity behind its planned schedule start time, might induce a cost. Constant rescheduling in order to improve the expected makespan, might strongly decrease the predictive value of the baseline schedule. Project schedules should also include some solution robustness to cope with the uncertainties during project execution such that the realized project schedule, i.e. the list of actually realized activity start times during project execution, will not differ too much from the original baseline schedule. In general terms, a baseline schedule that is rather ‘insensitive’ to disruptions that may occur during project execution is called robust. Many different types of robustness have been identified in the literature. In this thesis, we consider only solution robustness. Solution robustness aims at constructing a schedule that differs from the realized schedule in the least possible amount. Constructing solution robust schedules requires proactive and reactive scheduling techniques. With the risk information on hand, proactive scheduling aims at the construction of a protected initial schedule (baseline or predictive schedule) that anticipates possible future disruptions by exploiting statistical knowledge of uncertainties that have been detected and analyzed in the project planning phase. A change in the starting times of such activities could lead to infeasibilities at the organizational level or penalties in the form of higher costs. A possible measure for the deviation between the baseline schedule and the realized schedule is the weighted instability cost. It can be calculated by taking the sum of the expected weighted absolute deviations between the planned and the actually realized activity starting times. The weight w_i assigned to each activity i, reflects the activity’s importance of starting it at its planned starting time in the baseline schedule.

Minimizing instability then means looking for a schedule, which is able to accommodate disruptions without too much change in the activity starting times. Reactive scheduling, on the other hand, consists of defining a procedure to react to disruptions that cannot be absorbed by the baseline schedule.

1.7. SUBJECT AND THE ORGANIZATION OF THE THESIS

The literature on robust project scheduling is relatively scarce. To the best of our knowledge, still, there is no study on the preemptive version of the proactive–reactive multi-objective resource constrained multi-project scheduling problem considering uncertainty. In this thesis, we concentrated on the development of an efficient proactive-reactive solution approach for scheduling the R&D projects in a stochastic and dynamic environment present in the R&D department of a leading home appliances company in Turkey. A tree-phase model is developed incorporating data mining and project scheduling techniques to schedule these R&D projects. In Phase I, to consider the resource usage deviations of the projects as a risk measure in the proposed model, the projects are classified into four groups making use of feature subset selection, clustering and classification analysis. After project deviation level prediction, activities are classified into six groups and percentage resource usage deviation assignment procedure is developed to predict the deviation levels of activities. Phase II, the proactive project scheduling phase, proposes two scheduling approaches using a bi-objective genetic algorithm (GA). This bi-objective GA uses the output of the first phase with the aim of generating solution robust baseline project schedules. The objectives of the bi-objective GA are considering the possible resource requirement deviations of activities beforehand to minimize the completion time of projects and to minimize the total instability costs of projects. Phase III, using the scheduled order repair heuristic developed, aims at rescheduling the disrupted project plans.

Next chapter provides the literature review on the project scheduling problem, its extensions and solution methods. In Chapter 3 the problem and the problem environment are defined in detail together with the project scheduling process of the R&D Department under consideration –the owner of the problem. Mathematical formulation of the problem is also provided to formally describe the problem that will be tackled in the remaining chapters. In Chapter 4 we provide the three-phase approach for the problem on hand and in Chapter 5, Chapter 6 and Chapter 7 we present the implementation results of each phase in the suggested model. Finally, in Chapter 8 we conclude and provide suggestions for future work.

CHAPTER 2

PROJECT SCHEDULING LITERATURE REVIEW

In this chapter in order to position the research efforts made in this thesis into the broad project scheduling literature, first, project scheduling problem is introduced with definitions then following the solutions procedures for the single objective resource RCPSP and its extensions, solution procedures for RCPSP are given categorically. When presenting the solution procedures proposed in the literature, our focus was on the genetic algorithm based and evolutionary algorithm based solution procedures for the single objective RCPSP and multi-objective RCPSP, respectively. While going through these sections, we have tried to show how the research direction changed over time. The studies dealing with uncertainty in the project scheduling literature are presented in section 4 of this chapter and in section 5 we have presented the risk analysis studies in the literature.

2.1. PROJECT SCHEDULING PROBLEM

The scheduling of activities by considering the resource constraints and minimizing the completion time of the projects is a major task which is complex and challenging. The application of project scheduling can be seen in different applications such as construction engineering, software development and R&D projects. The project scheduling includes activities, precedence relations of activities, resources and objectives. Activity durations have generally two forms; deterministic and stochastic. In deterministic case, durations are known in advance. In the other case, some probability distributions are employed so as to represent the duration of an activity. In literature, four different precedence relationships exist that can take place between activities. SS (start-start) relationship shows that an activity cannot be started before the start of another activity. SF (start-finish), FS (finish-start) and FF (finish-finish) relationships can be interpreted in the same way. The most common relationship is FS with zero time lag. Minimal (maximal) time lags imply that an amount of time more than (less than) a predetermined level has to take between two events in any of these relationships. Resources can be defined in discrete form such as individual items or manpower or in continuous form such as energy. There are generally three types of resources. Renewable resources, nonrenewable resources and doubly constrained resources. While renewable resources are limited over all periods, nonrenewable resources are limited for all project duration and are consumed as it used by the activities. Doubly constrained resource includes the features of both renewable and nonrenewable resource.

The most frequently used objectives in project scheduling are the minimization of the project duration (makespan) and maximization of the net present value (NPV). Objectives of project scheduling can be classified into two general classes; regular and non-regular objectives. A regular measure of performance is a non-decreasing function of the activity completion times. Makespan minimization is a typical example of regular objectives. A non-regular objective is an objective for which regular objective definition does not hold. A typical example of a non-regular objective is the maximization of NPV. Objectives can also be grouped into four major sections; time-based, resource-based, financial objectives and tradeoff case. Minimization of makespan, total tardiness and quality robustness are some time-based objectives. Quality robustness is defined as the stability power of the completion time of the last activity against randomness.

Variance of the makespan can be regarded as another form of time-based objectives in stochastic project scheduling. Smoothness of the resource usage over project duration, minimization of total resource usage of activities are some resource-based objectives. Maximization of NPV is the most widely used financial objective. Multiple objectives can be any combination of the objectives for RCPSP that are conflicting each other. For example, the maximization of NPV and minimization of the makespan can be considered together as conflicting objectives under certain conditions.

In project scheduling, the projects are modeled as networks. A project network can be defined as a model representing the project through its events, activities, precedence relations among activities and as a planning method considering the duration, the cost, the resources and all other such factors in order to analyze, define and manage the duties. There are two types of network representation of projects: Activity-On-Node (AON) and Activity-On-Arc (AOA) representations. While in AOA representation, arcs represent activities and nodes represent events, in AON representation nodes represent activities and arcs represent precedence relations.

In unconstrained project scheduling problem the activities of a project have to be scheduled in order to achieve an objective without being subject to any resource constraints. After the development of the large scale projects during World War II, many researchers have started to show interest in project scheduling. Initially, the main purpose of the studies was to determine the start times of the activities and to satisfy the precedence relations such that the makespan is minimized without considering resource constraints. This effort led to Critical Path Method (CPM) and Program Evaluation and Review Technique (PERT). While CPM deals with projects which have deterministic task durations, PERT allows the projects to have stochastic task durations and tries to determine the completion time of the project in a probabilistic manner. When resource constraints are taken into consideration, this case gives rise to the RCPSP. Mathematical model of the deterministic RCPSP is given below. In the model below activity durations are assumed to be deterministic but it could be stochastic or fuzzy.

$$Min\ Z = \sum_{t=EFT_J}^{LFT_J} tx_{Jt} \text{ ....} (2.1)$$

s.t.

$$\sum_{t=EFT_j}^{LFT_j} x_{Jt} = 1, j = 1, \ldots .J \text{ ....} (2.2)$$

$$\sum_{t=EFT_i}^{LFT_i} tx_{Jt} \leq \sum_{t=EFT_j}^{LFT_j} (t - d_j) x_{Jt}, j = 2, \ldots .J, i \epsilon Pj \text{ ....} (2.3)$$

$$\sum_{j=1}^{J} k_{jn} \sum_{t=EFT_j}^{LFT_j} x_{Jt} \leq K_n, n = 1, \ldots .N \text{ ....} (2.4)$$

$$\sum_{j=1}^{J} k_{jr} \sum_{\tau=t}^{t+d_j-1} x_{J\tau} \leq K_r, r = 1, \ldots., R, t = 1, \ldots, T \text{ ....} (2.5)$$

$$x_{jt} = \begin{cases} 1, if\ activity\ j\ is\ finished\ at\ the\ end\ of\ period\ t \\ 0,\ otherwise \end{cases} \forall j, t \text{ ....} (2.6)$$

where,

$J =$ number of activities including dummy activities, if any.
$R =$ number of renewable resource types
$j =$ activity index
$r =$ renewable resource index
$t =$ time index
$d_j =$ duration of activity j
$P_j =$ set of immediate predecessors to activity j
$k_{jr} =$ per period renewable resource requirement of resource type r for activity j
$K_r =$ resource limit per period for renewable resource type r
$N =$ the number of non-renewable resources
$K_{jn} =$ non-renewable resource requirement of resource type r for activity j
$K_n =$ non-renewable resource limit per period for renewable resource type n

The objective of the problem is illustrated by expression (2.1). Although in the above formulation a single objective which is the minimization of project makespan is considered, a project scheduling problem can have one or more objectives which can be one of the objectives explained above such as NPV maximization or tardiness minimization. The constraint set (2.2) secures the completion of each activity in the range [EFT j , LFT j] once and only once. Although this model assumes that activities cannot be preempted, i.e., an activity in progress cannot be interrupted by another activity to be processed and the former activity is processed later starting from the last point or from the beginning, there are some studies in literature allowing activity preemption. Constraint set (2.3) insures that precedence relations are observed among the activities. The right-hand side of the constraint set (2.3) represents the starting period of activity j. Constraint set (2.4) expresses the upper bound constraints on the use of renewable resources at each period for each resource. Given we are in period t and the activity finishes in period ($t+d_j-1$), then activity j is in progress in period t and hence, occupies k_{jr} units of renewable resource r in period t. Activities generally require constant amount of renewable resources during the execution of the activity. However, this case can be generalized to varying amount of resource requirement. In the above formulation single mode case where only one combination of resource usage is considered is illustrated but there is also multi-mode case where an activity can be processed with more than one resource combination. In that case another decision on the selection of activity execution mode is made. Since multi-mode is possible for the activities, tradeoff cases may occur during project scheduling. The upper bound constraints on the use of nonrenewable resources at each period for each nonrenewable resource is given in constraint set (2.5). Constraint set (2.6) defines the zero-one variables.

2.2. SOLUTION PROCEDURES FOR THE SINGLE OBJECTIVE RCPSP AND ITS EXTENSIONS

2.2.1. Solution Procedures

The literature about solution approaches for RCPSP and its extensions such as multi-mode RCPSP (MRCPSP), the multi-project scheduling problem (RCMPSP), multi-skilled resource constrained project scheduling problem (MSPSP), resource constrained multi-project scheduling problem (RCMPSP) etc. is very extensive , including exact methods like branch and bound (B&B), branch and cut (B&C) techniques, heuristics like rule based procedures and meta-heuristics like simulated annealing (SA), tabu search (TS), ant colony optimization (ACO), and genetic algorithms(GA). It has been shown by Blazewicz et al. (1983) that the RCPSP, as a generalization of the classical job shop scheduling problem, belongs to the class of NP-hard optimization problems. Hence, the use of heuristics for solving large problem instances is indeed justified.

2.2.2. Genetic Algorithms

Most widely proposed solution procedure in the literature for RCPSP and its extensions is GA. Since they are widely proposed and the focus on solution procedures to be employed in this thesis is on GAs, next we will briefly explain the concept of GAs and then explain their usage in the RCPSP domain.

The concept of GA was developed by Holland (1975). GAs are inspired by the evolution theory suggested by Darwin. GAs belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using tools inspired by natural evolution, such as inheritance, mutation, selection, and crossover. The procedure of a generic GA given by Goldberg (1989) is as follows:

1. **[Start]** Generate random population of *n* chromosomes (suitable solutions for the problem)
2. **[Fitness]** Evaluate the fitness *f(x)* of each chromosome *x* in the population
3. **[New population]** Create a new population by repeating following steps until the new population is complete
    1. **[Selection]** Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)
    2. **[Crossover]** With a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.
    3. **[Mutation]** With a mutation probability mutate new offspring at each locus (position in chromosome).
    4. **[Accepting]** Place new offspring in a new population
4. **[Replace]** Use new generated population for a further run of algorithm
5. **[Test]** If the end condition is satisfied, **stop**, and return the best solution in current population
6. **[Loop]** Go to step **2**

In a simple GA, a solution (called chromosome) is represented by a string of numbers (genes). A chromosome's potential as a solution is measured by a fitness function which evaluates a chromosome with the objective function value. A judiciously selected set of chromosomes is called a population and the population at a given time is called a generation. Usually, the number of chromosomes remains fixed from generation to generation. The fundamental mechanism of GAs consists of three main operations: reproduction, crossover and mutation. Chromosomes resulting from these operations applied to those in the current population, often known as offspring or children, form a population of the next generation. The procedure of generating a next generation is repeated for a certain number of times, which can be determined in various ways or until a predefined condition holds. An application of a GA is characterized by an encoding scheme, an initial population, genetic operators (reproduction, crossover, and mutation), a fitness function, and a termination rule.

GA has been used for various problems. RCPSP is one of such problems. The components of GA varies from a problem type to another problem type. In other words, every problem has its own components in GA. Below, the components of GA are explained in the case that corresponding problem is RCPSP.

Chromosome Representation

There exist three main chromosome representation approaches; priority value based, priority rule based and activity list based. Priority value based chromosome representation depends on priority values assigned to activities. These values indicate the priority of activities. Priority rule based chromosome representation is achieved by assigning a rule for each activity. For example, let the number activities be N and scheduled activities be M. It means that algorithm will schedule $N - M$ activities. The activity $N + 1$ is scheduled as the best activity among remaining $N - M$ activities according to the priority rule $N + 1$. The other type of chromosome representation also known as the permutation based chromosome representation is activity list based representation in which each activity is represented in the chromosome as in the order to be scheduled. Besides from these commonly used chromosome representations, there is also a chromosome representation encoding a vector of random keys representing the number of activities followed by delay times used when scheduling activities. For

MRCPSP there are also some chromosome representations considering mode assignments as a string of mode numbers assigned for each activity.

Decoding Procedure

To solve the RCPSP with a GA, individuals have to be transformed into a schedule. This task is accomplished with schedule generation schemes (SGS). SGSs start from scratch and build a feasible schedule by stepwise extension of a partial schedule. There are two different classic SGSs. They can be distinguished into activity and time incremental. The so called serial SGS performs activity incremental and schedules the activities at their earliest feasible start time. The other classic SGS, so called parallel SGS performs time incremental and constructs active schedules. In other words, no activity can be left shifted without violating the constraints. Parallel schedule generation scheme (parallel SGS) works as follows: After scheduling the dummy sink activity at time 0, the parallel SGS computes a so-called decision point which is the time at which an activity to be scheduled is started. Earliest feasible finish time of the activities in progress determine this decision point. Eligible activities are determined by constructing the set of those activities that can start at this decision point. The eligible activities are selected successively and started until none are left. This process, constructing the eligible set and finding the decision point, is repeated until all activities are scheduled.

In addition to serial SGS and parallel SGS, the constructive heuristic so called parallel modified SGS makes use of the priorities and delay times of activities and constructs parameterized active schedules. This SGS is based on time incremental.

Beside constructive SGSs there are some local search techniques to improve the schedule obtained with the implementation of constructive SGSs. These local search techniques move from solution to solution in the space of candidate solution until the optimal solution or a stopping criterion is found and try to get improvements. Forward and backward local search algorithms are such local search techniques to improve the schedule.

Fitness Function

As stated before, GA works with a fitness function showing the quality of a solution. Thus, as the algorithm proceeds, the individuals having better fitness values stay in the population. In the literature, very large number of fitness functions have been used so far. While some of them are basically calculated from simple makespan or other type of indicator such as NPV, some are computed based on the functions of these performance indicators Any objective function that can be calculated from a schedule and any combination of these objectives can be fitness function for GA to solve RCPSP.

Initial Population

For a GA to start, an initial population has to be created. One of the mostly used methods for this task is random selection. Thus, each individual that will exist in the initial population is created randomly. Random sampling method can also be preferred for constructing the initial population.

Crossover

A variety of crossover methods to create feasible children from selected parents is present in the literature. One-point crossover, two-point crossover, uniform crossover are the most common ones. Generally crossover is applied with a predetermined crossover probability.

Mutation

The most widely used mutation technique is bit mutation and pairwise interchange mutation. Mutation is applied with a mutation probability to the children to be directly transformed to the next generation.

Selection Mechanism

Selection mechanisms are used for reproduction and selecting the individuals for crossover and mutation. Fitness proportionate selection, also known as roulette-wheel selection is one of the most used selection mechanism of GA. In roulette-wheel selection mechanism the fitness value is used to associate a probability of selection with each individual chromosome. It represents the survival probabilities for all the individuals in the population. Simple ranking method, proportional selection, 2-tournament selection and 3-tournament selection are the other common selection mechanisms.

2.2.3. Solution Procedures for RCPSP and Some Extensions

2.2.3.1. Resource Constrained Project Scheduling Problem

In recent years, the applications and challenges of the RCPSP and its extensions have attracted increasing interest from researchers. Below we will present an overview of recent literature addressing the RCPSP, especially focusing on the implementation of GAs suggested. For an overview of techniques developed to solve RCPSP we refer to the surveys provided by Icmeli et al. (1993), Ozdamar and Ulusoy (1995), Herroelen et al. (1998), Brucker et al. (1999), Klein (2000), Hartmann and Kolisch (2000), and Hartmann (2010).

Several exact methods to solve the RCPSP are proposed in the literature. Currently, the most competitive exact algorithms seem to be the ones of Demeulemeester and Herroelen (1997), Brucker et al. (1998), Klein and Scholl (1998), Mingozzi et al. (1998), and Sprecher (2000). Several authors propose procedures for computing lower bounds on the makespan to prune a set of solutions in B&B algorithms thus accelerating the algorithms. Brucker and Knust (2003) developed a destructive lower bound for the RCPSP, where the lower bound calculations are based on two methods for proving infeasibility of a given threshold value for the makespan. The first uses constraint propagation techniques, while the second is based on a linear programming formulation.

Most of the heuristic methods used for solving RCPSP belong to the class of priority rule based methods. These methods start with none of the activities being scheduled. Subsequently, a single schedule is constructed by selecting a subset of resource and precedence activities in each step and assigning starting times to these activities until all activities have been considered. This process is controlled by the SGS as well as priority rules with the latter being used for ranking the activities. Several approaches of this class have been proposed in the literature. Davis and Patterson (1975), Cooper (1976), Boctor (1990), Kolisch (1996a), Kolisch (1996b) and Tormos and Lova (2001) present such studies.

The other class of methods is based on the design of meta-heuristics which improve upon an initial solution. This is done by successively executing operations which transform one or several solutions into others. Several meta-heuristic approaches have been proposed in the literature. Some of these would be the following: simulated annealing procedures of Slowinski et al. (1994), Boctor (1996), and Bouleimen and Lecocq (2003); tabu search approaches of Pinson et al. (1994) and Thomas and Salhi (1998); local search-oriented approaches of Leon and Balakrishnan (1995), Fleszar and Hindi (2004), and Palpant et al. (2004). Since this thesis is focused on GAs as a solution methodology, we will analyze in more detail the GAs proposed to solve RCPSP in order to state the difference between those approaches.

Cheng and Gen (1994) proposed a GA using a list of activity/start time representation in which the chromosome represents a permutation of all activities associated with the start time. The gene is an ordered couple (activity, start time). A new discipline is addressed for designing the genetic operators, that is, for two main genetic operators, crossover and mutation, one is designed towards to perform blind search to try to explore the area beyond local optima, the other is designed towards to perform intensive search to try to find improved solution. A gamma augmented transformation of the natural objective function to a fitness function is proposed which can adjust the roulette wheel mechanism based selection behaviors from fitness-proportional selection to pure random selection.

Lee and Kim (1996) compared the use of GAs with TS and SA for the classical RCPSP. These three approaches are based on the random key representation with the parallel SGS as decoding procedure. While SA and TS make use of a restricted version of the pairwise interchange move, the GA employs the standard one-point crossover. Reproduction is done by randomly selecting and duplicating a chromosome with a probability which is proportional to the fitness value of the chromosome. The best results are obtained by the SA algorithm.

Kohlmorgen et al. (1999) studied the impact of parallel execution strategies for GAs in various combinatorial optimization problems, including RCPSP. They particularly tested the ‘‘island model” and the ‘‘neighborhood model”. In the ‘‘island model”, GAs are executed concurrently on several independent sub-populations with the added possibility of regularly exchanging good individuals between neighborhood islands. In the ‘‘neighborhood model”, the population is distributed over the processors of a large mesh connected array. This spatial arrangement of the population allows the natural use of local neighborhoods in the selection of parents for producing new individuals for the next generation. This study was especially focused on how the number and size of sub-populations and the migration rate, interval and strategy (for the first model) and the selection strategy and neighborhoods (for the second model) influenced the course of evolution and the quality of the generated solutions.

Hartmann (1998) proposed a competitive GA. The representation is based on a precedence feasible permutation of the set of the activities. The genotypes are transformed into schedules using a serial SGS. The initial population was determined with a randomized priority rule method. Three different crossover variants for the permutation based encoding are considered: one-point crossover, two-point crossover and uniform crossover. Mutation is done by a pair wise interchange of genes. Four alternative selection operators which follow a survival of the fittest strategy are considered: Simple ranking method, proportional selection, 2-tournament selection and 3-tournament selection. Among these alternative genetic operators, a ranking selection strategy, a mutation probability of 0.05, and a two-point crossover operator, which preserves precedence feasibility, were chosen for numerical experiments. In order to evaluate the proposed approach, they have compared it to two GA designs from the literature which make use of a priority value and a priority rule representation, respectively. As further benchmarks, seven other heuristics known from the literature were considered. The in-depth computational study revealed that the proposed GA outperformed the other GAs as well as the heuristics considered.

Alcaraz and Maroto (2001) designed a new type of representation based on the activity list representation for RCPSP, by adding an additional gene to the representation. This additional gene in the chromosome indicates the SGS used to build the schedule: serial forward or backward and allows the GA to adapt itself to the problem instance. Thus this new representation helps to reveal the problem specific knowledge. They have developed two new crossover techniques, two point forward and backward crossover applied to this new representation and another based on the standard activity list representation. These three crossover techniques are problem specific operators. They have implemented a GA with these three new crossover techniques and the two-point crossover developed by Hartmann (1998). As mutation operators, they have implemented the one developed by Hartmann (1998), and an adaptation of the procedure proposed by Boctor (1996) in his SA algorithm to generate a neighbor. Moreover, they have implemented three different selection strategies, stochastic sampling without replacement, 2-tournament and ranking, which were reported in previous studies to produce very good results. They also have used two different crossover probabilities and two different mutation probabilities in order to determine the most appropriate configuration.

Hartmann (2002) proposed a self adapting GA which employs the well-known activity list representation and considered two different decoding procedures. An additional gene in the representation determines which of the two decoding procedures, serial or parallel, is actually used to compute a schedule for an individual. This allows the GA to adapt itself to the problem instance actually solved. That is, the GA learns which of the alternative decoding procedures is the more successful one for this instance. In other words, not only the solution for the problem, but also the algorithm itself is subjected to evolution.

Hindi et. al. (2002) used activity list representation of a schedule in their evolutionary algorithm. Five types of crossover operations are designed and tested. These operators are one-point operator, two-point operator, multipoint operator, uniform operator and alternate operator. Mutation operator selects randomly a task in the string and calculates its feasible positions, then selects a position randomly within this range and exchanges the tasks.

Kim et. al. (2003) discussed how to solve a basic RCPSP problem using a fuzzy logic-hybrid GA initialized by the serial method. In their computational test they also considered experimenting with various genetic operators such as compounded partially mapped crossover, position-based crossover, swap mutation, and local search-based mutation.

Valls et. al. (2004) proposed a two-phase population based algorithm. The procedure incorporates different strategies for generating and improving a population of schedules. The first phase constructs an initial population of schedules and evolves them until high quality solutions are obtained. Quality of solutions is measured with their makespan. Random and rapid construction procedures are used to create a high quality and diverse initial population. The subsequent evolution is driven by the alternative application of an improving procedure and a schedule combination mechanism that blends the characteristics of scatter search and path relinking. The improving procedure applied to each population schedule is an iterative procedure for improving the local use of resources. Phase two begins from the best solution obtained in phase one. The objective of the second phase is to closely explore regions near high quality schedules. Exploring such a region means, firstly, generating a population by taking a random sample from a region near the schedule, and secondly, applying to the population the procedures used in the first phase – but with a variation. The on-going search is interrupted when a better schedule is obtained and a fresh search starts from the recently obtained better schedule.

Debels and Vanhoucke (2005) proposed a bi-population based GA which employs permutation based chromosome representation. In contrast to a regular GA, they use the bi-population GA that makes use of two different populations: a population LJS that only contains left-justified schedules and a population RJS that only contains right-justified schedules. Both populations have the same population size. The procedure starts with the generation of an initial LJS , followed by an iterative process that continues until the stopping criterion is satisfied. The iterative process consecutively adapts the population elements of RJS and LJS. RJS (LJS) is updated by feeding it with combinations of population elements taken from LJS (RJS) that are scheduled backwards (forwards) with the serial SGS.

Tseng and Chen (2006) developed a hybrid metaheuristic called ANGEL which uses an ACO and a GA. First, ACO searches the solution space and generates activity lists to provide the initial population for GA. Next, GA is executed and the pheromone set in ACO is updated when GA obtains a better solution. When GA terminates, ACO searches again by using a new pheromone set. ACO and GA search alternately and cooperatively in the solution space. This study also proposes an efficient local search procedure which is applied to yield a better solution when ACO or GA obtains a solution. A final search is applied upon the termination of ACO and GA.

Valls et. al. (2008) proposed a hybrid GA which employs activity list representation of chromosomes and serial SGS for decoding. The algorithm uses the peak crossover operator, whose objective is to exploit the knowledge of the problem to identify and combine those good parts of the solutions that have really contributed to its quality. Double justification operator, which is a local improvement operator is applied. Since applying the evolutionary scheme to one population, the initial set of solutions is restrictive a new population by using a biased random sampling of the neighborhood of the best solution found so far (a neighbor's population) is generated. The algorithm has two phases: an initial phase of general searching and a second phase of searching in the neighborhood of the best solution.

Mendes (2008) proposed a random key variant of GA. The schedule is constructed using a heuristic priority rule in which the priorities and delay times of the activities are defined by the GA. Each solution chromosome is made of 2n genes where n is the number of activities. The first n genes are used to calculate the priorities of the activities and the second n genes are used to calculate the delays of the activities. This heuristic makes use of the priorities and the delay times defined by the GA and constructs parameterized active schedules which are in a broader class of non-delay but narrower class of active schedules to reduce the solution space. The basic idea of parameterized active schedules consists in controlling the delay times that each activity is allowed thus reducing the solution space.

Gonçalves et. al. (2008) improved the study of Mendes (2008) and proposed a new GA which combines a GA and a schedule generator procedure that generates parameterized active schedules.. The GA evolves the chromosomes which represent the priorities of the activities, release dates and delay times. A new measure of performance attempting to capture reality by integrating due dates (tardiness), work in process(flow time) and inventory(earliness) is developed. A weighted function of these performance indicators is used to calculate the fitness of the individuals.

Montoya-Torres et al. (2010) proposed an alternative representation of the chromosomes using a multi-array object-oriented model in order to take advantage of programming features in most common languages for the design of decision support systems (DSS). A main difference between the proposed approaches and those previously existing in literature is that the proposed implementation utilizes object-oriented programming (OOP) paradigm to improve computational efficiency while maintaining effectiveness.

2.2.3.2. Multi-Mode Resource Constrained Project Scheduling Problem

Multi-mode resource constrained project scheduling problem, MRCPSP, is a generalized version of the RCPSP, where each activity can be performed in one out of a set of modes, with a specific activity duration and resource requirements.

The objective of the MRCPSP is to assign a mode and a start time for each activity such that an objective function is minimized and the schedule is feasible with respect to the precedence and renewable and nonrenewable resource constraints. As this problem is a generalization of the RCPSP, the MRCPSP is also NP-hard. Moreover, Kolisch and Drexl (1997) stated that if there is more than one nonrenewable resource, the problem of finding a feasible solution for the MRCPSP is NP-complete. Several exact and heuristic approaches to solve the MRCPSP have been proposed in the literature.

The first solution method for the multi-mode problem can be found in Slowinski (1980), who presented a one-stage and two-stage linear programming approach. Talbot (1982) and Patterson et al. (1989) presented an enumeration scheme-based procedure. Speranza and Vercellis (1993) proposed a depth-first B&B algorithm More recently, Sprecher et al. (1997), Hartmann and Drexl (1998) and Sprecher and Drexl (1998) presented B&B algorithms, while Zhu et al. (2006) proposed a B&C algorithm. However, none of these procedures can be used for solving large-sized realistic projects, since they are unable to find an optimal solution in a reasonable computation time. Therefore, different single-pass and meta-heuristic procedures are presented.

Talbot (1982) and Sprecher and Drexl (1998) proposed to impose a time limit on their exact B&B procedure computationally feasible. Boctor (1993) tested 21 heuristic scheduling rules and suggested a combination of 5 heuristics which have a high probability of giving the best solution. Drexl and Grünewald (1993) proposed a biased random sampling approach, while Özdamar and Ulusoy (1994) proposed a local constraint based analysis approach. Kolisch and Drexl (1997) suggested a local search method with a single-neighborhood search. Knotts et al. (2000) evaluated different agent-based algorithms and Lova et al. (2006) designed several multi-pass heuristics based on priority rules for solving the MRCPSP.

Slowinski et al. (1994), Boctor (1994), and Bouleimen and Lecocq (2003) used the SA approach, while Nonobe and Ibaraki (2001) proposed a TS procedure. Zhang et al. (2005) and Jarboui et al. (2008) presented the methodology of particle swarm optimization for solving the MRCPSP. Ranjbar et al. (2009) used a hybrid scatter-search to tackle the MRCPSP, using the path relinking methodology as a solution combination method.

GAs have been presented by many practitioners. Below, some papers proposing GA to solve MRCPSP are surveyed and most attracting points of them are described briefly.

Mori and Tseng (1997) suggested to use a GA for MRCPSP and since GAs are also stochastic they aimed at comparing it to stochastic heuristic scheduling rules. In the proposed method, representation of the individuals were executed by implementing direct representation. As for mutation, the algorithm chooses the activities other than the terminal node and changes their modes. It is shown that the proposed GA outperformed the stochastic heuristic scheduling rules.

Özdamar (1999) proposed a GA approach to a general category project scheduling problem. The algorithm incorporates problem-specific scheduling knowledge by an indirect chromosome encoding that consists of selected activity operating modes and an ordered set of scheduling rules. The scheduling rules in the chromosome are used in an iterative scheduling algorithm that constructs the schedule resulting from the chromosome. The proposed GA is designated as a hybrid GA (HGA) approach since it is integrated with traditional scheduling tools and expertise specifically developed for RCPSP. The essential feature of HGA is the integration of problem-specific knowledge within the GA's search strategy via an indirect encoding that maps onto the heuristic space, and a solution generator that thoroughly exploits the latter encoding.

Hartmann (2001) proposed a GA for MRCPSP. One of the very important modification implemented in this paper is the problem specific representation. Basically, the genetic encoding is based on precedence feasible list of activities and mode assignment. The second significant modification is to implement the local search concept in order to improve the schedules obtained by basic GA. Two local search methods are used. One is for dealing with feasibility problem of MRCPSP whereas the other one is for improving the schedules. GA is tested on the basis of a standard set of project instances. After analyzing the behavior of the proposed GA, it is compared to four heuristic approaches proposed in the literature.

Alcaraz et. al. (2003) extended the model proposed in Alcaraz et. al (2001) and suggested a new GA modifying the representation of individuals in order to make them available for multi mode case. Representation is performed by creating two lists and one additional gene. While activity list and mode assignment list make up two lists, SGS is presented in the additional gene. The second important modification of this paper is assigning the fitness function by considering the infeasible schedules. That is, infeasible schedules with respect to resources have worse fitness function value when compared to the feasible schedules.

Kim et. al. (2006) proposed an adaptive GA in order to solve the MRCPSP. There exist some significant characteristics of this algorithm. Activity and mode representation are performed by implementing priority-based encoding and multistage-based encoding, respectively. Crossover is executed by order-based crossover operator for activity list and local search-based mutation is preferred for activity mode list. Moreover, GA includes iterative hill-climbing method. Finally, auto-tuning approach is adopted for the rates of mutation and crossover. According to this approach, as long as mutation and crossover lead to better solutions, the occurrences of them increase.

Lova et. al. (2009) suggested a hybrid GA for MRCPSP. This algorithm uses powerful local search method to improve the solutions provided by GA. Its main contributions are mode assignment procedure, the fitness function and the use of efficient improving method. To consider the infeasible individuals in the selection process they proposed a new fitness function penalizing the infeasible individuals with a new upper bound to overcome the weaknesses of the upper bound proposed in Hartmann(2001). To increase the probability of obtaining feasible mode assignments in the initial population they proposed a procedure called Minimum Normalized Resources (MNR) which does not require any additional computational effort. To improve the solutions the application of multimode backward-forward method (MM-BF) or the application of multimode forward-backward method (MMFB) depending on whether the initial feasible schedule is obtained with forward or backward method is proposed.

Chen and Shahandashti (2009) suggested a hybrid algorithm including GA and SA. The main advantages of the proposed algorithm are (1) as the number of generation increases, the fitness value would increase and induce the algorithm to choose better-fitted solutions and (2) the mutation rate would decrease as the number of generations grows. Since the number of iterations at different temperatures vary so much, and this leads to local search algorithm, SA may not be beneficial. Instead, a modified SA has been developed by adjusting the number of iterations for each temperature.

Peteghem and Vanhoucke (2010) proposed bi-population GA extending serial SGS by introducing mode improvement procedure. Preemption is allowable in this study. Although non-preemptive version of RCPSP and multi mode RCPSP have been studied many times, preemptive version has been an open area for researchers. Bi-population GA has been used before for single mode RCPSP. This paper extends this application to multi-mode RCPSP in preemptive or non-preemptive version. In this paper, two population are used for GA. One of them is left-justified and the other one is right-justified. Moreover, an extended serial SGS which improves the mode selection is introduced by choosing that feasible mode of a certain activity that minimizes the finish time of that activity.

2.2.3.3. Multi-Skill Project Scheduling Problem

The multi-skill project scheduling problem, MSPSP, consists of determining a feasible schedule respecting the precedence constraints between activities and the resource constraints: a person cannot use a skill s/he does not master, cannot be assigned to more than one requirement at a given time and a person that satisfies a skill requirement must be assigned to the corresponding activity during its whole processing time. The aim is to minimize the total duration of the project. This model is an extension of the classical RCPSP: if we assume that all members only have one skill, we get the classical resource constraints. This problem may also be seen as an MRCPSP but with a very high number of modes, because of the number of eligible subsets of persons. As Bellenguez-Morineau and Néron (2007) point out, the large number of emerging modes make the MRCPSP not a viable option to model multiple skills. Hence, a number of papers have treated the multi-skill case explicitly. As Heimerl and Kolisch (2010) stated these papers can be distinguished according to (1) the scope, (2) the existence of efficiency of resources, and (3) the efficiency of a resource being dynamic or static due to learning and forgetting . The following explanations are largely based on the study of Bellengues (2008).

Bassett (2000) considered R&D-projects in the chemical industry where independent activities have to be scheduled within time windows and internal as well as external human resources have to be assigned to the activities. Efficiencies are assumed to be homogeneous and static.

Vairaktarakis (2003) defined a measure for the resource flexibility by the amount of resources that are capable of doing multi-skill work (with the extremes being single and completely skilled). Based on an MIP and heuristics he investigated how the increase of multi-skill capability improves the project performance measure project duration. Efficiencies are assumed to be homogeneous and static.

Bellenguez and Néron (2004) proposed a TS based on the priority list to solve MSPSP with homogenous and static skill efficiencies. A solution is evaluated either by serial SGS, or by parallel SGS. A solution S, built on a priority list L is considered to be a neighbor of a solution S’, from a list L’, if L’ is computed from L with only one swap of two activities. They also proposed two GAs to solve this problem, based on the priority list of activities and on the assignment of resources to requirements. But in their actual state of development, these methods are experimentally dominated by the tabu search.

Bellenguez and Néron (2005) proposed two lower bounds for MSPSP with homogenous and static skill efficiencies. The linear lower bound they present is based on a time-horizon decomposition into successive intervals. The second lower bound is based on energetic reasoning. For these two methods it is necessary that each activity has a time window. They have adapted these two lower bounds to MSPSP because their efficiency on a large range of RCPSP instances has been proved.

Wu and Sun (2006) presented a mixed-integer non-linear program for multi-project scheduling and staffing considering learning effect of staff which has not been addressed in the literature before. The work of the activities has to be allocated within the respective time windows (whereas precedence relations between activities are not taken into account). Efficiencies are assumed to be heterogeneous and dynamic and it is further assumed that each human resource can only be assigned to one project per period. A GA is designed to solve the nonlinear program. In the GA, a chromosome represents a staff allocation decision. For a particular chromosome, by reducing some variables and constraints, the nonlinear program becomes a linear program. The fitness of the chromosome is evaluated by solving the linear program.

Alba and Chicano (2007) present a project scheduling problem which is similar to an RCPSP. However, multiple objectives (time, cost, quality) are considered, resources can be assigned fractionally, resources are multi-skilled with homogeneous efficiencies, and tasks require multiple skills. The time required for completing a task depends on the number and fractions of assigned resources. The problem is not explicitly modeled as a mathematical program but only qualitatively described. A GA is proposed for solving the problem.

Bellenguez-Morineau and Néron (2007) propose a branch and bound procedure using a branching scheme based on the reduction of the slack of the activities to solve MSPSP. They introduce different branching strategies and define the way to treat the assignment problem in each leaf node. A heuristic method to calculate the upper bound and two lower bounds are proposed. In this study skill efficiencies are assumed to be homogeneous and static.

Valls et al. (2009) consider the scheduling of tasks and their assignment to a multi-skilled workforce with heterogeneous efficiencies in a service center under multiple criteria.

Besides the studies mentioned above there are some papers considering multi-skills in the scope of project selection, scheduling and staffing together, project staffing only and project selection and staffing.

Ballou and Tayi (1996) as well as Gutjahr et al. (2008), Gutjahr et al. (2010a) and Gutjahr and Reiter (2010b) treated the problem of project selection, scheduling, and staffing. In all these papers the entire problem is decomposed into cascaded sub problems which are then solved separately. Papers, which do consider the project staffing problem only, assume a project schedule as given. The latter defines for each period how many units of each skill are required. Cai and Li (2000) treated the assignment of days-off schedules to multi-skilled resources with two skills in order to cover the given resource demand within a service setting. Skill efficiencies are assumed to be static and homogeneous. Campbell (1999) as well as Campbell and Diaby (2002) consider the assignment of multi-skilled human resources with heterogeneous efficiencies to given demand in a health care service setting for a single period (static case). Corominas et al. (2005) assumed a single period where demand for different skills has to be covered by human resources with multiple skills as well as homogeneous and static efficiencies. Corominas et al. (2006) related to a service setting where multi-skill human resources with homogeneous efficiencies have to be assigned to activities over the periods of the planning horizon such that different objectives are met in the best way. A heuristic is proposed which solves a sequence of assignment problems where the elements of the assignment matrix take into account the different objectives.

2.2.3.4. Resource Constrained Multi-Project Scheduling Problem

RCPSP has been studied extensively in the literature. However, since it may be necessary to process more than one project in real-life, recent studies have been focusing on resource constrained multi-project scheduling problem, RCMPSP.

One of the most common ways to deal with multi-projects is to transform it into single project by adding a “super source” node and a “super sink” node. A different approach suggested in the literature is the decomposition approach. In this approach, each project is regarded as an activity. The scheduling of these activities corresponds then to the scheduling of the projects resulting in the starting time of the projects. Using this information the projects are scheduled individually to obtain the final schedule. The RCPSP has been treated extensively. In contrast, for the RCMPSP, there are a relatively limited number of studies involving the scheduling of several projects.

Exact methods to solve the RCMPSP are proposed in the literature. The pioneering work of multi-project scheduling by Pritsker et al. (1969) proposed a zero-one programming approach. Mohanthy and Siddiq (1989) studied the problem of assigning due dates to the projects in a multi-project environment. That study presents an integer programming model and simulation mechanism. The integer program generates the schedules. The simulation allows testing some heuristic rules and the system chooses the best schedule. Drexl (1991) considered a non-preemptive variant of the resource constrained assignment problem using a hybrid B&B / dynamic programming algorithm with a Monte Carlo-type upper bounding heuristic. Deckro et al. (1991) formulated the multi-project scheduling problem as a block angular general integer programming model and employed a decomposition approach to solve large problems. Speranza and Verceilles (2003) introduced the dual level approach which suggests a decomposition of the problem into a hierarchy of integer programming models. Vercellis (1994) described a Lagrangean decomposition technique for solving multi-project planning problems with resource constraints and alternative modes of performing each activity in the projects. Can and Ulusoy (2010) followed the approach of Speranza and Verceilles (2003) and suggested a 2-stage decomposition approach to formulate the problem as a hierarchy of 0-1 mathematical programming models. In the first stage they reduce each project to a macro activity with macro-modes resulting in a single project network and apply their genetic algorithm designed for this problem with the NPV maximization objective.

Most of the heuristics methods used for solving resource constrained multi-project scheduling problems belong to the class of priority rule based methods. Several approaches in this class have been proposed in the literature. For example, Fendley (1968) used multi-projects with three and five projects and considered three efficiency measurements in the computational analysis: project slippage, resource utilization, and in-process inventory. The most important conclusion of Fendley is that the priority rule minimum slack first (MINSLK) obtained the best efficiency with the three response variables. Kurtulus and Davis (1982) designed multi-project instances whose projects have between 34 and 63 activities and resource requirements for each activity between 2 and 6 units. They show six new priority rules and maximum total work content (MAX-TWK) and shortest activity from the shortest project (SASP) were the best algorithms to schedule multi-projects when the objective was to minimize the mean project delays, where the delays were measured in relation to the unconstrained critical path duration. Kurtulus and Narula (1985) considered penalties in their study and analyzed six penalty functions with four priority rules.

There are a large number of metaheuristics in the literature developed to solve RCMPSP. Kim et. al. (2005) implemented a hybrid GA (hGA) with adaptive abilities. Fuzzy logic controller (FLC) makes up the major point of this paper. Specifically, the authors use the mutation FLC. This procedure adaptively regulates the GA parameters, mainly mutation operator. The mutation FLC is implemented independently to adaptively regulate the mutation ratio during the genetic search process based on the fuzzy logic control. That is the main difference between the hGA and hGA with FLC. Adaptive hybrid GA uses nearly the same procedure as fuzzy logic control. During generating the population, mutation probability may change regarding the convergence to a solution.

Kumanan et. al. (2006) proposed the use of a simple and efficient GA with a heuristic for solving the RCMPSP with an objective to minimize the makespan of the projects.. Man et. al. (2008) suggested a hybrid algorithm consisting of SA and GA. Instead of transforming the project networks into a single project, which is a general approach, authors use Cross list to represent the project networks overall. First the ordinary GA and SA used in the proposed method are improved separately and then the SA operations which can overcome the defects of genetic algorithm are applied. The results show SA and GA to be better than the other heuristic algorithms proposed in the literature. Furthermore, proposed algorithm is better in terms of solution quality and speed.

2.2.3.5. Other Types of Related Problems using GA Based Approaches

Besides the RCPSP, different types of problems such as resource allocation and leveling, discounted cash flows and time-cost trade-off problem closely related to RCPSP have been studied in the literature. Below, some related studies have been given and their significant points have been mentioned shortly.

Hegazy (1999) proposed a GA for resource allocation and leveling. In their improved heuristics, random priorities are introduced into selected tasks and their impact on the schedule is monitored. The GA procedure then searches for an optimum set of tasks' priorities that produces shorter project duration and better leveled resource profiles. One major advantage of the procedure is its simple applicability within commercial project management software systems to improve their performance.

Ulusoy and Cebelli (2000) introduced a new approach in which the amount and timing of the payments made by the client and received by the contractor are determined so as to achieve an equitable solution to the payment scheduling approach. The authors defined an equitable solution as a solution where both the client and the contractor deviate from their respective ideal solution by an equal percentage. To find an equitable solution a double-loop GA approach is proposed where the outer loop represents the client and the inner loop the contractor. In the inner loop MRCPSP with the objective of maximizing the contractor's NPV for a given payment distribution is solved. While the payment distribution information flows from outer loop to inner loop, the information of timing of these payments flows from the inner loop to outer loop when searching for an equitable solution.

Leu et. al. (2000) addressed a GA-based resource leveling scheduling system called GARLS and a DSS architecture of the GARLS. Unlike heuristic models, it is not necessary for the GARLS to commit to any particular heuristic rules. Because of this, the GARLS has greater flexibility when solving complex resource leveling scheduling problems. The DSS enables to make ad-hoc analysis through "what-if" queries.

Ulusoy et al. (2001) developed models for the multi mode resource constrained project scheduling problem with discounted cash flows (MRCPSPDCF). Four payment models are considered: lump sum payment at the terminal event, payments at pre-specified event nodes, payments at pre-specified time points and progress payments. The GA uses a special crossover operator that can exploit the multi-component nature of the problem. The GA employing multi-component uniform order-based crossover (MCUOX) is shown to be applicable to both the MRCPSP and MRCPSPDCF with a relatively minor change in the definition of the fitness function.

Azaron et. al. (2005) proposed a GA for the time-cost trade off problem in PERT networks. They developed a new multi-objective model for the time-cost trade-off problem in PERT networks with generalized Erlang distributions for activity durations. To obtain the optimal resources allocated to the activities, a goal attainment model is developed with four conflicting objectives, minimization of the project direct cost, minimization of the mean of project completion time, minimization of the variance of project completion time and also maximization of the probability that the project completion time does not exceed the given threshold. The problem considered in that paper has continuous decision variables and involves non-linearity.

Ranjbar and Kianfar (2007) developed a GA procedure for solving the discrete time/resource trade-off problem. They use a GA in which a new method based on the resource utilization ratio is developed for the generation of crossover points and also a local search method is incorporated with the algorithm.

Wuliang and Chengen (2008) propose a GA to solve a multi-mode resource constrained discrete time-cost tradeoff problem (MRC-DTCTP). In that study, the general DTCTP was extended to a new multi-mode resource-constrained DTCTP model, which inherits some features of the well- known MRCPSP. In MRC-DTCTP, time constraints, renewable resource constraints, cost constraints were considered. The relationships among project duration, renewable resources as well as the direct project cost and the indirect project cost are balanced in the model. According to the characteristics of the proposed problem, an improved GA was presented to solve it.

2.3. SOLUTION PROCEDURES FOR THE MULTI-OBJECTIVE RCPSP

Most of the research in the RCPSP literature consider a single objective function. The two most commonly used objectives in RCPSP are minimizing the project makespan, and maximizing project NPV. While many authors have concentrated on minimizing the makespan, most of the recent research focused on maximizing the NPV of the project using the sum of positive and negative discounted cash flows throughout the life cycle of the project. However, the efficiency of a scheduling is not evaluated to satisfy a single objective in practice, but to obtain a trade-off schedule regarding multiple objectives. Therefore a need for multi-objective optimization has occurred. Currently, there are a large number of mathematical programming techniques for multi-objective optimization. However, these techniques tend to generate points in the Pareto optimal set one at a time. Additionally, most of them are very sensitive to the shape of the Pareto front. Hence, heuristics seem particularly suitable to solve multi-objective optimization problems, because they are less susceptible to the shape or continuity of the Pareto front , whereas this is a real concern for mathematical programming techniques. Additionally, many current heuristics (e.g., evolutionary algorithms, particle swarm optimization, etc.) are population-based. The two ideal goals of multi-objective optimization is to (1) Find a set of solutions which lie on the Pareto-optimal front, and (2) Find a set of solutions which are diverse enough to represent the entire range of the Pareto-optimal front. Evolutionary algorithms are the most widely used and appropriate algorithms that can achieve both goals.

In this part, first an introduction to evolutionary multi-objective optimization will be provided and then the evolution of evolutionary algorithms will be explained. Lastly, the studies considering multiple objectives for the RCPSP and its extensions will be explained briefly. It should be mentioned that most of the information given in the subsequent parts of multi-objective optimization is based on the book chapter of Deb (2001) and the paper of Coello (2006).

2.3.1. Evolutionary Multi-Objective Optimization

Evolutionary multi-objective optimization (EMO) algorithms use a population based approach in which more than one solution participates in an iteration and evolves a new population of solutions in each iteration. The reasons for their popularity are many. Some of them are:

(i) EMOs do not require any derivative information

(ii) EMOs are relatively simple to implement and

(iii) EMOs are flexible and have a wide-spread applicability.

An EMO begins its search with a population of solutions usually created at random within a specified lower and upper bound on each variable. Thereafter, the EMO procedure enters into an iterative operation of updating the current population to create a new population by the use of four main operators: selection, crossover, mutation and elite-preservation. The operation stops when one or more pre-specified termination criteria are met.

2.3.2. First Generation EMOs

The first implementation of a real multi-objective evolutionary algorithm vector-evaluated GA (VEGA) was suggested by Schaffer (1985). Schaffer modified the simple three-operator GA (with selection, crossover, and mutation) by performing independent selection cycles according to each objective. The selection method is repeated for each individual objective to fill up a portion of the mating pool. Then the entire population is thoroughly shuffled to apply crossover and mutation operators. This is performed to achieve the mating of individuals of different subpopulation groups. The algorithm works efficiently for some generations but in some cases suffers from its bias towards some individuals or regions. This does not fulfill the second goal of EMO. Ironically, no significant study was performed for almost a decade after the pioneering work of Schaffer, until a revolutionary 10-line sketch of a new non-dominated sorting procedure suggested by Goldberg (1989) in his seminal book on GAs. Since an EA needs a fitness function for reproduction, the trick was to find a single metric from a number of objective functions. Goldberg's suggestion was to use the concept of domination to assign more copies to non-dominated individuals in a population. Since diversity is the other concern, Goldberg and Richardson (1987) suggested the use of a niching strategy among solutions of a non-dominated class. Getting this clue, at least three independent groups of researchers developed different versions of multi-objective evolutionary algorithms during 1993-1994. Basically, these algorithms differ in the way a fitness assignment scheme is introduced to each individual.

Fonseca and Fleming (1993) suggested a multi-objective GA (MOGA), in which all non-dominated population members are assigned a rank one. Other individuals are ranked by calculating how many solutions (say k) dominated a particular solution. That solution is then assigned a rank (k+1). The selection procedure then chooses lower rank solutions to form the mating pool. Since the fitness of a population member is the same as its rank, many population members will have an identical fitness. MOGA applies a niching technique on solutions having identical fitness to maintain a diverse population. But instead of performing niching on the parameter values, they suggested niching on objective function values. The ranking of individuals according to their non-dominance in the population is an important aspect of the work.

Horn et al. (1994) used pareto domination tournaments in their niched-pareto GA (NPGA). In this method, a comparison set comprising of a specific number (t_{dom})

of individuals is picked at random from the population at the beginning of each selection process. Two random individuals are picked from the population for selecting a winner according to the following procedure. Both individuals are compared with the members of the comparison set for domination. If one of them is non-dominated and the other is dominated, then the non-dominated point is selected. On the other hand, if both are either non-dominated or dominated, a niche-count is calculated by simply counting the number of points in the entire population within a certain distance (σ_{share}) in the variable space from an individual. The individual with least niche-count is selected. The authors reported that the outcome of the algorithm depends on the chosen value of t_{dom} . Nevertheless, the concept of niche formation among the non-dominated points using the tournament selection is an important aspect of the work.

Srinivas and Deb (1994) developed a non-dominated sorting GA (NSGA), which differs from MOGA in the way of fitness assignment and the way niching is performed. After the population members belonging to the first non-domination class are identified, they are assigned a dummy fitness value equal to N (population size). A sharing strategy is then used on parameter values (instead of objective function values) to find the niche-count for each individual of the best class. For each individual, a shared fitness is then found by dividing the assigned fitness N by the niche-count. Thereafter, the second class of non-dominated solutions (obtained by temporarily discounting solutions of first non-domination class and then finding new non-dominated points) is assigned a dummy fitness value smaller than the least shared fitness of solutions of the previous non-domination class. This process is continued till all solutions are assigned a fitness value. This fitness assignment procedure ensured two matters: (i) a dominated solution is assigned a smaller shared fitness value than any solution which dominated it and (ii) in each non-domination class an adequate diversity is ensured. On a number of test problems and real-world optimization problems, NSGA has been found to provide a wide-spread Pareto-optimal or near Pareto-optimal solutions. However, one difficulty of NSGA is to choose an appropriate niching parameter, which directly affects the maximum distance between two neighboring solutions obtained by NSGA.

2.3.3. Second Generation EMOs

The second generation EMO algorithms implemented an elite-preserving operator in different ways and gave birth to the elitist EMO procedures. Since early EMO research concentrated on finding a set of well-converged and well-distributed set of near-optimal trade-off solutions, EMO researchers concentrated on developing better and computationally faster algorithms by developing scalable test problems and adequate performance metrics to evaluate EMO algorithms.

Zitzler and Thiele (1999) suggested an elitist multi-criterion EA with the concept of non-domination in their Strength Pareto EA (SPEA). They suggested maintaining an external population in every generation storing all non-dominated solutions discovered so far, beginning from the initial population. This external population participates in genetic operations. In each generation, a combined population with the external and the current population is first constructed. All non-dominated solutions in the combined population are assigned a fitness based on the number of solutions they dominate and all dominated solutions are assigned a fitness equal to one more than the sum of fitness of solutions which dominate it. This assignment of fitness makes sure that the search is directed towards the non-dominated solutions. Diversity among dominated and non-dominated solutions is maintained by performing a clustering procedure to maintain a fixed size archive. In their subsequent improved version (SPEA2), Zitzler et al. (2001) made three changes . First, the archive size is always kept fixed by adding dominated solutions from the EA population, if needed. Second, the fitness assignment procedure for the dominated solutions is slightly different and a density information s measuring the crowdedness of the region surrounded by an individual is used to resolve ties between solutions having identical fitness values. Third, a modified clustering algorithm is used from the k^{th} nearest neighbor distance estimates for each cluster and special attention is given to preserve the boundary elements.

Corne et. al. (2000) suggested pareto-archived ES (PAES) with one parent and one child. The child is compared with the parent. If the child dominates the parent, the child is accepted as the next parent and the iteration continues. On the other hand, if the parent dominates the child, the child is discarded and a new mutated solution (a new child) is found. However, if the child and the parent do not dominate each other, the choice between child or a parent is resolved by using a crowding procedure. To maintain diversity, an archive of non-dominated solutions found so far is maintained.

The child is compared with the archive to check if it dominates any member of the archive. If yes, the child is accepted as the new parent and the dominated solution is eliminated from the archive. If the child does not dominate any member of the archive, both parent and child are checked for their proximity (in terms of Euclidean distance in the objective space) to the archive members. If the child resides in the least crowded area in the objective space compared to other archive members, it is accepted as a parent and a copy is added to the archive. In their subsequent version, called the Pareto Envelope Based Selection Algorithm (PESA) Knowles and Corne (2000) combined good aspects of SPEA and PAES. Like SPEA, PESA carries two populations (a smaller EA population and a larger archive population). Non-domination and the PAES crowding concept is used to update the archive with the newly created child solutions.

The NSGA-II procedure developed by Deb et. al (2002) is extensively employed. EMO procedures which attempt to find multiple Pareto-optimal solutions. At any generation t, the offspring population (say, Q_t) is first created by using the parent population (say, P_t) and the usual genetic operators. Thereafter, the two populations are combined together to form a new population (say, R_t) of size $2N$. Then, the population R_t classified into different non-domination classes. Thereafter, the new population is filled by points of different non-domination fronts, one at a time. The filling starts with the first non-domination front (of class one) and continues with points of the second non-domination front, and so on. Since the overall population size of R_t is $2N$, not all fronts can be accommodated in N slots available for the new population. All fronts which could not be accommodated are deleted. When the last allowed front is being considered, there may exist more points in the front than the remaining slots in the new population. Instead of arbitrarily discarding some members from the last front, the points which will make the diversity of the selected points the highest are chosen. The crowded-sorting of the points of the last front which could not be accommodated fully is achieved in the descending order of their crowding distance (CD) values and points from the top of the ordered list are chosen. The CD value of point i (d_i) is a measure of the objective space around i which is not occupied by any other solution in the population. Here, we simply calculate this quantity d_i by estimating the perimeter of the cuboid formed by using the nearest neighbors in the objective space as the vertices.

There also exist other competent EMOs, such as multi-objective messy GA (MOMGA) developed by Veldhuizen and Lamont (2000), multi-objective micro-GA proposed by Coello and Toscano (2001); neighborhood constraint GA proposed by Loughlin and Ranjithan (1997); ARMOGA developed by Sasaki et. al. (2001); and others. Besides, there exists other EA based methodologies, such as particle swarm EMO proposed by Coello and Lechuga (2002); ant-based EMO proposed by McMullen (2001) and Gravel et. al. (2002); and differential evolution based EMO proposed by Babu and Jehan (2003).

2.3.4. Improved Diversity and Speed Based Studies

Pruning a set of non-dominated solutions is a common and essential part of multi-objective evolutionary algorithms (MOEAs). An idea is to prune a non-dominated set to have a desired number of solutions in such a way that the remaining solutions have as good diversity as possible, meaning that the spread of extreme solutions is as high as possible, and the relative distance between solutions is as equal as possible. Probably the best way to obtain a good distribution would be to use some clustering algorithm. However, this is computationally expensive since clustering algorithms take usually time $O(MN^2)$ to prune a set of size N with M objectives. In NSGA-II the pruning of non-dominated solutions is done in time $O(MN^2logN)$ based on CD. However, this method often gives non-optimal distribution as it is demonstrated in Kukkonen and Deb (2006a). Although maintaining diversity is easier for a two-objective objective space, the difficulty arises in the case of higher-dimensional objective spaces. This is the reason why researchers have developed different diversity measures, such as the hyper-volume measure, the spread measure, the chi-square deviation measure, the R-measures, and others. In maintaining diversity among population (or archive) members, several researchers have used different diversity-preserving operators, such as clustering, crowding, pre-specified archiving, and others. Interestingly, these diversity-preserving operators produce a trade-off between the achievable diversity and the computational time.

Deb et al. (2003) addressed this trade off and two approaches for a better spread is suggested. The first approach is a straightforward replacement of NSGA-II's crowding routine by the clustering approach used in SPEA. After the parent and offspring population are combined into a bigger population of size 2N and this combined population is sorted into different non-domination levels, only N good solutions are required to be chosen based on their non-domination levels and nearness to each other. In the clustering NSGA-II approach suggested, the authors replace the crowding procedure with the clustering approach. The solutions in the last permissible non-dominated level are used for clustering. Totally clustering is executed until as many required number of slots as clusters are created. The clustering algorithm used s exactly the same as that used in SPEA. Although this requires a larger computation time, the clustered NSGA-II finds a better distributed set of Pareto-optimal solutions than the original NSGA-II. In the second approach, the search space is divided into a number of grids (or hyper-boxes) and diversity is maintained by ensuring that a grid or hyper-box can be occupied by only one solution. Although, PAES and its variants are developed with the similar idea, ∈-dominance is a more general concept. The ∈-dominance does not allow two solutions with a difference $\in_i$ in the i-th objective to be non-dominated to each other, thereby allowing a good diversity to be maintained in a population. Besides, the method is quite pragmatic, because it allows the user to choose a suitable i depending on the desired resolution in the i^{th} objective. In the proposed ∈-MOEA, two populations (EA and archive) are evolved simultaneously. Using one solution each from both populations, two offspring solutions are created. Each offspring is then used to update both parent and archive populations. The archive population is updated based on the ∈€-dominance concept, whereas an usual domination concept is used to update the parent population. Since the ∈-dominance concept reduces the cardinality of the Pareto-optimal set and since a steady-state EA is proposed, the maintenance of a diverse set of solutions is possible in a small computational time. Simulation results of this study showed that for more than two objective problems the convergence of the ∈-MOEA is relatively better than that of the other two MOEAs (C-NSGA II and NSGA II) with identical function evaluations. Although the C-NSGA-II achieves the best distribution, it is also computationally the slowest of the three MOEA.

Kukkonen and Deb (2006a) suggested a modification to the diversity handling method of NSGA-II to obtain better diversity. The proposed algorithm first calculates CDs for the members of a non-dominated set. Instead of selecting n members having the largest CD values, N −n members having the smallest CD values are removed one by one, updating the CD values for the remaining members of the set after each removal. The efficient implementation of this algorithm needs an implementation of a priority queue such as a heap. The time complexity class of the new algorithm is estimated and in most cases it is the same as for the original pruning algorithm. Numerical results also support this estimate. For bi-objective test problems, the proposed pruning algorithm is demonstrated to provide better distribution compared to the original pruning algorithm of NSGA-II. However, with tri-objective test problems there is no improvement and this study reveals that CD does not estimate crowdedness well in this case and presumably also in cases of more objectives.

Since it is revealed in the earlier study of Kukkonen and Deb (2006a) that CD fails to approximate the crowding of the solutions when the number of the objectives is more than two, Deb et. al. (2006b) proposed a new algorithm with two different crowding estimation techniques for pruning non-dominated solutions in such a way that the obtained diversity is intended to be good also in the case of more than two objectives, and the consumed time is intended to be considerably less than in clustering. The basic idea of the proposed pruning method is to eliminate the most crowded members of a non dominated set one by one, and update the crowding information of the remaining members after each removal. For this, two approaches for crowding estimation based on the nearest neighbors of solution candidates are introduced, and then a technique for finding these nearest neighbors quickly is introduced. The first crowding estimation is called 2 Nearest neighbors and probably is the simplest crowding estimation technique to measure the distance between a solution and its nearest neighbor solution, and use this distance to estimate crowding. The solution having the smallest distance is considered as the most crowded. The second approach called M Nearest Neighbor uses a little bit more developed idea and use the k nearest neighbors for crowding estimation in such a way that distances to the k nearest neighbors are multiplied together, and the solution having the smallest product is considered the most crowded. According to the experimental results, the proposed pruning method provides a better diversity than the pruning method based on CD. Especially in the case of tri-objective problems, the obtained diversity is significantly better. The obtained distribution is observed to be similar to the distribution obtained with SPEA2. The execution time needed for the proposed pruning method is more than for the pruning method based on CD but significantly less than for the pruning method in SPEA2. Two different crowding estimation techniques provide similar diversity but simpler (the one, which uses 2 Nearest Neighbors) is also faster to execute. Based on the results, the proposed method provides near optimal distribution, which does not need improvement.

2.3.5. Multi-Objective Optimization Studies for the RCPSP Domain

Project scheduling problems are clearly multi-objective problems since managers in real life deal with several objectives at once. They want to finish the projects as soon as possible and with the minimum cost and with maximum quality. However, not many papers have been published in this field and this is perhaps due to the difficulty of solving multi-objective problems and the vast number of possible interesting generalizations.

Some authors considering the multi-objective project scheduling problem defined one overall objective as the weighted sum of all performance measures considered. Ulusoy and Özdamar (1995), Nudtasomboon and Randhawa (1997), Al-Fawzan and Haouari (2005), and Abbasi et al. (2006) took this approach.

Ulusoy and Özdamar (1995) proposed a heuristic iterative scheduling algorithm for RCPSP to minimize the project duration and to maximize NPV of the project. The iterative scheduling algorithm consists of forward/ backward scheduling passes, where consecutive scheduling passes are linked by updated activity time windows. The iterative algorithm was supported by a conflict-based activity selection technique called the local constraint based analysis (LCBA).

Nudtasomboon and Randhawa (1997) developed a zero-one integer programming model for the problem. The model considers many important characteristics of project scheduling including activity preemption, renewable and non-renewable resources, time-resource trade-offs, and multiple objectives (time minimization, cost minimization, and resource leveling). Solution algorithms are developed for these three single objective problems and for the preemptive goal programming model that includes time, cost and resource leveling objectives.

Al-Fawzan and Haouari (2005) addressed the issue of designing a project schedule which is not only short in time, but also less vulnerable to disruptions due to reworks and other undesirable conditions. Based on the concept of schedule robustness, they develop a bi-objective RCPSP model where the two objectives are the robustness maximization and makespan minimization. They proposed to measure robustness of a given schedule via the total amount of free slack of all activities. A TS algorithm is used to generate an approximate set of efficient solutions. Another bi-objective RCPSP with robustness and makespan criteria is investigated by Abbasi et al. (2006) via an SA algorithm. The free slack as a measure of robustness was also referred to as floating time in their study.

Chtourou and Haouari (2008) followed the approach of Al-Fawzan and Haouari but assigned weights to the free slack of an activity according to the number of its successors and/or the sum of its required resources. They presented a two-stage algorithm for robust RCPSP. The first stage of their algorithm solves the RCPSP for minimizing the makespan only using a priority-rule-based heuristic, namely an enhanced multi-pass random-biased serial SGS. The problem is then similarly solved for maximizing the schedule robustness while considering the makespan obtained in the first stage as an acceptance threshold. Selection of the best schedule in this phase is based on one out of 12 alternative robustness predictive indicators formulated for the maximization purpose.

Some other researchers considered the multi-objective RCPSP and obtained Pareto-optimal schedules. Davis et. al (1992), Slowinski et. al (1994), Hapke et. al (1998), Viana and de Sousa (2000), Hanne and Nickel (2005) and Xiong et. al (2012) are some of such studies.

Davis et al. (1992) formulated the RCPSP as a bi-objective problem consisting of the makespan and the overutilization of each renewable resource. An interactive decision support approach is introduced for explicitly considering multiple factors simultaneously in a multi-objective decision making framework. The approach enhances the traditional project scheduling framework in that it allows the decision maker to examine the minimization of project completion time as well as the balancing of available resources.

Slowinski et al. (1994) presented a DSS for multi-objective project scheduling under multiple-category resource constraints. They employed the multi-objective RCPSP with four objectives including makespan, mean weighted lateness, total number of tardy activities and smoothness of the resource profile. The DSS is based on three kinds of heuristics: parallel priority rules, SA and branch-and-bound. The last algorithm can even yield exact solutions when sufficient processing time is available. Some parts of the system are interactive, in particular, the search for the best compromise schedule.

Hapke et al. (1998) presented a multi-criteria approach allowing one to simultaneously consider several objectives based on time, resource, and money and consists of two stages. In the first stage, a large representative sample of approximately non-dominated schedules is generated by the Pareto Simulated Annealing (PSA) method. Then, in the second stage, an interactive search over the sample is organized by the `Light Beam Search' (LBS) procedure in its discrete version.

Viana and de Sousa (2000) added to these two objectives the minimization of overutilization of each non-renewable resource along with the mean weighted tardiness . They investigated the applicability of two metaheuristic approaches, PSA and multi-objective TS, to RCPSP.

Another study by Hanne and Nickel (2005) employing an evolutionary algorithm for scheduling and inspection planning in a software development project tries to minimize three objectives related to quality, time and cost. Quality of a project is measured by the number of defects; time by the project duration and cost by the total cost of time spent by the development team members

Ballestin and Blanco (2011) developed algorithms to obtain the pareto fronts for the case where all objective functions are regular. .

Xiong et.al. (2012) modeled the RCPSP as a three-objective optimization problem, where makespan minimization, robustness maximization, and stability maximization are simultaneously taken into account. A hybrid multi-objective evolutionary algorithm (H-MOEA) is proposed to solve the problem. In the process of the H-MOEA, the heuristic information is extracted periodically from the obtained non-dominated solutions, and a local search procedure based on the accumulated information is incorporated.

2.4. DEALING WITH UNCERTAINTY IN RCPSP DOMAIN

The vast majority of the research efforts in project scheduling assume complete information about the scheduling problem to be solved and a static deterministic environment within which the pre-computed baseline schedule will be executed. During project execution, however, project activities are subject to considerable uncertainty that may lead to numerous schedule disruptions. This uncertainty may stem from a number of possible sources: activities may take more or less time than originally estimated, resources may become unavailable, material may arrive behind schedule, ready times and due dates may have to be changed, new activities may have to be incorporated or activities may have to be dropped due to changes in the project scope, weather conditions may cause severe delays, etc. A disrupted schedule incurs higher costs due to missed due dates and deadlines, resource idleness, higher work-in-process inventory and increased system nervousness due to frequent rescheduling. Recognition of the fact that research results on problems based on certainty assumptions and static environments have only slim chances of implementation has recently led to an increasing interest in project management under uncertainty (Elmaghraby, 2005).

Different approaches like reactive scheduling, stochastic scheduling, and proactive (robust) scheduling, are employed to deal with uncertainty. In this part, we will discuss these approaches mainly from a project scheduling viewpoint and explain some related studies. For a detailed survey of studies dealing with uncertainty in scheduling domain we refer to Herroelen and Leus (2005).

2.4.1. Stochastic Project Scheduling

The stochastic RCPSP aims at scheduling project activities with uncertain durations in order to minimize the expected project duration subject to FS type precedence constraints with zero time lag and renewable resource constraints. The literature on stochastic project scheduling is rather sparse. Most of the research efforts on the stochastic RCPSP rely on so-called scheduling policies. Möhring et.al (1984 and 1985) are some studies using scheduling policies.

Pet-Edwards and Mollaghasemi (1996) proposed a simulation and a GA for the stochastic RCPSP aiming at minimizing the variance of the project duration. Simulation is used to estimate the mean and variance of the project duration generated through the application of the heuristics. A GA is used to help to improve the efficiency of a search over weighted linear combinations of scheduling heuristics. In this paper how a GA can be used to "breed" good weights for a combination of scheduling heuristics and how simulation is used to evaluate the fitness of the weights for a stochastic RCPSP are demonstrated.

Golenko-Ginzburg and Gonik (1997) considered PERT type project networks where the duration of an activity is a random variable with a given density function (beta, uniform and normal distributions are used) and where a pre-given lower and upper bound on the activity duration is available. At each decision point, when at least one activity is ready to be scheduled, resource conflict is resolved by solving a zero–one integer programming problem to maximize the total contribution of the accepted activities to the expected project duration. For each activity, this contribution is computed as the product of its average duration and the probability (determined by simulation) of it lying on the critical path.

Leu and Hung (2002) proposed the use of GAs and Monte Carlo simulation to develop the resource-constrained scheduling model under uncertainty. Monte Carlo simulation was used to model the uncertainty that is associated with time elements in project networks. GAs were then used to allocate multiple available construction resources to activities of a single project so as to achieve the objective of minimizing the project duration under uncertainty.

Ballestin and Leus (2009) investigated various objective functions related to timely project completion of RCPSP with stochastic activity durations. The authors developed a Greedy Randomized Adaptive Search Procedure (GRASP) to produce high-quality solutions.

Ashtiani et al. (2011) introduced a new class of scheduling policies for solving RCPSP with stochastic activity durations. The authors underlined the value of preprocessing in stochastic scheduling. In the process, a subset of sequencing choices at the beginning of the planning horizon was made and the rest of the scheduling decisions to future points were relegated in time.

2.4.2. Reactive Scheduling

Reactive scheduling does not try to cope with uncertainty in creating the baseline schedule but revises or re-optimizes the baseline schedule when an unexpected event occurs. Basically most efforts concentrate on ‘‘repairing’’ the baseline schedule (predictive-reactive scheduling) to take into account the unexpected events that have come up. The reactive scheduling action may be based on various underlying strategies. At one extreme, the reactive effort may rely on very simple techniques aimed at a quick schedule consistency restoration. We shall refer to these approaches as schedule repair actions. A typical example of such a simple control rule is the well-known right shift rule This rule moves forward in time all the activities that are affected by the schedule breakdown because they were executing on the resource(s) causing the breakage or because of the precedence relations. It should be clear that this strategy may lead to poor results as it does not re-sequence activities.

At the other extreme, the reactive scheduling approach may involve a full scheduling pass of that part of the project that remains to be executed at the time the reaction is initiated. Such an approach will be referred to as (full) rescheduling and may use any deterministic performance measure, such as the new project’s makespan.

However, in a stochastic environment, ensuring the timely completion of a project for a broad range of scenarios, i.e. quality robustness, is often not the only issue. In many cases where certain preparations have been made once the predictive schedule is established (ordering raw materials, acquiring necessary tools or equipment, organizing the workforce, fixing delivery dates for both subcontractors and customers, etc.), it is desirable that the activity starting times that are actually realized during project execution differ little from the planned activity starting times in the predictive schedule. Stability or solution robustness refers to the insensitivity of planned activity start times to schedule disruptions that may occur during project execution. The benefits of generating stable predictive schedules have been demonstrated by Van de Vonder et. al. (2005a). However, solution robustness or schedule stability should also be maintained when the predictive schedule breaks and needs to be repaired. Reactive procedures should try to repair the predictive schedule in such a way that the safety included in the original predictive schedule is preserved.

The literature concerning robust reactive project scheduling is virtually void. Artigues and Roubellat (2000) studied the case where, in a multi-project, multi-mode setting with ready times and due dates, it is desired to insert a new unexpected activity into a given baseline schedule such that the resulting impact on maximum lateness is minimized. The authors perform a clever rescheduling pass in which they restrict the solution to those schedules in which the resource allocation remains unchanged. Using a resource flow network representation they developed a step-wise procedure for generating a set of dominant 'insertion cuts' for the network. From each dominant insertion cut, they then derive the best execution mode and valid insertion arc subset for the new activity.

Yu and Qi (2004) described an integer linear programming model for the multi-mode RCPSP and report on computational results obtained by a hybrid mixed integer programming/constraint propagation approach for minimizing the schedule deviation caused by a single disruption induced by a known increase in the duration of a single activity.

Van de Vonder et. al. (2007) described new heuristic reactive project scheduling procedures that may be used to repair resource-constrained project baseline schedules that suffer from multiple activity duration disruptions during project execution. The objective is to minimize the deviations between the baseline schedule and the schedule that is actually realized.

Lambrechts et. al. (2008) modeled the uncertainty by means of resource availabilities that are subject to unforeseen breakdowns. The objective was to build a robust schedule that meets the project deadline and minimizes the schedule instability cost. They described how stochastic resource breakdowns can be modeled, which reaction is recommended, when a resource infeasibility occurs due to a breakdown.

Deblaere et. al. (2011) proposed and evaluated a number of dedicated exact reactive scheduling procedures as well as a TS for repairing a disrupted schedule under the assumption that no activity can be started before its baseline starting time.

2.4.3. Proactive Scheduling

Proactive scheduling aims at the construction of a protected initial schedule (baseline or predictive schedule) that anticipates possible future disruptions by exploiting statistical knowledge of uncertainties that have been detected and analyzed in the project planning phase. This protection is necessary because often project activities are subcontracted or executed by resources that are not exclusively reserved for the current project. A change in the starting times of such activities could lead to infeasibilities at the organizational level (in a multi-project context) or penalties in the form of higher subcontracting costs A possible measure for the deviation between the initial schedule and the realized schedule is the weighted instability cost. It can be calculated by taking the sum of the expected weighted absolute deviations between the planned and the actually realized activity starting times. The weight w_i , assigned to each activity i, reflects that activity's importance of starting it at its planned starting time in the initial schedule. More specifically, w_i denotes the marginal cost of deviating from the planned starting time of activity i during project execution.

Minimizing instability then means that we are looking for a schedule, which is able to accommodate disruptions without too much change in the activity starting times, i.e., a robust schedule that satisfies the precedence and resource constraints and does not exceed the deadline set by the project's client. Meeting this deadline during project execution is encouraged by giving a higher instability weight to the activity that signals the end of the project. Leus (2003) and Van de Vonder et al. (2006) considered this objective function for the case of project scheduling with stochastic activity durations.

Herroelen and Leus (2004) developed mathematical programming models for the generation of stable baseline schedules in a project environment. The authors make abstraction of resource usage, assuming that a proper allocation of resources has been performed. They used the concept of pairwise float for the activities, which are the elements of transitive closure of the arcs in the project network. The authors propose to use as stability measure the expected weighted deviation in start times in the realized schedule from those in the pre-schedule. The authors have extended the model to cope with multiple disturbances.

Van de Vonder et. al. (2006) proposed the resource flow dependent float factor (RFDFF) heuristic as a time buffering technique to produce robust schedules relying completely on the activity weights but did not exploit the available information offered by the activity duration distributions in making its buffering decisions. RFDFF starts from an unbuffered schedule and modifies it by adding safety buffers in front of activities. The hope is that the time buffers serve as a cushion to prohibit the propagation of the disruptions through the schedule.

In addition to reactive procedures Lambrechts et. al. (2008) also suggested proactive procedures in their paper. They focus on disruptions caused by stochastic resource availabilities and aim at generating stable baseline schedules. A schedule's robustness (stability) is measured by the weighted deviation between the planned and the actually realized activity starting times during project execution. They presented four procedures including resource buffering and a TS procedure that operates on a surrogate, free slack-based objective function.

Van de Vonder et. al. (2008) introduced multiple algorithms to include time buffers in a given schedule while a predefined project due date remains respected. Multiple efficient heuristic and meta-heuristic procedures were proposed to allocate buffers throughout the schedule. Following common practice in project scheduling they adopt a two-stage approach that first solves the RSPSP and afterwards adds safety to the initial schedule in a second stage. They compared the proposed time buffering techniques (the virtual activity duration extension (VADE), the starting time criticality (STC), improvement heuristics, TS procedure) with the RFDFF technique proposed by Van de Vonder et. al. (2006).

In a recent study, Lambrechts et. al. (2011) analytically determined the impact of unexpected resource breakdowns on activity durations. Furthermore, using this impact of unexpected resource breakdown information they developed an approach for inserting explicit idle time into the project schedule in order to protect it from possible disruptions caused by resource unavailabilities. In their procedure, first, one has to decide whether to schedule the project using the maximal, deterministic resource availability a_k or to use a buffered availability a_k*. This buffered availability is calculated by taking the expected value of the steady state availabilities. Next, an initial schedule is generated using either an optimal approach for minimizing the project makespan or by scheduling activities having a high Cumulative Instability Weight (CIW) as early as possible in time. After that, time buffering is applied. For time buffering; simulation based time buffering, resource flow network based techniques, and surrogate measure based techniques were suggested.

2.4.4. Integrated Procedures

Integrated procedures are comprised of proactive-reactive procedures and risk integrated methodologies. Proactive-reactive scheduling involves a proactive and a reactive phase. During the proactive phase, a baseline schedule is constructed that accounts for statistical knowledge of uncertainty and anticipates disruptions. The underlying idea is to protect the schedule as much as possible from the disruptions that may take place during the execution of the project. When disruptions do occur during actual project execution, it may be necessary to call upon reactive scheduling procedures to modify the baseline schedule in response to these disruptions.

Risk is defined by the Project Management Institute (PMI) (2000) as an uncertain event or condition that, if it occurs, has a positive or negative effect on a project objective. Most of the research approaches on project scheduling involving risk do not model risks explicitly, but try to evaluate the risk of schedule and/or budget overruns using stochastic models for activity durations and/or costs. In this part we will briefly mention the studies which consider risks explicitly as opposed the ones explained in the stochastic project scheduling part and integrated proactive-reactive studies.

Jaafari (2001) presented a general approach to risk management within project management. He conjectured that a paradigm shift is needed and introduced a strategy-based project management approach, which is called the life cycle project management. This is an integrated and collaborative framework, which installs life cycle objective functions as the basis of evaluation and decision making throughout project life cycle.

Shatteman et. al. (2008) developed an integrated methodology for planning construction projects under uncertainty. The methodology relies on a computer supported risk management system that allows to identify, analyze and quantify the major risk factors and derive the probability of their occurrence and their impact on the duration of the project activities. The system maintains a risk management database that is updated with the new subjective information generated by the project management teams of on-going projects through user interfaces and as such can serve input for a proactive scheduling system. Using project management estimates of the marginal cost of activity starting time disruptions, a proactive baseline schedule is developed that is sufficiently protected against the anticipated disruptions with acceptable project makespan performance. The methodology is illustrated on a real life application.

Creemers et. al. (2011) proposed a quantitative approach to project risk analysis that allows to address the risk response process in a scientifically-sound manner. They have shown that a risk-driven approach is more efficient than an activity-based approach when it comes to analyzing risks. Therefore, project risk management should focus on assessing the uncertainty caused by risks themselves (i.e., the root cause) rather than evaluating the uncertainty at the level of activities. In addition, they developed two new ranking indices to assist project managers in determining where to focus their risk mitigation efforts.

Herroelen (2013) proposed a risk integrated methodology for tactical and operational project planning under uncertainty. The methodology integrates quantitative risk analysis with reliable proactive/reactive project scheduling procedures. The integrated methodology relies on an iterative two-phase process. In phase one, he determines the number of regular renewable resource units to be allocated to the project and the so-called internal project due date. Phase two implements a proactive/reactive schedule generation methodology.

2.5. RISK ANALYSIS

The goal of risk analysis is to generate insights into the risk profile of a project and to use these insights in order to drive the risk response process. The insights generated include: the probability of achieving a specific project outcome, the distribution function of the project completion time, etc. The risk response process will use these insights to define practical risk responses that allow project managers to mitigate risks (i.e., to reduce the impact of risks on project objectives).

In literature, risk analysis process is divided into four main sub processes, namely, risk identification, risk prioritization, quantitative risk assessment and quantitative risk evaluation. Risk identification is the first and the most critical step in risk management, since failing to identify important risk factors can result in the failure of the whole risk management process. Checklists and interviewing are the basic risk identification tools. Risk prioritization is a qualitative procedure that allows to prioritize the risks that were identified in an earlier stage of the risk management process. It requires ordinal estimates of both the probability of occurrence and the impact of a risk. These ordinal estimates are then used to create a shortlist of high priority risks. Further risk analysis efforts focus on these high priority risks. The purpose of quantitative risk assessment is to measure risk exposure. The exposure can be expressed either in cost or in time. The benefit of risk assessment is that the project team comes up with a ranking of risks so that the team can focus on the important ones. In quantitative risk assessment step experts provide detailed estimates of the probability of occurrence and the impact of high priority risks. These estimates are used in the quantitative risk evaluation procedure to analyze the impact of the shortlisted risks on overall project objectives. Hubbard (2009) states that good risk management requires a risk analysis process that is scientifically sound and that is supported by quantitative techniques. A wide body of knowledge on such quantitative techniques has been accumulated over the last two decades. Monte Carlo Simulation is the predominant quantitative risk evaluation technique in both practice and in literature. Creemers et. al. (2013) introduced a quantitative new approach and showed that a risk-driven approach for project scheduling is more efficient than an activity-based approach to analyze risks. In addition, the authors propose two ranking indices, one activity-based index that ranks activities and another risk-driven index that ranks risks, allowing to identify the activities or risks that contribute most to the delay of a project to assist project managers in determining where to focus their risk mitigation efforts. Kirytopoulos et.al. (2001) introduces a knowledge system to identify risks and their assessments in project schedules for project risk management. Expert’s knowledge, checklists and corporate memory are the tools utilized for the identification of risks and risk breakdown structure is used as a risk guide.

CHAPTER 3

PROBLEM DEFINITION AND ENVIRONMENT

As mentioned earlier, the subject of this thesis is the scheduling of the R&D projects in a stochastic and dynamic environment present in the R&D Department of a leading home appliances company in Turkey. From now on, we will call the R&D Department of the firm as the R&D Department. To have a better understanding of the problem on hand and the problem environment, in this chapter first an introductory information on R&D project scheduling and the studies on this area are presented briefly. Then, R&D project management environment of the R&D Department is introduced. Finally, the scheduling problem of the R&D Department is defined. Special emphasis is given on underlining the main differences of the problem on hand from the project scheduling problems in the literature. A mixed integer linear programming (MILP) formulation of the problem is given.

3.1. R&D PROJECT SCHEDULING

R&D project selection and planning has been extensively studied in literature. Baker (1974), Souder and Mandakovic (1986), and Schmidt and Freeland (1992) all provide reviews. While there are literally hundreds of papers in this area, most are portfolio planning models, often with some multi-criteria objective. Most of the research on R&D project scheduling focuses on the selection of parallel versus sequential scheduling of project activities. The studies of Eppinger et al. (1995), Krishnan et al. (1997) and Dahan (1998) are some of the studies working on that topic.

Subramanian et. al. (2000) considered the stochastic project selection and scheduling problem for the R&D pipeline management and they developed a decision support tool, named Sim-Opt, that combines combinatorial optimization and discrete event system simulation to assess the uncertainty and control the risk present in the pipeline. Vanhoucke (2006) defined a set of time windows for each activity to be used as a time-based objective for scheduling the R&D projects in bio-technology sector. Carrying out an activity within one of its time windows is desired due to quality considerations. The objective function minimizes penalties that are caused by executing activities outside their time windows. Bartels and Zimmermann (2007) proposed an approach for scheduling the individual tests conducted in the R&D projects in the automotive industry such that the number of required experimental vehicles is minimized. The proposed approach is based on a new type of multi-mode resource-constrained project scheduling model with minimum and maximum time lags as well as renewable and cumulative resources.

Assigning human resources to work taking into account resource-specific skills and efficiencies is a general planning task which has to be performed in any organization. It is of particular importance for service firms such as R&D Departments where, compared to manufacturing firms, the labor intensity is higher and multi-skilled resources are more common. This makes the task more complicated in an R&D project planning and scheduling environment. Heimerl and Kolish (2010) address this situation and consider the problem of simultaneously scheduling projects that face uncertainty and assigning multi-skilled internal and external human resources with resource-specific efficiencies to the project work. In their study, they modeled the problem as a MILP problem with a tight LP-bound. The performance of the model with respect to the solution gap and computation time is assessed and managerial insight is given concerning different problem parameters such as the time window size of projects, the number of skills of human resources, and the workload.

Another important feature of R&D projects is that, apart from the commercial and market risks common to all projects, their constituent activities also carry the risk of technical failure. Therefore, besides projects overrunning their budgets or deadlines and the commercial returns not meeting their targets, R&D projects also carry the risk of failing altogether, resulting in time and resources spent without any tangible return. In a recent study, De Reyck and Leus (2008) considered this aspect of the R&D project scheduling problem and examined how to schedule R&D projects in order to maximize their expected net present value (NPV) when the project activities have a probability of failure and when an activity's failure leads to overall project termination. They formulated the problem, showed that it is NP-hard, developed a B&B algorithm that allows to obtain optimal solutions and provide extensive computational results in their study.

3.2. PROJECT MANAGEMENT ENVIRONMENT OF THE R&D DEPARTMENT

3.2.1. Organizational Structure of the R&D Department

In the R&D Department, research and development projects related to the technologies used in the production process of all kinds of products that the firm produces are conducted. The conducted projects are not product specific, but they are technology specific. Thus, the technology developed can be utilized in one or more product type in the product innovation process of the firm. R&D Department is organized in technology departments. Under the R&D directorship there are 5 technology departments and all of them have their own managers. The organizational chart of R&D Department is shown in Figure 3.1.

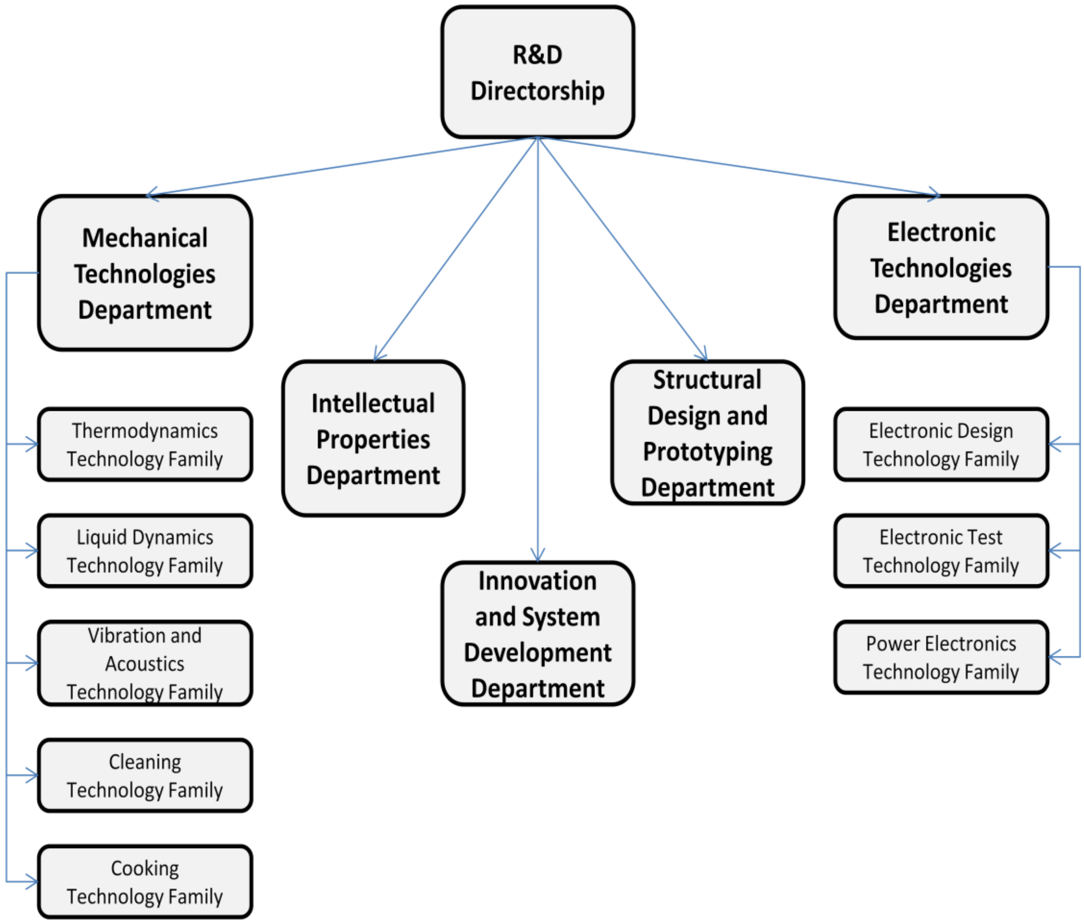

Figure 3. 1. Organizational Chart of R&D Department

Under the mechanical technologies department and electronic technologies department there are technology families each of which works on a different technology field. Each technology family has a technology family leader and these leaders are responsible for all of the resources that works under the corresponding technology family.

3.2.2. Project Structure

In the R&D Department, on the average there are 80 active projects in a year. Most of these projects are research based projects and they require approximately 1-3 years to be completed depending on the scope. The triggering processes for the project management process are the technology management process, idea management process and portfolio management process. Projects are managed with stage-gate approach. The details of the project management approach will be given in the next subsections.

Generally but not necessarily projects are conducted under a program. A program can be defined as a set of projects. The projects in this set are interrelated in terms of the topics they are working on. Besides their own objective(s) the projects all serve to a grand objective. Each program is under the control of a program leader and each project is under the control of a project leader. Each project has its own sponsor and the sponsor is always one of the department managers.

A project starts with the feasibility analysis of the idea that triggered the project and most of the time the idea owner is assigned as the project leader. The project leader forms the project team before the start of the project and after that the activities of the project and the job assignments are detailed. Each project is evaluated at its milestones. There is no precedence relation between projects but the completion of a project can trigger new projects. During the execution of the project, the scope and content of the work to be done can go through minor changes. The resource assignments are done on a "first come first serve" basis, i.e. previous projects have the priority upon the newly initiated projects. The work schedules of the resources that work on current projects does not change when a new project enters the system. The resources are assigned to the new project only when they are available.

The projects are recorded in a project management software and the progress of the projects can be monitored through this software. The software shows all the information related to project network, work plans and resource assignments and the resources use this software to see the workloads on them. Accomplished work is recorded in a weekly basis and thus, this enables to compare the actual project progress with the planned project progress in a weekly basis.

3.2.3. Resources

There are two types of resources in the R&D Department: human resources and equipments. Human resources consist of researchers and technicians. All the equipments, machines, mechanisms and laboratories are included under the equipment category. There are approximately 200 human resources each of whom is a member of one of the departments and works for one of the technology families. Each human resource works on the average in 2-4 projects actively but this number can go up to 10 or more for some of the human resources. The human resources are multi-skilled, i.e., each human resource has its own specialty and the degree of that specialty differs from one human resource to another. This makes human resources critical for the R&D Department since the human resources are not necessarily substitutable. Most of the time when a time conflict for human resources occurs during the project execution, using a different human resource instead of the originally assigned one is not an option to solve this conflict. The resource assignments to projects and to the activities of the projects requires communication and bargaining between the technology families.

Planned weekly work schedules of resources are saved in the project management software and the planned workloads of each resource can be seen from that software in a weekly basis. It is the responsibility of each human resource to enter at the end of each week the tasks accomplished during that week into the system. For the equipment type resources this recording is done by the project leaders. The software enables to make this records for the past four weeks and does not have limit on the recorded working hours.

3.2.4. Activities

A project consists of a number of activities that have to be performed in accordance with a set of precedence and resource constraints. The activities of the project are determined and the precedence relations between activities are detailed by the project leader and the project team. Since before the start of the project, in the planning phase, it is difficult to make correct estimations on the work content of the activities, the project leader and the project team define activities as aggregate activities. Most of the activities have finish-to-start (FS) type precedence relations but there can also be start-to-start (SS) relations with time lags between activities. During the execution of a project, an activity might be preempted. Preemption does not require starting from scratch. The activity starts from where it has left off. Each activity requires resources from different departments and different technology families for certain working hours thus they are conducted in a multidisciplinary environment. The estimated requirement of the resource hours are deterministic. The required resources do not have to work on that activity together or simultaneously, they can work separately and at different times. All the information about the activities and the project is recorded to the project management software by the project manager.

3.2.5. Project Management Process of the R&D Department

For the analysis of the project management process of the R&D Department, several interviews with the project management office employees are held and the process is analyzed in detail. In the analysis, taking the focus was on the process and information flow. In the R&D Department, for the initiation, implementation and finalization of projects, "*phase-gate*" model is adopted. A phase–gate model, also referred to as a stage–gate process, is a project management technique in which an initiative or project (e.g., new product development, process improvement is divided into stages or phases, separated by gates. At each gate, the continuation of the process is decided by (typically) a manager or a steering committee. The decision is based on the information available at the time, including the business case, risk analysis, and availability of necessary resources (e.g., money, human resources with required competencies). The phases of the R&D projects of the R&D Department are categorized as follows:

1. Idea Submission Phase (Gate <-1> Phase)
2. Technical Feasibility Analysis Phase (Gate <0> Phase)
3. Project Planning Phase (Gate <1> Phase)
4. Project Implementation and Monitoring Phase
5. Reporting and Ending Phase (Gate <2> Phase)
6. Post-Project Evaluation Phase

1. Idea Submission Phase

Idea Submission Phase is the phase in which the project ideas intended for the solutions of existing problems are turned into project suggestions. The demands that are revealed during the plant visits, outputs of the technology monitoring process, outputs of the long-term product planning process and the INTER process, an internal thematic idea generation process, are the main idea sources serving for this phase of the project management process. This phase starts with the submission of the idea suggestion form, Gate <-1> form, and continues with the Gate <-1> presentation. In the Gate <-1> form, the owner of the idea gives information about the project name, project aim, target product of the project, related long-term project planning title, related program or project group name and the leader of the program of project group, related technology field, potential project customers, related financial support information (if there is any), estimated completion time, estimated sale price, estimated human resource, equipment and laboratory requirements, the opinions of the related technology family leaders, potential contributions of the project to the company, project performance objectives of the project and other related additional information and submits it to the director of the R&D Department. Before submission, the owner of the idea has an agreement with the related technology family leaders on the planned resources after a bargaining process and forms the project team. The idea suggestions are evaluated by the process manager with respect to the criteria such as conformity with the strategic plans and technology roadmaps, potential to satisfy customer demands, availability of resources and the decision on the future status of the project idea is given. After this evaluation process, each idea suggestion can be (i) accepted, (ii) postponed to be evaluated in a later time, or (iii) rejected.

This phase finalizes with a decision on the future status of the project. Related information on the subject of the project idea suggestion is recorded to the system by a project management office employee and the project idea owner if the project idea is not rejected. When the project is accepted, the proposer of the project idea is generally assigned as the project leader but if the proposer does not have an experience on the subject or as a project leader, some other person with recognized experience as a project leader and knowledge on the subject is assigned as the leader of this project.

2. Technical Feasibility Analysis Phase

Technical Feasibility Analysis Phase is a phase that starts with the acceptance of the project idea suggestion and ends with the decision on the future status of the project after the presentation of the findings and studies of this phase. In this phase, the project team that is formed in the idea submission phase mainly describes the problem, identifies the project's scope, compiles the previous studies on the subject, analyses the technical feasibility, investigates the market forecasts and prepares the performance goals of the technical feasibility analysis phase. In case of any deviations on the resource and time requirements, that are reported in Gate <-1> form, the project leader and the project team can demand additional resources and time from the process manager. Project management office is responsible from the transfer of any revisions to the project management software. At the end of the phase, project leader presents the studies conducted in this phase and the future studies planned to be done in a later phase to the technology disciplines' and the customer's opinion with the gate <0> presentation. With this presentation, a common opinion on the significance of continuation of such a project, the scope and assignments of resources, etc. These opinions are taken into account in the project planning phase if the project is accepted for planning phase. Four types of decisions can be made on the future status of the project. These decisions are listed below:

1. The project continues with the new scope that is identified in Gate <0> presentation.
2. The project is closed.
3. Additional information is needed for decision.
4. The decision is postponed.

When a closure decision is made for the project, gate <0> presentation counted as the gate <2> presentation, project closure presentation, and project closure process starts. In case of additional information is required for the decision, additional resource and time are assigned to the project and the project is reevaluated later. When the project is accepted for the next phase, performance goals and achievements of the phase are evaluated by the process manager and the performance score of the technical feasibility phase is calculated and the project announcement form, which includes information on the project book publishing date, project book preparation team and the required equipment and some other performance and risk related information are confirmed and the project is announced to the related departments. Time and resource requirements of the preparation of the project book are determined by the project leader.

3. Project Planning Phase

In this phase, the steps that will be taken for the successful completion of the project, the activities and the milestones of the project are revised with the project leader and project team. If the project is under a program, the relationships between the other projects in the program are determined with the program leader. The resource requirements are detailed, if necessary, and the interviews with the technology family leaders are held to prevent any resource conflict that can occur in future. Project risk and contingency plans are prepared and agreements and contracts with the contractors are prepared. Considering the goals, target products, work packages, risks and the milestones, performance goals and corresponding weighting of the performance goals are determined for each year and for the completion of the project. Project leader is responsible from conducting the project as planned in this phase, process manager assists the project leader during this phase and implementation and monitoring phase. The output of project planning phase is the project book. All plans regarding the activities of the project are reported in this book. This book is confirmed by the program manager, process manager, R&D director and customer of the project after the gate <1> presentation.

4. Project Implementation and Monitoring Phase

The aim of *Project Implementation and Monitoring Phase*, is to control the project execution, to take the required precautions to keep the baseline plan valid and if there is an inevitable deviations, to revise the baseline plan. For this, project leader monitors the potential risks, technology family leaders monitor the resource usage levels of related resources, program manager monitors the project to check if there is any deviation that can affect other projects. The customer, the project team and the project leader meet at the milestones and at the times when meeting is required and all the accomplished work is compared with the goals. Remaining tasks are investigated and if needed, the plan is revised. Since the projects use the same resource pool and these revisions can affect other projects, confirmation from technology family leaders are required before any revision. In this phase, the project is analyzed with respect to quality, cost and time, compared with the predetermined goals and updated: (i) if the scope of the project changes, (ii) if there is more than 10% time deviation, (iii) if there is more than 20% resource usage deviation and (iv) if there is a change in the project budget. This analysis and comparison are accomplished through meetings and project management software. When an update is needed, this update is notified to the project management office. This phase ends with the review of the last milestone of the project.

5. Reporting and Ending Phase

This phase starts with the accomplishment of the last defined goal or closure of the project because of some internal or external reasons before the goals are achieved. In this phase, the information obtained during the execution of the project is disseminated through the R&D departments, project is evaluated and assessed, and the intellectual property management process is initiated. Findings and outputs of the project are shared with the project customer, process manager, project team and related specialists in the gate<2> presentation. During the presentation or after the presentation, and rarely, if there is a need, the project can be decided to be continued with additional activities that are thought to have important affect on the results and deliverables of the project. In this case, gate<2> presentation is repeated after the completion of these new activities. When the project is decided to be closed, required closure forms are filled and the performances of the project, and the project team are evaluated.

6. Post Project Evaluation Process

This phase aims at recording the information obtained during the project and the outputs. For this purpose, the “post project evaluation form” is filled by the project leader and these forms are reviewed by the process managers and program leaders on a regular basis. These information are published as a “research note” at the end of every year. In this note, the contribution of the closed projects to the knowledge base of the R&D Department is summarized.

3.3. PROBLEM DEFINITION AND FORMULATION

The problem on hand is the scheduling of the R&D projects with a priori assigned resources to the activities in a stochastic and dynamic environment present in the R&D Department of a leading home appliances company in Turkey. In this thesis, a three-phase model is developed for this problem. With this model, we will be contributing to the 3^{rd}, 4^{th}, 5^{th} and 6^{th} phases of project management process of the R&D Department.

A project consists of a number of events and activities that have to be performed in accordance with a set of precedence and resource constraints. The activities require two types of renewable resources: human resource and equipment. Equipment includes the equipments, machines, mechanisms and laboratories. Non-renewable resources are not considered. The problem environment under consideration contains multiple projects consisting of activities using multi-skilled renewable resources. The resource requirement of activities and hence the durations of activities are uncertain. The project network is of activity-on-node (AON) type with FS and SS precedence relations with zero or positive time lags. No precedence relation is assumed between projects. The problem on hand can be considered an extension of the resource constrained multi-project scheduling problem (RCMPSP) with generalized precedence relations and multi-skilled resources to include preemption, stochastic activity duration and resource availabilities and dynamic arrival of projects. The objective is, considering the possible activity time deviations beforehand, generating solution—robust baseline project schedules and minimizing the completion time for overall project makespan. Solution robustness is a measure of difference of the realized schedule with the baseline schedule. In our case, we use expected total instability for solution robustness and this total instability is expected with total sum of absolute deviations (TSAD) for the activities. TSAD is the sum of the absolute deviations between the actual starting time and the starting times realized in the simulations over all activities. The MILP formulation of the problem is introduced below.

<u>Sets and indices</u>

$K =$ set of all schedule realizations

k = realization index; $k \in K$

$V =$ set of all activities

i = activity index; $i \in V$ set of all resources

$R_i =$ set of all resources that activity i requires, $i \in V$

$r =$ resource index; $r \in R_i$

$I_i =$ set of all resources that have to work together in activity i, $i \in V$

$PFS =$ set of FS type precedence relations between activities i and j, $i, j \in V$

$LFS =$ set of lags for FS type precedence relations between activities i and j, $i, j \in V$

PSS = set of SS type precedence relations between activities i and j, $i, j \in V$

LSS = set of lags for SS type precedence relations between activities i and j, $i, j \in V$

$\mathcal{T}$ = set of time periods, t = time indices; $t \in \mathcal{T} = \{1, \dots, |\mathcal{T}|\}$

Parameters

s_i^k = starting time of activity i in realization k, $i \in V$, $k \in K$

$\bar{d}_{ir}$ = expected required working hours of activity i from resource r, $i \in V$, $r \in R_i$

a_{rt} = available working hours of resource r during time period t, $r \in$ R, $t \in$ T

w_i: marginal cost of deviating from baseline schedule for activity j

T = total length of the time horizon

M = a big numbe

Here, w_i is an activity specific flexibility factor and can be explained as the difficulty of performing an activity out of its predetermined performing time window.

Decision Variables

S_i = starting time of activity i, $i \in V$

F_{i} = starting time of activity i, $i \in V$

S_{r_i} = starting time of resource r to work on activity i, $i \in V$, $r \in R_i$

F_{r_i} =finishing time of resource r- to work on activity i, $i \in V$, $r \in R_i$

$$x_{rit} = \begin{cases} 1; \text{ if resource r is working on activity i during time t}, i \epsilon V, r \epsilon R_i, t \epsilon T \\ 0; \text{ otherwise} \end{cases}$$

y_{rit}= amount of hours that resource r spends on activity i during time t, $i \epsilon V, r \epsilon R_i, t \epsilon T$

Auxiliary variables

$$\delta_{rit} = \begin{cases} 1; \text{t is the min starting time of resource r's working on activity i}, i \epsilon V, r \epsilon R_i, t \epsilon T \\ 0; \text{otherwise} \end{cases}$$

$$\beta_{rit} = \begin{cases} 1; \text{t is the max starting time of resource r's working on activity i, i}\epsilon\text{V, r}\epsilon\text{R}_\text{i}\text{, t}\epsilon\text{T} \\ 0; \text{otherwise} \end{cases}$$

$$\mu_{ri} = \begin{cases} 1; \text{resource r is the first starts working on activity i}, i \epsilon V, r \epsilon R_i \\ 0; \text{otherwise} \end{cases}$$

$$\alpha_{ri} = \begin{cases} 1; \text{resource r is the resource that last finishes working on activity i}, i \epsilon V, r \epsilon R_i \\ 0; \text{otherwise} \end{cases}$$

Mathematical Model

$$\boldsymbol{Min}\ Completion\ Time\ of\ Projects = \boldsymbol{Min}\ \boldsymbol{F_{last}} \quad (3.1)$$

$$\boldsymbol{Min}\ Total\ Instability\ = \boldsymbol{Min}\ \boldsymbol{TSAD} =\ \sum_{i\in V} w_i \sum_{k\in K}(\left| \boldsymbol{s_i^k} - s_{\boldsymbol{i}} \right|) \quad (3.2)$$

s.t

$$S_j - S_i \geq LFS_{ij}, \forall (i,j) \epsilon PFS, LFS \quad (3.3)$$

$$S_j - S_i \geq LSS_{ij}, \forall (i,j) \epsilon PSS, LSS \quad (3.4)$$

$$\sum_{i\epsilon V} y_{rit} \leq a_{rt}, \forall r\ \epsilon R_i, \forall t\epsilon T \quad (3.5)$$

$$\sum_{i\epsilon V} y_{rit} \geq \bar{d}_{ir}, \forall i\epsilon V, \forall r\epsilon R_i \quad (3.6)$$

$$y_{rit} - Mx_{rit} \leq 0, \forall i\epsilon V, \forall r\epsilon R_i, t\epsilon T \quad (3.7)$$

$$t + M(1 - x_{rit}) - 2M(1 - \delta_{rit}) - S_{ri} \leq 0, \forall i\epsilon V, \forall r\epsilon R_i, t\epsilon T \quad (3.8)$$

$$S_{ri} - t - M(1 - x_{rit}) \leq 0, \forall i\epsilon V, \forall r\epsilon R_i, t\epsilon T \quad (3.9)$$

$$\sum_{t\in T} \delta_{rit} = 1, \forall i\epsilon V, \forall r\epsilon R_i, \quad (3.10)$$

$$t - M(1 - x_{rit}) - F_{ri} \leq 0, \forall i\epsilon V, \forall r\epsilon R_i, t\epsilon T \quad (3.11)$$

$$S_{ri} - t + M(1 - x_{rit}) + 2M(1 - \beta_{rit}) \leq 0, \forall i\epsilon V, \forall r\epsilon R_i, t\epsilon T \quad (3.12)$$

$$\sum_{t\in T} \beta_{rit} = 1, \forall i\epsilon V, \forall r\epsilon R_i \quad (3.13)$$

$$S_{ri} - M(1 - \mu_{ri}) - S_i \leq 0, \forall i\epsilon V, \forall r\epsilon R_i \quad (3.14)$$

$$S_i - S_{ir} \leq 0, \forall i\epsilon V, \forall r\epsilon R_i \quad (3.15)$$

$$\sum_{r\in R_i} \mu_{ri} = 1, \forall i\epsilon V \quad (3.16)$$

$$F_{ri} - F_i \leq 0, \forall i\epsilon V \quad (3.17)$$

$$F_i - F_{ri} - M(1 - \propto_{ri}) \leq 0, \forall i\epsilon V, \forall r\epsilon R_i \quad (3.18)$$

$$\sum_{r\in R_i} \propto_{ri} = 1, \forall i\epsilon V \quad (3.19)$$

$$x_{rit}\ \epsilon\ \{0,1\}, \forall r\epsilon R_i, \forall i \in V, \forall t\epsilon T \quad (3.20)$$

$$S_{ri}, F_{ri} \geq 0, \forall r\epsilon R_i, \forall i \in V \quad (3.21)$$

$$S_i, F_i \geq 0, \forall i \in V \quad (3.22)$$

$$\delta_{rit}, \beta_{rit} \epsilon\ \{0,1\}, \forall r\epsilon R_i, \forall i \in V, \forall t\epsilon T \quad (3.23)$$

$$\mu_{ri}, \propto_{ri}\ \epsilon\ \{0,1\}, \forall r\epsilon R_i, \forall i \in V \quad (3.24)$$

The mathematical model differentiates from the standard mathematical models formulated for the preemptive version of RCMPSP by the parameters that the model uses and its multi-objective nature. Besides the minimization of the completion time of the projects (3.1), a second objective considered is the minimization of TSAD (3.2). While constraint set (3.3) implies FS type precedence relation, constraint set (3.4) implies SS type precedence relations with zero or larger time lags. Constraint set (3.5) reflects the resource limits on each resource and for every time period t. Constraint set (3.6) expresses the resource requirement constraints of the activities. Here, notice that, since the resource requirements of the activities are uncertain, estimated resource requirements of the activities are used in the right hand side of the constraint. Constraint set (3.7) guarantees that if a resource is not active in a time instance, that resource cannot spend any time on that activity. Constraint sets (3.8)-(3.10) and (3.11)-(3.13) guarantee the correct assignment of S_{ir} values to determine the starting times of resource r's working on activity i and F_{ir} values to determine the finishing times of resource r's working on activity i, respectively. In the same manner, constraint set (3.14)-(3.16) and constraint sets (3.17)-(3.19) using the S_{ir} and F_{ir} values, determine the S_i, starting time of activity i and F_i, finishing time of activity i, respectively. Constraint sets (3.20) through (3.24) represent the feasible ranges for the decision and auxiliary variables.

With this problem definition and formulation, we have created a different project scheduling environment that has different data requirement than the project scheduling environments existing in the literature. In the literature, instead of requiring working hours, the activities require resources for deterministic or stochastic durations. Most of the time, it is assumed that resources are dedicated and cannot work on more than one activity in a defined time period. In our case, the resources can work on more than one activity in a time period and the working hours of resources for an activity can differ over the periods that the activity is executed. Additionally, the concept of preemption of activities with the preemption of resources' workings on activities is also introduced.

CHAPTER 4

A THREE-PHASE APPROACH FOR ROBUST PROJECT SCHEDULING

In this chapter we have presented efficient proactive-reactive solution approaches to schedule the R&D projects in a stochastic and dynamic environment. A three-phase model is developed incorporating data mining and project scheduling techniques to schedule the R&D projects. Section 4.1 introduces Phase I of the proposed approach. In Phase I, the aim is, considering the percentage resource usage deviations of the projects and the percentage resource usage deviations of the activities in the projects as a risk measure, to construct a percentage resource usage deviation assignment procedure to predict the percentage resource usage deviation levels of activities. Then, Phase II, proactive bi-objective project scheduling phase, aiming at generating non-dominated robust baseline project schedules with a bi-objective GA is presented in Section 4.2. This chapter concludes with Phase III of the proposed approach that suggests a rescheduling procedure that will be triggered when a disruption occurs in the baseline schedule.

4.1. PHASE I: DEVIATION ANALYSIS PHASE

In this section our focus will be Phase I of the three-phase approach proposed as a solution to the preemptive version of the multi-objective RCMPSP with generalized precedence relations. It should be noted that, this phase is not problem-specific, i.e., it can be implemented for any stochastic project scheduling problem. In this phase, to consider the percentage resource usage deviations of the projects as a risk measure in the proposed model , risk tables of randomly selected 40 R&D projects in the firm are constructed and analyzed. The lack of some required components of the risk data precluded the implementation of risk-driven approach proposed in the literature for robust project scheduling. As an alternative to considering the risk-based deviations in project scheduling, we have proposed making use of the most known data mining tools which are feature subset selection, clustering and classification. The aim of each data mining technique and the related literature are given briefly in the first step of Phase I as the solution methodology. We kindly suggest the interested readers to refer to Tan et. al.(2006), and Du (2010) for detailed information on the data mining tools.

Phase I of the proposed model suggests classifying the projects with respect to their percentage resource usage deviations, then classifying the activities with respect to their percentage resource usage deviations, thus constructing a resource usage deviation assignment procedure for the prediction of the percentage resource usage deviation levels of the activities. This resource usage deviation assignment procedure will be used later during the proactive project scheduling phase and whenever needed will trigger the reactive project scheduling phase on a disrupted project plan.

Phase I, which is referred to as the deviation analysis phase, is comprised of two steps: (i) Deviation Analysis of Projects and (ii) Activity Deviation Assignment Step. It aims at predicting the deviation level of the projects and the deviation level of the activities of the projects.

4.1.1. Step I: Deviation Analysis of Projects

The objective of the first step of Phase I is establishing a classification model based on real data collected from the R&D Department, in order to classify the R&D projects with respect to their percentage resource usage deviation from mean and predict the percentage resource usage deviation levels of projects. Thus, by using the classification model, in the planning phase that is to say before the project actually starts, predicting its resource deviation level can be possible, and the needed precautions can be taken. Furthermore, this information will be used in the second step of the proposed methodology in order to obtain the percentage resource usage deviation distributions of activities. The resulting resource deviation distributions of the activities will be used later in Phase II when assigning start and finish times to projects and their activities.

4.1.1.1. Feature Subset Selection

Construction of the classification model for the prediction of a project's percentage resource usage deviation levels starts with determining the features that can have a positive or negative effect on the percentage resource usage deviations of the projects and finding the best subset of these features in terms of prediction. Feature subset selection is the process of choosing a subset of the original predictive features by eliminating redundant and uninformative ones. Feature subset selection is an important problem in knowledge discovery. The objective of feature subset selection is three-fold. It not only improves the prediction performance of the predictors, but also provides faster and more cost-effective predictors and provides an insight from determining relevant modeling features. The techniques of feature subset selection differ from each other in the way they incorporate the search method in the added space of feature subsets in the model selection. As defined by John et. al. (1994), feature subset selection methods essentially divide into wrappers and filters depending on how they combine the feature subset selection search with the construction of the classification model.

Filters select subsets of variables as a pre-processing step, independently of the chosen predictor. Filter techniques assess the relevance of features by looking only at the intrinsic properties of the data. In most cases a feature relevance score is calculated, and low-scoring features are removed. Advantages of filter techniques are that they easily scale to very high-dimensional datasets, they are computationally simple and fast, and they are independent of the classification algorithm. As a result, feature subset selection needs to be performed only once, and then different classifiers can be evaluated. A common disadvantage of filter methods is that they ignore the interaction with the classifier (the search in the feature subset space is separated from the search in the hypothesis space), and that most proposed techniques are univariate. This means that each feature is considered separately, thereby ignoring feature dependencies, which may lead to worse classification performance when compared to other types of feature subset selection techniques. In order to overcome the problem of ignoring feature dependencies, a number of multivariate filter techniques were introduced, aiming at the incorporation of feature dependencies to some degree. On the other hand, the wrapper methodology, popularized by Kohavi and John (1997), offers a simple and powerful way to address the problem of variable selection, regardless of the chosen learning machine. In fact, the learning machine is considered a perfect black box and the method lends itself to the use of off-the-shelf machine learning software packages. In its most general formulation, the wrapper methodology consists in using the prediction performance of a given learning machine to assess the relative usefulness of subsets of variables.

It is argued that, compared to wrappers, filters are faster. Still, recently proposed efficient wrapper methods are competitive in that respect. Another argument is that some filters (e.g. those based on mutual information criteria) provide a generic selection of variables, not tuned for/by a given learning machine. Another compelling justification is that filtering can be used as a preprocessing step to reduce space dimensionality and overcome overfitting. In our approach, for the feature subset selection, we suggest utilizing an open source data mining tool, namely WEKA developed by Hall et. al. (2009) and comparing the performances of different feature subset selection algorithms that WEKA supports and select the best ones in terms of prediction to proceed.

4.1.1.2. Clustering

The next step after feature subset selection is using the percentage resource usage deviations of projects from their mean, to obtain a label output representing the percentage resource usage deviations of projects since most of the classification algorithms work on nominal (labeled) output. To obtain nominal output values, we propose application of a clustering algorithm on the percentage resource usage deviations of projects from their mean. Clustering is the process of dividing a data set into mutually exclusive groups such that the members of each group are as "close" as possible to one another, and different groups are as "far" as possible from one another, where distance is measured with respect to all available variables. Clustering has a wide range of applications some of which are spatial data analysis, customer profiling, market research, web-log record analysis for websites, pattern recognition, and so on. A clustering solution has three essential elements: (i) a sensible measure of proximity, which is a similarity or dissimilarity among data objects; (ii) a goodness of fit function to evaluate the quality of the resulting clusters; and (iii) an effective clustering algorithm. The clustering algorithm uses the similarity measuring function repeatedly to determine which group a data point should belong to. The most popular proximity metric for continuous features is the "Euclidean distance". There exist a large number of clustering algorithms in literature. The choice of clustering algorithm depends on the data type and particular purpose and application. Clustering methods can be classified into four categories: hierarchical methods, non-hierarchical partition-based methods, density-based methods, and model-based methods.

In hierarchical clustering all of the instances are organized into a hierarchy that describes the degree of similarity between those instances. A hierarchical clustering algorithm yields a dendrogram representing the nested groupings. Such representation may provide a great deal of information and many algorithms have been proposed. Partitional clustering, on the other hand, simply creates a single partition of the data, where each instance falls into one single cluster. The most known partitional clustering method is the classic and still popular K-means algorithm developed by MacQueen (1967). The basic idea of the K-means algorithm is to divide the data into K partitions. K-means is a simple iterative algorithm starting with randomly selected instances as centers. Each instance is first assigned to the closest center. Closeness is often measured by Euclidean distance but can be measured by some other closeness measures as well.

Given those assignments, the cluster centers are recalculated and each instance is again assigned to the closest center. This is repeated until no instance changes clusters after the centers are recalculated; that is, the algorithm converges to a local optimum. Since the algorithm is very sensitive to the selection of the initial centroid, running the algorithm with different initial centroids may be a good idea to improve the solutions obtained. The most used evaluation measure of clusterers is the sum of squared error (SSE) measure. Beside this simple K-means algorithm, there are several variants of the simple K-means algorithms. Against the backdrop limitations of the basic methods, a large number of more sophisticated clustering methods have been developed in the literature. As stated in Du (2010), these sophisticated methods can be categorized as prototype-based methods, density-based methods, graph based methods and model-based methods.

Prototype-based methods group the data points as prototypes and modify them through the clustering process. In addition to simple K-means and variants of the simple K-means algorithm, fuzzy C-means method proposed by Bezdek (1981) is one of the prototype based methods. Density based methods use a density function to measure the density of either a region in the multi-dimensional space of the data or a neighborhood of a data point. DBSCAN algorithm developed by Ester et. al. (1996) is one of the density-based clustering methods. Model-based methods try to discover hidden statistical models that best fit the given data set and describe the data clusters and probabilistic memberships of data points to the clusters. Expectation Maximization method (EM) introduced by Lauritzen (1995) is one of the most known model-based clustering techniques. Similar to K-means method, the EM algorithm first randomly decides the initial estimates for the parameters for a set of distributions, then it refines the estimates in an iterative process. For a detailed discussion on clustering methods we refer to Jain et. al. (1999) and Berkhin (2006). In our approach, since the clustering of the numeric output is not the most essential part of Phase I, we propose the use of simple K-means of MacQueen (1967) to obtain the nominal output values for each project from the numeric ones.

4.1.1.3. Classification

After we obtain the nominal output, we are ready to develop the classification model. In that stage, we propose the use of both numeric and nominal output that both represent the percentage resource usage deviations of projects from their mean. By doing so, we will have more than one classification model to select, one model for each output type-feature subset combination, each having a different performance on the data. Classification is the task of assigning objects to one of several predefined categories which serve as an explanatory tool to distinguish between objects in different classes. A classification model is used to predict the class label of unknown records. As stated in Tan et. al. (2006) classification models are most suited for predicting or describing data sets with binary or nominal categories. They are less effective for ordinal categories.

Classification is a two step process. In the first step, a model is built describing the predetermined set of data classes. Since the class label of each sample is known in advance, this step is also known as supervised learning. Typically the learned model is represented in the form of classification rules, decision trees, or mathematical formulae. In the second step, the model is used for classification. First, the predictive accuracy of the model is estimated. In order to develop a reliable model in classification, the data set is normally divided into two subsets: a training set and a test set. Data points from the training set are used to induce a model whereas data points from the test set are used to evaluate the accuracy of the model. The accuracy is measured by the error rate of classification. Classification methods can be grouped into several categories: decision tree induction approaches, rule based approaches, nearest neighbor approaches, Bayesian classifiers and artificial neural networks. These approaches are categorized according to the form in which the classification model is represented. Each technique employs a learning algorithm to identify a model that best fits the relationship between the feature set and class label of the input data.

A typical decision tree consists of leaf nodes, internal nodes and links. A leaf node represents a class label, an internal node represents the name of the attribute and the link from a parent node to a child node represents a value of the feature of the parent node. As a major approach to classification, decision tree induction has received a great amount of attention from researchers over the last three decades. Examples of some well known decision- tree induction algorithms include C4.5 developed by Quinlan (1993) and CART developed by Breiman et. al. (1984). For an overview of decision tree induction algorithms, we refer to the survey articles by Moret (1982), Buntine (1993), Murthy (1998), and Safavian and Landgrebe (1998).

The nearest neighbor classification approach considers a classification model as a memory space of representative examples. IB1 and IBk developed by Aha and Kibbler (1991) known as lazy algorithms are some of the neighbor classification algorithms. IB1 uses normalized euclidian distance to find the training instance closest to the given test instance, and predicts the same class as this training instance. IBk which is a K-nearest classifier can select appropriate value of k and can also do distance weighting. K*, another nearest neighbor classification method, developed by Cleary and Trigg (1995) is an instance-based classifier, that is the class of a test instance is based upon the class of those training instances similar to it, as determined by some similarity function. For an overview of the nearest neighbor classification methods we refer to Cover and Hart (1995).

Rule based classification algorithms construct a sequence of rules discovered directly or indirectly from the training set of instances. While the direct approach discovers the rules directly from the training set, the indirect approach first discovers a model in some other form and translates this form into rules. Inducing a tree into a set of classification rules is an example of indirect approach. The ripper algorithm developed by Cohen (1995) is an incremental rule induction algorithm with pruning that works for both categorical and numeric features and Prism method developed by Cendrowska (1987), which can deal with only nominal features is an example of rule based classification methods.

Bayesian classification models describe probabilistic relationships between feature values based on the Bayes Theorem: the posterior probability of the class that a point belongs to is approximated using prior probability drawn from the training set. The classification model estimates the likelihood of the point belonging to each class then it gets the label of the class with the highest probability. The naive Bayes method makes the assumption that descriptive features are conditionally independent of each other given the class labels are known and widely used for different applications.

Among the basic Bayesian classification methods, while the Naïve Bayes method of Duda and Hart (1973) is a simple naive Bayes classifier where numeric features are modeled by normal distribution, the Naïve Bayes method of John and Langley (1995) uses a kernel estimator to estimate the distribution of the numeric features rather than assuming a normal distribution.

Artificial neural network approach for classification takes the trained network with the right weights attached to the links as the classification model. Developing a neural network model requires the definition of a network topology and the training of the defined network.

In this step, instead of selecting the classification model that performs best on the given data, we propose to use all the prediction results of different classification models that will be obtained and produce probabilistic predictions for the percentage resource usage deviation levels of the projects. That way, we will be providing probabilistic membership of the projects to the predetermined percentage resource usage deviation classes. This approach is more robust than selecting a single classification model and making a deterministic predictions since providing a probabilistic predictions precludes the missing of the actual deviation class of projects and tolerates the error caused by model selection. In reality, instead of making a class prediction, giving a closeness value to each deviation class is more understandable by the project managers thus it makes sense both in terms of convenience of perception and correctness.

To summarize, the input of Step 1 of Phase I consists of various features that are thought to be relevant for determining the percentage resource usage deviations of the projects and the values that these features take for each project. First, with the application of feature subset selection algorithms, the most important features are determined then clustering is applied to the percentage resource usage deviation of projects to generate actual nominal class labels of the projects. Afterwards, these nominal output values and the percentage resource usage deviation values of the projects are used in the learning stage of classification model construction. For each feature subset and output combination, a classification model is constructed and using the predictions resulting in each classification model, membership to a deviation class is predicted probabilistically for each project. This information constitutes one of the inputs for Step 2 of Phase I. In this step, as a side contribution, relationships between important features on the prediction of deviation classes are also identified to have a better understanding of the system and to enable making fine-tuning on the important feature values of a project in order to bring the project's deviation at a desired level.

4.1.2. Step II: Activity Deviation Assignment Procedure

In Step I, we develop a model to predict the percentage resource usage deviation level of a newly arrived project based on its various input features. Using this information, in Step II, we also develop a model to predict the percentage resource deviation of the activities of this newly arrived project. The aim of Step II of Phase I is to obtain percentage resource usage deviation distributions for each project deviation class - activity class combination to be used in Phase II of the proposed solution approach for robust project scheduling. Step II of Phase I starts with the classification of all the activities, thus forming a number of activity subsets. Forming a distribution requires sufficient number of replications. Since we are dealing with R&D projects and the activities of R&D projects are usually unique and the work content is characteristic among all the activities, to obtain sufficiently large amount of data for a valid distribution of percentage resource usage deviation from the mean requirement, such an aggregation and classification is compulsory.

Using the model developed in the previous step, for each activity class of a newly arrived project, using the percentage resource deviation information of already completed activities in the corresponding activity class we form the resource deviation distribution of that activity class. Note that, the resource usage deviation classes of already completed projects are known, thus we know the frequency and deviation level information for the activities in each project deviation class- activity class combination. To form the resource usage deviation distribution for an activity class, we set a minimum and maximum value on the percentage resource usage deviation from its mean requirement that an activity can take and then this relatively large range is divided into smaller intervals. After that, for each activity class, frequency of the number of activities in each project class and in each interval is obtained. In this case, since the project's percentage resource deviation prediction is probabilistic, we cannot directly use either the frequency distribution for the activity class or the frequency distribution for the activity class-project deviation class combination. We need to adjust the frequency distribution for the activity classes using the project's resource deviation classes. In each interval, we will be knowing how many activities are observed and the allocation of these activities to the project deviation classes. The adjusted frequency information for an interval is obtained summing the multiplications of the activity numbers in each project deviation class with the probability of the membership of the newly arrived project to that project deviation class. As an illustration, assume that we have two percentage resource usage deviation from the mean requirement classes for the projects and the newly arrived project is predicted to be the member of class one and class two with a probability of 50% each. Also assume that there are 5 activity classes in the newly arrived project, the percentage resource usage deviation range is divided into 4 intervals. Now let's consider the first interval and try to find the adjusted frequency information of this interval. In the first interval, 40 activities belonging to projects in project class one and 80 activities belonging to projects in project class two. Adjusted frequency of an activity's having a percentage resource usage deviation from the mean requirement value in the first interval is the sum of the multiplications of 40 and 50% and 80 and 50%. Thus, the adjusted frequency is calculated as 60.

After obtaining these adjusted frequency distributions, the probabilities of an activity having a deviation level in each range is calculated and the piecewise linear percentage resource usage deviation distributions of each activity class in the newly arrived project is formed. This distribution is used to assign percentage resource usage deviation level to the to-be-scheduled activities. Step I of Phase I is called from Phase II of the proposed robust project scheduling approach whenever a new project enters the project management system and Step II of Phase I is called from Phase II whenever robust project schedules need to be obtained.

4.2. PHASE II: PROACTIVE PROJECT SCHEDULING WITH A BI-OBJECTIVE GA

In this section, we present a bi-objective genetic algorithm that uses the output of the Phase I and two scheduling approaches each using the bi-objective GA. The aim of these approaches is to generate non-dominated solution robust project schedules, i.e., baseline schedules that do not differ much from the actually realized schedules, with the minimum makespan for the completion of all projects scheduled. This expected difference between the baseline schedule and the actually realized schedule is measured with the total sum of absolute deviations (TSAD) of the schedule through K number of possible schedule realizations in both approaches. The proposed approaches, the single project scheduling approach, and the multi-project scheduling approach differ in the way they adopt for the scope of scheduling. Single project scheduling approach considers the remaining part of the schedules of the already active projects as fixed and schedules only the newly arrived project using the currently available resources. The multi-project scheduling approach, on the other hand, schedules all the active projects in the system again together with the newly arrived project. Since the two scheduling approaches differ in the way they adopt for the scope of scheduling, the definitions of TSAD and makespan, thus, the objectives considered in the bi-objective GA also differ although they both try to minimize TSAD and makespan. Note that in the proactive project scheduling approaches, a set of non-dominated robust project schedules are generated. From these non-dominated robust schedules, the decision maker can choose the schedule that best fits the current project management environment in the system.

In the following subsections, first, we present the single project scheduling approach and the multi-project scheduling approach along with the differences between them. After that, we have given the basic scheme of the proposed bi-objective GA procedure and gave the common grounds of the procedure in the two scheduling approaches proposed.

4.2.7. Single Project Scheduling Approach

As we have stated in the previous chapter, when a new project is initiated in the R&D Department, it is scheduled with the available resources at that time. While assigning a time for a resource for the new project, the work schedules of the resource stay the same for the projects that it currently is assigned to. This resource can work for the new project only in its available times. This policy is reasonable, since changing the schedule of the currently active projects might lead to system nervousness. Our single project scheduling approach leans on this rationale and in this multi-project environment it considers only the newly arrived project and schedules it with the currently available resources without any change in the project plans of the existing projects. The objectives considered in the single project scheduling approach are minimizing the TSAD and the makespan of the newly arrived project. The output of this single project scheduling approach is the set of non-dominated robust project schedules for the newly arrived project.

4.2.8. Multi- Project Scheduling Approach

Although it can increase system nervousness to change the schedules of the current projects when a new project enters the system, these changes, even if they are small, can yield much better schedules for the newly arrived project. For this reason, in our multi- project scheduling approach, we suggest considering all the active projects with the newly arrived project and scheduling them together. To do so, when a new project enters the system, all the not attempted and attempted but incomplete activities of the current projects are added to the project network of the newly arrived project with their remaining work and a composite project network is obtained. Then, this composite project is scheduled as a single project. The objectives considered are minimizing the makespan and the TSAD of this composite project. This time TSAD of the composite project is not only the TSAD of the newly arrived project. The difference in the starting times of the existing activities between the new schedule and the old schedule also contributes to the TSAD value. Thus, TSAD of a schedule is the sum of the TSAD of existing activities and the TSAD of newly arrived project. Since this time, the starting time of the existing activity is compared with only the previous starting time of that activity instead of comparing K possible realizations, this difference is multiplied with K and summed over all existing activities to obtain the TSAD of existing activities. With this new TSAD definition, the multi-project scheduling approach searches for schedules which do not differ too much from the original baseline schedules of the current projects and robust baseline schedules for the newly arrived project. The output of this multi-project scheduling approach is a set of non-dominated project schedules that differ as little as possible from the original baseline plans for the current activities and robust baseline schedules comprising all the activities active in the system.

4.2.1. Basic Scheme of the Bi-Objective Genetic Algorithm

Proposed bi-objective GA is an adopted version of NSGA-II suggested by Deb et al. (2002), which uses an explicit diversity generation procedure along with an elite-preservation procedure. The GA framework of the procedure starts with the computation of an initial population, i.e., the first generation, which is described in subsection 4.2.4. The number of individuals in the population is referred to as POP, which is assumed to be an even integer. After each solution is decoded as a schedule (see subsection 4.2.3) the population is sorted based on the non-domination levels (see subsection 4.2.5.1), then each solution is assigned a fitness (rank) equal to its non-domination level (1 is the best level, 2 is the next-best level, and so on) with respect to its objective function values. Thus, minimization of fitness is assumed. After that, the population is partitioned into pairs of individuals. To each resulting pair of (parent) individuals, we apply the crossover operator (see subsection 4.2.5.4) to produce two new (a daughter and a son) individuals. Subsequently, we apply the mutation operator (see subsection 4.2.5.5.) to the genotypes of the newly produced children. Since elitism is introduced by comparing current population with previously found best non-dominated solutions, the procedure is different after the initial generation. After computing the fitness of each child individual, we add the children to the current population, leading to a population size of 2*POP. Then the population is sorted into a different non-domination levels (frontiers) and the reduction process (see 4.2.5.6) which makes use of a selection operator to reduce the population to its former size POP is applied. Doing so, we obtain the next generation to which we again apply the crossover operator and so on. This process is repeated for a pre-specified number of generations which is denoted as *Total Generations*. The steps of population management are given in Figure 4.1. P_t, Q_t, R_t and F denote the current population, offspring population, combined population and F a frontier of a population, respectively.

```
Set populations R_t, P_t, Q_t → 0;
Generate initial Population;
Pop = InitialPopulation
Perform non-dominated sorting on Pop
    for; generation ∈ TotalGenerations do
        for q = 0 → NewBornCount do
            Crowded Tournament Selection for selecting 2 Parents;
            Generate a daughter chromosome d by one-point crossover;
            if RandomNo ≤ MutationProbability then ⇒ SwapMutation
            end if
            Evaluate daughter d and find fitness pairs
            Q_t.Add(d);
            Generate a son chromosome s by one-point crossover;
            if RandomNo ≤ MutationProbability then ⇒ SwapMutation
            end if
            Evaluate son s and find fitness pairs
            Q_t.Add(s);
        end for
        Combine Population and Offspring: R_t = P_t + Q_t;
        Perform non-dominated sorting on R_t ;
        Identify different frontiers Fall in R_t ;
        Set new Population P_t → 0, count → POP;
        while count > 0 do
            for Frontier F ∈ F_all do
                if |F| < count then
                    Add members of F → P_t
                    count ⇐ count − |F|
                else
                    Calculate crowding distance (F)
                    Add k = count individuals with highest cd_i to P_t , count ⇐ 0
                end if
            end for
        end while
    end for
```

Figure 4. 1. Pseudocode for Population Management of Proposed Bi-objective GA

4.2.2. Chromosome Representation

The chromosome structure in the proposed bi-objective GA is composed of a precedence feasible activity sequence list, A, of the activities of the project network. Simulation experiments performed by Hartmann and Kolisch (2000) reveal that performance of activity-list representation is superior to other discussed representations.

4.2.3. Schedule Generation Scheme

In our problem, the activities require specific resources for certain hours. Thus, the resources that an activity requires do not need to either work together or work simultaneously. Furthermore, they can stop working on that activity for a while and then continue later, i.e., the work of the resources on activities are preemptive. Hence, this might lead to preemptive activities. The starting times, ending times and the durations of the activities are determined by the work schedules of the resources that each activity requires. The starting time (ending time) of an activity is basically the starting time (ending time) of the resource that first (last) starts (finishes) his work on that activity. Therefore, to generate a schedule from a chromosome, we mainly schedule resources instead of activities. In our resource schedule generation procedure, a schedule is represented with the lists of resource, activity, week and amount (r,a,t,k) quadruple. Each (r,a,t,k) quadruple in this list shows that resource r works on activity a at time instant t for k working hours. Our resource schedule generation scheme starts with scheduling the resources of the first activity in the chromosome. Note that, resource order for scheduling is not important since all the orders give the same work schedule for that activity. Using the available working hours of resources in each time instant and taking into account the earliest precedence feasible starting time of activities, starting at the first available time instant, the resources are scheduled until they reach the required working hours. After all the resources that the first activity in the chromosome are scheduled, the starting and ending time of that activity is determined by simply checking the work schedules of the resources that activity requires. Then, the earliest starting time of the successor activities are updated. This procedure is repeated until all the activities in the chromosome are scheduled.

4.2.4. Chromosome Evaluation

The calculation of the fitness value for a chromosome is based on a set of K realizations reflecting the uncertainty around the activity resource requirements. For a given order of activities both the overall makespan and solution robustness are assessed through a set of K realizations mimicking the implementation phase, where a realization corresponds to a sample instance obtained by a simulation run using the activities' percentage resource requirement deviation distributions. For this purpose, two alternative fitness calculation procedures with the objective of quality robustness represented with makespan and solution robustness, expressed in terms of TSAD value of the robust activity starting times from their counterparts in all K realizations are considered.

4.2.4.1. Fitness Calculation Procedure1

In the first alternative, robust project schedule of a chromosome is obtained by solving the relaxed TSAD model, since the number of variables are large in the TSAD model. Using the realization information, this relaxed TSAD model aims at finding a schedule that minimizes the TSAD value of the scheduled activities. In this relaxed TSAD model, the constraints are ignored. A pseudocode for this alternative is presented with Algorithm 2 in Figure 4.2. K represents the number of simulations, A represents the set of activities.

```
FOR K simulations do
    Generate activity resource requirements for each a ∈ A using deviation assignment procedure
    FOR ALL a ∈ A DO
        Schedule a to its earliest sequence, precedence and resource feasible starting time
END FOR
SOLVE Relaxed Minimum TSAD Model and compute the robust starting times
SCHEDULE the activities using these robust starting times as the earliest starting times.
OBTAIN starting times of activities, makespan and the TSAD of the schedule
```

Figure 4. 2. Pseudocode for Fitness Calculation Procedure

Fitness calculation procedure1, first, performs a set of K realizations. For each such realization, the activity list is scheduled with the schedule generation scheme explained in the previous subsection. Hence, K precedence and resource feasible schedules each having their own makespan and starting times for all activities are obtained. In the last step of the pseudocode, the TSAD minimization model is called for to obtain the robust activity start times. Using the resulting robust starting times, first, feasibility of these starting times with respect to resource requirements and precedence relations is checked and if it is found infeasible, the schedule is fixed with deferring the activities that have infeasibility. The makespan and the TSAD values of the resulting schedule are used as the performance measures of the chromosome. Note that the TSAD in the multi-project scheduling approach includes the deviations of the starting times of the existing activities as well.

4.2.4.2. Fitness Calculation Procedure2

In the fitness calculation procedure2, again, first, a set of *K* realizations are performed. For each such realization, the activity list is scheduled with the schedule generation scheme explained in the previous subsection. Hence, *K* precedence and resource feasible schedules each having its own makespan and starting times for all activities are obtained. These *K* realizations are then sorted in their non-domination levels using the corresponding makespan and TSAD values and among the schedules that have a rank value of 1, the schedule having the minimum TSAD is selected as the robust schedule of the chromosome and makespan and the TSAD value of this schedule are used as the performance measures of the chromosome. Note that the TSAD in the multi-project scheduling approach includes the deviations of the starting times of the existing activities as well.

4.2.5. Generating the Initial Population

Initial population is comprised of randomly selected precedence feasible activity lists. Precedence feasible activity list is formed using dynamic eligible activity lists and adding one activity to the precedence feasible activity list in each iteration. An eligible activity is an activity whose predecessor activities have already been scheduled in the feasible activity list. It means that it can be selected for the next position of the chromosome. In each iteration, current eligible activity set is created. Afterwards, an activity belonging to this set is selected randomly to be placed into the next position in the precedence feasible activity list. This operation is repeated until all activities are included in the precedence feasible activity list.

4.2.6. Construction of the Next Generation

Compared to single objective GA's, the presence of multiple objectives complicates population management. The construction of the next generation basically starts with the non-dominated sorting of the current population P_t and the calculation of crowding distances for each individual in P_t. Then parent pairs are selected from the current population and the offspring population Q_t is generated. Before the combination of the current population P_t with the offspring population Q_t, schedule generation scheme is applied to all the individuals in the offspring population Q_t and the performance measures (makespan and TSAD) are calculated for each offspring. After the combined population R_t with a size on 2N is obtained, this population is reduced to size of N with the reduction procedure. The steps of the construction of the next generation are given in the following subsections.

4.2.6.1. Non-dominated Sorting Procedure

The aim of non-dominated sorting is to obtain the ranks of the individuals that they belong to. This information is used in the parent selection and reduction process. For the non-dominated sorting, an approach with $O(MN^2)$ time complexity is adopted. In this approach, for each solution two entities are calculated: (1) domination count, the number of solutions which dominate the solution, and (2) a set (S_p) of solutions that the solution dominates. This requires $O(MN^2)$ comparisons. All solutions in the first non-dominated front will have their domination count as zero. Now, for each solution with, each member of its set S_p is visited and its domination count is reduced by one. In doing so, if for any member the domination count becomes zero, it is put in a separate list. These members belong to the second non-dominated front. Now, the above procedure is continued with the remaining members of the population and the third front is identified. This process continues until all fronts are identified. For each solution in the second or higher level of non-domination, the domination count can be at most N-1. Thus, each solution will be visited at most N-1 times before its domination count becomes zero. At this point, the solution is assigned a non-domination level and will never be visited again. Since there are at most N-1such solutions, the total complexity is $O(N^2)$. Thus, the overall complexity of the procedure is $O(MN^2)$.

4.2.6.2. Crowding Distance Calculation

Construction of the next generation starts with the crowding distance calculation. We use the crowding distance metric to compare individuals when selecting the parents and later in the reduction of the population process. To get an estimate of the density of solutions surrounding a particular solution in the population, we calculate the average distance of two points on either side of this point along each of the objectives. This quantity serves as an estimate of the perimeter of the cuboid formed by using the nearest neighbors as the vertices and is called as the crowding distance. In Figure 4.3, the crowding distance of the i^{th} solution in its front (marked with solid circles) is the average side length of the cuboid (shown with a dashed box).

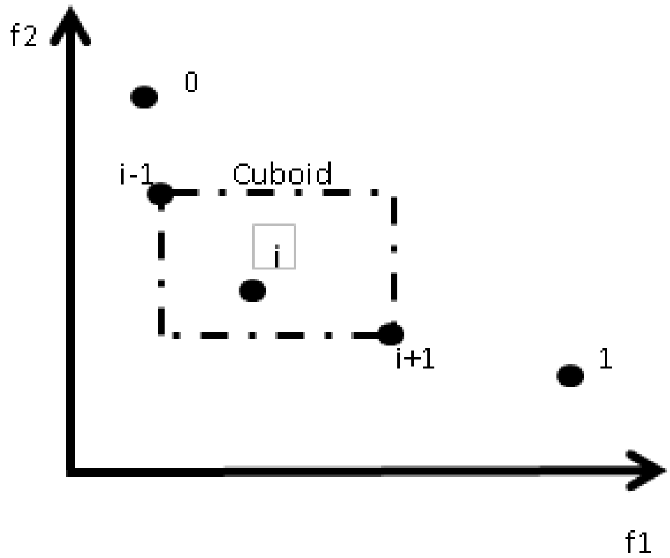

Figure 4. 3. Crowding Distance of the i^{th} Solution (Deb et.al., 2002)

The crowding distance computation requires sorting the population according to each objective function value in ascending order of magnitude. Thereafter, for each objective function, the boundary solutions (solutions with smallest and largest function values) are assigned an infinite distance value. All other intermediate solutions are assigned a distance value equal to the absolute normalized difference in the function values of two adjacent solutions. This calculation is continued with other objective functions. The overall crowding distance value is calculated as the sum of the individual distance values corresponding to each objective. Each objective function is normalized before calculating the crowding distance.

This quantity is an estimate of the density of solutions surrounding a particular solution . A solution with high crowding distance is not surrounded by other solutions in close proximity. Hence, we may want to keep this solution for the next generation, as we aim for dispersed solutions. Figure 4.4 outlines the crowding distance computation procedure of all solutions in a non-dominated set .

$\boldsymbol{f_m^l}$ denotes the $\boldsymbol{m^{th}}$ objective value of solution $\boldsymbol{l}$.
$\boldsymbol{fmax_m}$, $\boldsymbol{fmin_m}$ denote the maximum and minimum $\boldsymbol{m^{th}}$ objective values of all solutions in $\boldsymbol{F}$.
For each solution $\boldsymbol{i} \in \boldsymbol{F}$, initialize $\boldsymbol{cd_i = 0}$.
    for Objective $\boldsymbol{m} \in \boldsymbol{Obj}$ **do**
    Sort the set in increasing order of $\boldsymbol{f_m}$
    $\boldsymbol{cd_1 = cd_{|F|}} = \infty$
    for Solution $\boldsymbol{l = 2 \rightarrow |F| - 1}$
        $\boldsymbol{cd_l = cd_l + \frac{f_m^{l+1} - f_m^{l-1}}{fmax_m - fmin_m}}$
    end for
end for

Figure 4. 4. Pseudocode for Crowding Distance Calculation (Deb et.al., 2002)

4.2.6.3. Selection of Parent Pairs

Construction of the next generation continues with the selection of parent pairs. To obtain these pairs, from the population on hand, using binary tournament selection procedure and crowded-comparison operator, mother population and father population are generated .

Before explaining the parent selection mechanism we need to define crowded-comparison operator. After all the population members in the set are assigned a distance metric (crowding distance) we can compare two solutions for their extent of proximity with other solutions. A solution with a smaller value of this distance measure is in some sense more crowded by other solutions. This is exactly what we compare in the proposed crowded-comparison operator described below. The crowded-comparison operator guides the selection process at the various stages of the algorithm. Assume that every individual in the population has two attributes: non-domination rank and crowding distance. Between two solutions with different non-domination ranks, we prefer the solution with the lower rank. Otherwise, if both solutions belong to the same front, then we prefer the solution that is located in a lesser crowded region.

There are many selection schemes for GAs, each with different characteristics. An ideal selection scheme would be simple to code, and efficient for both non-parallel and parallel architectures. Furthermore, a selection scheme should be able to adjust its selection pressure so as to tune its performance for different domains. Tournament selection is increasingly being used as a GA selection scheme because it satisfies all of the above criteria. Tournament selection can also adjust the selection pressure to adapt to different domains. Tournament selection pressure is increased (decreased) by simply increasing (decreasing) the tournament size. Note that a tournament size of one would be equivalent to selecting individuals randomly from the population and a tournament size equal to the population size would be equivalent to selecting the best individual at any given point. If the tournament size is larger, weak individuals have a smaller chance to be selected. All of these factors have contributed to the increased usage of tournament selection as a selection mechanism for GAs. Here, in our solution approach, we have adopted binary tournament selection mechanism for the selection of parents and for the selection of individuals that will be transferred to the next generation along with the crowding distance operator. In binary tournament selection, two individuals are randomly selected from the population as a candidate to become a mother and compared using the crowded-comparison operator. The one with the larger crowding distance metric wins the tournament and is chosen as the mother. Again this procedure is applied for the selection of the father but this time, the selection is repeated, if the father is the same as the mother until they are different from each other for one mother-father pair.

4.2.6.4. Crossover Operator

Let us assume that parent pairs are selected for crossover, i.e, we have a mother individual *M* and a father individual *F*. Now two child individuals have to be constructed, a daughter *d* and a son *s*. We make use of one-point crossover operator in which one point is randomly selected. To create the daughter, first the genes are chosen from the mother until the randomly created point and the rest are chosen from the father. To assure precedence feasibility, while choosing the genes from the father, we start checking, if a gene is already chosen starting from the beginning of the precedence feasible activity list of the father. The son is formed with the same logic but this time the genes are first chosen from the father. Note that since the first and second parents are precedence feasible, the resulting off-spring's chromosome is also precedence feasible. This crossover operator is applied for all mother-father pair and child population is obtained.

4.2.6.5. Mutation Operator

Modification of the newly produced chromosomes plays an important part in increasing a population's diversity. The following mutation operator is applied to each newly produced child individual. The mutation operator modifies the genes of the genotype with a probability of *MutationProbability*. First, we show how the mutation operator modifies the individual's activity list. We move through the activity list from left to right. Thereby, we apply a right shift to each activity with a probability of *MutationProbability*. This mutation operator is called as the swap mutation operator. Consider a current position $i \epsilon \{1, . . . , J-1\}$ in the activity list $\Lambda=(j_1, j_2,.,j_i,.j_n)$. Now activity j_i can be shifted after some position $h \epsilon \{i+1, . . . ,J\}$, which leads to activity list $\Lambda'=(j_1,.j_{i-1},j_{i+1},.j_h, j_i,j_{h+}1,.,j_n)$. That is, activity j_i is right shifted within the activity list and inserted immediately after some activity j_h . Clearly, such a shift is executed only if the resulting activity list is still precedence feasible.

4.2.6.6. Reducing the Population Size

After the usual binary tournament selection, recombination, and mutation operators are used to create an offspring population (Q_t) of size N, and the solution decoding procedure is applied to the offspring individuals to obtain the performance measures, a combined population $R_t=P_tUQ_t$ is formed. The population R_t is of size 2N. Then, the population is sorted according to non-domination and crowding distance calculations are done for the combined population Since all previous and current population members are included in elitism is ensured. Now, solutions belonging to the best non-dominated set F_1 are of best solutions in the combined population and must be emphasized more than any other solution in the combined population. If the size of F_1 is smaller then , we definitely choose all members of the set for the new population P_{t+1} The remaining members of the population P_{t+1} are chosen from subsequent non-dominated fronts in the order of their ranking. Thus, solutions from the set F_2 are chosen next, followed by solutions from the set F_3, and so on. the set F_l be the first non-dominated set such that the count of solutions in all sets from F_1 to F_l would be larger than the population size N. To choose exactly N population members, we sort the solutions of the last front F_l using the crowded-comparison operator in descending order and choose the best solutions needed to fill all population slots. This reduced population is the next population.

In Phase II of the proposed three-phase approach for robust project scheduling we presented two scheduling approaches with two different fitness calculation procedures. Each scheduling approach provides a set of robust non-dominated baseline schedules for the scheduled activities to the decision maker. The decision maker can choose one of these non-dominated robust baseline schedules considering the dynamics of the current project management environment to be used as the main baseline plan for the activities. This baseline plan is used as the reference point in the implementation and monitoring phase of the projects and can be revised if needed.

4.3. PHASE III: REACTIVE PROJECT SCHEDULING PHASE

After the baseline schedule has been determined, project execution can start. However, no matter how much we try to protect the baseline schedule against possible disruptions, we can never totally eliminate their occurrence since proactive scheduling only employs statistical knowledge of disruptions, the process of obtaining this knowledge is subject to estimation errors and anticipating all possible disruptions would simply be impossible. A proactive scheduling procedure must therefore be combined with a reactive scheduling procedure, which allows during schedule execution to react to schedule disturbances that cannot be absorbed by the proactive schedule.

In this section, we present Phase III, reactive project scheduling phase of the proposed three-phase approach for robust project scheduling. In the following subsections, we first present the suggested rescheduling approach for the revision of schedules when a disruption occurs and then we present the features of the two repaired schedules obtained with the suggested rescheduling approach.

4.3.1. Scheduled Order Repair Heuristic

Phase III, the final phase, using the scheduled order repair heuristic developed by Lambrechts et.al (2008) aims at rescheduling the activities of projects to revise the project schedules when a disruption occurs. The scheduled order repair heuristic is a list scheduling heuristic that reschedules the activities in the order dictated by the baseline schedule (using the lowest activity number as a tie-breaker), while taking into account the observed disruption. When a disruption occurs in time period t, first the project networks and the components of the projects is updated taking into account the changes caused by that disruption, then a priority list L is created including the activities that are not yet completed at t, listed in the scheduling order of the baseline plan. This priority list is then decoded into a feasible schedule using the schedule generation scheme that takes into account the updated parameters of the project network for the times after t. This fixing procedure gives two repaired schedules: repaired schedule1 and repaired schedule2. The details of these repaired schedules are presented in the next subsection.

4.3.2. Repaired Schedule1 and Repaired Schedule2

When a disruption occurs, our scheduled order repair heuristic proposes two alternative repaired schedules enabling the decision maker compare and select one of them. In the first alternative, the earliest starting time of all activities are set to their robust baseline starting times generated in the proactive project scheduling phase. By doing so, the schedule keeps the buffers to be used in a future disruption in the case that the baseline schedule is able to absorb the current disruption. In the second alternative repaired schedule, the time buffers contained the baseline schedule generated in the proactive project scheduling phase are used to absorb the completion time delays of the activities. To do so, the earliest starting times of all the activities are set to the time at which the disruption occurs. Using these earliest starting times, scheduled order repair heuristic first tries the current time (t) for the ongoing activities. If this is infeasible, the procedure tries the next time period (t +1) and subsequent time periods if necessary. For the activities that did not yet start, it only considers the earliest precedence feasible starting time.

The superiority of the two repaired schedules changes from case to case. Although keeping the time buffers seems to be more reasonable when fixing the schedule, repaired schedule2 may give better fixed schedules in some situations. We will consider two cases to illustrate this situation. Consider a scenario such that there are three active activities in the system and three require the same resource. Assume that activity 1 requires 50 hours, activity 2 requires 30 hours and activity 3 requires 40 hours from resource 1. Assume that with the daily availability of 10 hours for resource 1, the robust baseline starting times for activity 1, activity 2 and activity 3 are 0, 7 and 12, respectively yielding a makespan of 16. The Gantt chart for these activities are shown in Figure 4.5.

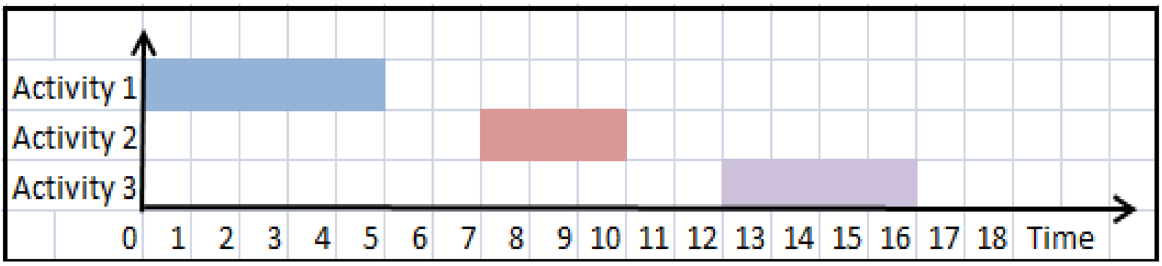

Figure 4. 5. Gantt Chart of an Example Robust Baseline Schedule

Now consider a case that at the beginning of time 2, a disruption occurs and resource 1 says that s/he will be unavailable during time 4 and time 5. Repaired schedule1 and repaired schedule2 for Case 1 is shown in Figure 4.6 and 4.7.

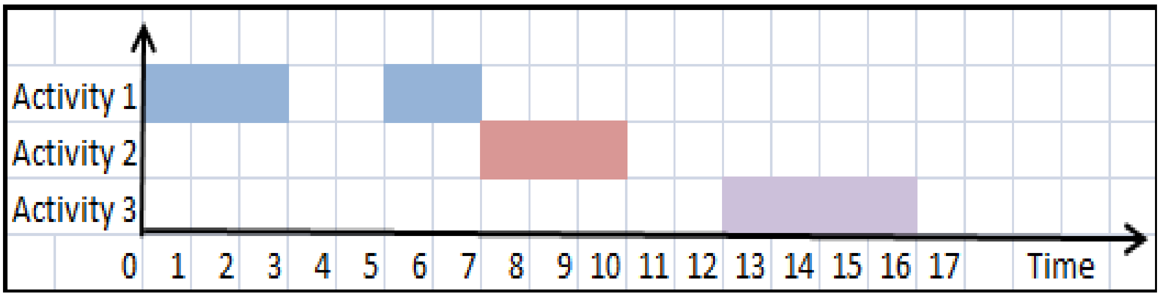

Figure 4. 6. Gantt Chart of an Example Repaired Schedule1: Case 1

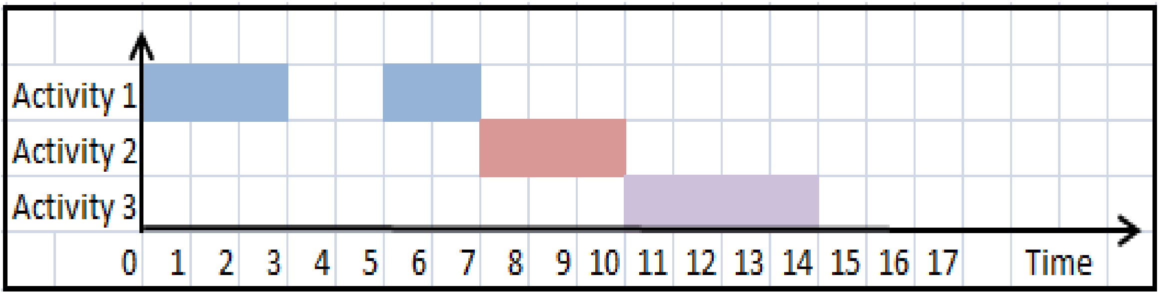

Figure 4. 7. Gantt Chart of an Example Repaired Schedule2: Case 1

It is seen from the Figures 4.6 and 4.7 that although this disruption does not cause an increase in the completion time of these activities, the robust starting time of activity 3 changes in repaired schedule2, using the buffer in front of activity 3 and presents a fixed schedule with a smaller makespan value. In such a scenario, a fixed schedule that has the same makespan value with the makespan value in the baseline plan might be more preferable since it changes the schedules of activities only if it is inevitable.

Now consider Case 2. In this case, resource 1 says that he/she will be unavailable during time 4 and during times 9, 10 and 11 instead of during time 4 and time 5. Repaired schedule1 and repaired schedule2 are shown in Figure 4.8 and Figure 4.9 for this baseline plan.

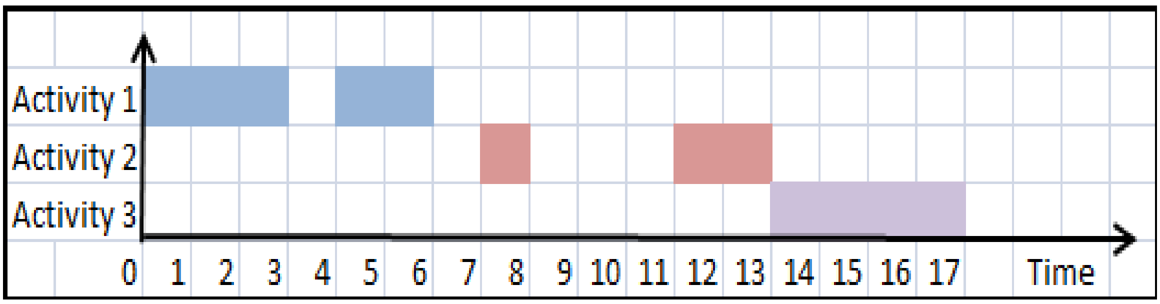

Figure 4. 8. Gantt Chart of an Example Repaired Schedule1: Case 2

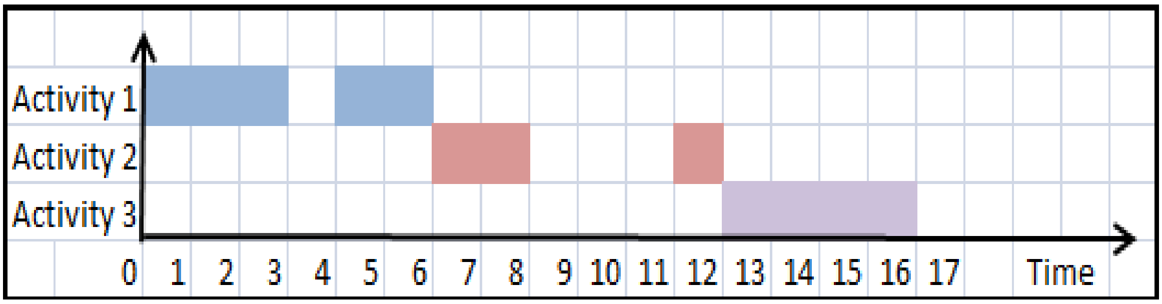

Figure 4. 9. Gantt Chart of an Example Repaired Schedule2: Case 2

Now the completion time for these activities is 17 in repaired schedule1 and 16 in repaired schedule2. In such a scenario, it seems that the repaired schedule2 gives a better fixed schedule. Note that these two repaired schedules are the schedules that have the two extreme makespan values. Between these two extreme schedules, there are many fixed schedules with different activity starting times and some of them may be better in terms of both makespan and the closeness to the baseline schedule. Although at additional computational expense, complete set of fixed schedules can be obtained, we present here only the two extreme fixed schedules in our implementation routine.

With these two alternative repaired schedules, the decision maker will be able to observe the possible effects of the disruptions on the work schedules of the resources and the starting and ending time of the activities for each reaction strategy. We advise to react whenever the current project schedule becomes infeasible. Essentially, proactive scheduling aims to reduce schedule nervousness by limiting the need for rescheduling decisions. Furthermore, the developed procedure for reactive scheduling enables the project managers to make “what-if analysis” and thus to generate a set of contingency plans for better preparation.

In this Chapter, a detailed explanation of the three-phase approach for robust project scheduling is presented. First, Phase I of the three-phase approach providing a model for estimating the deviation levels of projects and their activities is presented. How this model is used to obtain robust project schedules is explained in Phase II with the two alternative robust project scheduling approaches. Finally, a schedule repair heuristic is presented to fix the baseline plans in case of a disruption.

CHAPTER 5

DEVIATION ANALYSIS WITH REAL DATA

In the implementation of phase I of the proposed three-phase approach for robust R&D project scheduling, real R&D project data of a leading home appliances company in Turkey is used. In the following, the only resource considered is the various types of human resource. This is due to the relatively high importance of human resource as well as the relatively unrestricted availability of other resources such as laboratory and equipment in the problem that is dealt with. In the implementation, in order to consider the human resource usage deviations of the projects as a risk measure, in the proposed model, the projects are classified into four groups making use of the feature selection, clustering and classification analysis. This chapter first introduces the data used in the implementation of this phase then gives the implementation steps of Phase I on real data with the findings and results.

5.1. DATA

To obtain the real data used in the implementation phase, first a project set that is used in all the implementation phases of the proposed three phase approach is determined. Then the relevant input features that might have a positive or negative effect on the deviation levels of projects are determined and the values of these features for each project is obtained. After that, feature subset selection is applied to the data as a preprocessing step to determine the most relevant and most important factors on the deviation levels of projects and thus to construct a better classification model in the step I of phase I. Then the activities of the projects in the project set are classified into groups to develop a better activity deviation level prediction procedure for the activities of a newly arrived project. Data section ends with the presentation of the activity data analysis results.

5.1.1. Determining the Project Set

To determine the set of completed R&D projects used in the implementation of the proposed three phase approach, a project suggested by the R&D project manager of the firm is considered. The reason why this project is suggested is that the resources worked on that project are thought to be critical for the R&D Department. The most critical six resources with respect to average workloads that worked on the suggested project are listed and the projects that these resources are worked during the execution of the suggested procedure are filtered from the project database of the firm. A total of 117 projects is obtained after this filtering. To decrease the project set three-month time range starting with the starting time of the suggested project and the three-month time range ending with the ending time of the suggested project are cut from the considered time range and a total of 33 project are removed from the project set. From this 84 remaining projects, the projects started before 2007 and the support projects are removed since the project plans was not detailed for the projects started before 2007 and the support projects cannot be counted as projects because they are just small work packages. Consequently, a project set comprised of 43 projects which are interrelated with respect to the required resources is obtained. This project set is adopted to be used the implementation of proactive project scheduling phase. Note that in the implementation of all the phases of the proposed three phase approach, we used this project set.

5.1.2. Determining the Relevant Features that Might Affect Deviation Level of Projects

After several interviews with the project managers of the firm, the factors that might affect project risk levels and cause time overruns are determined and the values that these features takes for each project is obtained. The input features determined after these interviews with the data types and ranges are presented in Table 5.1.

Table 5. 1. Input Features Used in Feature Subset Selection

Feature ID	Feature Name	Type	Min. Value	Max. Value
FA1	Existence of the technology family “Liquid Dynamics”	(binary)	0	1
FA2	Existence of the technology family “Material Science”	(binary)	0	1
FA3	Existence of the technology family “Thermodynamics”	(binary)	0	1
FA3	Existence of the technology family “Cleaning”	(binary)	0	1
FA5	Existence of the technology family “Vibration and Acoustics”	(binary)	0	1
FA6	Existence of the technology family “Structural Design”	(binary)	0	1
FA7	Existence of the technology family “Power Electronics”	(binary)	0	1
FA8	Existence of the technology family “Electronic Assessment”	(binary)	0	1
FA9	Number of collaborative internal plants	(integer)	0	5
F1	Number of Technology families involved in the project	(integer)	2	9
F2	Required size of project team in numbers	(integer)	5	27
F3	Number of required equipment and machine type	(integer)	0	5
F4	Number of collaborations	(integer)	0	3
F5	First Usage of infrastructure	(binary)	0	1
F6	Existence of similar projects worked on before	(binary)	0	1
F7	Planned man-months needed	(double)	6.1	88.69
F8	Planned equipment-months needed	(double)	0	119,97
F9	Expected cost of the project	(integer)	32064	506825
F10	Technology maturity of the Project	(integer)	1	25
F11	Position of the project in the r&D-R&d spectrum	(integer)	1	3

In the analysis, two types of output features are considered, i.e., Numeric Output and Nominal Output. Note that for the numerical output case various well known classification algorithms such as J48 Decision Tree or *Naive Baye*s were not applicable and limited to only regression like algorithms The numeric output is basically the percentage human resource usage deviations. On the other hand the nominal output is determined by the application of a simple K-Means clustering algorithm developed by

Mac Queen [10] to the numeric output. Based on the resulting clusters, four deviation levels (Negative High Deviation-NHD, Negative Low Deviation-NLD, Positive Low Deviation-PLD and Positive High Deviation-PHD) are determined and each project is labeled accordingly. As a result a data set with 20 input features and two output features is obtained.

5.1.3. Data Preprocessing: Feature Subset Selection

Not only missing some of the significant input features but also the existence of abundant number of irrelevant features makes it difficult (if not impossible) to establish the relation between the inputs and the output. Therefore, feature subset selection is an essential step in data mining process and directly influences the classification performance.

In the analysis reported here, 20 input features and the numeric output, i.e., the percentage human resource deviation of the projects is utilized. Various different filtering and wrapper algorithms with n-fold cross validation is utilized. Note that different folds (i.e., different training and test combinations) yield different subsets of significant inputs hence a threshold value of 70% is set in order to make a final decision for inclusion of a feature for the further analysis in the case of wrappers. On the other hand, for the filtering techniques 0.007 ±0.004 are assumed as threshold values for the merits in the final decision. The results of the feature selection analysis are given in Table 5.2.

Table 5. 2. Results of Feature Subset Selection Analysis

Results for 11 Input Features		Results for 20 Input Features	
Used Output			
% Human Resource Usage Deviation	% Human Absolute Resource Usage Deviation	% Human Resource Usage Deviation	% Human Absolute Resource Usage Deviation
F1,F4,F5,F6,F10	F1,F2,F4,F5,F6	FA1,FA6,F4,F5,F6	FA1,FA4,FA8,F5,F6

As a result of the analysis, four different feature subsets are determined as significant; namely, {F1, F4, F5, F6, F10}, {F1, F2, F4, F5, F6}, {FA1, FA6, F4, F5, F6} and {FA1, FA4, FA8, F5, F6}. In order to evaluate the influence of the feature subset selection stage to the classification performance two extra feature sets are also included in the further analysis, i.e., one set with all of the features proposed by the managers, and the second set, which consists of 11 features { F1, …, F11}.

5.1.4. Data Preprocessing: Activity Classification

As stated in the previous chapter the aim of Step II of Phase I is to obtain percentage human resource deviation distributions to be used in Phase II . Since we are dealing with R&D projects and the activities of R&D projects are unique and the work content is characteristic among all the activities, in order to obtain sufficiently large amount of data for a valid percentage human resource activity deviation distribution we have grouped the activities of projects in the project set in six activity classes. The classification of the activities was based on the work contents and the density of required resource types of the activities. The list of activity classes are as follows:

- Class1: Meeting and Reporting Activity Class
- Class 2: Design Modeling and Visualizing Activity Class
- Class 3: Test, Measurement and Analysis Activity Class
- Class 4: Prototyping/Production Activity Class
- Class 5: Literature and Patent Search Activity Class
- Class 6: Other Activity Class

5.1.5. Activity Data Analysis

Before starting the implementation we have analyzed the activity data to be more familiar with the data we are dealing with. In this part, we will give some statistics concerning the activities in our project set. Figure 5.1 shows the number of projects of each project deviation class and Figure 5.2 shows the number of activities in each project deviation class.

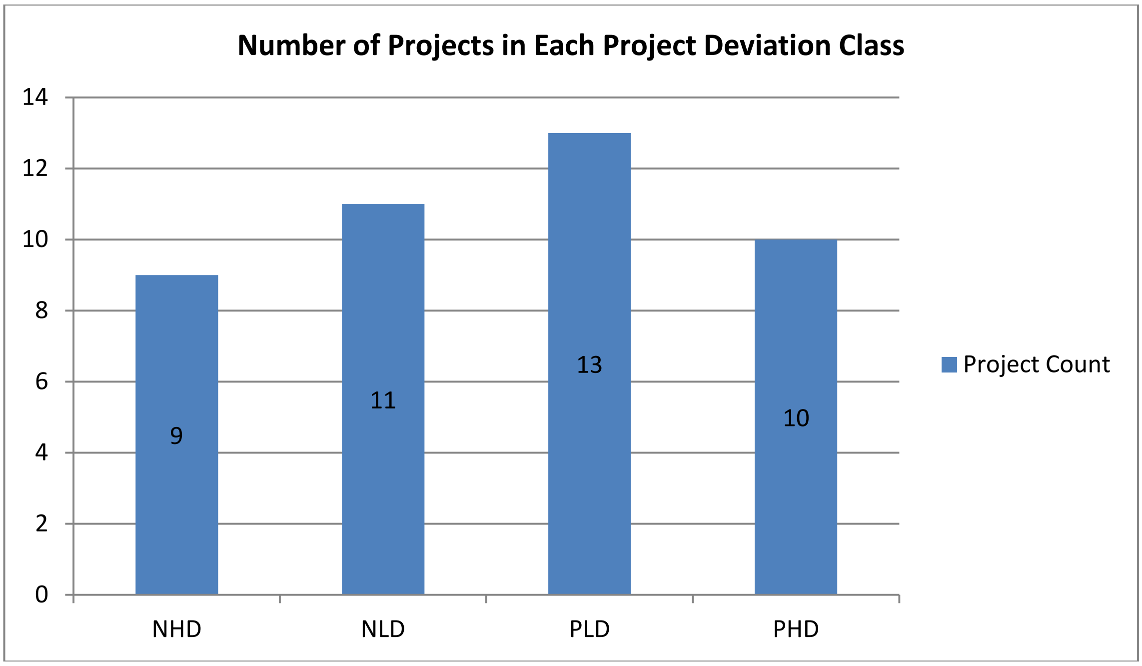

Figure 5. 1. Number of Projects in Each Project Deviation Class

It is seen from Figure 5.1 that the numbers of projects in each type of project deviation class are very similar with each other. It shows that we almost have a homogeneous project set in terms of project deviation classes.

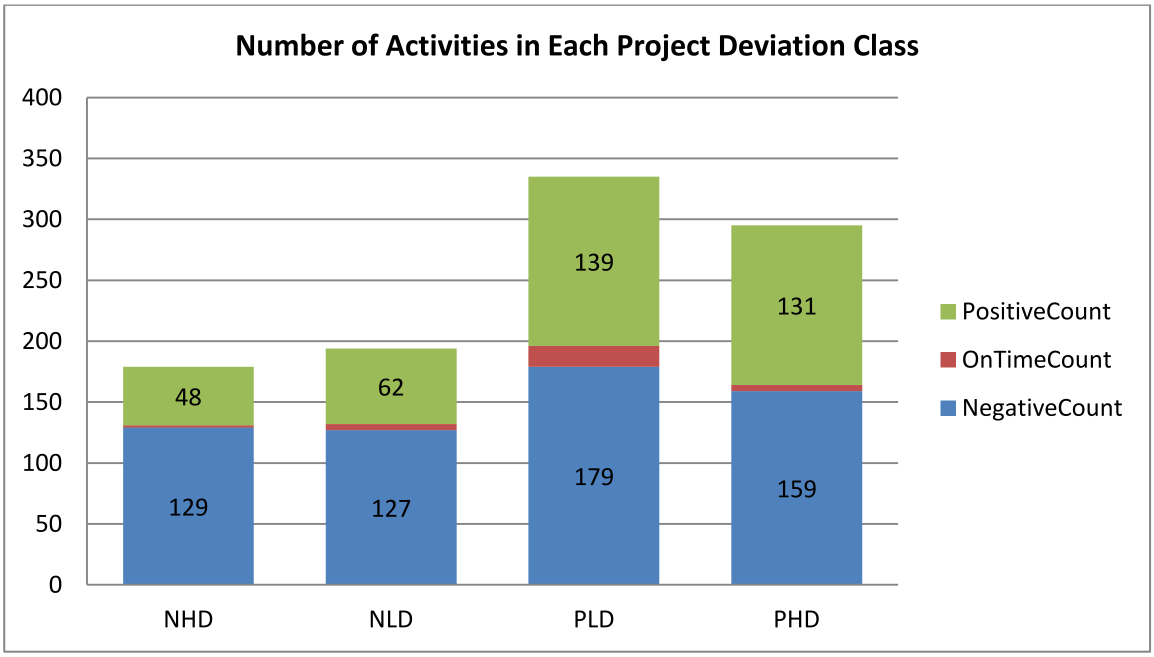

Figure 5. 2. Number of Activities in Each Project Deviation Class

Figure 5.2 indicates that the activity counts in each project deviation class are not very different from each other. and while the percentages of the activities having negative deviations is noticeably higher than the percentages of the activities having positive deviations in the project classes of NHD and NLD, this difference is not so big for the project deviation classes of PLD and PHD.

Table 5.3 shows the number of activities in each activity class and the number of activities with positive and negative deviations and the probabilities of them.

Table 5. 3. Activity Statistics

Activity Type	Activity Type Name	Activity Count	Negative Count	Positive Count	Negative Probability	Positive Probability
1	Meeting and Reporting	237	134	103	56.54%	43.46%
2	Test Measurement and Analysis	609	364	245	59.77%	40.23%
3	Literature and Patent Search	27	17	10	62.96%	37.04%
4	Design Modeling and Visualizing	57	33	24	57.89%	42.11%
5	Prototyping/ Production	56	34	22	60.71%	39.29%
6	Other	19	9	10	47.37%	52.63%

Table 5.3 reveals that the activities in all classes except in the “other” activity class have the tendency of having negative deviation. This shows that the project managers are generally overestimating the durations of the activities on which the regular workers of the firm are working and they behave risk-aversely not to be blamed in case of any unforeseen event causes delay.

The same analysis is done for the activities in each project deviation class. Since all the projects that we're dealing with are completed projects, we know the actual project deviation class of them. Table 5.4 tabulates the activity statistics when the activities are also classified according to their project deviation classes.

Table 5. 4. Project Deviation Class Based Activity Statistics

Project Type	Project Count	Activity Count	Negative	On Time	Positive	% Negative	% On Time	% Positive
NHD	9	179	129	2	48	0.72	0.01	0.27
NLD	11	195	127	5	62	0.65	0.03	0.32
PLD	13	338	179	17	139	0.53	0.05	0.41
PHD	10	296	159	5	131	0.54	0.02	0.44

Table 5.4 indicates that the activities in the project class type NHD and NLD have the tendency of having negative deviation as expected. On the contrary, the activities in the project class type PHD and PLD have also the tendency of having negative deviation with a lower probability than the activities in the project class type NHD and NLD. This situation can be explained by the dominance of the activities with negative deviation in the activity set.

Table 5.5 shows the number of the activities on time, with negative deviation and with positive deviation together with their project deviation classes and the probability of being in these project classes.

Table 5. 5. Activities' Project Deviation Class Statistics Based on Their Deviation Type

		% Negative	% On Time	% Positive
On Time Activity Count	**NHD**	21.72%	6.90%	12.63%
29	**NLD**	21.38%	17.24%	16.32%
Negative Deviation Activity Count	**Negative Class**	43.10%	24.14%	28.95%
594	**PLD**	30.13%	58.62%	36.58%
Positive Deviation Activity Count	**PHD**	26.77%	17.24%	34.47%
380	**Positive Class**	56.90%	75.86%	71.05%

Table 5.5 reveals that among the activities having negative deviation, about half of them are in the class of negative deviation projects (NHD and NLD), among the activities on time, 75% of them belongs to the project class of low deviation projects (NLD and PLD) and among the activities having positive deviation 71% of them belongs to the class of positive deviation projects (PLD and PHD).

5.2. STEP I: DEVIATION ANALYSIS OF PROJECTS

Recall that the objective of the first step is establishing a classification model in order to classify the R&D projects on hand with respect to their percentage resource usage deviation from mean. For this purpose, each R&D project in the data set is labeled as NHD (negative high deviation), NLD (negative low deviation), PLD (positive low deviation) and PHD (positive high deviation) based on threshold levels which are determined by consulting the experts and the projects' percentage human resource deviations realized. Next a feature selection process is applied to the data in order to determine the relevant features. The resulting data is used to construct the classification model. Note that in the analysis an open source data mining tool, namely WEKA developed by Hall et. al. (2009) is utilized.

For each one of the six feature sets that was determined as the result of the feature subset selection analysis two different classification analysis were conducted; one with the numerical output and one with the nominal output.

5.2.1. Classification Analysis with Numeric Output

As stated earlier, in the classification analysis with numeric output, only regression based classification algorithms were applied, namely, Linear Regression, Least Median Squared Linear Regression, Pace Regression and M5P Algorithm.

Table 5.6 tabulates the predictive performance of these algorithms based on various metrics, namely, Count of Exact Class Matches (True Count), Accuracy Rate and the Mean Squared Error (MSE), for each of the six input feature sets determined as the result of the Data Preprocessing Stage. Note that, for the numerical output analysis the True Counts are calculated based on the intervals determined as the labels of the numeric output using threshold values. The thresholds that were used to label the projects with four class labels (NHD, NLD, PLD and NLD) are as -0.20, 0.00 and 0.20. That is to say, the projects having percentage human resource deviation level less than or equal to -0.20 were labeled as NHD, the projects having percentage human resource deviation between -0.20 and 0.00 were labeled as NLD, the projects having percentage human resource deviation between 0.00 and 0.20 were labeled as PLD and the rest were labeled as PHD.

In order to calculate the MSE of classification methods, the labels of the projects are converted into numbers. The numbers 1, 2, 3, and 4 are used for the labels “NHD”, “NLD”, “PLD”, and “PHD”, respectively. In this manner, the error is simply the difference between the corresponding number of prediction and corresponding number of actual label.

In addition to the performance metrics, Table 5.6 also presents the features used in the class label assignment procedure of the corresponding classification method for each feature subset used in the analysis.

Table 5. 6. Classification Results for the Numeric Output

Performance	INPUT FEATURES					
	11 Feature	20 Feature	F1,F4,F5 ,F6,F10	F1,F2,F4 , F5,F6	FA1,FA6, F4,F5,F6	FA1,FA4, FA8,F5,F6
Classification Method: Linear Regression						
True Count	21	9	17	16	17	12
Accuracy Rate	48.84%	20.93%	39.53%	37.21%	39.53%	27.91%
MSE	54	94	57	63	64	66
Selected Features	F2,F4,F5 , F10	FA1,FA4,F A6,F2,F4,F 7,F9,F10	F1,F5,F10	F2,F4	FA1,FA6,F 4	FA1,FA4
Classification Method: Least Median Squared LR						
True Count	18	10	18	17	21	24
Accuracy Rate	39.53%	23.26%	39.53%	37.21%	46.51%	55.81%
MSE	34	93	37	47	42	34
Selected Features	ALL	ALL	ALL	ALL	ALL	ALL
Classification Method: Pace Regression						
True Count	17	21	22	16	18	22
Accuracy Rate	39.53%	48.84%	51.16%	37.21%	41.86%	51.16%
MSE	44	37	36	45	42	39
Selected Features	F1,F2,F4 ,F5,F10, F11	FA1,FA4,F A6,F2,F4,F 10	F1,F4,F5, F10	ALL	ALL	ALL
Classification Method: M5P						
True Count	20	24	19	16	17	12
Accuracy Rate	46.51%	55.81%	44.19%	37.21%	39.53%	27.91%
MSE	35	49	36	45	64	55
Selected Features	F2,F4,F7 ,F10,F11	FA6,F2,F4, F10	F1,F4,F5, F10	F2,F4	FA1,FA6,F4	FA1,FA4

Table 5.6 shows that the best true count values, accuracy rates and MSE values are obtained with the Pace Regression classification method. Besides being good, the true count values, accuracy rates and MSE values are more robust among the input feature subsets.

5.2.2. Classification Analysis with Nominal Output

The classification algorithms applied to the data set with nominal output were J48 Decision Tree classification method and Naïve Bayes classification method. Again the same predictive performance metrics are used. The results for the data set with nominal output are presented in Table 5.7.

Performance	INPUT FEATURES					
	11 Feature	**20 Feature**	**F1,F4,F5,F6,F10**	**F1,F2,F4,F5,F6**	**FA1,FA6,F4,F5,F6**	**FA1,FA4,FA8,F5,F6**
Classification Method: J48 DECIDION TREE						
True Count	37	37	28	29	27	22
Accuracy Rate	83.72%	83.72%	65.12%	67.44%	62.79%	48.84%
MSE	12	20	30	20	41	45
Selected Features	F2,F3,F4,F5,F9,F10	FA1,FA3,FA4,FA5,FA6, F3,F4,F5,F6,F8,F11	F1,F4,F5, F10	F1,F2,F4,F5	FA1,F4,F5	FA1,FA4,FA8,F5
Classification Method: NAIVE BAYES						
True Count	26	30	23	23	25	22
Accuracy Rate	60.47%	67.44%	53.49%	53.49%	55.81%	48.84%
MSE	29	22	44	38	33	45

Table 5.7 demonstrates that the best true count values, accuracy rates and MSE values are obtained with J48 Decision Tree classification method. Besides being good, the true count values, accuracy rates and MSE values are more robust among the input feature subsets

.

5.2.3. Further Results

In this part, we have suggested two further ways of producing classification results. Other than selecting a feature subset and a classification method, one other way of prediction of the deviation risk and deviation levels of projects is using all the analysis done so far and producing probabilistic results.

Using the prediction results obtained with each feature subset and classification model combination we can provide probabilistic percentage human resource deviation estimation for each project by simply counting each label assigned to projects and dividing this number to the number of prediction methods. For the numeric percentage human resource deviation and corresponding deviation labels, we have 24 prediction for each project. The probabilistic results for a subset of projects are depicted in Table 5.8 for the percentage human resource deviation of projects and corresponding deviation level class of projects.

Table 5. 7. Probabilistic Classification Results for Numeric Output

	Prediction Count				Probability			
Project ID	**NHD**	**NLD**	**PLD**	**PHD**	**NHD**	**NLD**	**PLD**	**PHD**
10-015	19	5	0	0	79.17%	20.83%	0.00%	0.00%
09-045	16	8	0	0	66.67%	33.33%	0.00%	0.00%
08-054	10	10	4	0	41.67%	41.67%	16.67%	0.00%
09-018	6	5	7	5	25.00%	20.83%	29.17%	20.83%
09-023	9	12	3	0	37.50%	50.00%	12.50%	0.00%
09-036	3	13	8	0	12.50%	54.17%	33.33%	0.00%
11-009	3	8	11	1	12.50%	33.33%	45.83%	4.17%
10-049	6	10	8	0	25.00%	41.67%	33.33%	0.00%
08-040	2	14	5	3	8.33%	58.33%	20.83%	12.50%
08-022	2	12	7	2	8.33%	50.00%	29.17%	8.33%

Table 5.8 is interpreted as follows. For example, for project with project id of 10-015, among the 24 classification combinations 19, 5, 0, and 0 of them give the result of NHD, NLD, PLD and PHD, thus this project is with the probability of 79%, 21%, 0% and 0% has the deviation level of NHD, NLD, PLD, and PHD, respectively. By using probabilistic results we will be decreasing the prediction error arising from the selected classification method. Table 5.9 shows the performances of the analysis for prediction of the classes of projects as NHD, NLD, PLD and PHD using probabilistic results, with the aim of prediction of the deviation level classes of projects.

Table 5. 8. Performances of Probabilistic Results for the Deviation Level Prediction

Output	Applied Classification Algorithms	Negative True Count	Positive True Count	Exact Class Match Count	% Negative True	% Positive True	% Exact Class Match
Labels obtained from using Threshold Values on the Deviations	Regression Based Classification Algorithms	15	16	22	75.00%	69.57%	51.16%
Labels obtained from using simple K-Means Algorithm	Tree Based and Bayesian Classification Algorithms	19	13	29	95.00%	56.52%	67.44%

Table 5.9 shows that the proposed probabilistic approach performs good for the prediction of deviation level classes of projects but performs much more better for the prediction of negative and positive deviation levels of projects . The table also shows that compared with the output labels obtained from K-Means Clustering algorithm, the classification methods performs better.

Another way of prediction of the deviation risk and deviation levels of projects could be obtaining one-take-out results. In the one-take-out procedure, the classification approach is applied iteratively. In each iteration one data point is disregarded from the analysis and the learning is obtained from the remaining 42 projects. From these 42 projects a classification rule is learned and this classification rule is applied to the project previously taken out from the analysis to predict its deviation risk level class or deviation class.

5.2.4. Comparisons of Classification Approaches

In the previous three parts we have provided the classification accuracy results using each of output and feature subset combinations. To make a better decision on selecting the classification approach, only looking at the accuracy results and selecting the approach giving the best accuracy might not be reliable enough. In this part, we will suggest further ways of comparing classification approaches explained in the previous parts.

One way of comparing the classification approaches other than comparing accuracy performances is using average variability of each classification approach among the other approaches. This variability attribute is specific for each feature subset and classification method combination and can be calculated using the label numbers associated with the projects and in the same manner that was adopted while calculating MSE for the nominal analysis. The variability of a project for a feature subset and classification method is simply the sum of the squared difference between the corresponding label number of the result obtained from the combination in question and corresponding number labels of the results obtained from the other feature subsets and classification methods. The average variability is obtained summing these variability values of the projects among 43 projects and simply taking the average. Since the number of combinations for each output type is different (due to number of algorithms used in the analysis for the corresponding output type) in order to make the comparisons consistent we have divided the average variability values to the number of combinations. In this way, we were able to compare the feature subset and classification method combinations. The average variability values of the feature subset and classification method combinations for the prediction of percentage human resource deviation levels of projects as NHD, NLD, PLD and PHD are demonstrated in Table 5.10.

Table 5. 9. Average Variability Results of the Classification Approaches

	Labels Obtained by Using Threshold Values on the Percentage Human Resource Usage Deviation		Labels Obtained by Applying Simple K-Means Algorithm to the Percentage Human Resource Usage Deviation	
Feature Subset	**Classification Method**	**Average Variability**	**Classification Method**	**Average Variability**
11 Feature	Linear Regression	0.94	J48 Decision Tree Method	0.64
	Least Median Squared LR	0.94		
	Pace Regression	0.80	Naive Bayes Method	0.86
	MP5	0.73		
20 Feature	Linear Regression	1.76	J48 Decision Tree Method	0.71
	Least Median Squared LR	1.89		
	Pace Regression	0.73	Naive Bayes Method	0.86
	MP5	1.00		
F1,F4,F5, F6,F10	Linear Regression	1.14	J48 Decision Tree Method	0.53
	Least Median Squared LR	0.78		
	Pace Regression	1.02	Naive Bayes Method	0.65
	MP5	0.98		
F1, F2, F4,F5, F6	Linear Regression	1.01	J48 Decision Tree Method	0.66
	Least Median Squared LR	0.78		
	Pace Regression	0.90	Naive Bayes Method	0.59
	MP5	1.01		
FA1, FA6 F4,F5,F6	Linear Regression	1.80	J48 Decision Tree Method	1.02
	Least Median Squared LR	0.99		
	Pace Regression	0.97	Naive Bayes Method	0.52
	MP5	1.80		
FA1, FA4,FA8 ,F5,F6	Linear Regression	1.26	J48 Decision Tree Method	0.69
	Least Median Squared LR	0.76		
	Pace Regression	0.88	Naive Bayes Method	0.70
	MP5	1.48		

Table 5.10 reveals that among the classification approaches the feature subset of F1,F4,F5,F6,F10 and the classification method of J48 Decision Tree Method and the feature subset of FA1,FA6,F4,F5,F6 and the Naive Bayes classification method combinations give the lowest average variability results. In parallel with the accuracy results, using the labels obtained by applying Simple K-Means clustering algorithm to the percentage human resource deviations of projects yields better results than the results using the percentage human resource deviation of projects.

Another consideration we need to take into account while comparing classification approaches is the interpretability of the results. Since the Naive Bayes classification method is a black box only giving the classes of the given projects, it is hard to convince the decision-maker about the reliability of the method. Decision tree based algorithms are better for interpretability, since they also give a tree as a rule of classification to the decision maker for the classification of the newly added data point (new project in our case). When selecting a classification approach the other consideration is the number of features used in the classification and the ease of obtaining them.

5.3. STEP II: ACTIVITY DEVIATION ASSIGNMENT PROCEDURE

In Step I, we have developed a model to predict the percentage human resource deviation level of a newly arrived project based on its various input features. Using this information, in Step II, and the activity class information obtained in the data preprocessing stage, we also developed a model to predict the percentage human resource deviation of the activities of this newly arrived project. In this section first we presented activity deviation assignment procedure for a project whose deviation class is deterministically predicted, then for a project whose project deviation class is probabilistically predicted.

5.3.1. Activity Deviation Assignment with Deterministic Project Deviation Class Prediction

Using the model developed in Step I, for a newly arrived project we predict its percentage human resource deviation class and for each activity class in the corresponding project; using the percentage human resource deviations of already completed activities in the associated activity class belonging the predicted project deviation class we form the human resource deviation distribution of that project deviation class - activity class combinations. As an example, Table 5.11 shows the frequency information used to obtain the deviation distribution for NHD Project Class -

Test, Measurement and Analysis Activity Class combination and Figure 5.3 depicts the corresponding deviation distribution.

Table 5. 10. Frequency and Probability Information for the NHD-Test, Measurement and Analysis Class Combination

Activity Class	Count of Activities in NHD Project Deviation Class	% Deviation Range	Count of Activities	Probability of Being in the Range
2. Test Measurement and Analysis	102	(-1)-(-0,67)	23	22.55%
		(-0,67)-(-0,33)	30	29.41%
		(-0,33)-(0)	23	22.55%
		0-0,33	14	13.73%
		0,33-0,67	6	5.88%
		0,67-1	1	0.98%
		1-1,33	3	2.94%
		1,33-1,66	0	0.00%
		1,66-2	2	1.96%
		SUM	102	

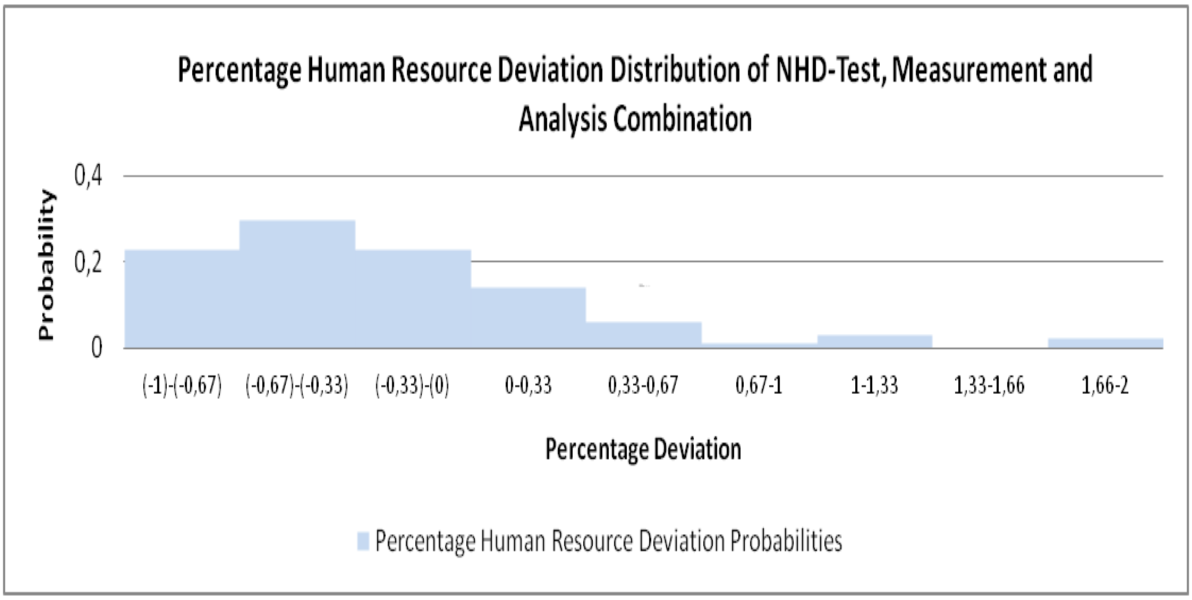

Figure 5. 3. Distribution the NHD - Test, Measurement and Analysis Combination

The percentage human resource usage deviation distributions of the activities belonging to each activity class - project deviation class combinations are obtained following the same procedure and percentage human resource deviations are assigned all the activities belonging to the existing project set in order to compare the actual percentage human resource deviations with the percentage deviations assigned using the procedure we have just suggested.

5.3.2. Activity Deviation Assignment with Probabilistic Project Deviation Class Prediction

The same model can be modified to use the probabilistic results obtained in the first step of Phase I. In this way we would not ignore the possibility of the newly arrived project's belonging to another project deviation class from predicted. To do so, for each activity class, frequency of project classes information is obtained. Table 5.12 tabulates the frequency information for the activity class "Meeting and Reporting" and Figure 5.4 depicts the corresponding frequency chart.

Table 5. 11. Frequency Information for the Activity Class "Meeting and Reporting"

Activity Class	Activity Count	Deviation Range (%)	NHD	NLD	PLD	PHD
1. Meeting and Reporting	187	(-1)-(-0,67)	13	16	7	18
		(-0,67)-(-0,33)	5	5	6	4
		(-0,33)-(0)	10	9	8	6
		0-0,33	6	11	6	7
		0,33-0,67	1	4	4	2
		0,67-1	0	2	6	5
		1-1,33	1	0	2	1
		1,33-1,66	0	1	2	1
		1,66-2	2	2	6	5
		SUM	38	50	47	49

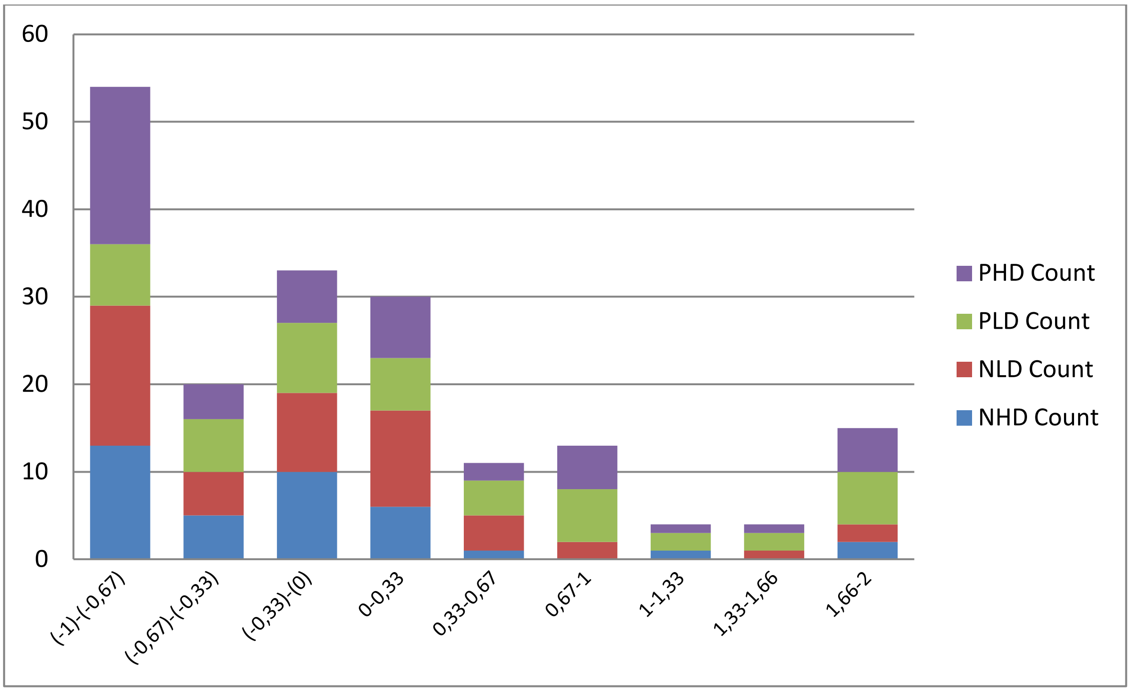

Figure 5. 4. Frequency Distribution for Activity Class "Meeting and Reporting"

In this case, since the project's percentage human resource deviation prediction is probabilistic, i.e. the project is predicted as the member of each of the four deviation level class with a probability, we cannot directly use either the frequency distribution for the activity class or frequency distribution for the activity class-project deviation class combination. We need to adjust the frequency distribution for the activity classes using the project's human resource deviation classes. To obtain a distribution, for each range we take the summation of the multiplications of the probability of belonging to a project deviation class with the number of activities in corresponding activity class-project deviation class combination in that range.

Figure 5.5 shows the human resource usage deviation distributions for the activities belonging to "Test-Measurement" activity class- 50% PHD, 50% PLD project deviation class and Figure 5. 6 shows the corresponding probability distribution for "Test-Measurement" activity class- 50% PHD, 50% PLD project deviation class.

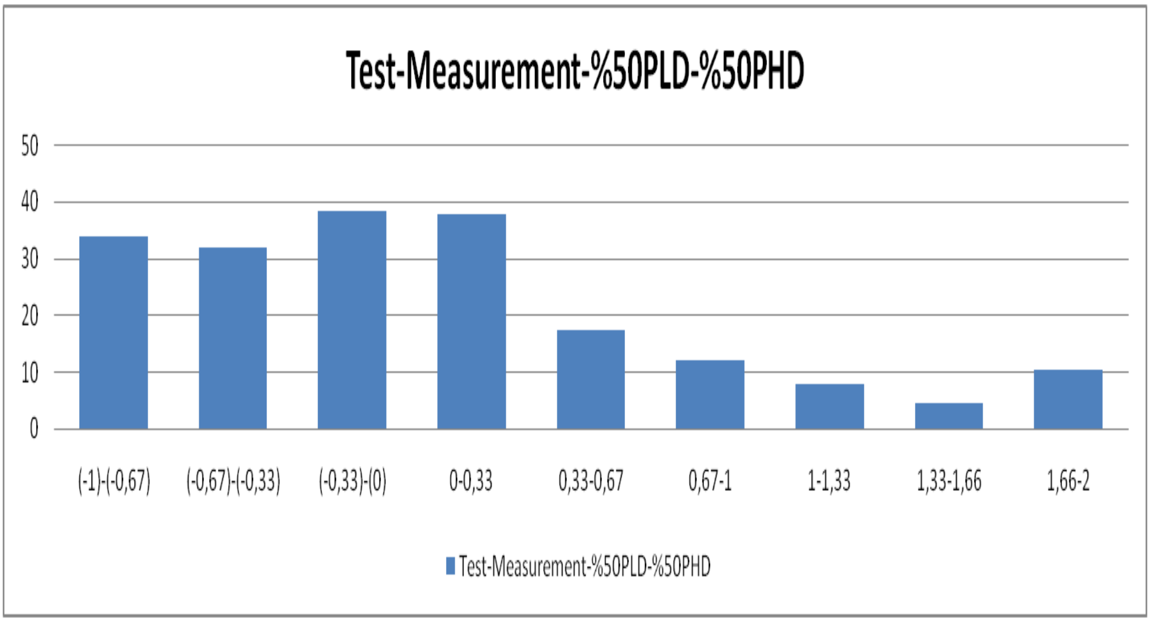

Figure 5. 5. Frequency Distribution for "Test-Measurement" activity class- 50% PHD, 50% PLD Project Deviation Class Combination

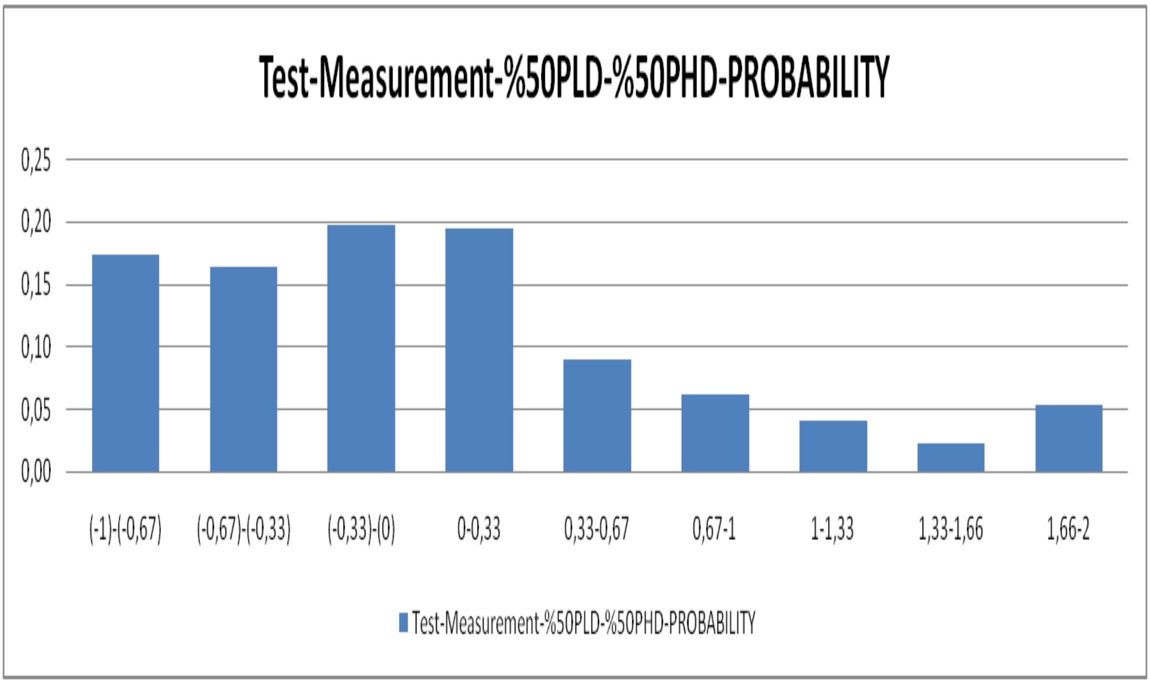

Figure 5. 6. Probability Distribution for "Test-Measurement" Activity Class- 50% PHD, 50% PLD Project Deviation Class Combination

5.3.3. Activity Deviation Assignment Performance Analysis

To test our activity deviation assignment procedure we have used the actual project deviation class information and actual activity deviations. The performance is measured in terms of match number of activities having negative deviation, and match number of activities having positive deviation and to reduce the randomness effect, the activity deviation assignment procedure is called to obtain average performances. Table 5.13 shows the results of the activity deviation assignment procedure when project deviation labels are exactly given; Table 5.14 shows the results of activity deviation assignment procedure with deterministic project deviation level prediction and Table 5.15 shows the results of activity deviation assignment procedure with probabilistic project deviation level prediction.

Table 5. 12. Activity Deviation Assignment Results for the Actual Project Deviation Classes

	Assignment No					
	1	2	3	4	5	AVG.
Total Negative Match Count	377	366	373	365	353	366.8
Total Positive Match Count	168	161	183	153	146	162.2
Total Match Count	545	516	556	518	498	526.6
% Negative Match	60.03%	58.28%	59.39%	58.12%	56.21%	58.41%
%Positive Match	44.21%	42.37%	48.16%	40.26%	38.42%	42.68%
%Match	54.07%	51.19%	55.16%	51.39%	49.40%	52.24%
Total Activity Count						1008
Total Activity Count Having Negative Deviation						628
Total Activity Count Having Positive Deviation						380

Table 5. 13. Activity Deviation Assignment Results with Deterministic Project Deviation Class Prediction

	Assignment No					
	1	**2**	**3**	**4**	**5**	**AVG.**
Total Negative Match Count	369	371	375	360	369	368.8
Total Positive Match Count	147	143	160	160	140	150
Total Match Count	516	514	535	520	509	518.8
% Negative Match	58.76%	59.08%	59.71%	57.32%	58.76%	58.73%
%Positive Match	38.68%	37.63%	42.11%	42.11%	36.84%	39.47%
%Match	51.19%	50.99%	53.08%	51.59%	50.50%	51.47%
Total Activity Count					1008	
Total Activity Count Having Negative Deviation					628	
Total Activity Count Having Positive Deviation					380	

Table 5. 14. Activity Deviation Assignment Results with Probabilistic Project Deviation Class Prediction

	Assignment No					
	1	2	3	4	5	AVERAGE
Total Negative Match Count	375	362	360	353	379	365.8
Total Positive Match Count	160	155	156	159	151	156.2
Total Match Count	535	517	516	512	530	522
% Negative Match	59.71%	57.64%	57.32%	56.21%	60.35%	58.25%
%Positive Match	42.11%	40.79%	41.05%	41.84%	39.74%	41.11%
%Match	53.08%	51.29%	51.19%	50.79%	52.58%	51.79%
Total Activity Count					1008	
Total Activity Count Having Negative Deviation					628	
Total Activity Count Having Positive Deviation					380	

Table 5.14 indicates that using the procedure that we suggested with deterministic project deviation level prediction, on the average with the probability of 51 % we are able to make correct predictions on the percentage deviations of activities. Our predictions are much better to predict the negative percentage deviations of activities than the positive percentage activity deviations of activities. Table 5.15 also shows that using the procedure that we suggested with probabilistic project deviation level predictions, on the average with the probability of 52 % we are able to make

correct predictions on the percentage deviations of activities. Again, our predictions are much better to predict the negative percentage deviations of activities than the positive percentage activity deviations of activities. This correct prediction rates cannot be underrated since the correct prediction rates of activity deviation levels even if the deviation level of the projects are exactly known in advance very similar with the results presented above.

To see how the activity deviation assignment procedure will work for two different projects having different deviation prediction profiles, the probabilities of activities' tendency of having negative and positive deviations are illustrated in Table 5.16.

Table 5. 15. The Probabilities of Activities' Tendency of Having Negative and Positive Deviations for Two Different Projects

	Project Type (0.00, 0.10, 0.10, 0.80)		Project Type (0.80, 0.10, 0.10, 0.00)	
Activity Class	**Negative Probability**	**Positive Probability**	**Negative Probability**	**Positive Probability**
1. Meeting and Reporting	56.24%	43.76%	68.58%	31.42%
2. Test Measurement and Analysis	54.35%	45.65%	71.14%	28.86%
3. Literature and Patent Search	68.57%	31.43%	53.33%	46.67%
4. Design Modeling and Visualizing	40.23%	59.77%	54.97%	45.03%
5. Prototyping/Production	56.25%	43.75%	75.00%	25.00%
6. Other	56.00%	44.00%	40.00%	60.00%

Table 5.16 shows that the activities in the "Meeting and Reporting" activity class, the activities in the "Test Measurement and Analysis" class, the activities in the "Literature and Patent Search" class, the activities in the "Prototyping and Production" activity class and the activities in the " other" activity class belonging both most probably in PHD project class and most probably in NHD class tends to have negative deviation. The activities in the "Design Modeling and Visualizing" activity class belonging most probably in PHD project class tends to have positive deviation while the ones belonging most probably to NHD class have the tendency of having negative deviation.

In this chapter, we have implemented the model developed to predict the deviation percentages of activities. Using the feature subset selection, clustering and classification tools, with this model, we made predictions on the percentage human resource usage deviation of activities that we have in our data set. Besides this main output, we have determined the main features that have an effect on the human resource usage deviations of projects, we developed a classification model to classify the newly arrived projects with respect to resource usage deviations and we classified the projects in our project set. The results show that, our activity deviation assignment procedure works well on the test data. Since we have obtained better results with the probabilistic predictions, we decided to use them in the implementation of proactive project scheduling phase.

CHAPTER 6

PROACTIVE PROJECT SCHEDULING WITH REAL DATA

For the implementation of Phase II of the proposed three-phase approach for robust project scheduling, 37 completed R&D projects of the R&D Department are used as test instances to compare the performances of the two proactive project scheduling approaches developed. All codes are written in Microsoft Visual Studio C# and CPLEX 12.5 is used as the MILP solver. All tests are performed on a computer with a 3.20 GHz Intel(R) Core(TM) i7 CPU 960 processor and 8 GB of RAM.

For our study, we have implemented the two approaches, single project scheduling approach and multi-project scheduling approach, that we suggested with the two fitness calculation procedures, fitness calculation procedure1 and fitness calculation procedure2, on the real project data. The activity weight w_i values in the TSAD calculation are taken as 1 for all activities in the implementation. To reflect the dynamic environment of the R&D Department, an implementation routine with a time loop is considered. In this routine, each time period represents a week and this routine works on an active activity list. This activity list is comprised of scheduled activities that are not completed yet. Starting from the beginning of the time range considered, in each time instant the routine first checks if an activity is completed in the previous time instant and if there is any activity completed in the previous time instant, the routine removes these activities from the active activity list. Then, it updates the remaining work for each activity. After this update, the routine checks if a new project is initiated at that time instant. If there is any new project initiated in the system, the implementation routine calls one of the proactive project scheduling approaches with one of the fitness calculation procedures and obtains the robust baseline schedules for the scheduled activities. After the robust baseline schedules are obtained, the routine updates the active activity list by adding the activities of the newly initiated project. If there is more than one project initiated at the same time, the routine selects one of them randomly. It should be noted that, in this implementation routine, it is assumed that no disruption occurs during the execution of the projects. The GA parameters employed in the implementation of the proactive project scheduling approaches are selected after a fine-tuning process.

In section 6.1 we present the fine-tuning procedure to determine the best bi-objective GA parameter combinations and in Section 6.2 we introduce the project data we have used with the explanation of the data preprocessing stage. Then in section 6.3 we present the results of the single project scheduling approach with fitness calculation procedure1 and fitness calculation procedure2 and compare the results of them with respect to some performance measures. In section 6.4 we present the results of the multi-project scheduling approach with two fitness calculation procedures and compare the results of them with respect to some performance measures. Finally, in section 6.5 we compare the results of the suggested two approaches. Note that each of the approaches produces a set of non-dominated schedules for the scheduled activities instead of a single schedule at the end. Thus, the performance comparisons of the schedules obtained with suggested approaches are based on the schedules randomly selected from the non-dominated schedules.

6.1. FINE-TUNING OF THE BI-OBJECTIVE GA PARAMETERS

Since the parameters used in GAs have a direct effect on the performance of the GAs, they cannot be selected randomly. There is a need for a fine-tuning procedure to select the best parameter combinations to be used in the implementation of the proposed solution approaches. The best combination of the parameters to be used in the bi-objective GA are determined through extensive experimentation. For this experiment, four projects representing the whole project set in terms of the number of activities that the projects have, the number of different resources that the projects require and the number of precedence relations in the networks of projects, i.e., in terms of the size of the projects, are selected. Selected projects are project 10-015, project 08-024, project 08-059 and project 09-028, representing the relatively small, medium, large and very large projects, respectively. Each project is solved with the single project scheduling approach with fitness calculation procedure1 using each parameter combination. The values for the GA parameters that we tested and used to construct the parameter combinations are shown in Table 6.1.

Table 6. 1. Parameter Values Tested in Selection of the Best Parameter Combinations

Crossover Rate	0.6	0.75	0.95
Mutation Rate	0.05	0.1	
Population Size	30	50	70
Number of Generations	30	50	70
Number of Schedule Realizations for a Chromosome	100	200	

Using each value of the GA parameters in Table 6.1, a total of 108 parameter combinations is obtained and tested by solving the determined four projects five times to reduce the undesired effect of randomness. Thus, we have run our single project scheduling approach five times for each project-parameter combinations, yielding a total of 2160 runs.

To compare the performances of the parameter combinations Ballestin and Blanco (2011) provide a number of performance measures used in the literature to express the quality of the solutions obtained with multi-objective optimization algorithms with their advantages and disadvantages. In our parameter selection procedure , we have used the following four different performance measures:

- Scaled Extreme Hyperarea Ratio (Scaled EHR)
- Maximum Spread (MS)
- Number of Non-dominated Solutions
- CPU Time

Scaled extreme hyperarea ratio (Scaled EHR) that we use to compare the performances of different GA parameter combinations is based on the hypervolume indicator, defined by Zitzler (1999) and extreme hyperarea ratio (EHR) developed by Kılıç et.al. (2008). The hypervolume indicator of Zitzler (1999) measures the area between the reference point and the solutions of the non-dominated front. In our case, since the objectives of the GA are both minimization type, we have taken the point (0,0) as the reference point. The union of all m-dimensional regular prisms formed by all non-dominated solutions constitute the hypervolume, where *m* represents the number of objectives. For the bi-objective case, two examples are shown in Figures 6.1 (a) and (b) by the cross-hatched areas. The smaller the area is, the better the corresponding parameter combinations performs. As stated in Ballestin and Blanco (2011), this metric became a very popular metric for comparing non-dominated fronts and it receives considerable attention from researchers in the area.

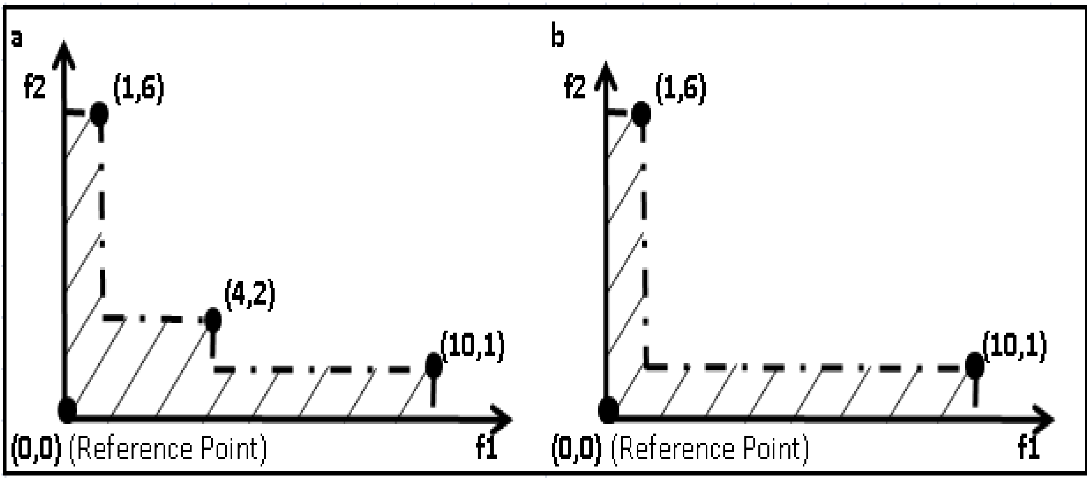

Figure 6. 1. Hypervolume Values in Two Different Non-domination Fronts

It is seen from Figure 6.1 (a) and (b), that although there is no difference in terms of the closeness to the reference point between the non-dominated fronts in Figure 6.1 (a) and (b), the non-dominated front in Figure 6.1 (b) seems superior to the non-dominated front in Figure 6.2 (a), if we use the usual hypervolume as a performance measure. But note that the benefit of the additional non-dominated solution to the decision making environment in the case in Figure 6.2 (a) is ignored. On the other hand, for the bi-objective case, of Kılıç et.al. (2008) define what they call as the extreme hyperarea ratio (EHR) as the ratio of the hyperarea (hypervolume indicator) of the front (area H in Figure 6.2 (a)) to the area bounded by the reference point and the maximum values of the two objective functions (area A in Figure 6.2 (b)). Hence, the smaller EHR, the better. However, this metric can yield wrong comparisons as a result of the same deficiency mentioned above for the hypervolume performance metric.

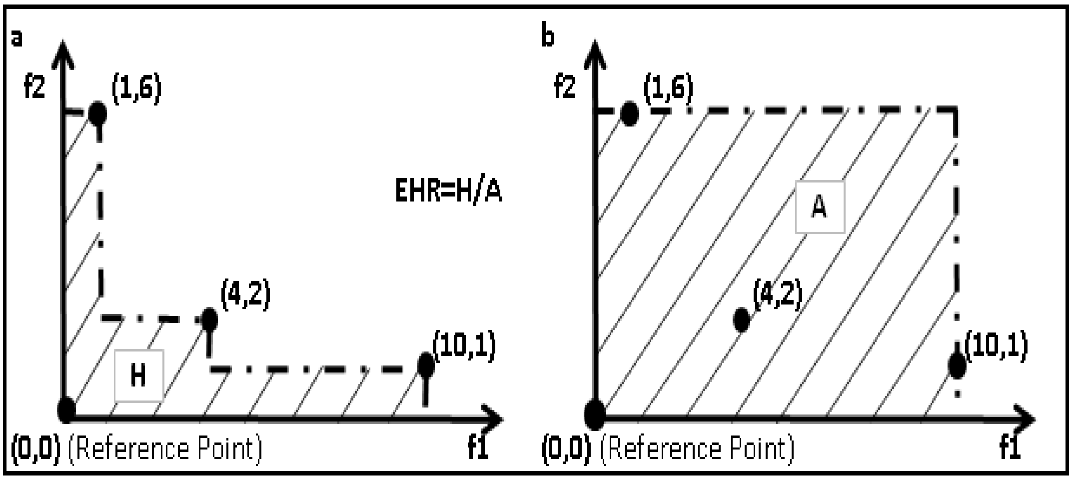

Figure 6. 2. Illustration of the Performance Metric Denoted as HER

To overcome this kind of deficiency, we adjusted EHR by dividing it to the number of non-dominated solutions in the non-dominated front and called this metric as the scaled EHR. By doing so, the non-dominated fronts with a better spread are given priority over the non-dominated fronts with less spread even if the solutions in the non-dominated fronts have pairwise non-domination.

The maximum spread metric (MS), developed by Zitzler (1999) how much the non-dominated solutions in the non-dominated front spread and calculated as in equation 6.1.

$$MS = \sqrt{\sum_{i=1}^{m} \max_{(z_0,z_1)\in MXM} (f_i(z_0) - f_i(z_1))^2} \qquad 6.1$$

z_k denotes the kth solution in the non-dominated front and $f_i(z_k)$ denotes the scaled objective value of z_k in ith objective, i.e. the objective value of z_k divided by the worst objective value realized in the solutions of the non-dominated front. In our case, since there are only two objectives, MS value can be calculated using only the two extreme solutions in terms of objective values in the non-dominated front and is the squared root of the summations of the squared differences of them over the two scaled objective values. The larger the MS value, the better the spread the corresponding parameter combination achieves.

The number of non-dominated solutions is another performance measure that we use to compare the results of the GAs employing different parameter combinations. The solutions in the non-dominated front are the possible solutions provided to the decision maker to select from. Therefore, the higher the number of these solutions is, the better it is. The final performance measure that we used is the CPU time that the GAs employing different parameter combinations require to obtain the non-dominated fronts. All in all, we prefer parameter combinations with smaller scaled EHR values, higher MS values, higher number of non-dominated solutions and smaller CPU times. The aim of this fine-tuning procedure is to obtain the non-dominated parameter combinations in terms of these performance measures.

The results of GA runs employing different parameter value combinations are compared on the basis of average EHR values, average MS values, average non-dominated solutions obtained and average CPU times required over the four projects and over five replications. To do so, for each parameter combination and project we applied single project scheduling approach with fitness calculation procedure1 five times and obtained five quadruples of performance measures for the corresponding parameter combination and project. Then, we took the averages of the performance measures over the five replications to obtain only one quadruple of average performance measures for a parameter combination and a project. After that, to calculate the average performance of a parameter combination, this time we took the averages of average performance measures over all projects and normalize these values. Hence, we have one quadruple of normalized average performance measures for each parameter combination. To determine the best parameter combinations, non-dominated sorting is applied to these quadruples. Then, among the non-dominated quadruples one is chosen with its corresponding parameter combination being the one to be implemented in the computational study to follow. The pseudocode of the fine-tuning procedure to determine the best GA parameter combination is given in Figure 6.3.

```
FOR EACH parameter combination pc
    FOR EACH project p
        FOR replication i = 0 → replicationCount do
            OBTAIN the non-dominated solutions using single project scheduling
            approach with fitness calculation procedure1 employing parameter
            combination pc for project p
            CALCULATE (EHR,MS,Number of Non-dominated Solutions,CPU Time)
            quadruples for parameter combination pc and project p pairs for replication i.
        END FOR
        CALCULATE average (EHR,MS,Number of Non-dominated Solutions,CPU Time)
        quadruples for  parameter combination pc and project p pairs over all replications
    END FOR EACH
    CALCULATE average (EHR,MS,Number of Non-dominated Solutions,CPU Time)
    quadruples for  parameter combination pc over all projects
END FOR EACH
NORMALIZE average (EHR,MS,Number of Non-dominated Solutions,CPU Time)
  quadruples for  all parameter combinations
PERFORM  non-dominated sorting  to determine the best parameter combinations
```

Figure 6. 3. Pseudocode for Fine-Tuning of the GA Parameters

Table 6.2 tabulates the best parameter combinations and Table 6.3 tabulates the normalized performance measure values for these parameter combinations over the tested 108 parameter combinations obtained after the fine tuning procedure.

Table 6. 2. Non-dominated GA Parameters Obtained with the Fine-Tuning Procedure

Non-dominated Parameter Combination	Crossover Rate	Mutation Rate	Population Size	Number of Generations	Number of Schedule Realizations
1	0.60	0.05	30	30	100
2	0.75	0.05	30	30	100
3	0.60	0.10	30	30	100
4	0.60	0.10	50	30	100
5	0.60	0.10	30	50	100

Table 6. 3. Normalized Performance Measure Values for the Non-dominated GA Parameters

Non-dominated Parameter Combination	Normalized Performance Measures			
	MS	Scaled EHR	Number Of Non-dominated Solutions	CPU Time
1	0.93	0.86	0.82	0.05
2	0.88	0.82	0.87	0.07
3	0.72	0.86	0.82	0.06
4	1.00	0.67	1.00	0.09
5	0.82	0.74	0.85	0.09

Project based non-dominated parameter combinations and the best parameters combinations in terms of single performance measures are given in Appendix A and Appendix B, respectively. From the non-dominated parameter combinations listed in Table 6.2, we can randomly select a combination to be used in the implementation. It should be noted that parameter combination 1 is common for the four representative projects as the non-dominated parameter combination. We can also obtain the scores for the non-dominated parameter combinations and then choose the one which gives the maximum weighted overall score. Table 6.4 presents the normalized values for the performance measures of the non-dominated parameter combinations by linear approximation between the extreme values in the corresponding performance measure column and the weighted overall scores for each non-dominated parameter combination when the weights for all performance measures are taken as 0.25.

Table 6. 4. Overall Weighted Scores for Each Non-dominated Parameter Combination

Non-dominated Parameter Combination	**Weighted Scores**				
	MS	**Scaled EHR**	**Number Of Non-dominated Solutions**	**CPU Time**	**Overall Weighted Score**
1	**0.93**	**0.00**	**0.00**	**1.00**	**0.48**
2	**0.88**	**0.21**	**0.28**	**0.50**	**0.47**
3	**0.72**	**0.00**	**0.00**	**0.25**	**0.24**
4	**1.00**	**1.00**	**1.00**	**0.00**	**0.75**
5	**0.82**	**0.63**	**0.17**	**0.00**	**0.40**

It is seen from Table 6.4 that if the weights for the performance measures are taken equal, non-dominated parameter combination 4 gives the best overall weighted score among the non-dominated parameter combinations. However, In the following sections, the results of the scheduling approaches obtained employing the parameters listed in Table 6.5.

Table 6. 5. GA Parameters Used in the Implementation

Crossover Rate	0.95
Mutation Rate	0.05
Population Size	50
Number of Generations	50
Number of Schedule Realizations for a Chromosome	100

6.2. DATA

In the implementation of proactive project scheduling phase, we have used 37 of the projects in the same project set comprising 43 projects that we have used in the implementation of deviation analysis phase. In the subsequent sections, we will first introduce the projects in the project set with their network structure and give details on the activities of the project, resource requirements of the activities and how we obtained the data related to projects and activities. After that, we will give information on the resources that the activities of the projects in the project set require and explain how we collected and preprocessed the resource data.

6.2.1. Projects and Project Networks

The projects in the project set are the projects initiated between 2007 and 2011. The number of projects initiated in each year is shown in the Table 6.6.

Table 6. 6. The Number of Projects by Their Initiation Year

2007	1
2008	14
2009	16
2010	8
2011	4

Each project requires activities that have to be performed in accordance with a set of precedence and resource constraints. For the implementation, for each project, the project plans showing the project teams, names of the activities, required resource lists with the budgeted human resource hours and equipment hours of the activities, budgeted starting and ending times of the activities, actual starting and ending times of activities, the precedence relations between the activities for the corresponding project are obtained from the database of the project management software that the R&D Department uses. Total number of activities each project contains and the number of activities in each activity class that the projects require are illustrated in Table 6.7. The projects that are written in italic in the table are excluded from the project set because of the reasons that will be explained in the following subsections.

Table 6. 7. Total Number of Activities for Each Project the Project Set

	Activity Numbers						
	Total	**Class1**	**Class2**	**Class3**	**Class4**	**Class5**	**Class6**
07-030	*66*	*5*	*43*	*1*	*1*	*14*	*2*
08-022	*44*	*4*	*31*	*1*	*9*	*0*	*0*
08-024	17	4	10	1	3	1	0
08-030	73	23	40	2	1	7	2
08-031	14	5	7	1	0	2	1
08-033	17	4	14	1	0	0	0
08-035	19	4	16	1	0	0	0
08-036	20	4	16	1	0	1	0
08-040	13	4	10	1	0	0	0
08-052	*40*	*29*	*4*	*2*	*1*	*3*	*2*
08-054	9	4	4	1	2	0	0
08-058	17	4	12	1	1	1	0
08-059	27	8	16	1	1	3	0
08-060	39	3	34	1	1	1	1
08-063	22	2	20	1	0	0	0
09-004	34	5	15	1	8	3	4
09-005	15	5	7	1	3	1	0
09-006	25	4	16	1	2	4	0
09-010	9	4	6	1	0	0	0
09-016	22	4	16	1	1	2	0
09-017	5	4	2	1	0	0	0
09-018	20	10	11	1	0	0	0
09-023	15	5	10	1	1	0	0
09-028	31	6	23	1	3	0	0
09-031	*45*	*6*	*30*	*1*	*3*	*2*	*4*
09-036	25	3	18	1	4	1	0
09-040	19	9	11	1	0	0	0
09-045	21	5	15	1	0	0	2
09-047	18	5	14	1	0	0	0
09-050	16	5	12	1	0	0	0
09-051	*63*	*7*	*34*	*1*	*4*	*10*	*3*
10-009	11	4	7	1	1	0	0
10-011	24	3	21	1	1	0	0
10-015	10	4	5	1	2	0	0
10-016	11	4	4	1	4	0	0
10-037	15	5	10	1	0	1	0
10-042	12	5	5	1	1	2	0
10-045	12	4	7	1	1	0	1
10-049	12	5	7	1	1	0	0
11-002	18	4	13	1	0	2	0
11-009	11	5	2	1	0	5	0
11-017	8	4	5	1	0	0	0
11-043	9	4	5	1	1	0	0

The project networks are of AON type FS and SS precedence relations with zero and positive time lags. There is no precedence relation between projects. When we analyzed the precedence relation information between the activities of the project that we obtained from the software, we have realized that the precedence information is incomplete for all the projects in the project set. To have a complete network information for the projects, for each project we analyzed the work content of the activities and formed new project networks using the work content information and planned and actual start and finish time information for activities to complete the missing precedence information. When completing the project networks, we have used FS and SS type precedence relations with zero and positive time lags since the project leaders in the R&D Department also use these types of precedence relations. Completing the project networks was difficult for some projects since they have a great number of activities. Hence, we have removed these projects from the project set in order to work with realistic data. The activity information about the projects removed are written italic in Table 6.7. The remaining project set is comprised of 37 projects with complete project network information. Additionally, since the literature and patent search type and meetings and administrative type activities are executed when needed, they cannot be planned before the start of the project. Furthermore, they require relatively less working hours when compared to the other activities of the projects. Hence, we excluded these activities from the project networks as well.

The required number of resources and working hours differ from activity to activity depending on the work content of the activities. While some activities require only one human resource, some activities require a total of more than 11 human resources and equipments.

6.2.2. Resources

There are two types of renewable resources required for the execution of the activities: Human resource and equipments. The projects in the project set require a total of 91 different equipment type resources and 183 different human resources. It should be noted that the resources in the equipment category can be an equipment group comprised of a number of equipments and/or a laboratory or simply can be an equipment or laboratory. In the implementation, we have used two types of information related to resources: weekly capacities and weekly availabilities.

Obtaining the Capacity Information for the Resources

Each resource has a weekly capacity that shows the maximum regular working hours that the resource with a unique id can work in a week. These capacities differ from resource to resource. While the weekly capacity of human resources is 45 working hours, these capacity values differ from 9 working hours to 672 working hours for the resources in the equipment category. Since this capacity information for the equipments is not recorded in the project management software, we have obtained this information by visiting each department owning the corresponding equipment type resource and interviewing with the responsible employee. To complete the capacity information a total of 3 R&D departments are visited and 5 responsible employee are interviewed. Visited R&D Departments are the "Mechanical Technologies Department", "Structural Design and Prototyping Department" and ""Electronic Technologies Department".

Obtaining Budgeted Resource Availabilities

We have decided to use the budgeted availability information in the implementation when scheduling the resources. For this, we have obtained the budgeted weekly work schedule of all resources working in the R&D Department for a certain time range. This time range starts with the initiation time of the first project in our project set and ends in the time after one year the last completed project in the project set is completed. Thus, we have obtained weekly budgeted workloads of the resources between 05/05/2008 and 01/01/2013 and we have filtered the information related to 274 resources that have participated in the projects in our project set. When we analyzed this information, we have realized that for some weeks, recorded workload of some resources exceed their capacity. This was expected since the project management software enables to make workload records for the past four weeks and does not have limit on the recorded working hours. Since there is such flexibility, some resources may record a monthly workload in one week. To fix this situation, we transferred the surplus workload in a week to the previous three weeks for each resource. To do so, for each resource, starting from 05/05/2008, we have checked the budgeted weekly workload until 01/01/2013. If the recorded workload exceeded the weekly capacity of the resource, we calculated the surplus and transferred this surplus to the previous weeks. While transferring this surplus, we first tried to place the surplus to the previous week. If the remaining capacity of that resource for the previous week was enough we moved the whole surplus to the previous week and terminated the transfer process. Otherwise, we transferred the surplus to the previous week until we reached the capacity of the previous week and we calculated the remaining surplus, After that we tried to place this remaining surplus to the second previous week. If the remaining capacity of that resource for the second previous week was enough, we transferred the whole remaining surplus to the second previous week and terminated the transfer process. Otherwise, we transferred the remaining surplus to the second previous week until we reached the capacity of the second previous week and we re-calculated the remaining surplus, After that, we tried to place this remaining surplus to the third previous week. If the remaining capacity of that resource for the third previous week was enough we moved the whole current remaining surplus to the third previous week and terminated the transfer process. Otherwise, we transferred the current remaining surplus to the third previous week until we reached the capacity of the third previous week and we terminated the transfer process. After this fixing, to obtain the weekly availability of resources, the weekly workloads of the resources related to the projects in our project set are subtracted from the weekly capacity of the resources.

6.3. RESULTS OBTAINED WITH THE SINGLE PROJECT SCHEDULING APPROACH

In this section, to show how our single project scheduling approach performs with fitness calculation procedure1 and fitness calculation procedure2 we have scheduled project 08-040. This project is a medium size one and thus, it is relatively easy to interpret the results obtained with the suggested approaches. It has a total of 14 activities and it requires a total of 14 different resources. The corresponding project network is given in Figure 6.4.

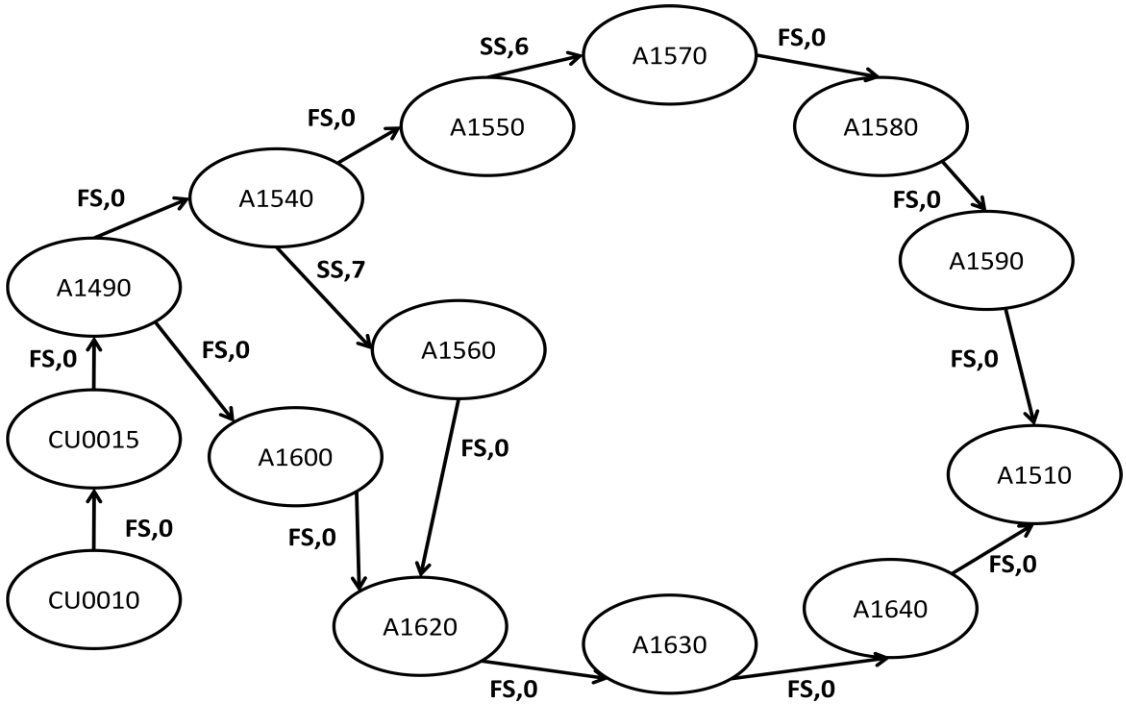

Figure 6. 4. Project Network of Project 08-040

6.3.1. Results of Single Project Scheduling Approach with Fitness Calculation Procedure1

To obtain the baseline schedule for project 08-040 with the single project scheduling approach with fitness calculation procedure1, our simulation routine is run from the beginning until all the projects are scheduled. A total of 3 different non-dominated robust baseline schedules are obtained with this approach for project 08-040. The makespan and TSAD values of these robust baseline schedules are presented in Table 6.8.

Table 6. 8. Performance Values of Non-Dominated Baseline Schedules Obtained with Single Project Scheduling Approach and Fitness Calculation Procedure1

Non-dominated Schedules	Makespan (weeks)	TSAD
1	91	2557
2	90	2631
3	89	2945

From these robust schedules, non-dominated schedule 2 is selected randomly. The robust starting and ending times for the activities in the selected schedule are given in Table 6.9 and the corresponding start and end times for the resources needed in each activity and the resource schedules are given in Appendix C and Appendix D, respectively. Note that, since our scheduling approaches yields weekly schedules, the starting and ending times represents the first day (Monday) of the starting week and ending week and they do consider the weekends.

Table 6. 9. Robust Activity Schedule of Project 08-040 Obtained with Single Project Scheduling Approach and Fitness Calculation Procedure1

Activity ID	Man-Hour Estimated by the R&D Department	Allocated Man-Hour in the Robust Schedule	Estimated Equipment-Hours	Robust Starting Time	Robust Ending Time	Duration of the Activity (weeks)
08-040CU0010	413	350	1170	07/14/08	09/29/08	11
08-040CU0015	10	8	0	10/20/08	11/10/08	3
08-040A1490	45	39	0	11/10/08	12/01/08	3
08-040A1600	135	111	0	12/01/08	12/29/08	4
08-040A1540	666	574	1440	12/01/08	05/04/09	22
08-040A1560	288	267	180	01/19/09	03/30/09	10
08-040A1620	362	318	2880	03/30/09	07/20/09	16
08-040A1630	252	246	2880	07/20/09	12/07/09	20
08-040A1550	477	448	1440	05/04/09	07/06/09	9
08-040A1570	225	202	0	06/15/09	08/24/09	10
08-040A1640	522	561	2880	12/07/09	04/05/10	17
08-040A1580	363	347	540	08/24/09	03/08/10	28
08-040A1590	38	34	0	03/08/10	03/15/10	1

The robust starting and ending time for project 08-040 in the selected non-dominated baseline schedule is 07/14/08 and 03/15/10, respectively. Notice that, the makespan values for the non-dominated baseline schedules for project 08-040 do not differ much from each other.

Table 6.10 shows the selected baseline schedules for each project in the project set with their initiation times and CPU times that the single project scheduling approach with fitness calculation procedure1 needed to obtain robust non-dominated baseline schedules.

Table 6. 10. Robust Baseline Schedules of Projects Obtained with Single Project Scheduling Approach and Fitness Calculation Procedure1

Project ID	Initiation Time	Activity Count	Robust Starting Time	Robust Ending Time	Makespan (weeks)	TSAD	CPU Time (minutes)
08-024	05/05/08	17	05/05/08	04/19/10	102	6191	15.46
08-031	05/19/08	14	05/19/08	10/26/09	75	12402	8.92
08-033	05/21/08	17	05/26/08	01/25/10	87	1533	22.65
08-035	05/29/08	19	06/02/08	08/16/10	115	4137	25.89
08-036	06/09/08	20	06/09/08	02/14/11	140	1421	21.41
08-040	07/10/08	13	07/14/08	04/05/10	90	2631	10.74
08-054	10/06/08	9	10/06/08	01/18/10	67	5070	16.14
08-058	12/12/08	17	12/15/08	12/27/10	106	857	26.32
08-059	12/15/08	27	12/15/08	03/08/10	64	7927	21.72
08-060	12/15/08	39	12/15/08	07/12/10	82	2442	32.98
08-063	01/12/09	22	01/12/09	10/04/10	90	733	23.68
09-004	01/12/09	34	01/12/09	09/06/10	86	15907	21.40
09-005	02/09/09	15	02/09/09	09/06/10	82	2200	11.63
09-006	02/23/09	25	02/23/09	08/02/10	75	1348	11.71
09-010	03/02/09	9	03/02/09	02/01/10	48	1547	15.89
09-016	01/01/09	22	01/05/09	05/24/10	72	13230	24.12
09-017	03/26/09	5	03/30/09	02/08/10	45	2302	6.40
09-018	03/16/09	20	03/16/09	09/12/11	130	2609	17.48
09-023	12/07/09	15	12/07/09	05/07/12	126	2695	17.66
09-028	04/27/09	31	04/27/09	07/25/11	117	2622	30.77
09-036	09/07/09	25	09/07/09	02/28/11	77	7310	17.48
09-040	02/02/09	19	02/02/09	03/29/10	60	5887	8.88
09-045	12/07/09	21	12/07/09	09/06/10	39	1600	12.23
09-047	11/02/09	18	11/02/09	11/29/10	56	7494	17.11
09-050	12/21/09	16	12/21/09	05/23/11	74	4436	13.33
10-009	01/11/10	11	01/11/10	04/25/11	67	4378	14.73
10-011	02/08/10	24	02/08/10	03/28/11	59	4188	13.53
10-015	02/17/10	10	02/22/10	08/16/10	25	1917	6.30
10-016	01/04/10	11	01/04/10	03/14/11	62	883	13.43
10-037	03/22/10	15	03/22/10	04/18/11	56	2182	11.10
10-042	07/19/10	12	07/19/10	10/24/11	66	1540	9.43
10-045	07/12/10	12	07/12/10	08/08/11	56	1671	15.76
10-049	09/27/10	12	09/27/10	05/21/12	86	5303	10.22
11-002	01/31/11	18	01/31/11	07/16/12	76	3166	12.04
11-009	01/24/11	11	01/24/11	10/29/12	92	3061	15.63
11-017	01/17/11	8	01/17/11	11/28/11	45	1833	9.08
11-043	05/23/11	9	05/23/11	12/05/11	28	2505	14.49

As you see from Table 6.10, all the projects are scheduled starting with their initiation time. There is no direct relationship between the activity numbers of the projects and the CPU time required to schedule the project since these CPU times are also closely related with the network complexities. If we think all these 37 projects as a one composite project, the expected starting and ending times for this composite project is 5/5/08 and 10/29/12, respectively.

6.3.2. Results of the Single Project Scheduling Approach with Fitness Calculation Procedure2

To obtain the baseline schedule for project 08-040 with the single project scheduling approach with fitness calculation procedure2, our simulation routine is run from the beginning until all the projects are scheduled. A total of 11 different non-dominated robust baseline schedules are obtained with this approach for project 08-040. The makespan and TSAD values are reported in Table 6.11.

Table 6. 11. Performance Values of Non-Dominated Baseline Schedules Obtained with Single Project Scheduling Approach and Fitness Calculation Procedure2

Non-dominated Schedules	Makespan (weeks)	TSAD
1	87	3364
2	84	3606
3	83	3717
4	82	3823
5	81	4495
6	80	4888
7	79	5231
8	77	5334
9	76	6410
10	75	6768
11	74	7499

Table 6.11 indicates that the range of the makespan values vary between 74 and 87 and the TSAD values vary between 3364 and 7499. From these robust schedules, non-dominated schedule 4 is selected randomly. The robust starting and ending times for the activities of project 08-040 for this selection are presented in Table 6.12.

Table 6. 12. Robust Activity Schedule of Project 08-040 Obtained with Single Project Scheduling Approach and Fitness Calculation Procedure2

Activity ID	Man-Hour Estimated by the R&D Department	Allocated Man-Hour in the Robust Schedule	Estimated Equipment-Hours	Robust Starting Time	Robust Ending Time	Duration of the Activity (weeks)
08-040CU0010	413	396	1170	07/14/08	09/29/08	11
08-040CU0015	10	4	0	09/29/08	10/13/08	2
08-040A1490	45	41	0	10/13/08	11/03/08	3
08-040A1540	666	162	1440	11/03/08	05/04/09	26
08-040A1560	288	186	180	12/22/08	02/02/09	6
08-040A1600	135	28	0	11/17/08	12/15/08	4
08-040A1620	362	303	2880	02/02/09	05/25/09	16
08-040A1550	477	305	1440	05/04/09	06/22/09	7
08-040A1570	225	673	0	06/15/09	08/17/09	9
08-040A1580	363	268	540	08/17/09	09/07/09	3
08-040A1590	38	10	0	09/07/09	09/14/09	1
08-040A1630	252	182	2880	05/25/09	10/12/09	20
08-040A1640	522	1001	2880	10/12/09	02/08/10	17

The robust starting and ending time for project 08-040 in the selected non-dominated baseline schedule is 07/14/08 and 02/08/10, respectively.

Table 6.13 shows the selected baseline schedules for each project in the project set with their initiation times and CPU times that the single project scheduling approach employing the fitness calculation procedure2 needs to obtain robust non-dominated baseline schedules.

Table 6. 13. Robust Baseline Schedules of Projects Obtained with Single Project Scheduling Approach and Fitness Calculation Procedure2

Project ID	Initiation Time	Activity Count	Robust Starting Time	Robust Ending Time	Makespan (weeks)	TSAD	CPU Time (minutes)
08-024	05/05/08	17	05/05/08	03/22/10	98	8768	13.29
08-031	05/19/08	14	05/19/08	06/15/09	56	25658	7.06
08-033	05/21/08	17	05/26/08	02/22/10	91	292	21.70
08-035	05/29/08	19	06/02/08	10/26/09	73	7929	23.63
08-036	06/09/08	20	06/09/08	03/07/11	143	1153	19.63
08-040	07/10/08	13	07/14/08	02/08/10	82	3823	9.25
08-054	10/06/08	9	10/06/08	10/05/09	52	6954	15.68
08-058	12/12/08	17	12/15/08	12/06/10	103	1277	24.85
08-059	12/14/08	27	12/15/08	04/12/10	69	7282	18.53
08-060	12/15/08	39	12/15/08	06/28/10	80	2786	28.74
08-063	01/12/09	22	01/12/09	08/16/10	83	1263	23.06
09-004	01/12/09	34	01/12/09	08/30/10	85	11758	16.99
09-005	02/09/09	15	02/09/09	08/09/10	78	3141	10.32
09-006	02/23/09	25	02/23/09	03/15/10	55	2080	9.51
09-010	03/02/09	9	03/02/09	04/05/10	57	501	15.35
09-016	01/01/09	22	01/05/09	02/15/10	58	18624	22.12
09-017	03/26/09	5	03/30/09	02/08/10	45	2982	5.96
09-018	03/16/09	20	03/16/09	09/12/11	130	2972	15.16
09-023	12/07/09	15	12/07/09	04/30/12	125	4097	16.62
09-028	04/27/09	31	04/27/09	06/20/11	112	11065	27.86
09-036	09/07/09	25	09/07/09	11/29/10	64	8494	15.01
09-040	02/02/09	19	02/02/09	01/18/10	50	9018	6.42
09-045	12/07/09	21	12/07/09	10/04/10	43	2000	9.56
09-047	11/02/09	18	11/02/09	09/13/10	45	11416	14.71
09-050	12/21/09	16	12/21/09	03/28/11	66	4866	11.29
10-009	01/11/10	11	01/11/10	12/20/10	49	4600	13.39
10-011	02/08/10	24	02/08/10	10/18/10	36	6405	10.64
10-015	02/17/10	10	02/22/10	05/24/10	13	3376	5.35
10-016	01/04/10	11	01/04/10	12/06/10	48	5000	11.94
10-037	03/22/10	15	03/22/10	02/21/11	48	2676	11.09
10-042	07/19/10	12	07/19/10	07/18/11	52	2154	10.19
10-045	07/12/10	12	07/12/10	08/29/11	59	1300	17.32
10-049	09/27/10	12	09/27/10	10/03/11	53	7421	11.07
11-002	01/31/11	18	01/31/11	09/26/11	34	13859	11.01
11-009	01/24/11	11	01/24/11	10/29/12	92	3740	15.70
11-017	01/17/11	8	01/17/11	10/03/11	37	3669	10.70
11-043	05/23/11	9	05/23/11	11/14/11	25	2195	14.72

As you see from Table 6.13, all the projects are scheduled starting with their initiation time. If we think all these 37 projects as a one composite project, the expected starting and ending times for this composite project is 5/5/08 and 10/29/12, respectively.

6.3.3. Comparison of Results Obtained with the Single Project Scheduling Approaches

In this subsection, we have compared the results obtained with single project scheduling approaches with respect to CPU time, diversity of the solutions and solution quality. It is seen from Table 6.10 and Table 6.13 that the CPU times required to schedule the projects is less for almost all projects when fitness calculation procedure2 is used in the single project scheduling approach instead of fitness calculation procedure1 since fitness of a chromosome is calculated using an already generated schedule in the fitness calculation procedure2. Thus, it seems sorting the schedules generated in the simulation with respect to their non-domination level requires less computational time than solving the relaxed TSAD model and generating a new schedule using the output of the TSAD model.

When we compare the activity schedules for project 08-040 obtained with suggested approaches, we see a difference in the scheduling order of activities as well as the starting and ending times of the activities. Note that, besides randomness of the GA and the used fitness calculation procedure, the difference in the allocated man-hours contribute to the difference of the resulting activity schedules obtained with each approach. Table 6.14 shows the makespan and TSAD values of the two extreme solutions of the first non-domination front obtained with single project scheduling approach with each fitness calculation procedure each approach for each project. These two extreme solutions are called as the minimum makespan schedule and the minimum TSAD schedule. Minimum makespan schedule is the schedule that has the minimum makespan value among the non-dominated schedules and the minimum TSAD schedule is the schedule that has the minimum TSAD value among the non-dominated schedules, respectively. Table 6.14 also shows the number of non-dominated schedules obtained with single project scheduling approach with each fitness calculation procedure.

Table 6. 14. Performance Comparison of Results obtained with Single Project Scheduling Approaches

Project ID	SINGLE PROJECT SCHEDULING WITH FITNESS CALCULATION PROCEDURE1					SINGLE PROJECT SCHEDULING WITH FITNESS CALCULATION PROCEDURE2				
	Min Makespan Schedule		Min. TSAD Schedule		Number of Non-dominated Schedules	Min. Makespan Schedule		Min TSAD Schedule		Number of Non-dominated Schedules
	Makespan	TSAD	Makespan	TSAD		Makespan	TSAD	Makespan	TSAD	
08-024	102	6191	102	6191	1	97	9286	103	5966	4
08-031	75	12402	76	11482	2	55	25914	76	11818	11
08-033	87	1533	90	330	3	91	292	91	292	1
08-035	98	6806	115	4137	4	70	9398	95	4760	9
08-036	140	1421	140	1421	1	139	1191	143	1153	3
08-040	89	2945	91	2557	3	74	7499	87	3364	11
08-054	66	5071	73	3879	6	45	10162	71	4222	15
08-058	106	857	106	857	1	103	1277	106	390	3
08-059	64	7927	66	5499	3	63	10017	70	7015	6
08-060	82	2442	82	2442	1	80	2786	80	2786	1
08-063	88	920	91	724	4	82	1396	86	1032	5
09-004	86	15907	87	14241	2	81	14461	85	11758	5
09-005	82	2200	83	2057	2	71	4853	86	2921	10
09-006	75	1348	76	1330	2	54	2309	61	1889	7
09-010	48	1547	48	1547	1	57	501	58	338	2
09-016	71	13587	76	11142	5	48	31015	78	12810	14
09-017	45	2302	45	2302	1	41	4407	61	2283	8
09-018	130	2609	130	2609	1	129	3177	131	2869	3
09-023	125	2819	126	2695	2	100	9725	126	3319	9
09-028	117	2622	117	2622	1	105	20437	125	7941	15
09-036	76	8005	79	6247	4	61	13491	72	6398	10
09-040	59	6346	61	5513	3	47	11171	64	6338	9
09-045	39	1600	39	1600	1	42	2060	43	2000	2
09-047	54	8184	56	7494	3	41	16450	58	6457	7
09-050	72	4897	74	4436	3	55	8905	69	4748	9
10-009	66	4389	67	4378	2	45	5193	57	3838	8
10-011	51	5491	60	3945	5	32	10348	42	5049	9
10-015	23	2249	25	1917	3	11	4521	26	2057	9
10-016	62	883	62	883	1	46	9741	56	3864	6
10-037	56	2182	58	1533	3	34	6644	48	2676	13
10-042	64	1593	66	1540	2	44	4161	58	2043	9
10-045	56	1671	58	1594	3	54	2192	62	1033	8
10-049	70	7383	86	5303	5	45	8901	61	4062	10
11-002	75	3258	76	3166	2	34	13859	45	7336	7
11-009	92	3061	93	2781	2	91	3772	93	3491	3
11-017	44	1845	45	1833	3	37	3669	45	1617	5
11-043	27	2628	29	2452	3	15	3757	25	2195	10

Table 6.14 indicates that, using the single project scheduling approach with fitness calculation procedure1, less number of non-dominated schedules are obtained for each project. Table 6.14 also indicates that, while single project scheduling approach with fitness calculation procedure1 tends to find schedules with less TSAD, single project scheduling approach with fitness calculation procedure2 tends to find schedules with smaller makespan values. The blue cells indicate the minimum TSAD values and the green cells indicates the minimum makespan values for the corresponding projects. By looking at these values, we conclude that while single project scheduling approach with fitness calculation procedure1 has found a better schedule only for project 09-045, single project scheduling approach with fitness calculation procedure2 has found better schedules for projects 08-024, 08-036, 08-058, 09-004, 09-017, 09-047, 10-009, 10-045, 10-049, 11-0017, and 11-043, since both the makespan and TSAD values are superior. For the 23 of remaining 25 projects, while the TSAD values are less in the schedules obtained with the fitness calculation procedure1, the makespan values are less in the schedules obtained with fitness calculation procedure2. The situation is just the opposite for the two of the remaining 25 projects,

6.4. RESULTS OBTAINED WITH MULTI-PROJECT SCHEDULING APPROACH

In this section, to show how our multi-project scheduling approach performs with fitness calculation procedure1 and fitness calculation procedure2 we have scheduled project 08-035. This project is a medium size one. It has a total of 19 activities and it requires a total of 43 different resources. There are three active projects in the system and the total number of active activities is 48 when project 08-035 is initiated. In the following, we report robust baseline project plans for all the projects in our project set with fitness calculation procedure1 and fitness calculation procedure2 using multi-project scheduling approach. As an example, we show the robust baseline schedule for project 08-035 obtained with each fitness calculation procedure.

6.4.1. Results of the Multi-Project Scheduling Approach with Fitness Calculation Procedure1

In this subsection, the baseline schedule for project 08-035 obtained with the multi-project scheduling approach with fitness calculation procedure1 and the changes in the schedules of the current active activities of the previously initiated projects are presented. The expected completion times for these projects are shown in Table 6.15.

Table 6. 15. Expected Completion Times of Active Projects Scheduled with Multi-Project Scheduling Approach and Fitness Calculation Procedure1 when Project 08-035 Initiated

Project ID	Completion Time Before Project 08-035 Initiated
08-024	04/05/10
08-031	11/02/09
08-033	02/15/10

When project 08-035 is initiated, now in our multi-project scheduling approach, all the active activities have the probability to have a change in their current schedule. A total of 1 different non-dominated robust schedule is obtained with this approach for all the active projects and the activities of 08-035. The starting and ending times for project 08-035 are given in Table 6.16.

Table 6. 16. Activity Schedule of Project 08-035 Obtained with Multi-Project Scheduling Approach and Fitness Calculation Procedure1

Activity ID	Man-Hour Estimated by the R&D Department	Allocated Man-Hour in the Robust Schedule	Estimated Equipment -Hours	Robust Starting Time	Robust Ending Time	Duration of the Activity (weeks)
08-035A1000	800	833	1194	06/02/08	09/29/08	17
08-035A1010	15	15	0	09/29/08	10/06/08	1
08-035A1060	27	30	0	10/06/08	10/13/08	1
08-035A1075	153	155	0	10/13/08	01/12/09	13
08-035A1080	332	340	585	01/12/09	03/23/09	10
08-035A1180	269	247	750	10/13/08	04/20/09	27
08-035A1090	282	280	483	12/08/08	05/25/09	24
08-035A1120	674	688	2443	10/13/08	06/08/09	34
08-035A1140	621	612	1845	10/13/08	07/13/09	39
08-035A1130	261	268	390	11/03/08	07/20/09	37
08-035A1100	392	405	249	05/25/09	07/06/09	6
08-035A1190	278	298	485	07/20/09	08/24/09	5
08-035A1200	293	328	620	08/17/09	09/14/09	4
08-035A1160	493	518	572	07/27/09	09/21/09	8
08-035A1220	304	317	247	09/21/09	11/30/09	10
08-035A1170	406	378	527	09/14/09	01/25/10	19
08-035A1150	580	615	737	07/27/09	01/04/10	23
08-035A1210	530	566	397	01/25/10	03/01/10	5
08-035A1030	27	24	0	03/08/10	03/15/10	1

Initiation of project 08-035 has affected 17 of 48 existing activities. The affected activities and their schedules before and after project 08-035 is initiated are illustrated in Table 6.17.

Table 6. 17. Schedule Change of Existing Activities Affected by the Initiation of Project 08-040 Scheduled with Multi-Project Scheduling Approach and Fitness Calculation Procedure1

		ACTIVITY SCHEDULES BEFORE PROJECT 08-035 IS INITIATED			ACTIVITY SCHEDULES AFTER PROJECT 08-035 IS INITIATED		
Project ID	Activity ID	Starting Time	Ending Time	Duration (weeks)	Starting Time	Ending Time	Duration (weeks)
08-024	08-024A1100	09/08/08	01/05/09	17	09/08/08	02/09/09	22
08-024	08-024A1061	12/08/08	02/09/09	9	12/08/08	01/26/09	7
08-024	08-024A1220	02/09/09	04/13/09	9	02/09/09	04/20/09	10
08-024	08-024A1190	02/09/09	03/23/09	6	02/09/09	03/02/09	3
08-024	08-024A1170	02/09/09	04/27/09	11	02/09/09	08/17/09	27
08-024	08-024A1260	04/13/09	08/31/09	20	04/20/09	08/31/09	19
08-031	08-031A1090	09/08/08	04/06/09	30	09/08/08	03/02/09	25
08-031	08-031A1230	05/11/09	06/29/09	7	05/11/09	06/22/09	6
08-031	08-031A1150	06/29/09	08/03/09	5	06/22/09	08/03/09	6
08-033	08-033A1050	12/08/08	03/16/09	14	12/08/08	04/06/09	17
08-033	08-033A2010	12/29/08	04/20/09	16	12/29/08	04/13/09	15
08-033	08-033A1055	03/16/09	03/30/09	2	04/06/09	04/20/09	2
08-033	08-033A3010	06/22/09	07/27/09	5	06/22/09	08/03/09	6
08-033	08-033A3030	07/27/09	08/17/09	3	08/03/09	08/24/09	3
08-033	08-033A3040	09/07/09	09/21/09	2	09/07/09	12/14/09	14
08-033	08-033A4010	09/07/09	10/05/09	4	09/07/09	11/16/09	10
08-033	08-033A4020	09/07/09	12/14/09	14	09/07/09	10/26/09	7

Table 6.17 shows that most of the time, the activities have ending time delay when project 08-035 is initiated, since the bi-objective GA tries to keep the original starting times of the existing activities when a new project is initiated. Notice that while some activities have positive time delays, some activities have negative time delays. The reason for these negative time delays is the change in the order of scheduling for activities. Table 6.18 shows the effect of these changes in the activity schedules to the completion times of active projects.

Table 6. 18. Effect of Initiation of Project 08-040 to the Completion Times of Existing Projects Scheduled with Multi-Project Scheduling Approach and Fitness Calculation Procedure1

Project ID	Completion Time Before 08-035 Initiated	Completion Time After 08-035 Initiated
08-024	04/05/10	04/12/10
08-031	11/02/09	11/02/09
08-033	02/15/10	02/15/10
08-035	-	03/15/10

As you see from Table 6.18 that, to obtain a good completion time for the composite project consisting of all the active activities in the system, project 08-024 is delayed one week. The completion time of the composite project is 03/15/10.

Final selected project schedules (project schedules after project 08-060 initiated and scheduled) for the first 10 projects in the project set are presented in Table 6.19. For the purpose of presenting a simplified demonstration, instead of scheduling all the 37 project in the project set, we decided to schedule only the first 10 projects, since scheduling the projects with the multi-project scheduling approach requires a great amount of computation time.

Table 6. 19. Final Robust Baseline Schedules of Projects Obtained with Multi-Project Scheduling Approach and Fitness Calculation Procedure1

Project ID	Starting Time	Ending Time	Makespan (weeks)
08-024	05/05/08	07/19/10	115
08-031	05/19/08	01/25/10	88
08-033	05/26/08	04/05/10	97
08-035	06/02/08	03/29/10	95
08-036	06/09/08	08/16/10	114
08-040	07/14/08	06/07/10	99
08-054	10/06/08	03/01/10	73
08-058	12/15/08	09/27/10	93
08-059	12/15/08	04/12/10	69
08-060	12/15/08	06/21/10	79

The completion time of the composite project when project 08-060 is initiated and scheduled is 09/27/10. Table 6.20 shows the selected baseline schedules for the first 10 project in the project set and Table 6.21 presents the total number of activities scheduled, robust completion times for the composite project and TSAD values. Note that these values for a project are assigned values when that project is initiated.

Table 6. 20. Robust Baseline Schedules of Projects Obtained with Multi-Project Scheduling Approach and Fitness Calculation Procedure1 When Each Project is Initiated

Project ID	Activity Count	Starting Time	Ending Time	Makespan (weeks)
08-024	17	05/05/08	04/19/10	102
08-031	14	05/19/08	11/02/09	76
08-033	17	05/26/08	02/15/10	90
08-035	19	06/02/08	03/15/10	93
08-036	20	06/09/08	09/13/10	118
08-040	13	07/14/08	04/19/10	92
08-054	9	10/06/08	03/01/10	73
08-058	17	12/15/08	09/27/10	93
08-059	27	12/15/08	04/19/10	70
08-060	39	12/15/08	06/21/10	79

Table 6. 21. Performance Values of Projects Obtained with Multi-Project Scheduling Approach and Fitness Calculation Procedure1

Project ID	Activity Count	Total Number of Activities Scheduled	Robust Ending Time of Composite Project	TSAD of Active Activities	TSAD of Initiated Project	TSAD of Scheduled Activities	CPU Time (minu tes)
08-024	17	17	04/19/10	0	6191	6191	15.46
08-031	14	31	04/12/10	800	7253	8053	41.60
08-033	17	48	04/12/10	300	210	510	74.23
08-035	19	67	04/12/10	600	5032	5632	100.57
08-036	20	87	09/13/10	5700	3837	9537	115.84
08-040	13	100	09/13/10	13800	4533	18333	129.52
08-054	9	96	08/02/10	18900	4605	23505	159.96
08-058	17	107	09/27/10	18200	650	18850	219.15
08-059	27	134	09/27/10	28000	6628	34628	238.21
08-060	39	173	09/27/10	23400	1887	25287	278.20

It is seen from Table 6.21 that when the scheduled activity number grows, the contribution of TSAD of the existing active activities to TSAD of scheduled activities grows as well. To keep the contribution of the TSAD of the existing activities and the activities of the newly initiated project roughly equal, when calculating the TSAD of a schedule in the bi-objective GA this time, we have used TSAD values divided by the corresponding activity numbers before summing them to obtain the TSAD of the scheduled activities and we called this strategy as divided TSAD strategy. The final starting and ending times for the projects and the starting and ending times for the projects when they are initiated and the performance values obtained using the multi-project scheduling approach with fitness calculation procedure with divided TSAD strategy are shown in Table 6.22, 6.23, and Table 6.24, respectively.

Table 6. 22. Final Robust Baseline Schedules of Projects Obtained Using the Multi-Project Scheduling Approach with Fitness Calculation Procedure1 and Divided TSAD Strategy

Project ID	Starting Time	Ending Time	Makespan (weeks)
08-024	05/05/08	07/19/10	115
08-031	05/19/08	01/04/10	85
08-033	05/26/08	04/05/10	97
08-035	06/02/08	03/29/10	95
08-036	06/09/08	08/16/10	114
08-040	07/14/08	04/26/10	93
08-054	10/06/08	02/22/10	72
08-058	12/15/08	09/27/10	93
08-059	12/15/08	03/22/10	66
08-060	12/15/08	06/21/10	79

When Table 6.22 is compared with Table 6.19, it is seen that, the divided TSAD strategy gives schedules with smaller makespan values for most of the projects but the completion time over all projects is still 9/27/10.

Table 6. 23. Robust Baseline Schedules of Projects Obtained Using Multi-Project Scheduling Approach with Fitness Calculation Procedure1 and Divided TSAD Strategy When Each Project is Initiated

Project ID	Activity Count	Starting Time	Ending Time	Makespan (weeks)
08-024	17	05/05/08	04/19/10	102
08-031	14	05/19/08	11/02/09	76
08-033	17	05/26/08	02/15/10	90
08-035	19	06/02/08	03/22/10	94
08-036	20	06/09/08	09/27/10	120
08-040	13	07/14/08	04/19/10	92
08-054	9	10/06/08	02/22/10	72
08-058	17	12/15/08	10/04/10	94
08-059	27	12/15/08	03/22/10	66
08-060	39	12/15/08	06/21/10	79

When Table 6.23 is compared with Table 6.20, it is seen that there is not a big difference in the project plans. The makespan values are very close for each project when they are initiated.

Table 6. 24. Performance Values of Projects Obtained with Multi-Project Scheduling Approach with Fitness Calculation procedure1 and Divided TSAD Strategy

Project ID	Activity Count	Total Number of Activities Scheduled	Robust Ending Time of Composite Project	TSAD of Active Activities	TSAD of Initiated Project	TSAD of Scheduled Activities	CPU Time (minutes)
08-024	17	17	4/19/2010	0	364	364	15.46
08-031	14	31	4/12/2010	29	509	538	42.76
08-033	17	48	4/12/2010	29	21	50	74.7
08-035	19	67	4/12/2010	29	259	288	100.96
08-036	20	87	9/27/2010	140	51	191	113.9
08-040	13	100	7/19/2010	442	170	612	131.5
08-054	9	98	7/19/2010	175	453	628	162.03
08-058	17	106	8/2/2010	174	17	191	211.23
08-059	27	133	9/27/2010	218	193	411	242.1
08-060	39	172	9/27/2010	145	46	191	275.67

Table 6.24 shows that, now the contribution of the TSAD of active activities to the TSAD of the scheduled activities does not grow when the with the increasing number of active activities since we have normalized the TSAD of the active activities and the TSAD of the initiated project with the corresponding activity numbers.

6.4.2. Results of the Multi- Project Scheduling Approach with Fitness Calculation Procedure2

In this subsection, the baseline schedule for project 08-035 obtained with the multi- project scheduling approach with fitness calculation procedure2 and the changes in the schedules of the current active activities of the previously initiated projects are presented. When project 08-035 is initiated, there are 3 active projects in the system and the total number of active activities is 48. The expected completion times for these projects are shown in Table 6.25.

Table 6. 25. Expected Completion Times of Active Projects Scheduled with Multi-Project Scheduling Approach and Fitness Calculation Procedure2 When Project 08-035 Initiated

Project ID	Completion Time Before 08-035 Initiated
08-024	3/22/2010
08-031	10/19/2009
08-033	2/1/2010

A total of 4 non-dominated robust schedules are obtained with this approach for all the active projects and the activities of 08-035. The makespan and the TSAD values obtained for each non-dominated schedule are shown in Table 6.26.

Table 6. 26. Performance Values Obtained with Multi-Project Scheduling Approach when Project 08-035 Initiated

Non-dominated Schedules	Makespan of the Composite Project (weeks)	TSAD of the Scheduled Activities	TSAD of Active Activities	TSAD of Project 08-035
1	99	5444	100	5344
2	98	5532	0	5532
3	96	5538	200	5338
4	94	5675	0	5675

From these non-dominated schedules, non-dominated schedule4 is selected randomly. The starting and ending times for project 08-035 are given in Table 6.27.

Table 6. 27. Activity Schedule of Project 08-035 Obtained with Multi-Project Scheduling Approach and Fitness Calculation Procedure2

Activity ID	Man-Hour Estimated by the R&D Departme nt	Allocated Man-Hour in the Robust Schedule	Estimated Equipment -Hours	Robust Starting Time	Robust Ending Time	Duration of the Activity (weeks)
08-035A1000	800	859	1194	06/02/08	08/25/08	12
08-035A1010	15	10	0	08/25/08	09/01/08	1
08-035A1060	27	20	0	09/01/08	09/08/08	1
08-035A1075	153	192	0	09/08/08	11/24/08	11
08-035A1180	269	48	750	09/08/08	03/23/09	28
08-035A1130	261	472	390	09/08/08	04/13/09	31
08-035A1090	282	563	483	11/03/08	05/25/09	29
08-035A1080	332	451	585	11/24/08	06/15/09	29
08-035A1120	674	226	2443	09/08/08	06/29/09	42
08-035A1100	392	356	249	06/15/09	07/06/09	3
08-035A1140	621	589	1845	09/08/08	07/20/09	45
08-035A1170	406	246	527	09/14/09	02/08/10	21
08-035A1150	580	652	485	07/27/09	08/24/09	4
08-035A1190	278	133	620	07/20/09	08/17/09	4
08-035A1200	293	200	247	08/17/09	02/08/10	25
08-035A1220	304	52	620	02/08/10	02/15/10	1
08-035A1160	493	850	572	07/27/09	12/21/09	21
08-035A1210	530	358	397	02/08/10	03/08/10	4
08-035A1030	27	52	0	03/08/10	03/22/10	2

Initiation of project 08-035 has affected 8 of 48 existing activities. The affected activities and their schedule before and after project 08-035 is initiated are reported in Table 6.28.

Table 6. 28. Schedule Change of Existing Activities Affected by the Initiation of Project 08-040 Scheduled with Multi-Project Scheduling Approach and Fitness Calculation Procedure2

		ACTIVITY SCHEDULES BEFORE PROJECT 08-035 IS INITIATED			ACTIVITY SCHEDULES AFTER PROJECT 08-035 IS INITIATED		
Project ID	Activity ID	Starting Time	Ending Time	Duration (weeks)	Starting Time	Ending Time	Duration (weeks)
08-024	08-024A1150	08/25/08	01/19/09	21	08/25/08	01/12/09	20
08-024	08-024A1170	02/02/09	04/20/09	11	02/02/09	04/27/09	12
08-024	08-024A1235	01/04/10	01/11/10	1	01/04/10	02/01/10	4
08-033	08-033A2010	12/22/08	04/20/09	17	12/22/08	06/15/09	25
08-033	08-033A4010	08/31/09	09/14/09	2	08/31/09	11/30/09	13
08-033	08-033A3040	08/31/09	11/30/09	13	08/31/09	10/26/09	8
08-033	08-033A7100	11/30/09	01/04/10	5	11/30/09	02/01/10	9
08-033	08-033A5010	11/30/09	02/01/10	9	11/30/09	01/11/10	6

Table 6.28 shows that all the active activities have ending time delay when project 08-035 is initiated, since the bi-objective GA tries to keep the original starting times of the existing activities when a new project is initiated. Table 6.29 shows the effect of these changes in the activity schedules to the completion times of active projects.

Table 6. 29. Effect of Initiation of Project 08-040 to the Completion Times of Existing Projects Scheduled with Multi-Project Scheduling Approach and Fitness Calculation Procedure2

Project ID	Completion Time Before 08-035 Initiated	Completion Time After 08-035 Initiated
08-024	3/22/2010	3/22/2010
08-031	10/19/2009	10/19/2009
08-033	2/1/2010	2/1/2010
08-035	-	03/22/10

As you see from Table 6.29 that the completion times of projects did not change. The completion time of the composite project is 03/22/10. Final selected project schedules (project schedules after project 08-060 initiated and scheduled) for the first 10 projects in the project set are presented in Table 6.30.

Table 6. 30. Final Robust Baseline Schedules of Projects Obtained with Multi-Project Scheduling Approach and Fitness Calculation Procedure2

Project ID	**Starting Time**	**Ending Time**	**Makespan (weeks)**
08-024	05/05/08	06/28/10	112
08-031	05/19/08	12/28/09	84
08-033	05/26/08	05/17/10	103
08-035	06/02/08	06/28/10	108
08-036	06/09/08	09/27/10	120
08-040	07/14/08	04/12/10	91
08-054	10/06/08	02/15/10	71
08-058	12/15/08	09/27/10	93
08-059	12/15/08	03/01/10	63
08-060	12/15/08	06/21/10	79

The completion time of the composite project, when project 08-060 is initiated and scheduled, is 09/27/10. Table 6.31 shows the selected baseline schedules for the first 10 projects in the project set and Table 6.32 presents the total number of activities scheduled, robust completion time for the composite project and the TSAD values. Note that these values for a project are assigned values when that project is initiated.

Table 6. 31. Robust Baseline Schedules of Projects Obtained with Multi-Project Scheduling Approach and Fitness Calculation Procedure2 When Each Project is Initiated

Project ID	Activity Count	Starting Time	Ending Time	Makespan (weeks)
08-024	17	05/05/08	03/22/10	98
08-031	14	05/19/08	10/19/09	74
08-033	17	05/26/08	02/01/10	88
08-035	19	06/02/08	03/22/10	94
08-036	20	06/09/08	10/04/10	121
08-040	13	07/14/08	04/12/10	91
08-054	9	10/06/08	02/15/10	71
08-058	17	12/15/08	09/27/10	93
08-059	27	12/15/08	03/08/10	64
08-060	39	12/15/08	06/21/10	79

Table 6. 32. Performance Values of Projects Obtained with Multi-Project Scheduling Approach and Fitness Calculation Procedure2

Project ID	Activity Count	Total Number of Activities Scheduled	Robust Ending Time of Composite Project	TSAD of Active Activities	TSAD of Initiated Project	TSAD of Scheduled Activities	CPU Time (minutes)
08-024	17	17	03/22/10	0	8768	8768	13.29
08-031	14	31	03/22/10	0	7595	7595	31.93
08-033	17	48	03/22/10	0	541	541	57.82
08-035	19	67	03/22/10	0	5675	5675	75.90
08-036	20	87	10/04/10	10200	1340	11540	85.06
08-040	13	100	09/27/10	13000	6133	19133	99.89
08-054	9	94	09/27/10	16300	4621	20921	136.11
08-058	17	105	09/27/10	16900	2451	19351	174.11
08-059	27	132	09/27/10	33500	8993	42493	186.87
08-060	39	171	09/27/10	21700	2938	24638	217.91

It is seen from Table 6.32 that when the scheduled activity number grows, the contribution of TSAD of the existing activities to TSAD of the scheduled activities increases. The final starting and ending times for the projects and the starting and ending times for the projects when they are initiated and the performance values obtained using the divided TSAD strategy is shown in Table 6.33, 6.34, and Table 6.35, respectively.

Table 6. 33. Final Robust Baseline Schedules of Projects Obtained with Multi-Project Scheduling Approach and Fitness Calculation Procedure2 and Divided TSAD Strategy

Project ID	Activity Count	Starting Time	Ending Time	Makespan (weeks)
08-024	17	5/5/2008	6/28/2010	112
08-031	14	5/19/2008	11/2/2009	76
08-033	17	5/26/2008	2/22/2010	91
08-035	19	6/2/2008	1/4/2010	83
08-036	20	6/9/2008	10/4/2010	121
08-040	13	7/14/2008	2/15/2010	83
08-054	9	10/6/2008	2/8/2010	70
08-058	17	12/15/2008	9/27/2010	93
08-059	27	12/15/2008	3/8/2010	64
08-060	39	12/15/2008	6/21/2010	79

When Table 6.31 is compared with Table 6.19, it is seen that, the divided TSAD strategy gives schedules with less makespan values for half of the projects but the completion time over all projects is still 9/27/10.

Table 6. 34. Robust Baseline Schedules of Projects Obtained with Multi-Project Scheduling Approach and Fitness Calculation Procedure2 with Divided TSAD Strategy When Each Project is Initiated

Project ID	Starting Time	Ending Time	Makespan (weeks)
08-024	5/5/2008	3/22/2010	98
08-031	5/19/2008	11/2/2009	76
08-033	5/26/2008	2/22/2010	91
08-035	6/2/2008	1/4/2010	83
08-036	6/9/2008	10/4/2010	121
08-040	7/14/2008	2/15/2010	83
08-054	10/6/2008	2/8/2010	70
08-058	12/15/2008	9/27/2010	93
08-059	12/15/2008	3/8/2010	64
08-060	12/15/2008	6/21/2010	79

When Table 6.34 is compared with Table 6.31, it is seen that there is not much difference in the project plans. The makespan values are very close for each project when they are initiated.

Table 6. 35. Performance Values of Projects Obtained with Multi-Project Scheduling Approach with Fitness Calculation Procedure2 and Divided TSAD Strategy

Project ID	Activity Count	Total Number of Activities Scheduled	Robust Ending Time of Composite Project	TSAD of Active Activities	TSAD of Initiated Project	TSAD of Scheduled Activities	CPU Time (minut es)
08-024	17	17	03/22/10	0	515	515	13.29
08-031	14	31	03/22/10	5	572	577	32.13
08-033	17	48	03/22/10	3	20	23	57.93
08-035	19	67	03/22/10	33	268	301	76.73
08-036	20	87	10/04/10	128	65	193	85.42
08-040	13	100	09/06/10	295	245	540	100.12
08-054	9	94	06/28/10	482	473	955	144.31
08-058	17	106	09/27/10	265	13	278	178.29
08-059	27	133	09/27/10	385	175	560	196.95
08-060	39	172	09/27/10	204	46	250	229.33

Table 6.35 shows that, now the contribution of the TSAD of active activities to the TSAD of the scheduled activities does not grow when the with the increasing number of active activities since we have normalized the TSAD of the active activities and the TSAD of the initiated project with the corresponding activity numbers.

6.5. COMPARISON OF THE RESULTS OBTAINED WITH SINGLE PROJECT AND MULTI-PROJECT SCHEDULING APPROACHES

In the previous sections, we have presented the project schedules and the performance values obtained with the suggested approaches. In this section, we will compare the results of the suggested approaches with respect to CPU time, solution quality and manageability. Table 6.36 shows the final project completion times obtained with each of the suggested approaches.

Table 6. 36. Final Completion Times of Projects Obtained with Suggested Approaches

Project ID	FINAL ENDING TIMES OF THE PROJECTS OBTAINED WITH			
	SINGLE PROJECT SCHEDULING		MULTI-PROJECT SCHEDULING	
	WITH FITNESS CALCULATION PROCEDURE1	WITH FITNESS CALCULATION PROCEDURE2	WITH FITNESS CALCULATION PROCEDURE1	WITH FITNESS CALCULATION PROCEDURE2
08-024	04/19/10	03/22/10	07/19/10	06/28/10
08-031	10/26/09	06/15/09	01/25/10	12/28/09
08-033	01/25/10	02/22/10	04/05/10	05/17/10
08-035	08/16/10	10/26/09	03/29/10	06/28/10
08-036	02/14/11	03/07/11	08/16/10	09/27/10
08-040	04/05/10	02/08/10	06/07/10	04/12/10
08-054	01/18/10	10/05/09	03/01/10	02/15/10
08-058	12/27/10	12/06/10	09/27/10	09/27/10
08-059	03/08/10	04/12/10	04/12/10	03/01/10
08-060	07/12/10	06/28/10	06/21/10	06/21/10

Table 6.36 shows that for all the projects except 08-035, 08-036,08-058, and 08-060 single project scheduling approach gives better completion times. On the other hand, if we think all the projects as a composite project, the completion time of this composite project obtained with multi-project scheduling approaches approximately 5 months with fitness calculation procedure1 and approximately 6 months earlier with fitness calculation procedure2. We can clearly state that, if completing the composite project is more important than completing the projects individually, multi-project scheduling approach is better. On the other hand, there are some disadvantages of multi-project scheduling. First, it re-schedules all the active activities with a new project initiation, so an activity is scheduled more than once even if there is no disruption affecting that activity. This re-scheduling increases system nervousness and demotivates the resources that work on the activities. Additionally, when we approach the situation from the perspective of project leaders and project teams, each project leader and each project team wants to complete the schedule as soon as possible to be satisfied with the success they get. In such an R&D project management an environment, adopting multi-project scheduling and managing projects might not be so easy. Besides, multi-project scheduling approach more CPU time than the single project scheduling approach.

In this Chapter, first we explained the real data we have used in the implementation of the proactive project scheduling phase of the three-phase approach that we have developed and then we presented project scheduling results obtained with each of the suggested proactive project scheduling approaches. Finally, after comparing the suggested approaches with each other, we have compared the results obtained with our proactive project scheduling approaches with the actual project baseline schedules of the R&D Department. The results show that, using the fitness calculation procedure2 in both project scheduling approaches, we obtain better results both in makespan and instability for most of the projects. The results also indicated that the project schedules obtained with the suggested robust project scheduling approaches are better in terms of makespan when they are compared with the actual project plans of the R&D Department.

CHAPTER 7

REACTIVE PROJECT SCHEDULING WITH REAL DATA

For the implementation of Phase III of the proposed three-phase approach for robust R&D project scheduling an implementation routine is developed and a subset of the projects in our project set is used. The developed implementation routine is exactly the same as the implementation routine used in the implementation of the proactive project scheduling procedure but this time it also considers random disruptions that can occur during the project execution. In this implementation routine, single project scheduling approach with fitness calculation procedure1 using the same GA parameters employed in the implementation of Phase II is adopted to schedule the newly arrived projects. Six types of disruptions are considered and the scheduled order repair heuristic is used to fix the project schedules when a disruption occurs. The scheduled order repair heuristic gives two alternative repaired schedules. In this chapter, we first give the basic scheme of the implementation routine in section 7.1. Then, in section 7.2, we give information about the disruption types we have considered and explain how we generate these disruptions. After that, we generate each type of disruption on a baseline project plan and present the repaired schedules with a comparison in subsection 7.3. Finally, in section 7.4 a possible project execution scenario obtained with the implementation routine is presented.

7.1. BASIC SCHEME OF THE IMPLEMENTATION ROUTINE

In our implementation routine we have considered a time loop. In this routine, each time instant represents a week. The steps of the implementation routine are given in Figure 7.1.

```
FOR t=0→totalTimeCount do
    DELETE completed activities from the active activity list
    UPDATE remaining workloads of the active activities
        IF a new project is initiated
            CALL proactive baseline schedule for the initiated project
            ADD the activities of the project to active activity list
        END IF
    GENERATE a random probability disrProb
    WHILE (disrProb<p_threshold)
        SELECT the disruption type
        MAKE necessary updates in the inputs affected by the disruption
        FIX the schedules
        COMPARE the fixed schedule with the schedule before disruption
        DIVIDE p_threshold by 2
        UPDATE the random probability disrProb
    END WHILE
    INCREMENT t
END FOR
```

Figure 7. 1. Pseudocode of the Implementation Routine Developed

Our implementation routine works on an active activity list. This activity list is comprised of scheduled activities that are not completed yet. The activities in this list may be started but not finished or may be planned to start in future. Starting from the beginning of the time range considered, in each time instant the routine first checks if an activity is completed in the previous time instant and if there is any activity completed in the previous time instant, the routine removes these activities from the active activity list. Then, it updates the progress of the active activities by updating the remaining work for each activity. After this update, the routine checks if a new project is initiated at that time instant. If there is any new project initiated in the system, the implementation routine calls the single project scheduling approach with fitness calculation procedure1 and obtains a set of non-dominated robust baseline schedules for this new project. After one of the robust baseline schedule is selected randomly from these non-dominated robust baseline schedules, the routine adds the activities of this newly initiated project to the active activity list. If there is more than one project initiated simultaneously, the routine selects one of them randomly. Then, a random probability *disrProb* is generated to decide if a disruption occurs at that time instant. If this *disrProb* is smaller than a pre-specified threshold value *p_threshold*, it means that there is a disruption at time t and the disruption type is chosen randomly. Depending on the disruption type chosen, necessary changes are made for the activities. The disruption types and the procedure that we follow to make the necessary changes for each disruption type will be explained in the following subsections. After making the necessary changes, the routine calls the scheduled order repair heuristic to fix the baseline schedule and provides two repaired schedules for the active activities. After selecting one of the repaired schedules, the routine shows the affected resources and affected activities with the impact that the disruption created on the affected resources and affected activities. Since there can be more than one disruption in a single time instant in real life, the routine also allows this situation to happen by generating a new *disrProb* for the same time instant. But this time the threshold value *p_threshold* is divided by 2 before checking if an additional disruption is occurred in the same time instant. With this decrease, the routine decreases the possibility of having another disruption in the same time instant after a disruption occurs. The routine continue until the time reaches the predefined time *totalTimeCount*.

7.2. DISRUPTIONS

In the implementation routine, we have considered six type of internal disruptions. We believe that these disruptions cover a big portion of the internal disruptions that can occur during project execution and reflects the dynamic and stochastic project management environment present in the R&D Department. In our routine, we considered disruption types that can cause not only positive time delays, but also negative time delays on the baseline schedule. The disruption types considered in the implementation routine are listed as follows:

- Type 1: Unavailability of a resource in a time instant for certain hours.
- Type 2: Inadequacy of the planned working hours for a resource to complete an activity.
- Type 3: Removal of an activity.
- Type 4: Insertion of an activity.
- Type 5: Replacing an activity with a more efficient activity.
- Type 6: Change in the activity parameters.

The details related to each disruption type and the required changes to be done when that disruption occurs are given in the following subsections.

7.2.1. Unavailability of a Resource in a Time Instant for Certain Hours

This disruption type is one of the most common disruption type that is faced in the R&D Department. This disruption type occurs when a human resource gets his/her annual leave, needs a day off or when s/he is called from the production department to solve an urgent problem arises in the production process. For equipment type resources, breakdowns or maintenance requirements can be the reasons for this type of disruption. This type of unavailability is called as disruption if this unavailability is more than nine hours in a week since the unavailability less than nine hours can be absorbed with overtime working.

In our implementation routine, to generate this kind of disruption in a week, we consider all the resources that actively work on some activities more than nine hours in that week and select one of them randomly. Then we generate a new random availability of that resource for that week. This new availability can range from zero to nine hours less than the planned workload of the resource at that week. After this update, all the active activities are rescheduled using the new availability information since this unavailability can affect all the active activities through precedence relations.

7.2.2. Inadequacy of the Planned Working Hours for a Resource to Complete an Activity

This disruption type is the most common disruption type that is faced in the R&D Department. Since most the projects that the resources work on are research-based projects, making correct estimations for the required working hours needed to complete an activity for a resource is a very difficult task and these estimations are subject to great variability. Most of the time, the planned required working hours are not enough for the successful completion of an activity. This type of disruption especially happens when the activities are test, measurement or analysis type activities. This type of inadequacy is called disruption, if the additional required working hours are more than nine hours for a resource.

In our implementation routine, we generated this kind of disruption by first listing all the resource-activity couples that are active at the current week and then selecting a random resource-activity couple that needs additional working hours for completion. After that, a random additional working hour requirement is generated. This additional working hour requirement ranges from nine to the estimated total working hour requirement of that resource for that activity. By doing so, we restrict the additional working hours that a resource requires to complete his/her job on an activity with the amount budgeted in the baseline plan. In our implementation routine, this upper bound is just a parameter and can be changed, if needed. After a resource and an activity couple is selected, the remaining working hours of that resource to complete that activity is updated and all the active activities are rescheduled using the updated required working hour information since this additional requirement will change the work schedule of the resource in question and this change can affect all the active activities through precedence relations.

7.2.3. Removal of an Activity

This disruption type rarely happens in the R&D Department under consideration since the activities are defined as a combined activity most of the time. Thus, removal of the whole activity has a small possibility to occur in the current environment. Nevertheless, we consider this type of disruption since it is one of the basic disruptions in project management environments. Additionally, there is a tendency of defining the projects as small as possible comprising smaller activities in the R&D Department under consideration. Thus, in future, the possibility of facing this kind of disruption will increase. The reason of a removal can be two fold. First, it might be realized that execution of a defined activity is not useless anymore, i.e. the activity will not have any contribution to the project. This can happen if the requirement of an activity is subject to the results of a previous activity and the results indicate that there is no need for the activity anymore. Second, it might be decided to outsource the activity. If it is thought that outsourcing is better during the progress of the project, the activity is removed from the project since it does not require any internal resource anymore.

In our implementation routine, to generate this kind of disruption we select one of the activities not yet started from the active activity list and simply remove it from that list. Since this removal can yield incomplete project networks, the project networks are repaired, if necessary. To update the project networks, new precedence relations between the predecessor activities of the removed activity and the successor activities of the removed activity need to be adjusted. In this precedence relation update process, we consider only the precedence relation types between the removed activity and the predecessor activities of the removed activity, i.e., we ignored the precedence relation between the removed activity and successor activities. If the relation between a predecessor activity and the removed activity is of FS type with time lag l, the new relations between all the successor activities and the predecessor activity are of FS type with a lag of l. Similarly, if the relation between a predecessor activity and the removed activity is of SS type with time lag l, the new relations between all the successor activities and the predecessor activity are of SS type with a lag of l. After we update the project networks, since removal of the activity will release some resources for the time periods that they are assigned to the removed activity, we update the availability information for these resources. After that, we reschedule all the active activities since any activity can be affected from this disruption through precedence relations and through the updated resource availability information.

7.2.4. Insertion of an Activity

This disruption type is one the most common disruption types that is encountered in the R&D Department. During the execution of a project, when knowledge and experience about the project environment accumulates, the project manager and the project team may need to define additional activities to achieve the goals of the project. For example, if a new technology becomes available during the execution of the project, and if this technology is highly related to the subject of the project, the project manager and the project team might define a new activity for the analysis and benchmarking of this newly developed technology. The examples for the reasons of an activity insertion can be varied. For example, if the project leader and the project team realizes that they have forgotten to define an activity, they need to insert a new activity to the project network.

In our implementation routine, to generate this kind of disruption, we generate a new activity requiring a random number of resources that exist in the resource pool and add this new activity to the list of active activities. For the new activity, the lower bounds on the required human resources and equipment resources are 1 and 0 and the upper bounds for on the required human resources and equipment resources are 5 and 3, respectively. After obtaining random number of required resources, working hours that the new activity requires for each resource are randomly generated and assigned. These requirements vary from 9 to 21 and from 9 to 46 for the human resources and equipments, respectively. All these lower and upper bounds and the ranges of required working hours are parameters for the implementation routine and can be changed if required. After generating the parameters for the new activity, an active project is selected and this new activity is inserted in a random place in the current project network of the selected project. To insert the new activity, random numbers for the number of predecessors and the number of successors is generated. Note that, the upper bounds on the number of predecessors and the number of successors depends on the randomly selected place that the new activity is inserted to. These numbers can vary from zero to the number of active activities that belongs to the selected project. The predecessors and the successors of the new activity are selected from the candidate active activities of the selected project to be predecessor and successor, respectively and the relation types and the time lags are assigned randomly. Randomly assigned time lags vary from 0 to 4 weeks for each created relation. After we make all the required updates, we reschedule all the active activities.

7.2.5. Replacement of an Activity

This disruption can occur when the work content of a defined activity can be accomplished through a more efficient resource combination. Consider the case that the firm buys a new integrated CNC machine. In this case, if there is an activity whose work content covers the job that this new machine can handle by its own, the same job can be accomplished through different procedures and with less resources. To generate such a disruption in our implementation routine, we change the parameters of a randomly selected active activity. We select a random resource from our resource pool and assign it to the selected activity. With this new resource, some resources will be removed from the required resource list of the selected activity and/or some resources will be able to accomplished their work on that activity in less time. The number of removed and affected resources of the activity is selected randomly and new time requirements are given randomly in the range between zero and the amount planned in the baseline schedule. The availability information for the removed and affected resources is updated and all the active activities are rescheduled after the required updates.

7.2.6. Change in the Activity Parameters

A change in the activity parameters can be a change in the required number of resources to complete the activity, a change in the required resource combinations for an activity or a change in the required hours of the resources to complete the work that they need. The need for these kind of changes during the execution of the projects is common in project management environments. To generate these kind of disruptions, we randomly select an active activity and select a random number of resources to remove and a random number of resources that needs an update of required working hours. We make the changes on these activities and update the availability information of the affected activities if needed. After all the required updates, we reschedule the active activities.

7.3. IMPLEMENTATION

For the implementation of the reactive project scheduling, to show how the scheduled order repair heuristic behaves in case of a disruption for a scheduled project, first, we took the baseline schedule of project 08-024 obtained with single project scheduling approach with fitness calculation procedure1 and we created each type of disruptions in a random time instants. After that, we compared the repaired schedules that the scheduled order repair heuristic presents with the baseline schedule of project 08-024 and each other. Second, we have generated a scenario using the implementation routine. Note that, this scenario is one of the infinitely many scenarios that can happen during project execution and is created just to show how the implementation routine works and reports the results of the changes caused by disruptions. Considering initiation of only 2 project is thought to be enough to show how the implementation routine generates baseline schedules and how they behave in case of disruption, thus the scenario covers a time range such that in this time range 2 R&D projects are initiated and a number of disruptions are occurred. In the following subsections, first, for each type of disruption the results of the scheduled order repair heuristic gives for the baseline schedule of project 08-024 when this type of disruption occurs and the comparison of them between each other and with the baseline schedule are given. Then a sample scenario generated by the implementation routine is presented.

7.3.1. Results of the Scheduled Order Repair Heuristic Applied to a Baseline Project Plan

To show how the scheduled order repair heuristic behaves in case of a disruption for a scheduled project, we selected project 08-024 which is a medium size project as a test project and used the baseline schedule of that project generated with single project scheduling approach with fitness calculation procedure1. The project network of this project is shown in Figure 7.2 .

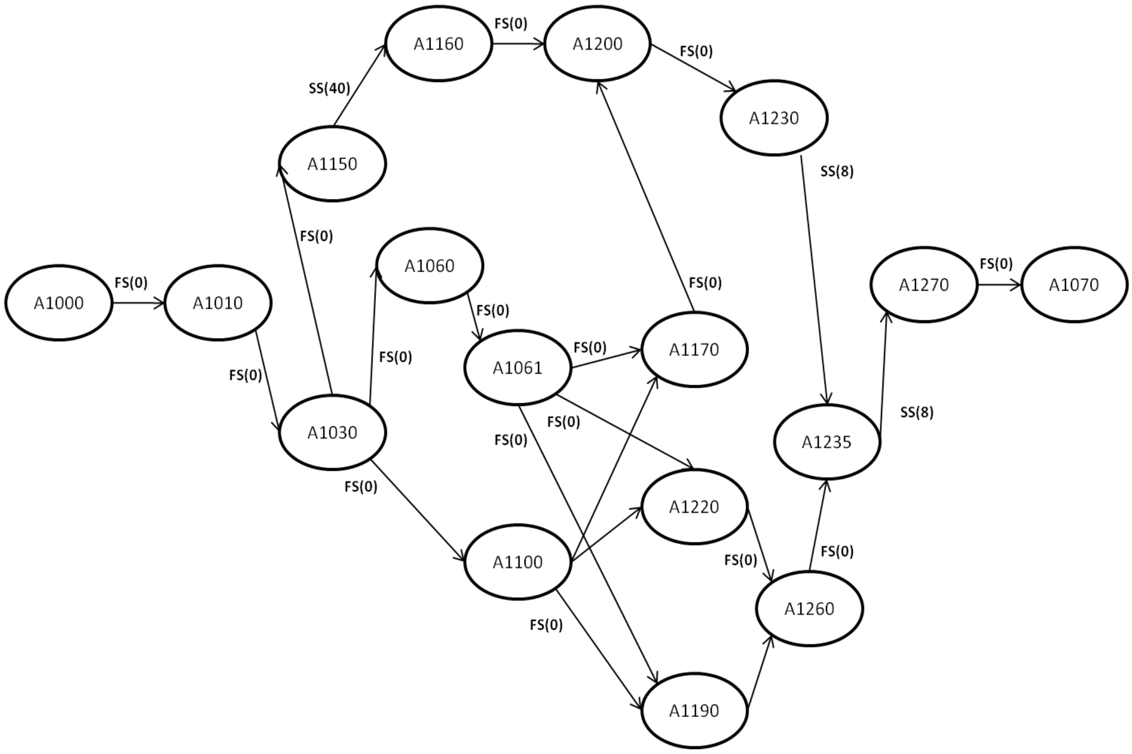

Figure 7. 2. Project Network of the Test Project 08-024

The robust baseline starting and ending times for the activities and for the resources of this project are shown in Table 7.1 and Table 7.2, respectively.

Table 7. 1. Robust Baseline Activity Schedules for Project 08-024

Activity ID	Robust Starting Time	Robust Ending Time	Duration (weeks)
A1000	5/5/2008	8/18/2008	15
A1010	8/18/2008	8/25/2008	1
A1030	8/25/2008	9/8/2008	2
A1150	9/8/2008	12/29/2008	16
A1060	9/8/2008	12/29/2008	16
A1061	1/5/2009	2/9/2009	5
A1100	9/8/2008	2/2/2009	21
A1160	6/15/2009	7/6/2009	3
A1190	2/16/2009	3/9/2009	3
A1220	2/16/2009	4/13/2009	8
A1170	2/16/2009	4/27/2009	10
A1200	11/2/2009	11/30/2009	4
A1230	11/30/2009	12/14/2009	2
A1260	4/13/2009	9/7/2009	21
A1235	1/25/2010	2/8/2010	2
A1270	3/22/2010	4/5/2010	2
A1070	4/12/2010	4/19/2010	1

Table 7. 2. Robust Baseline Activity-Resource Schedules for Project 08-024

Activity ID	Resource ID	Robust Starting Time	Robust Ending Time	Activity ID	Resource ID	Robust Starting Time	Robust Ending Time
A1000	AR001819	05/05/08	06/02/08	A1160	A40C.AFD	06/15/09	07/06/09
A1000	AR002080	05/05/08	08/18/08	A1190	AR110975	02/16/09	03/02/09
A1000	A40C.CFDM	05/05/08	05/19/08	A1190	AR002080	02/16/09	03/09/09
A1000	A90C.W-ID	05/05/08	05/19/08	A1190	AR001819	02/16/09	03/02/09
A1000	A40C.AD	05/05/08	08/11/08	A1190	AR002040	02/16/09	03/02/09
A1000	A40C.CFDA	05/05/08	05/12/08	A1220	AR110975	02/23/09	03/16/09
A1010	AR002080	08/18/08	08/25/08	A1220	AR206522	02/16/09	02/23/09
A1030	AR002080	08/25/08	09/08/08	A1220	AR001296	02/16/09	03/30/09
A1030	AR310113	08/25/08	09/01/08	A1220	AR001813	02/16/09	02/23/09
A1150	AR110963	11/10/08	12/15/08	A1220	AR001819	02/23/09	03/30/09
A1150	AR110975	09/08/08	10/06/08	A1220	AR109545	02/16/09	03/09/09
A1150	AR002081	09/15/08	11/24/08	A1220	AR002080	03/02/09	03/30/09
A1150	AR002080	09/08/08	10/13/08	A1220	A40C.AFD	02/16/09	03/30/09
A1150	AR001819	09/08/08	10/06/08	A1220	A45C.SAR_x	02/16/09	02/23/09
A1150	AR109545	09/08/08	12/15/08	A1220	A40C.PIV	02/16/09	02/23/09
A1150	A90C.W-ID	09/08/08	09/22/08	A1220	A40C.LDA	04/06/09	04/13/09
A1150	A40C.AFD	09/08/08	10/13/08	A1220	A44C.AC1	02/16/09	02/23/09
A1150	A40C.CFDA	09/08/08	09/15/08	A1220	A44C.AC2	02/16/09	02/23/09
A1150	A40C.CFDM	09/08/08	09/15/08	A1220	A44C.W1	02/16/09	02/23/09
A1150	A40C.PIV	09/08/08	09/15/08	A1220	A44C.W2	02/16/09	02/23/09
A1150	A40C.LDA	12/01/08	12/29/08	A1220	A45C.PU1	02/16/09	02/23/09
A1060	AR110975	09/29/08	10/06/08	A1170	AR110975	03/09/09	03/23/09
A1060	AR002080	10/06/08	11/03/08	A1170	AR002080	03/23/09	04/20/09
A1060	AR001819	09/29/08	10/27/08	A1170	AR001819	03/23/09	04/13/09
A1060	AR109545	12/08/08	12/29/08	A1170	AR109545	03/02/09	03/23/09
A1060	A40C.AD	09/08/08	12/01/08	A1170	A40C.CFDA	02/16/09	02/23/09
A1061	AR110975	01/05/09	01/12/09	A1170	A40C.CFDM	02/16/09	03/02/09
A1061	AR002080	01/05/09	02/02/09	A1170	A40C.PIV	02/16/09	03/02/09
A1061	AR001819	01/05/09	02/02/09	A1170	A40C.AFD	03/23/09	04/27/09
A1061	AR109545	01/05/09	01/26/09	A1200	AR002080	11/02/09	11/30/09
A1061	A40C.AD	01/05/09	02/09/09	A1200	A40C.CFDA	11/02/09	11/09/09
A1100	AR110975	09/29/08	10/13/08	A1200	A40C.CFDM	11/02/09	11/09/09
A1100	AR002080	10/27/08	11/17/08	A1230	AR001819	11/30/09	12/14/09
A1100	AR001296	09/08/08	09/22/08	A1230	AR002081	11/30/09	12/14/09
A1100	AR001819	10/20/08	12/01/08	A1230	AR002080	11/30/09	12/14/09
A1100	AR001813	09/29/08	01/05/09	A1230	A40C.CFDM	11/30/09	12/07/09
A1100	AR109545	12/22/08	02/02/09	A1230	A90C.W-ID	11/30/09	12/14/09
A1100	A44C.AC1	09/08/08	09/15/08	A1260	AR002080	04/13/09	05/04/09
A1100	A44C.AC2	09/08/08	09/15/08	A1260	AR001819	04/13/09	04/20/09
A1100	A44C.W1	09/08/08	09/29/08	A1260	AR109545	04/13/09	04/27/09
A1100	A44C.W2	09/08/08	10/06/08	A1260	A40C.AFD	04/20/09	09/07/09
A1100	A45C.PU1	09/08/08	09/15/08	A1235	AR206522	01/25/10	02/01/10
A1100	A40C.AD	11/24/08	01/05/09	A1235	AR002080	01/25/10	02/08/10
A1100	A45C.SAR_x	09/08/08	09/15/08	A1235	A44C.AC1	01/25/10	02/01/10
A1160	AR110975	06/15/09	06/22/09	A1235	A44C.AC2	01/25/10	02/01/10

A1160	AR002080	06/15/09	06/22/09	A1235	A44C.W1	01/25/10	02/01/10
A1160	AR001819	06/15/09	06/22/09	A1235	A44C.W2	01/25/10	02/01/10
A1160	AR109545	06/15/09	06/29/09	A1270	AR110975	03/22/10	04/05/10
A1160	A40C.CFDA	06/15/09	06/22/09	A1270	AR002080	03/22/10	04/05/10
A1160	A40C.CFDM	06/15/09	06/22/09	A1270	A40C.CFDA	03/22/10	03/29/10
A1160	A40C.PIV	06/15/09	06/22/09	A1270	A40C.CFDM	03/22/10	04/05/10
				A1070	AR002080	04/12/10	04/19/10

To show the behavior of the scheduled order repair heuristic it is assumed that a disruption occurs at a random week during the execution of project 08-024.

7.3.1.1. Schedule Change When an Unexpected Resource Unavailability Occurs

Assume that a disruption of type 1 occurs at time 10/13/08. Resource A40C.AD is unexpectedly unavailable for 33 hours at the week starting in 10/13/08. While Table 7.3 and Table 7.4 shows the affected activities and how their schedule is changed for the repaired schedule1 and for the repaired schedule2, respectively, Table 7.5 gives the schedule change statistics for each repaired schedule.

Table 7. 3. Affected Activity Schedules Obtained in Repaired Schedule1 after Type 1 Disruption

	Before Disruption			After Disruption			Delays	
Activity ID	Starting Time	Ending Time	Duration (weeks)	Starting Time	Ending Time	Duration (weeks)	Start Delay (weeks)	End Delay (weeks)
A1100	09/08/08	02/02/09	21	09/08/08	02/16/09	23	0	2
A1160	06/15/09	07/06/09	3	07/20/09	08/17/09	4	5	6
A1200	11/02/09	11/30/09	4	12/07/09	12/21/09	2	5	3
A1230	11/30/09	12/14/09	2	12/21/09	01/18/10	4	3	5
A1235	01/25/10	02/08/10	2	02/15/10	03/01/10	2	3	3
A1270	03/22/10	04/05/10	2	04/12/10	04/26/10	2	3	3
A1070	04/12/10	04/19/10	1	04/26/10	05/03/10	1	2	2

Table 7. 4. Affected Activity Schedules Obtained in Repaired Schedule2 after Type 1 Disruption

	Before Disruption			After Disruption			Delays	
Activity ID	**Starting Time**	**Ending Time**	**Duration (weeks)**	**Starting Time**	**Ending Time**	**Duration (weeks)**	**Start Delay (weeks)**	**End Delay (weeks)**
A1150	09/08/08	12/29/08	16	10/13/08	12/29/08	11	5	0
A1060	09/08/08	12/29/08	16	10/13/08	12/29/08	11	5	0
A1061	01/05/09	02/09/09	5	12/29/08	02/09/09	6	-1	0
A1100	09/08/08	02/02/09	21	10/20/08	02/09/09	16	6	1
A1160	06/15/09	07/06/09	3	07/20/09	08/17/09	4	5	6
A1190	02/16/09	03/09/09	3	02/09/09	03/02/09	3	-1	-1
A1220	02/16/09	04/13/09	8	02/09/09	04/13/09	9	-1	0
A1170	02/16/09	04/27/09	10	02/09/09	04/27/09	11	-1	0
A1200	11/02/09	11/30/09	4	12/07/09	12/21/09	2	5	3
A1230	11/30/09	12/14/09	2	12/21/09	01/18/10	4	3	5
A1260	04/13/09	09/07/09	21	04/13/09	08/31/09	20	0	-1
A1235	01/25/10	02/08/10	2	02/15/10	03/01/10	2	3	3
A1270	03/22/10	04/05/10	2	04/12/10	04/26/10	2	3	3
A1070	04/12/10	04/19/10	1	04/26/10	05/03/10	1	2	2

Table 7. 5. Schedule Change Statistics after Type 1 Disruption Occurs

	Repaired Schedule1	**Repaired Schedule2**
Affected Activity Count	7	14
Affected Activity-Resource Pairs	30	67
Delay of the Project Completion Time (weeks)	2	2
Active Activity Count	14	
Active Activity-Resource Couple Count	78	

Table 7.5 shows that when repaired schedule1 is adopted, 7 of 14 active activities and 30 of 78 active activity-resource couples will be affected and if repaired schedule2 is adopted, 14 of 14 active activities and 67 of 78 active activity-resource couples will be affected from this disruption. It is also seen from Table 7.5 that whichever repaired schedule is adopted, the increase in the completion time of project 08-024 will be two weeks later than the completion time in the baseline plan.

7.3.1.2. Schedule Change When a Resource Needs Additional Time to Complete the Job on an Activity

Assume that type 2 disruption occurs at time 10/6/2008. Resource AR002080 says that s/he needs additional 26 hours to complete the activity A1060 s/he is working at the week starting in 10/6/2008. His/her remaining working hour on activity A1060 at time 10/6/2008 is increased to 84 from 58. Table 7.6 and Table 7.7 show the affected activities and how their schedule is changed for the repaired schedule1 and for the repaired schedule2, respectively. Table 7.8 gives the schedule change statistics for each repaired schedule.

Table 7. 6. Affected Activity Schedules Obtained in Repaired Schedule1 after Type 2 Disruption

	Before Disruption			After Disruption			Delays	
Activity ID	Starting Time	Ending Time	Duration (weeks)	Starting Time	Ending Time	Duration (weeks)	Start Delay (weeks)	End Delay (weeks)
A1160	06/15/09	07/06/09	3	07/13/09	08/03/09	3	4	4
A1200	11/02/09	11/30/09	4	11/30/09	12/14/09	2	4	2
A1230	11/30/09	12/14/09	2	12/14/09	01/04/10	3	2	3
A1235	01/25/10	02/08/10	2	02/08/10	02/22/10	2	2	2
A1270	03/22/10	04/05/10	2	04/05/10	04/19/10	2	2	2
A1070	04/12/10	04/19/10	1	04/19/10	04/26/10	1	1	1

Table 7. 7. Affected Activity Schedules Obtained in Repaired Schedule2 after Type 2 Disruption

	Before Disruption			After Disruption			Delays	
Activity ID	Starting Time	Ending Time	Duration (weeks)	Starting Time	Ending Time	Duration (weeks)	Start Delay (weeks)	End Delay (weeks)
A1150	09/08/08	12/29/08	16	10/06/08	12/29/08	12	4	0
A1060	09/08/08	12/29/08	16	10/06/08	12/29/08	12	4	0
A1061	01/05/09	02/09/09	5	12/29/08	02/09/09	6	-1	0
A1100	09/08/08	02/02/09	21	10/06/08	02/09/09	18	4	1
A1160	06/15/09	07/06/09	3	07/13/09	08/03/09	3	4	4
A1190	02/16/09	03/09/09	3	02/09/09	03/02/09	3	-1	-1
A1220	02/16/09	04/13/09	8	02/09/09	04/13/09	9	-1	0
A1170	02/16/09	04/27/09	10	02/09/09	04/27/09	11	-1	0
A1200	11/02/09	11/30/09	4	11/30/09	12/14/09	2	4	2
A1230	11/30/09	12/14/09	2	12/14/09	01/04/10	3	2	3
A1260	04/13/09	09/07/09	21	04/13/09	08/31/09	20	0	-1
A1235	01/25/10	02/08/10	2	02/08/10	02/22/10	2	2	2
A1270	03/22/10	04/05/10	2	04/05/10	04/19/10	2	2	2
A1070	04/12/10	04/19/10	1	04/19/10	04/26/10	1	1	1

Table 7. 8. Schedule Change Statistics after Type 2 Disruption Occurs

	Repaired Schedule1	Repaired Schedule2
Affected Activity Count	6	14
Affected Activity-Resource Pairs	30	71
Delay of the Project Completion Time (weeks)	1	1
Active Activity Count	14	
Active Activity-Resource Couple Count	81	

Table 7.8 shows that when repaired schedule1 is adopted, 6 of 14 active activities and 30 of 81 active activity-resource couples will be affected and if repaired schedule2 is adopted, 14 of 14 active activities and 71 of 81 active activity-resource couples will be affected from this disruption. It is also seen from Table 7.8 that whichever repaired schedule is adopted, the increase in the completion time of project 08-024 will be one week later than the completion time in the baseline plan.

7.3.1.3. Schedule Change When an Activity is Removed from The Project Network

Assume that type 3 disruption of type 3 occurs at time 8/25/2008. Activity A1030 is decided to be removed from the project at the week starting in 8/25/2008. Table 7.9 and Table 7.10 show the affected activities and how their schedule is changed for the repaired schedule1 and for the repaired schedule2, respectively. Table 7.11 gives the schedule change statistics for each repaired schedule.

Table 7. 9. Affected Activity Schedules Obtained in Repaired Schedule1 after Type 3 Disruption

	Before Disruption			After Disruption			Delays	
Activity ID	**Starting Time**	**Ending Time**	**Duration (weeks)**	**Starting Time**	**Ending Time**	**Duration (weeks)**	**Start Delay (weeks)**	**End Delay (weeks)**
A1150	09/08/08	12/29/08	16	08/25/08	12/29/08	18	-2	0
A1060	09/08/08	12/29/08	16	08/25/08	12/22/08	17	-2	-1
A1061	01/05/09	02/09/09	5	12/22/08	02/02/09	6	-2	-1
A1100	09/08/08	02/02/09	21	08/25/08	02/02/09	23	-2	0
A1160	06/15/09	07/06/09	3	06/01/09	06/22/09	3	-2	-2
A1190	02/16/09	03/09/09	3	02/02/09	02/23/09	3	-2	-2
A1220	02/16/09	04/13/09	8	02/02/09	04/13/09	10	-2	0
A1170	02/16/09	04/27/09	10	02/02/09	04/20/09	11	-2	-1
A1200	11/02/09	11/30/09	4	10/19/09	11/30/09	6	-2	0
A1260	04/13/09	09/07/09	21	04/13/09	08/31/09	20	0	-1
A1070	04/12/10	04/19/10	1	04/05/10	04/12/10	1	-1	-1

Table 7. 10. Affected Activity Schedules Obtained in Repaired Schedule2 after Type 3 Disruption

Activity ID	Starting Time	Ending Time	Duration (weeks)	Starting Time	Ending Time	Duration (weeks)	Start Delay (weeks)	End Delay (weeks)
A1000	05/05/08	08/18/08	15	07/07/08	08/18/08	6	9	0
A1150	09/08/08	12/29/08	16	08/25/08	12/29/08	18	-2	0
A1060	09/08/08	12/29/08	16	08/25/08	12/22/08	17	-2	-1
A1061	01/05/09	02/09/09	5	12/22/08	02/02/09	6	-2	-1
A1100	09/08/08	02/02/09	21	08/25/08	02/02/09	23	-2	0
A1160	06/15/09	07/06/09	3	06/01/09	06/22/09	3	-2	-2
A1190	02/16/09	03/09/09	3	02/02/09	02/23/09	3	-2	-2
A1220	02/16/09	04/13/09	8	02/02/09	04/13/09	10	-2	0
A1170	02/16/09	04/27/09	10	02/02/09	04/20/09	11	-2	-1
A1200	11/02/09	11/30/09	4	10/19/09	11/30/09	6	-2	0
A1260	04/13/09	09/07/09	21	04/13/09	08/31/09	20	0	-1
A1070	04/12/10	04/19/10	1	04/05/10	04/12/10	1	-1	-1

Table 7. 11. Schedule Change Statistics after Type 3 Disruption Occurs

	Repaired Schedule1	Repaired Schedule2
Affected Activity Count	11	12
Affected Activity-Resource Pairs	72	75
Delay of the Project Completion Time (weeks)	-1	-1
Active Activity Count	16	
Active Activity-Resource Couple Count	98	

Table 7.11 shows that when repaired schedule1 is adopted, 11 of 16 active activities and 72 of 98 active activity-resource couples will be affected and if repaired schedule2 is adopted, 12 of 14 active activities and 75 of 98 active activity-resource couples will be affected from this disruption. It is also seen from Table 7.11 that whichever repaired schedule is adopted, the increase in the completion time of project 08-024 will be one week earlier than the completion time in the baseline plan.

7.3.1.4. Schedule Change When a New Activity is Inserted to the Project Network

Assume that a disruption of type 4 occurs at time 8/25/2008. A new activity with a name 08-024X1 is defined for successful completion of the project. This new activity requires a total of 17 hours from resource AR903360X and a total of 29 hours from resource A41C.FTC. The only predecessor activity is A1030 with a relation type SS and with a lag of four weeks. The successors of these new activity are A1150, A1160, A1190, A1220, A1260 and A1070 with relation type SS, SS, FS, SS, FS and SS and with time lags 4, 1, 2, 3, 4, and 2, respectively. Table 7.12 and Table 7.13 show the affected activities and how their schedule is changed for the repaired schedule1 and for the repaired schedule2, respectively. Table 7.14 gives the schedule change statistics for each repaired schedule.

Table 7. 12. Affected Activity Schedules Obtained in Repaired Schedule1 after Type 4 Disruption

	Before Disruption			After Disruption			Delays	
Activity ID	**Starting Time**	**Ending Time**	**Duration (weeks)**	**Starting Time**	**Ending Time**	**Duration (weeks)**	**Start Delay (weeks)**	**End Delay (weeks)**
A1150	09/08/08	12/29/08	16	10/20/08	12/29/08	10	6	0
A1160	06/15/09	07/06/09	3	07/27/09	08/24/09	4	6	7
A1200	11/02/09	11/30/09	4	12/14/09	01/04/10	3	6	5
A1230	11/30/09	12/14/09	2	01/04/10	02/01/10	4	5	7
A1235	01/25/10	02/08/10	2	03/01/10	03/15/10	2	5	5
A1270	03/22/10	04/05/10	2	04/26/10	05/10/10	2	5	5
A1070	04/12/10	04/19/10	1	05/10/10	05/17/10	1	4	4

Table 7. 13. Affected Activity Schedules Obtained in Repaired Schedule2 after Type 4 Disruption

	Before Disruption			After Disruption			Delays	
Activity ID	Starting Time	Ending Time	Duration (weeks)	Starting Time	Ending Time	Duration (weeks)	Start Delay (weeks)	End Delay (weeks)
A1150	09/08/08	12/29/08	16	10/20/08	12/29/08	10	6	0
A1061	01/05/09	02/09/09	5	12/29/08	02/09/09	6	-1	0
A1100	09/08/08	02/02/09	21	09/08/08	02/09/09	22	0	1
A1160	06/15/09	07/06/09	3	07/27/09	08/24/09	4	6	7
A1190	02/16/09	03/09/09	3	02/09/09	03/02/09	3	-1	-1
A1220	02/16/09	04/13/09	8	02/09/09	04/13/09	9	-1	0
A1170	02/16/09	04/27/09	10	02/09/09	04/27/09	11	-1	0
A1200	11/02/09	11/30/09	4	12/14/09	01/04/10	3	6	5
A1230	11/30/09	12/14/09	2	01/04/10	02/01/10	4	5	7
A1260	04/13/09	09/07/09	21	04/13/09	08/31/09	20	0	-1
A1235	01/25/10	02/08/10	2	03/01/10	03/15/10	2	5	5
A1270	03/22/10	04/05/10	2	04/26/10	05/10/10	2	5	5
A1070	04/12/10	04/19/10	1	05/10/10	05/17/10	1	4	4

Table 7. 14. Schedule Change Statistics after Type 4 Disruption Occurs

	Repaired Schedule1	Repaired Schedule2
Affected Activity Count	7	13
Affected Activity-Resource Pairs	44	77
Delay of the Project Completion Time (weeks)	4	3
Active Activity Count	16	
Active Activity-Resource Couple Count	98	

Table 7.14 shows that when repaired schedule1 is adopted, 7 of 16 active activities and 44 of 98 active activity-resource couples will be affected and if repaired schedule2 is adopted, 13 of 16 active activities and 77 of 98 active activity-resource couples will be affected from this disruption. It is also seen from Table 7.14 that if repaired schedule2 is adopted, the delay in the expected project completion time is one week earlier than the completion time in the baseline plan..

7.3.1.5. Schedule Change When an Activity is Replaced with a more Efficient One

Assume that a disruption of type 5 occurs at time 4/13/2009. It is realized that now it is possible to make activity A1260 in a more efficient way with a new resource A41C.FTIR instead of A40C.AFD. Resource A41C.FTIR is capable of doing the same job in 248 hours while A40C.AFD was doing it in 800 hours. While Table 7.15 and Table 7.16 shows the affected activities and how their schedule is changed for the repaired schedule1 and for the repaired schedule2, respectively, Table 7.17 gives the schedule change statistics for each repaired schedule.

Table 7. 15. Affected Activity Schedules Obtained in Repaired Schedule1 after Type 5 Disruption

	Before Disruption			After Disruption			Delays	
Activity ID	Starting Time	Ending Time	Duration (weeks)	Starting Time	Ending Time	Duration (weeks)	Start Delay (weeks)	End Delay (weeks)
A1160	06/15/09	07/06/09	3	06/15/09	06/29/09	2	0	-1
A1170	02/16/09	04/27/09	10	02/16/09	03/16/09	4	0	-6
A1200	11/02/09	11/30/09	4	11/02/09	11/16/09	2	0	-2
A1230	11/30/09	12/14/09	2	11/30/09	12/07/09	1	0	-1
A1260	04/13/09	09/07/09	21	04/13/09	02/01/10	42	0	21
A1235	01/25/10	02/08/10	2	02/01/10	02/08/10	1	1	0
A1270	03/22/10	04/05/10	2	03/29/10	04/05/10	1	1	0

Table 7. 16. Affected Activity Schedules Obtained in Repaired Schedule2 after Type 5 Disruption

	Before Disruption			After Disruption			Delays	
Activity ID	Starting Time	Ending Time	Duration (weeks)	Starting Time	Ending Time	Duration (weeks)	Start Delay (weeks)	End Delay (weeks)
A1160	06/15/09	07/06/09	3	04/13/09	04/27/09	2	-9	-10
A1170	02/16/09	04/27/09	10	04/13/09	05/04/09	3	8	1
A1200	11/02/09	11/30/09	4	08/31/09	09/21/09	3	-9	-10
A1230	11/30/09	12/14/09	2	09/21/09	11/09/09	7	-10	-5
A1260	04/13/09	09/07/09	21	04/13/09	02/01/10	42	0	21
A1235	01/25/10	02/08/10	2	02/01/10	02/08/10	1	1	0
A1270	03/22/10	04/05/10	2	03/29/10	04/05/10	1	1	0
A1070	04/12/10	04/19/10	1	04/05/10	04/12/10	1	-1	-1

Table 7. 17. Schedule Change Statistics after Type 5 Disruption Occurs

	Repaired Schedule1	Repaired Schedule2
Affected Activity Count	7	8
Affected Activity-Resource Pairs	22	31
Delay of the Project Completion Time (weeks)	1	-1
Active Activity Count	8	
Active Activity-Resource Couple Count	34	

Table 7.17 shows that when repaired schedule1 is adopted, 7 of 8 active activities and 22 of 34 active activity-resource couples will be affected and if repaired schedule2 is adopted, 8 of 8 active activities and 31 of 34 active activity-resource couples will be affected from this disruption. It is also seen from Table 7.17 that if repaired schedule2 is adopted, the expected project completion time is one week earlier while if repaired schedule1 is adopted, the delay in the expected completion time is one week.

7.3.1.6. Schedule Change When Some Parameters of an Activity Changes

Assume that a disruption of type 6 occurs at time 8/11/2008. Now The new requirements for resources AR002080, AR110975, AR109545, AR001813, A45C.PU1, A44C.AC1, A44C.AC2, A45C.SAR_x, A44C.W2, A40C.LDA, A44C.W1, A40C.PIV, and A40C.AFD are changed to 56, 1, 60, 11, 32, 10, 13, 6, 12, 13, 10, 28, 18, from 97, 32, 78, 21, 50, 25, 25, 50, 25, 25, 25, 40 and 200, respectively. Table 7.18 and Table 7.19 shows the affected activities and how their schedule is changed for the repaired schedule1 and for the repaired schedule2, respectively. Table 7.20 gives the schedule change statistics for each repaired schedule.

Table 7. 18. Affected Activity Schedules Obtained in Repaired Schedule1 after Type 6 Disruption

	Before Disruption			After Disruption			Delays	
Activity ID	Starting Time	Ending Time	Duration (weeks)	Starting Time	Ending Time	Duration (weeks)	Start Delay (weeks)	End Delay (weeks)
A1170	02/16/09	04/27/09	10	02/16/09	04/13/09	8	0	-2
A1260	04/13/09	09/07/09	21	04/13/09	08/24/09	19	0	-2

Table 7. 19. Affected Activity Schedules Obtained in Repaired Schedule2 after Type 6 Disruption

	Before Disruption			After Disruption			Delays	
Activity ID	Starting Time	Ending Time	Duratio n (weeks)	Starting Time	Ending Time	Duratio n (weeks)	Start Delay (weeks)	End Delay (weeks)
A1000	05/05/08	08/18/08	15	08/11/08	08/18/08	1	14	0
A1061	01/05/09	02/09/09	5	12/29/08	02/09/09	6	-1	0
A1100	09/08/08	02/02/09	21	09/08/08	02/09/09	22	0	1
A1190	02/16/09	03/09/09	3	02/09/09	03/02/09	3	-1	-1
A1220	02/16/09	04/13/09	8	02/09/09	04/13/09	9	-1	0
A1170	02/16/09	04/27/09	10	02/09/09	03/30/09	7	-1	-4
A1260	04/13/09	09/07/09	21	04/13/09	08/24/09	19	0	-2
A1070	04/12/10	04/19/10	1	04/05/10	04/12/10	1	-1	-1

Table 7. 20. Schedule Change Statistics after Type 6 Disruption Occurs

	Repaired Schedule1	Repaired Schedule2
Affected Activity Count	2	8
Affected Activity-Resource Pairs	8	31
Delay of the Project Completion Time (weeks)	0	-1
Active Activity Count	17	
Active Activity-Resource Couple Count	99	

Table 7.20 shows that when repaired schedule1 is adopted, 2 of 17 active activities and 8 of 99 active activity-resource couples will be affected and if repaired schedule2 is adopted, 8 of 17 active activities and 31 of 99 active activity-resource couples will be affected from this disruption. It is also seen from Table 7.20 that if repaired schedule2 is adopted, the expected project completion time is one week earlier than the completion time in the baseline schedule. On the other hand, if repaired schedule1 is adopted, the expected completion time for project 08-024 remains the same.

7.3.2. Results Obtained with the Implementation Routine Employing the Single Project Scheduling Approach

In this subsection, to create a project progress scenario, we have used our implementation routine with the single project scheduling approach. This implementation routine is started with the initiation of project 08-024, the first project in our project set and ended when project 08-031, the second project in our project set, is completed. It is assumed that until project 08-031 is completed, no new project is initiated. Disruption threshold value *p_threshold* is taken as 0.10 for each time instant and repaired schedule1 is adopted for the fixed schedule. This routine covers generating baseline schedules for projects 08-024 and 08-031 and random disruptions that may affect these projects. A possible scenario with possible events that can occur during project execution obtained with the implementation routine that uses single project scheduling approach to generate baseline schedule for a newly arrived project is shown in the following tables. Table 7.21 shows the events that happened during the implementation with the affected projects, number of affected activities and with the number of affected resource-activity couples and table 7.22 shows the detailed explanation of events.

Table 7. 21. A Possible Project Execution Scenario Obtained with Implementation Routine

Event No	Event Occurence Time	Event Type	Affected Projects	Total Number of Active Activities	Total Number of Affected Activities	Total Number of Affected Activity-Resource Couple
1	05/05/08	Project Initiation	-			
2	05/19/08	Project Initiation	-			
3	09/08/08	Type 4 Disruption	0	29	0	2
4	12/08/08	Type 6 Disruption	08-024, 08-031	24	17	73
5	02/02/09	Type 2 Disruption	08-031	22	10	43
6	8/24709	Type 3 Disruption	08-031	12	3	12
7	10/12/09	Type 5 Disruption	-	9	0	3
8	11/23/09	Type 5 Disruption	-	9	0	2
9	12/14/09	Type 3 Disruption	08-024, 08-031	5	2	9
10	12/28/09	Project Completion	-			
11	03/01/10	Type 5 Disruption	-	3	0	2
12	03/08/10	Type 1 Disruption	08-024	3	2	6
13	07/05/10	Project Completion	-			

Table 7. 22. Event Explanations of the Possible Project Execution Scenario

Event No	Explanation
1	Project 08-024 is initiated
2	Project 08-031 is initiated
3	A new activity with a name08-031X1 is added to project 08-031. This new activity requires a total of 15 hours from resource AAR0011746X and a total of 40 hours from resource A45C.CIN_X. There is no predecessor activity. Successor activities for this activity with relation type and required lag are : (08-031A1180,FS,1),(08-031A1150,SS,3),(08-031A1240,SS,4)
4	Activity 08-024A1220 does not need resource A40C.AFD anymore. Additionally, The new requirements for resources AR001813, AR002080, AR001296, AR110975, AR001819, AR206522, A40C.LDA, A44C.AC2, A44C.W2, A40C.PIV, A45C.SAR_x, A44C.AC1, A45C.PU1 are changed to 8, 52, 16, 14, 20, 7, 1, 4, 15, 27, 12, 14, 18, from 18, 83, 41, 27, 41, 23, 25, 25, 25, 40, 50, 25, 50.
5	Resource AR109545 says he/she needs additional 33 working hours to complete his/her job on activity 08-031A1090.
6	Activity 08-031A1240 is removed from the project network
7	Activity 08-024A1235 is more efficient with resource A44C.CA3 instead of resource A44C.AC1. Resource A44C.CA3 is capable of doing the same job in 21 hours while A44C.AC1 was doing it in 25 hours Now there is no need to use resources AR206522, A44C.AC1
8	Activity 08-024A1200 is more efficient with resource A45C.GDA instead of resource A40C.CFDM Resource A45C.GDA is capable of doing the same job in 62 hours while A40C.CFDM was doing it in 70 hours.
9	Activity 08-024A1270 is removed from the project network
10	Project 08-031 is completed
11	Activity 08-024A1230 is more efficient with resource A44C.FR2 instead of resource A90C.W-ID resource A44C.FR2 is capable of doing the same job in 144 hours while A90C.W-ID was doing it in 180 hours
12	Resource AR002080's availability at week 3/8/2010 is decreased to 19 from 38
13	Project 08-024 is completed

In this chapter, we have presented the implementation results of Phase III of our three-phase approach for robust R&D project scheduling. First, we have created each of the disruptions that we have considered for a baseline project plan, then we used the scheduled order repair heuristic to fix the baseline plans when the disruption occurs. Then, we have compared the resulting two repaired schedules that our implementation routine yields. Finally, we have presented a possible project execution scenario consisting of project initiations, disruption occurrences and project completions.

CHAPTER 8

CONCLUSION AND FUTURE WORK

In this thesis, we have proposed a three-phase approach to obtain robust project schedules for R&D projects. In Phase I, we have developed an activity deviation prediction procedure to be used when obtaining non-dominated robust baseline schedules in Phase II. To obtain robust baseline schedules, in Phase II, we suggested two scheduling approaches each using a bi-objective GA with two different fitness calculation procedures. Solution robustness is assured with solving a TSAD minimization model after a pre-specified number of schedule realizations are obtained for a chromosome of the bi-objective GA. The other objective is the minimization of the project completion time over all projects. Although we have used these two objectives, some other objectives could be used or added to the model. Furthermore, in Phase III, we have suggested a reactive project scheduling approach using scheduled order repair heuristic. This heuristic gives two alternative repaired schedules. Whereas in the first repaired schedule keeps the buffers to be used in a future disruption in the case that the baseline schedule is able to absorb the current disruption, the second repaired schedule makes use of the buffers immediately to absorb the disruption. The proposed three-phase approach is implemented on the real data from the R&D Department of a leading home appliances company in Turkey. We could not produce any results for comparisons between the project schedules of the suggested approaches with the actual project plans of the R&D Department. The reason for this is that the way the history of the data has been kept makes it impossible to achieve a comparison on the same basis, however, the results of the suggested approaches show that the proposed model is effective in producing robust non-dominated project schedules.

To the extent of our knowledge, this study is the first study considering multiple objectives on proactive-reactive project scheduling literature for the problem of the preemptive version of the RCMPSP with generalized precedence relations. However, the study can be improved further. First of all, within the scope of this study, we have implemented Phase I of the three-phase approach with a small data set. To obtain a more reliable project classification and activity deviation prediction model, the data set should be enlarged. Another suggestion for future study is the testing of different crossover operators, mutation operators and selection mechanisms in the fine-tuning procedure of the bi-objective GA parameters. Since including them as parameters increases the number of parameter combinations tested to a very huge numbers, we selected them based on the literature. Although we have presented the best GA parameter combinations, we have employed in the implementation of the proactive and reactive project scheduling phases with real data a different set of parameters than the ones obtained with the fine-tuning procedure. Choosing the best parameter set and applying it using real data will give us the opportunity to compare the contribution of the fine-tuning procedure of GA parameter selection.

An extension of our work could be considering the concepts of activity flexibility, project flexibility, activity priority and project priority while scheduling the projects. Besides these, we are planning to work on obtaining better repaired schedules when a disruption occurs. For this, an improvement search could be applied to the presented two repaired schedules. Moreover, in addition to the scheduled order repair heuristic, some other activity list based heuristics, such as earliest baseline activity starting time (EBST), earliest projected starting time (EPST) or largest activity weight (LW), could be applied to obtain repaired schedules in the reactive phase of the proposed approach.

We hope this initial study on proactive-reactive project scheduling under uncertainty considering multiple objectives would motivate researchers to this intriguing area of research.

BIBLIOGRAPHY

Abbasi, B., Shadrokh, S., Arkat, J. (2006). Bi-objective resource-constrained project scheduling with robustness and makespan criteria. *Applied mathematics and computation*, *180*(1), 146-152.

Aha, D. W., Kibler, D., Albert, M. K. (1991). Instance-based learning algorithms. *Machine learning*, *6*(1), 37-66.

Alba, E., Francisco Chicano, J. (2007). Software project management with GAs. *Information Sciences*, *177*(11), 2380-2401.

Alcaraz, J., Maroto, C. (2001). A robust genetic algorithm for resource allocation in project scheduling. *Annals of Operations Research*, *102*(1), 83-109.

Alcaraz, J., Maroto, C., Ruiz, R. (2003). Solving the multi-mode resource-constrained project scheduling problem with genetic algorithms. *Journal of the Operational Research Society*, *54*(6), 614-626.

Al-Fawzan, M. A., Haouari, M. (2005). A bi-objective model for robust resource-constrained project scheduling. *International Journal of Production Economics*, *96*(2), 175-187.

Artigues, C., Roubellat, F. (2000). A polynomial activity insertion algorithm in a multi-resource schedule with cumulative constraints and multiple modes. *European Journal of Operational Research*, *127*(2), 297-316.

Ashtiani, B., Leus, R., Aryanezhad, M. B. (2011). New competitive results for the stochastic resource-constrained project scheduling problem: exploring the benefits of pre-processing. *Journal of Scheduling*, *14*(2), 157-171.

Azaron, A., Perkgoz, C., Sakawa, M. (2005). A genetic algorithm approach for the time-cost trade-off in PERT networks. *Applied Mathematics and Computation*, *168*(2), 1317-1339.

Babu, B. V., Jehan, M. M. L. (2003, December). Differential evolution for multi-objective optimization. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on* (Vol. 4, pp. 2696-2703). IEEE.

Baker, N. R. (1974). R D project selection models: an assessment. *Engineering Management, IEEE Transactions on*, (4), 165-171.

Ballestin, F., Blanco, R. (2011). Theoretical and practical fundamentals for multi-objective optimization in resource-constrained project scheduling problems. *Computers Operations Research*, *38*(1), 51-62.

Ballestin, F., Leus, R. (2009). Resource-Constrained Project Scheduling for Timely Project Completion with Stochastic Activity Durations. *Production and Operations Management*, *18*(4), 459-474.

Ballou, D. P., Tayi, G. K. (1996). A decision aid for the selection and scheduling of software maintenance projects. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, *26*(2), 203-212.

Bartels, J. H., Zimmermann, J. (2009). Scheduling tests in automotive R&D projects. *European Journal of Operational Research*, *193*(3), 805-819.

Bassett, M. (2000). Assigning projects to optimize the utilization of employees' time and expertise. *Computers Chemical Engineering*, *24*(2), 1013-1021.

Bellenguez O, Néron E (2004) Methods for solving the multi-skill project scheduling problem. Ninth international workshop on project management and scheduling, Nancy, pp 66–69.

Bellenguez, O., Néron, E. (2005). Lower bounds for the multi-skill project scheduling problem with hierarchical levels of skills. *Practice and Theory of Automated Timetabling V*, 229-243.

Bellenguez-Morineau, O. (2008). Methods to solve multi-skill project scheduling problem. *4OR: A Quarterly Journal of Operations Research*, *6*(1), 85-88.

Bellenguez-Morineau, O., Néron, E. (2007). A branch-and-bound method for solving multi-skill project scheduling problem. *RAIRO-Operations Research*, *41*(02), 155-170.

Berkhin, P. (2006). A survey of clustering data mining techniques. *Grouping multidimensional data*, 25-71.

Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers.

Blazewicz, J., Lenstra, J. K., Kan, A. H. G. (1983). Scheduling subject to resource constraints: classification and complexity. *Discrete Applied Mathematics*, *5*(1), 11-24.

Boctor, F. F. (1990). Some efficient multi-heuristic procedures for resource-constrained project scheduling. *European Journal of Operational Research*, *49*(1), 3-13.

Boctor, F. F. (1993). Heuristics for scheduling projects with resource restrictions and several resource-duration modes. *International Journal of Production Research*, *31*(11), 2547-2558.

Boctor, F. F. (1994). An adaptation of the simulated annealing algorithm for solving resource-constrained project scheduling problems.

Boctor, F. F. (1996). A new and efficient heuristic for scheduling projects with resource restrictions and multiple execution modes. *European Journal of Operational Research*, *90*(2), 349-361.

Bouleimen, K. L. E. I. N., Lecocq, H. O. U. S. N. I. (2003). A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. *European Journal of Operational Research, 149*(2), 268-281.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and regression trees*. Chapman and Hall/CRC.

Brucker, P., Knust, S. (2003). Lower bounds for resource-constrained project scheduling problems. *European Journal of Operational Research*, *149*(2), 302-313.

Brucker, P., Drexl, A., Möhring, R., Neumann, K., Pesch, E. (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, *112*(1), 3-41.

Brucker, P., Knust, S., Schoo, A., Thiele, O. (1998). A branch and bound algorithm for the resource-constrained project scheduling problem. *European Journal of Operational Research*, *107*(2), 272-288.

Buntine, W. (1992). Learning classification trees. *Statistics and computing*,*2*(2), 63-73.
Cai, X., Li, K. N. (2000). A genetic algorithm for scheduling staff of mixed skills under multi-criteria. *European Journal of Operational Research*, *125*(2), 359-369.

Campbell, G. M. (1999). Cross-utilization of workers whose capabilities differ. *Management Science*, *45*(5), 722-732.

Campbell, G. M., Diaby, M. (2002). Development and evaluation of an assignment heuristic for allocating cross-trained workers. *European Journal of Operational Research*, *138*(1), 9-20.

Can, A., Ulusoy, G. (2010). Multi-project scheduling with 2-stage decomposition.

Cendrowska, J. (1987). PRISM: An algorithm for inducing modular rules. *International Journal of Man-Machine Studies*, *27*(4), 349-370.

Chen, P. H., Shahandashti, S. M. (2009). Hybrid of genetic algorithm and simulated annealing for multiple project scheduling with multiple resource constraints. *Automation in Construction*, *18*(4), 434-443.

Cheng, R., Gen, M. (1994, June). Evolution program for resource constrained project scheduling problem. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on* (pp. 736-741). IEEE.

Chtourou, H., Haouari, M. (2008). A two-stage-priority-rule-based algorithm for robust resource-constrained project scheduling. *Computers Industrial Engineering*, *55*(1), 183-194.

Cleary, J. G., Trigg, L. E. (1995, July). K^*: An Instance-based Learner Using an Entropic Distance Measure. In *ICML* (pp. 108-114).

Coello Coello Coello, C., Toscano Pulido, G. (2001). A micro-genetic algorithm for multi-objective optimization. In *Evolutionary Multi-Criterion Optimization* (pp. 126-140). Springer Berlin/Heidelberg.

Coello Coello, C. A., Lechuga, M. S. (2002). MOPSO: A proposal for multiple objective particle swarm optimization. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on Evolutionary Computation* (Vol. 2, pp. 1051-1056). IEEE.

Cohen, W. W. (1995, July). Fast effective rule induction. In *ICML* (Vol. 95, pp. 115-123).

Cooper, D. F. (1976). Heuristics for scheduling resource-constrained projects: An experimental investigation. *Management Science*, *22*(11), 1186-1194.

Corne, D., Knowles, J., Oates, M. (2000). The Pareto envelope-based selection algorithm for multi-objective optimization. In *Parallel Problem Solving from Nature PPSN VI* (pp. 839-848). Springer Berlin/Heidelberg.

Corominas, A., Ojeda, J., Pastor, R. (2004). Multi-objective allocation of multi-function workers with lower bounded capacity. *Journal of the Operational Research Society*, *56*(6), 738-743.

Corominas, A., Pastor, R., Rodríguez, E. (2006). Rotational allocation of tasks to multifunctional workers in a service industry. *International Journal of Production Economics*, *103*(1), 3-9.

Cover, T., and Hart, P. (1967). Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, *13*(1), 21-27.

Creemers, S., Demeulemeester, E., Van de Vonder, S. (2011, December). Project risk management: A new approach. In *Industrial Engineering and Engineering Management (IEEM), 2011 IEEE International Conference on* (pp. 839-843). IEEE.

Creemers, S., Demeulemeester, E., Van de Vonder, S. (2013). A new approach for quantitative risk analysis. *Annals of Operations Research*, 1-39.

Dahan, E. (1998). *Reducing technical uncertainty in product and process development through parallel design of prototypes*. Working Paper, Graduate School of Business, Stanford University.

Davis, E. W., Patterson, J. H. (1975). A comparison of heuristic and optimum solutions in resource-constrained project scheduling. *Management science*, *21*(8), 944-955.

Davis, K. R., Stam, A., Grzybowski, R. A. (1992). Resource constrained project scheduling with multiple objectives: A decision support approach. *Computers operations research*, *19*(7), 657-669.

De Reyck, B., Leus, R. (2008). R&D project scheduling when activities may fail. *IIE transactions*, *40*(4), 367-384.

Deb, K. (2001). Multi-objective optimization. *Multi-objective optimization using evolutionary algorithms*, 13-46.

Deb, K., Mohan, M., Mishra, S. (2003). Towards a quick computation of well-spread pareto-optimal solutions. In *Evolutionary Multi-Criterion Optimization* (pp. 68-68). Springer Berlin/Heidelberg.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. A. M. T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, *6*(2), 182-197.

Debels, D., Vanhoucke, M. (2005). *A decomposition-based heuristic for the resource-constrained project scheduling problem* (No. 05/293). Ghent University, Faculty of Economics and Business Administration.

Deblaere, F., Demeulemeester, E., Herroelen, W. (2011). Reactive scheduling in the multi-mode RCPSP. *Computers Operations Research*, *38*(1), 63-74.

Deckro, R. F., Winkofsky, E. P., E Hebert, J., Gagnon, R. (1991). A decomposition approach to multi-project scheduling. *European Journal of Operational Research*, *51*(1), 110-118.

Demeulemeester, E. L., Herroelen, W. S. (1997). New benchmark results for the resource-constrained project scheduling problem. *Management Science*, *43*(11), 1485-1492.

Drexl, A. (1991). Scheduling of project networks by job assignment. *Management Science*, *37*(12), 1590-1602.

Drexl, A., Gruenewald, J. (1993). Non-preemptive multi-mode resource-constrained project scheduling. *IIE transactions*, *25*(5), 74-81.

Du, Hongbo (M. Phil.). (2010). *Data Mining Techniques and Applications: An Introduction*. Course Technology Cengage Learning.

Duda, R. O., Hart, P. E. (1973). *Pattern classification and scene analysis*(Vol. 3). New York: Wiley.

Elmaghraby, S. E. (2005). On the fallacy of averages in project risk management. *European Journal of Operational Research*, *165*(2), 307-313.

Eppinger, S. D., Whitney, D. E., Smith, R. P., Gebala, D. A. (1994). A model-based method for organizing tasks in product development. *Research in Engineering Design*, *6*(1), 1-13.

Ester, M., Kriegel, H. P., Sander, J., and Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. Kdd.

Fendley, L. G. (1968). Towards the development of a complete multi-project scheduling system. *Journal of Industrial Engineering*, *19*(10), 505-515.

Fleszar, K., Hindi, K. S. (2004). Solving the resource-constrained project scheduling problem by a variable neighborhood search. *European Journal of Operational Research, 155*(2), 402-413.

Fonseca, C. M., Fleming, P. J. (1993, June). Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization. In *Proceedings of the fifth international conference on genetic algorithms* (Vol. 1, p. 416).

Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning.

Goldberg, D. E., Richardson, J. (1987, October). Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application* (pp. 41-49). L. Erlbaum Associates Inc.

Golenko-Ginzburg, D., Gonik, A. (1997). Stochastic network project scheduling with non-consumable limited resources. *International Journal of Production Economics, 48*(1), 29-37.

Gonçalves, J. F., Mendes, J. J., Resende, M. G. (2008). A genetic algorithm for the resource constrained multi-project scheduling problem. *European Journal of Operational Research*, *189*(3), 1171-1190.

Gravel, M., Price, W. L., Gagné, C. (2002). Scheduling continuous casting of aluminum using a multiple objective ant colony optimization metaheuristic. *European Journal of Operational Research*, *143*(1), 218-229.

Gutjahr, W. J., Reiter, P. (2010b). Bi-objective project portfolio selection and staff assignment under uncertainty. *Optimization*, *59*(3), 417-445.

Gutjahr, W. J., Katzensteiner, S., Reiter, P., Stummer, C., Denk, M. (2008). Competence-driven project portfolio selection, scheduling and staff assignment. *Central European Journal of Operations Research, 16*(3), 281-306.

Gutjahr, W. J., Katzensteiner, S., Reiter, P., Stummer, C., Denk, M. (2010a). Multi-objective decision analysis for competence-oriented project portfolio selection. *European Journal of Operational Research*, *205*(3), 670-679.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, *11*(1), 10-18.

Hanne, T., Nickel, S. (2005). A multi-objective evolutionary algorithm for scheduling and inspection planning in software development projects. *European Journal of Operational Research, 167*(3), 663-678.

Hapke, M., Jaszkiewicz, A., Słowiński, R. (1998). Interactive analysis of multiple-criteria project scheduling problems. *European Journal of Operational Research, 107*(2), 315-324.

Hartmann, S. (1998). A competitive genetic algorithm for resource-constrained project scheduling. *Naval Research Logistics (NRL), 45*(7), 733-750.

Hartmann, S. (2001). Project scheduling with multiple modes: a genetic algorithm. *Annals of Operations Research, 102*(1), 111-135.

Hartmann, S. (2002). A self-adapting genetic algorithm for project scheduling under resource constraints. *Naval Research Logistics (NRL), 49*(5), 433-448.

Hartmann, S., Briskorn, D. (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research, 207*(1), 1-14.

Hartmann, S., Drexl, A. (1998). Project scheduling with multiple modes: A comparison of exact algorithms. *Networks, 32*(4), 283-297.

Hartmann, S., Kolisch, R. (2000). Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research, 127*(2), 394-407.

Hegazy, T. (1999). Optimization of resource allocation and leveling using genetic algorithms. *Journal of construction engineering and management, 125*(3), 167-175.

Heimerl, C., Kolisch, R. (2010). Scheduling and staffing multiple projects with a multi-skilled workforce. *OR spectrum, 32*(2), 343-368.

Herroelen, W., Leus, R. (2004). Robust and reactive project scheduling: a review and classification of procedures. *International Journal of Production Research, 42*(8), 1599-1620.

Herroelen, W., Leus, R. (2005). Project scheduling under uncertainty: Survey and research potentials. *European journal of operational research, 165*(2), 289-306.

Herroelen, W., De Reyck, B., Demeulemeester, E. (1998). Resource-constrained project scheduling: a survey of recent developments. *Computers Operations Research, 25*(4), 279-302.

Herroelen, W. (2013) "A risk integrated methodology for project planning under uncertainty",Essays in Production, Project Planning and Scheduling: A Festschrift in Honor of Salah Elmaghraby, Springer Verlag, Berlin (to appear in 2013)

Hindi, K. S., Yang, H., Fleszar, K. (2002). An evolutionary algorithm for resource-constrained project scheduling. *Evolutionary Computation, IEEE Transactions on, 6*(5), 512-518.

Holland, J. H. (1975). Adaptation in natural and artificial systems, University of Michigan press. *Ann Arbor, MI*, *1*(97), 5.

Horn, J., Nafpliotis, N., Goldberg, D. E. (1994, June). A niched Pareto genetic algorithm for multi-objective optimization. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on* (pp. 82-87). IEEE.

Hubbard, D. W. (2009). *The failure of risk management: why it's broken and how to fix it*. John Wiley and Sons.

Icmeli, O., Erenguc, S. S., Zappe, C. J. (1993). Project scheduling problems: a survey. *International Journal of Operations Production Management*, *13*(11), 80-91.

Jaafari, A., (2001). Management of risks, uncertainties and opportunities on projects: time for a fundamental shift. International Journal of Project Management 19, 89–101.

Jain, A. K., Mao, J., and Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. *Computer*, *29*(3), 31-44.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, *31*(3), 264-323.

Jarboui, B., Damak, N., Siarry, P., Rebai, A. (2008). A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems. *Applied Mathematics and Computation*, *195*(1), 299-308.

John, G. H., Langley, P. (1995, August). Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence* (pp. 338-345). Morgan Kaufmann Publishers Inc..

John, G. H., Kohavi, R., and Pfleger, K. (1994, July). Irrelevant features and the subset selection problem. In *Proceedings of the eleventh international conference on machine learning* (Vol. 129, pp. 121-129).

Kılıç, M., Ulusoy, G., Şerifoğlu, F. S. (2008). A bi-objective genetic algorithm approach to risk mitigation in project scheduling. *International Journal of Production Economics*, *112*(1), 202-216.

Kim, K. W., Gen, M., Yamazaki, G. (2003). Hybrid genetic algorithm with fuzzy logic for resource-constrained project scheduling. *Applied Soft Computing*, *2*(3), 174-188.

Kim, K., Gen, M., Kim, M. H. (2006). Adaptive genetic algorithms for multi-resource constrained project scheduling problem with multiple modes. *International Journal of Innovative Computing, Information Control*, *2*(1), 41-49.

Kim, K., Yun, Y., Yoon, J., Gen, M., Yamazaki, G. (2005). Hybrid genetic algorithm with adaptive abilities for resource-constrained multiple project scheduling. *Computers in Industry*, *56*(2), 143-160.

Kirytopoulos, K., Leopoulos, V., Malandrakis, C., Tatsiopoulos, I. (2001). Innovative knowledge system for project risk management. *Advances in Systems Science 2001*, 505-510.

Klein, R. (2000). *Scheduling of resource-constrained projects* (Vol. 10). Springer.

Klein, R., Scholl, A. (1998). *Scattered branch and bound: An adaptive search strategy applied to resource-constrained project scheduling* (No. 14076). Darmstadt Technical University, Department of Business Administration, Economics and Law, Institute for Business Studies (BWL).

Knotts, G., Dror, M., Hartman, B. C. (2000). Agent-based project scheduling. *Iie Transactions*, *32*(5), 387-401.

Knowles, J. D., Corne, D. W. (2000). Approximating the non-dominated front using the pareto archived evolution strategy. *Evolutionary computation*, *8*(2), 149-172.

Kohavi, R., and John, G. H. (1997). Wrappers for feature subset selection. *Artificial intelligence*, *97*(1), 273-324.

Kohlmorgen, U., Schmeck, H., Haase, K. (1999). Experiences with fine-grained parallel genetic algorithms. *Annals of Operations Research*, *90*, 203-219.

Kolisch, R. (1996a). Efficient priority rules for the resource-constrained project scheduling problem. *Journal of Operations Management*, *14*(3), 179-192.

Kolisch, R. (1996b). Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. *European Journal of Operational Research*, *90*(2), 320-333.

Kolisch, R., Drexl, A. (1997). Local search for non-preemptive multi-mode resource-constrained project scheduling. *IIE transactions*, *29*(11), 987-999.

Krishnan, V., Eppinger, S. D., Whitney, D. E. (1997). A model-based framework to overlap product development activities. *Management science*,*43*(4), 437-451.

Kukkonen, S., Deb, K. (2006a). Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems. In *Proceedings of the World Congress on Computational Intelligence (WCCI-2006)(IEEE Press). Vancouver, Canada* (pp. 1179-1186). Proceedings of the World Congress on Computational Intelligence (WCCI-2006)(IEEE Press). Vancouver, Canada.

Kukkonen, S., Deb, K. (2006b). A fast and effective method for pruning of non-dominated solutions in many-objective problems. *Lecture Notes in Computer Science, 4193*, 553-562.

Kumanan, S., Jegan Jose, G., Raja, K. (2006). Multi-project scheduling using an heuristic and a genetic algorithm. *The International Journal of Advanced Manufacturing Technology*, *31*(3), 360-366.

Kurtulus, I. S., Narula, S. C. (1985). Multi-project scheduling: Analysis of project performance. *IIE transactions*, *17*(1), 58-66.

Kurtulus, I., Davis, E. W. (1982). Multi-project scheduling: Categorization of heuristic rules performance. *Management Science, 28*(2), 161-172.

Lambrechts, O., Demeulemeester, E., Herroelen, W. (2008). A tabu search procedure for developing robust predictive project schedules. *International Journal of Production Economics, 111*(2), 493-508.

Lambrechts, O., Demeulemeester, E., Herroelen, W. (2011). Time slack-based techniques for robust project scheduling subject to resource uncertainty. *Annals of Operations Research, 186*(1), 443-464.

Lauritzen, S. L. (1995). The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis, 19*(2), 191-201.

Lee, J. K., Kim, Y. D. (1996). Search heuristics for resource constrained project scheduling. *Journal of the Operational Research Society*, 678-689.

Leon, V. J., Balakrishnan, R. (1995). Strength and adaptability of problem-space based neighborhoods for resource-constrained scheduling. *OR Spectrum*, *17*(2), 173-182.

Leu, S. S., Hung, T. H. (2002). A genetic algorithm-based optimal resource-constrained scheduling simulation model. *Construction Management Economics, 20*(2), 131-141.

Leu, S. S., Yang, C. H., Huang, J. C. (2000). Resource leveling in construction by genetic algorithm-based optimization and its decision support system application. *Automation in construction*, *10*(1), 27-41.

Leus, R. (2003). The generation of stable project plans. PhD thesis. Department of applied economics, Katholieke Universiteit Leuven, Belgium.

Loughlin, D. H., Ranjithan, S. (1997, July). The neighborhood constraint method: a genetic algorithm-based multi-objective optimization technique. In *Proceedings of the Seventh International Conference on Genetic Algorithms* (Vol. 673). San Francisco, California.

Lova, A., Tormos, P., Barber Sanchís, F. (2006). Multi-mode resource constrained project scheduling: scheduling schemes, priority rules and mode selection rules. *Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial*, *10*(30), 69-86.

Lova, A., Tormos, P., Cervantes, M., Barber, F. (2009). An efficient hybrid genetic algorithm for scheduling projects with resource constraints and multiple execution modes. *International Journal of Production Economics*, *117*(2), 302-316.

MacQueen, J. (1967, June). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (Vol. 1, No. 281-297, p. 14).

Man, Z., Wei, T., Xiang, L., Lishan, K. (2008, December). Research on multi-project scheduling problem based on hybrid genetic algorithm. In *Computer Science and Software Engineering, 2008 International Conference on* (Vol. 1, pp. 390-394). IEEE.

McMullen, P. R. (2001). An ant colony optimization approach to addressing a JIT sequencing problem with multiple objectives. *Artificial Intelligence in Engineering*, *15*(3), 309-317.

Mendes, J. J. D. M .(2008). Project scheduling under multiple resources constraints using a genetic algorithm. *way*, *1*, 1.

Mingozzi, A., Maniezzo, V., Ricciardelli, S., Bianco, L. (1998). An exact algorithm for project scheduling with resource constraints based on a new mathematical formulation. *Management Science*, *44*(5), 714-729.

Mohanty, R. P., Siddiq, M. K. (1989). Multiple projects-multiple resources-constrained scheduling: Some studies. *International Journal Of Production Research*, *27*(2), 261-280.

Montoya-Torres, J. R., Gutierrez-Franco, E., Pirachicán-Mayorga, C. (2010). Project scheduling with limited resources using a genetic algorithm. *International Journal of Project Management*, *28*(6), 619-628.

Moret, B. M. (1982). Decision trees and diagrams. *ACM Computing Surveys (CSUR)*, *14*(4), 593-623.

Mori, M., Tseng, C. C. (1997). A genetic algorithm for multi-mode resource constrained project scheduling problem. *European Journal of Operational Research*, *100*(1), 134-141.

Möhring, R., Radermacher, F., Weiss, G. (1984). Stochastic scheduling problems I – set strategies. Zeitschrift fu¨r Operations Research, 28, 193–260.

Möhring, R., Radermacher, F., Weiss, G. (1985). Stochastic scheduling problems II – general strategies. Zeitschrift fur Operations Research, 29, 65–104.

Murthy, S. K. (1998). Automatic construction of decision trees from data: A multi-disciplinary survey. *Data mining and knowledge discovery*, *2*(4), 345-389.

Nonobe, K., Ibaraki, T. (2001). An improved tabu search method for the weighted constraint satisfaction problem. *Infor-Information Systems and Operational Research*, *39*(2), 131-151.

Nudtasomboon, N., Randhawa, S. U. (1997). Resource-constrained project scheduling with renewable and non-renewable resources and time-resource tradeoffs. *Computers industrial engineering*, *32*(1), 227-242.

Özdamar, L. (1999). A genetic algorithm approach to a general category project scheduling problem. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, *29*(1), 44-59.

Özdamar, L., Ulusoy, G. (1994). A local constraint based analysis approach to project scheduling under general resource constraints. *European Journal of Operational Research*, *79*(2), 287-298.

Özdamar, L., Ulusoy, G. (1995). A survey on the resource-constrained project scheduling problem. *IIE transactions*, *27*(5), 574-586.

Palpant, M., Artigues, C., Michelon, P. (2004). LSSPER: solving the resource-constrained project scheduling problem with large neighbourhood search. *Annals of Operations Research*, *131*(1), 237-257.

Patterson, J. H., Slowinski, R., Talbot, F. B., Weglarz, J. (1989). An algorithm for a general class of precedence and resource constrained scheduling problems. *Advances in project scheduling*, (Part I), 3-28.

Payne, J. H. (1995). Management of multiple simultaneous projects: a state-of-the-art review. *International Journal of Project Management*, *13*(3), 163-168.

Pet-Edwards, J., Mollaghesemi, M. (1996, June). A simulation and genetic algorithm approach to stochastic research constrained project scheduling. In *Southcon/96. Conference Record* (pp. 333-338). IEEE.

Peteghem, V. V., Vanhoucke, M. (2010). A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem. *European Journal of Operational Research*, *201*(2), 409-418.

Pinson, E., Prins, C., Rullier, F. (1994). Using tabu search for solving the resource-constrained project scheduling problem. In *Proceedings of the fourth international workshop on project management and scheduling* (pp. 102-106).

Pritsker, A. A. B., Waiters, L. J., Wolfe, P. M. (1969). Multi-project scheduling with limited resources: A zero-one programming approach. *Management Science*, *16*(1), 93-108.

Project Management Institute (PMI), 2000. Guide to the Project Management Body of Knowledge. PMI, Pittsburgh.

Quinlan, J. R. (1993). *C4. 5: programs for machine learning* (Vol. 1). Morgan kaufmann.

Ranjbar, M. R., Kianfar, F. (2007). Solving the discrete time/resource trade-off problem in project scheduling with genetic algorithms. *Applied Mathematics and Computation*, *191*(2), 451-456.

Ranjbar, M., De Reyck, B., Kianfar, F. (2009). A hybrid scatter search for the discrete time/resource trade-off problem in project scheduling. *European Journal of Operational Research*, *193*(1), 35-48.

Safavian, S. R., Landgrebe, D. (1991). A survey of decision tree classifier methodology. *Systems, Man and Cybernetics, IEEE Transactions on*, *21*(3), 660-674.

Sasaki, D., Morikawa, M., Obayashi, S., Nakahashi, K. (2001). Aerodynamic shape optimization of supersonic wings by adaptive range multi-objective genetic algorithms. In *Evolutionary Multi-Criterion Optimization* (pp. 639-652). Springer Berlin/Heidelberg.

Schaffer, J. D. (1985, July). Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st international Conference on Genetic Algorithms* (pp. 93-100). L. Erlbaum Associates Inc..

Schatteman, D., Herroelen, W., Van de Vonder, S., Boone, A. (2008). Methodology for integrated risk management and proactive scheduling of construction projects. *Journal of Construction Engineering and Management*, *134*(11), 885-893.

Schmidt, R. L., Freeland, J. R. (1992). Recent progress in modeling R&D project-selection processes. *Engineering Management, IEEE Transactions on*,*39*(2), 189-201.

Slowinski, R. (1980). Two Approaches to Problems of Resource Allocation among Project Activities--A Comparative Study. *Journal of the Operational Research Society*, 711-723.

Slowinski, R., Soniewicki, B., Węglarz, J. (1994). DSS for multi-objective project scheduling. *European Journal of Operational Research*, *79*(2), 220-229.

Souder, W. E., Mandakovic, T. (1986). R&D project selection models. *Research Management*, *29*(4), 36-42.

Speranza, M. G., Vercellis, C. (1993). Hierarchical models for multi-project planning and scheduling. *European Journal of Operational Research*, *64*(2), 312-325.

Sprecher, A. (2000). Scheduling resource-constrained projects competitively at modest memory requirements. *Management Science*, *46*(5), 710-723.

Sprecher, A., Drexl, A. (1998). Multi-mode resource-constrained project scheduling by a simple, general and powerful sequencing algorithm. *European Journal of Operational Research*, *107*(2), 431-450.

Sprecher, A., Hartmann, S., Drexl, A. (1997). An exact algorithm for project scheduling with multiple modes. *OR spectrum*, *19*(3), 195-203.

Srinivas, N., Deb, K. (1994). Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, *2*(3), 221-248.

Subramanian, D., Pekny, J. F., Reklaitis, G. V. (2000). A simulation—optimization framework for addressing combinatorial and stochastic aspects of an R&D pipeline management problem. *Computers & Chemical Engineering*,*24*(2), 1005-1011.

Talbot, F. B. (1982). Resource-constrained project scheduling with time-resource tradeoffs: The nonpreemptive case. *Management Science*, *28*(10), 1197-1210.

Tan, P. N., Steinbach, M. and Kumar, V.(2006). Introduction to Data Mining, Addison-Wesley.

Thomas, P. R., Salhi, S. (1998). A tabu search approach for the resource constrained project scheduling problem. *Journal of Heuristics*, *4*(2), 123-139.

Tormos, P., Lova, A. (2001). A competitive heuristic solution technique for resource-constrained project scheduling. *Annals of Operations Research*, *102*(1), 65-81.

Tseng, L. Y., Chen, S. C. (2006). A hybrid metaheuristic for the resource-constrained project scheduling problem. *European Journal of Operational Research*, *175*(2), 707-721.

Ulusoy, G., Cebelli, S. (2000). An equitable approach to the payment scheduling problem in project management. *European Journal of Operational Research*, *127*(2), 262-278.

Ulusoy, G., Özdamar, L. (1995). A heuristic scheduling algorithm for improving the duration and net present value of a project. *International Journal of Operations Production Management*, *15*(1), 89-98.

Ulusoy, G., Sivrikaya-Şerifoğlu, F., Şahin, Ş. (2001). Four payment models for the multi-mode resource constrained project scheduling problem with discounted cash flows. *Annals of Operations Research*, *102*(1), 237-261.

Vairaktarakis, G. L. (2003). The value of resource flexibility in the resource-constrained job assignment problem. *Management Science*, *49*(6), 718-732.

Valls, V., Ballestín, F., Quintanilla, S. (2004). A population-based approach to the resource-constrained project scheduling problem. *Annals of Operations Research*, *131*(1), 305-324.

Valls, V., Ballestin, F., Quintanilla, S. (2008). A hybrid genetic algorithm for the resource-constrained project scheduling problem. *European Journal of Operational Research*, *185*(2), 495-508.

Valls, V., Pérez, Á., Quintanilla, S. (2009). Skilled workforce scheduling in Service Centres. *European Journal of Operational Research*, *193*(3), 791-804.

Van de Vonder, S., Ballestin, F., Demeulemeester, E., Herroelen, W. (2007). Heuristic procedures for reactive project scheduling. *Computers industrial engineering*, *52*(1), 11-28.

Van de Vonder, S., Demeulemeester, E., Herroelen, W. (2005a). Heuristic procedures for generating stable project baseline schedules. In Proceedings of 3rd Euro conference for young OR researchers and practitioners (ORP3), Valencia, Spain, pp. 11–19.

Van de Vonder, S., Demeulemeester, E., Herroelen, W. (2008). Proactive heuristic procedures for robust project scheduling: An experimental analysis. *European Journal of Operational Research*, *189*(3), 723-733.

Van de Vonder, S., Demeulemeester, E., Herroelen, W., Leus, R. (2006). The trade-off between stability and makespan in resource-constrained project scheduling. International Journal of Production Research, 44(2), 215-236.

Vanhoucke, M. (2006). Scheduling an R&D project with quality-dependent time slots. *In Computational Science and Its Applications-ICCSA* 2006 (pp. 621-630). Springer Berlin Heidelberg.

Veldhuizen, D. A., Lamont, G. B. (2000, March). multi-objective optimization with messy genetic algorithms. In *Proceedings of the 2000 ACM symposium on Applied computing-Volume 1* (pp. 470-476). ACM.

Vercellis, C. (1994). Constrained multi-project planning problems: A Lagrangean decomposition approach. *European Journal of Operational Research*, *78*(2), 267-275.

Viana, A., Pinho de Sousa, J. (2000). Using metaheuristics in multi-objective resource constrained project scheduling. *European Journal of Operational Research*, *120*(2), 359-374.

Wu, M. C., Sun, S. H. (2006). A project scheduling and staff assignment model considering learning effect. *The International Journal of Advanced Manufacturing Technology*, *28*(11), 1190-1195.

Wuliang, P., Chengen, W. (2009). A multi-mode resource-constrained discrete time–cost tradeoff problem and its genetic algorithm based solution. *International journal of project management*, *27*(6), 600-609.

Xiong, J., Chen, Y. W., Yang, K. W., Zhao, Q. S., Xing, L. N. (2012). A hybrid multi-objective genetic algorithm for robust resource-constrained project scheduling with stochastic durations. *Mathematical Problems in Engineering*, *2012*.

Yu, G., Qi, X. (2004). Disruption management – Framework, models and applications. New Jersey: World Scientific.

Zhang, H., Tam, C. M., Li, H. (2005). Multimode project scheduling based on particle swarm optimization. *Computer‐Aided Civil and Infrastructure Engineering*, *21*(2), 93-103.

Zhu, G., Bard, J. F., Yu, G. (2006). A branch-and-cut procedure for the multimode resource-constrained project-scheduling problem. *INFORMS Journal on Computing*, *18*(3), 377-390.

Zitzler, E., Thiele, L. (1999). multi-objective evolutionary algorithms: A comparative case study and the strength pareto approach. *Evolutionary Computation, IEEE Transactions on*, *3*(4), 257-271.

APPENDIX A. Project-Based Non-Dominated GA Parameters

Non-dominated Parameter GA Combinations for Project 08-024								
Parameter Combination					Normalized Performance Measures			
Cross over Rate	Mutation Rate	Population Size	Number of Generations	Number of Schedule Realizations	MS	Scaled EHR	Number Of Non-dominated Solutions	CPU Time
0.6	0.05	30	30	200	0.85	0.50	1.00	0.10
0.6	0.05	50	50	200	0.92	0.50	1.00	0.35
0.75	0.1	30	70	200	1.00	0.50	1.00	0.37
0.6	0.05	30	30	100	0.39	0.70	0.80	0.05
0.75	0.05	30	50	100	0.85	0.50	1.00	0.11
0.95	0.1	30	30	100	0.64	0.70	0.80	0.09
0.6	0.1	50	30	100	0.54	0.70	0.80	0.09

Non-dominated Parameter GA Combinations for Project 08-059								
Parameter Combination					Normalized Performance Measures			
Cross over Rate	Mutation Rate	Population Size	Number of Generations	Number of Schedule Realizations	MS	Scaled EHR	Number Of Non-dominated Solutions	CPU Time
0.6	0.1	70	30	200	0.82	0.47	0.80	0.29
0.75	0.1	70	30	200	0.96	0.51	0.85	0.35
0.75	0.1	70	70	200	0.75	0.38	1.00	0.76
0.6	0.05	30	30	100	0.72	0.77	0.60	0.05
0.75	0.05	30	30	100	1.00	0.56	0.75	0.06
0.6	0.05	70	30	100	0.95	0.47	0.85	0.12
0.6	0.1	30	30	100	0.79	0.50	0.80	0.05
0.95	0.1	30	30	100	0.71	0.50	0.80	0.08
0.6	0.1	50	30	100	0.79	0.45	0.85	0.09

Non-dominated Parameter GA Combinations for Project 09-028								
Parameter Combination					Normalized Performance Measures			
Cross over Rate	Mutation Rate	Population Size	Number of Generations	Number of Schedule Realizations	MS	Scaled EHR	Number Of Non-dominated Solutions	CPU Time
0.6	0.05	30	30	100	0.83	0.38	1.00	0.05
0.95	0.05	30	30	100	0.85	0.39	0.96	0.08
0.75	0.05	70	50	100	1.00	0.40	0.91	0.26

Non-dominated Parameter GA Combinations for Project 10-015								
Parameter Combination					Normalized Performance Measures			
Cross over Rate	Mutation Rate	Population Size	Number of Generations	Number of Schedule Realizations	MS	Scaled EHR	Number Of Non-dominated Solutions	CPU Time
0.6	0.05	30	30	100	0.67	0.72	0.62	0.06
0.75	0.05	30	30	100	0.90	0.44	0.81	0.07
0.95	0.05	70	70	100	1.00	0.38	0.95	0.42
0.6	0.1	30	30	100	0.72	0.55	0.71	0.06
0.75	0.1	30	30	100	0.92	0.45	0.81	0.07
0.6	0.1	50	30	100	0.98	0.37	1.00	0.09

APPENDIX B. Best GA Parameters in Terms of Single Performance Measures

Overall Results						
Best Parameter Combinations in Terms of	**Crossover Rate**	**Mutation Rate**	**Population Size**	**Number of Generations**	**Number of Schedule Realizations**	**Best Normalized Value**
MS	0.6	0.1	50	30	100	0.16
Scaled EHR	0.6	0.1	50	30	100	0.38
Number of Non-Dominated Solutions	0.6	0.1	50	30	100	3.40
CPU Time	0.6	0.05	30	30	100	
Project 08-024						
Best Parameter Combinations in Terms of	**Crossover Rate**	**Mutation Rate**	**Population Size**	**Number of Generations**	**Number of Schedule Realizations**	**Best Normalized Value**
MS	0.75	0.1	30	70	200	0.09
Scaled EHR	0.6	0.05	50	50	200	0.50
Number of Non-Dominated Solutions	0.75	0.05	30	50	100	2.00
CPU Time	0.6	0.05	30	30	100	230.34
Project 08-059						
Best Parameter Combinations in Terms of	**Crossover Rate**	**Mutation Rate**	**Population Size**	**Number of Generations**	**Number of Schedule Realizations**	**Best Normalized Value**
MS	0.75	0.05	30	30	100	0.18
Scaled EHR	0.75	0.1	70	70	200	0.26
Number of Non-Dominated Solutions	0.75	0.1	70	70	200	4.00
CPU Time	0.6	0.05	30	30	100	321.45
Project 09-028						
Best Parameter Combinations in Terms of	**Crossover Rate**	**Mutation Rate**	**Population Size**	**Number of Generations**	**Number of Schedule Realizations**	**Best Normalized Value**
MS	0.75	0.05	70	50	100	0.35
Scaled EHR	0.6	0.05	30	30	100	0.23
Number of Non-Dominated Solutions	0.6	0.05	30	30	100	4.60
CPU Time	0.6	0.05	30	30	100	449.12
Project 10-015						
Best Parameter Combinations in Terms of	**Crossover Rate**	**Mutation Rate**	**Population Size**	**Number of Generations**	**Number of Schedule Realizations**	**Best Normalized Value**
MS	0.75	0.05	70	50	100	0.20
Scaled EHR	0.6	0.05	30	30	100	0.25
Number of Non-Dominated Solutions	0.6	0.05	30	30	100	4.20
CPU Time	0.6	0.05	30	30	100	99.84

APPENDIX C. Activity-Resource Schedule for Project 08-040 Obtained with Single Project Scheduling Approach and Fitness Calculation Procedure1

Activity ID	Resource ID	Robust Starting Time	Robust Ending Time	Duration (weeks)	Activity ID	Resource ID	Robust Starting Time	Robust Ending Time	Duration (weeks)
08-040CU0010	AR001749	07/14/08	07/28/08	2	08-040A1550	AR001749	05/04/09	05/18/09	2
08-040CU0010	AR001993	08/04/08	09/01/08	4	08-040A1550	AR002162	05/04/09	05/18/09	2
08-040CU0010	AR110962X	08/04/08	09/15/08	6	08-040A1550	AR001835	05/04/09	05/25/09	3
08-040CU0010	AR001835	08/11/08	09/01/08	3	08-040A1550	AR001993	05/04/09	06/01/09	4
08-040CU0010	AR001863	09/15/08	09/29/08	2	08-040A1550	A44C.AC1	05/04/09	05/18/09	2
08-040CU0010	A44C.TOC	07/14/08	08/11/08	4	08-040A1550	A44C.AC2	05/04/09	05/25/09	3
08-040CU0010	A44C.AC1	07/21/08	08/11/08	3	08-040A1550	A44C.W1	05/04/09	06/15/09	6
08-040CU0010	A44C.DD3	07/14/08	07/28/08	2	08-040A1550	A44C.W2	05/04/09	06/22/09	7
08-040CU0015	AR001993	09/29/08	10/13/08	2	08-040A1550	A44C.DD3	05/04/09	05/11/09	1
08-040A1490	AR001993	10/13/08	10/27/08	2	08-040A1570	AR001863	06/15/09	06/29/09	2
08-040A1490	AR001863	10/13/08	11/03/08	3	08-040A1570	AR110962X	06/15/09	08/10/09	8
08-040A1490	AR110962X	10/13/08	10/20/08	1	08-040A1570	AR001749	06/15/09	07/20/09	5
08-040A1540	AR110640	11/17/08	11/24/08	1	08-040A1570	AR001993	06/15/09	08/17/09	9
08-040A1540	AR001863	11/03/08	11/24/08	3	08-040A1580	AR001863	08/31/09	09/07/09	1
08-040A1540	AR110962X	11/03/08	11/24/08	3	08-040A1580	AR110962X	08/17/09	09/07/09	3
08-040A1540	AR001749	11/03/08	12/01/08	4	08-040A1580	AR001351X	08/17/09	08/24/09	1
08-040A1540	AR001835	11/17/08	01/12/09	8	08-040A1580	AR001749	08/17/09	08/31/09	2
08-040A1540	AR001351X	11/03/08	11/10/08	1	08-040A1580	AR001835	08/17/09	09/07/09	3
08-040A1540	AR002162	11/03/08	12/01/08	4	08-040A1580	AR001993	08/17/09	08/31/09	2
08-040A1540	AR001993	11/24/08	12/15/08	3	08-040A1580	A44C.AC1	08/17/09	08/24/09	1
08-040A1540	A44C.AC1	11/17/08	01/26/09	10	08-040A1580	A44C.AC2	08/17/09	08/31/09	2
08-040A1540	A44C.AC2	02/02/09	03/02/09	4	08-040A1580	A44C.W1	08/17/09	09/07/09	3
08-040A1540	A44C.W1	03/09/09	05/04/09	8	08-040A1580	A44C.W2	08/24/09	09/07/09	2
08-040A1540	A44C.W2	03/02/09	03/30/09	4	08-040A1580	A44C.DD3	08/17/09	08/24/09	1
08-040A1540	A44C.DD3	11/03/08	11/17/08	2	08-040A1590	AR001863	09/07/09	09/14/09	1
08-040A1560	AR001993	12/22/08	02/02/09	6	08-040A1590	AR110962X	09/07/09	09/14/09	1
08-040A1560	AR001863	12/22/08	12/29/08	1	08-040A1590	AR001351X	09/07/09	09/14/09	1
08-040A1560	AR110962X	12/22/08	01/12/09	3	08-040A1590	AR001993	09/07/09	09/14/09	1
08-040A1560	AR001749	12/22/08	12/29/08	1	08-040A1630	AR001749	05/25/09	06/08/09	2

08-040A1560	A44C.DD3	12/22/08	12/29/08	1	08-040A1630	AR110962X	05/25/09	06/08/09	2
08-040A1600	AR001993	12/08/08	12/15/08	1	08-040A1630	AR001993	05/25/09	06/15/09	3
08-040A1600	AR110962X	11/17/08	12/01/08	2	08-040A1630	AR001863	05/25/09	06/01/09	1
08-040A1600	AR001863	11/17/08	11/24/08	1	08-040A1630	AR001835	05/25/09	06/08/09	2
08-040A1620	AR110640	02/02/09	03/09/09	5	08-040A1630	A44C.FR4	05/25/09	10/12/09	20
08-040A1620	AR001749	02/02/09	02/23/09	3	08-040A1640	AR001749	10/12/09	11/09/09	4
08-040A1620	AR110962X	02/02/09	02/23/09	3	08-040A1640	AR110962X	10/12/09	11/09/09	4
08-040A1620	AR001993	02/02/09	02/23/09	3	08-040A1640	AR001993	10/12/09	11/02/09	3
08-040A1620	AR001863	02/02/09	02/16/09	2	08-040A1640	AR001863	10/12/09	11/02/09	3
08-040A1620	AR001835	02/02/09	02/16/09	2	08-040A1640	AR001835	10/12/09	11/02/09	3
08-040A1620	AR109312	02/02/09	03/02/09	4	08-040A1640	AR109312	10/12/09	11/02/09	3
08-040A1620	A44C.FR4	02/02/09	05/25/09	16	08-040A1640	AR110640	10/12/09	11/02/09	3
08-040A1550	AR001863	05/18/09	05/25/09	1	08-040A1640	AR903358	10/12/09	12/14/09	9
08-040A1550	AR110962X	05/04/09	05/18/09	2	08-040A1640	A44C.FR4	10/12/09	02/08/10	17

APPENDIX D. Resource Work Schedules for Project 08-040 Obtained with Single Project Scheduling Approach and Fitness Calculation Procedure1

Activity ID	Resource ID	Week	Allocated Hours	Activity ID	Resource ID	Week	Allocated Hours
08-040CU0010	AR001749	07/14/08	29	08-040A1620	A44C.FR4	06/29/09	180
08-040CU0010	AR001749	07/21/08	9	08-040A1620	A44C.FR4	07/06/09	180
08-040CU0010	AR001993	07/14/08	14	08-040A1620	A44C.FR4	07/13/09	180
08-040CU0010	AR001993	07/21/08	14	08-040A1630	AR001749	07/20/09	29
08-040CU0010	AR001993	07/28/08	15	08-040A1630	AR001749	07/27/09	15
08-040CU0010	AR001993	08/04/08	40	08-040A1630	AR110962X	07/20/09	32
08-040CU0010	AR001993	08/11/08	23	08-040A1630	AR110962X	07/27/09	30
08-040CU0010	AR110962X	08/04/08	10	08-040A1630	AR110962X	08/03/09	8
08-040CU0010	AR110962X	08/11/08	45	08-040A1630	AR001993	07/20/09	25
08-040CU0010	AR110962X	08/18/08	37	08-040A1630	AR001993	08/03/09	34
08-040CU0010	AR110962X	08/25/08	24	08-040A1630	AR001863	07/20/09	8
08-040CU0010	AR110962X	09/01/08	24	08-040A1630	AR001863	08/10/09	22
08-040CU0010	AR110962X	09/08/08	12	08-040A1630	AR001835	07/20/09	39
08-040CU0010	AR001835	08/18/08	2	08-040A1630	AR001835	08/03/09	5
08-040CU0010	AR001835	08/25/08	17	08-040A1630	A44C.FR4	07/20/09	180
08-040CU0010	AR001835	09/01/08	17	08-040A1630	A44C.FR4	07/27/09	94
08-040CU0010	AR001835	09/08/08	2	08-040A1630	A44C.FR4	08/03/09	94
08-040CU0010	AR001863	09/15/08	10	08-040A1630	A44C.FR4	08/10/09	94
08-040CU0010	AR001863	09/22/08	5	08-040A1630	A44C.FR4	08/17/09	94
08-040CU0010	A44C.TOC	07/14/08	148	08-040A1630	A44C.FR4	08/24/09	94
08-040CU0010	A44C.TOC	07/21/08	148	08-040A1630	A44C.FR4	08/31/09	94
08-040CU0010	A44C.TOC	07/28/08	148	08-040A1630	A44C.FR4	09/07/09	94
08-040CU0010	A44C.TOC	08/04/08	96	08-040A1630	A44C.FR4	09/14/09	180
08-040CU0010	A44C.AC1	07/21/08	96	08-040A1630	A44C.FR4	09/21/09	180
08-040CU0010	A44C.AC1	07/28/08	101	08-040A1630	A44C.FR4	09/28/09	180
08-040CU0010	A44C.AC1	08/04/08	73	08-040A1630	A44C.FR4	10/05/09	180
08-040CU0010	A44C.DD3	07/14/08	272	08-040A1630	A44C.FR4	10/12/09	180
08-040CU0010	A44C.DD3	07/21/08	88	08-040A1630	A44C.FR4	10/19/09	180
08-040CU0015	AR001993	10/20/08	1	08-040A1630	A44C.FR4	10/26/09	180
08-040CU0015	AR001993	10/27/08	3	08-040A1630	A44C.FR4	11/02/09	180
08-040CU0015	AR001993	11/03/08	4	08-040A1630	A44C.FR4	11/09/09	180
08-040A1490	AR001993	11/10/08	6	08-040A1630	A44C.FR4	11/16/09	180
08-040A1490	AR001993	11/17/08	7	08-040A1630	A44C.FR4	11/23/09	180
08-040A1490	AR001993	11/24/08	2	08-040A1630	A44C.FR4	11/30/09	62
08-040A1490	AR001863	11/10/08	1	08-040A1550	AR001863	05/04/09	17
08-040A1490	AR001863	11/17/08	7	08-040A1550	AR110962X	05/04/09	36
08-040A1490	AR110962X	11/10/08	11	08-040A1550	AR110962X	05/11/09	36
08-040A1490	AR110962X	11/17/08	4	08-040A1550	AR110962X	05/18/09	12

08-040A1600	AR001993	12/01/08	9	08-040A1550	AR001749	05/04/09	27
08-040A1600	AR001993	12/08/08	28	08-040A1550	AR001749	05/11/09	27
08-040A1600	AR110962X	12/01/08	15	08-040A1550	AR001749	05/18/09	14
08-040A1600	AR110962X	12/08/08	14	08-040A1550	AR002162	05/04/09	44
08-040A1600	AR110962X	12/15/08	28	08-040A1550	AR002162	05/11/09	24
08-040A1600	AR110962X	12/22/08	2	08-040A1550	AR001835	05/04/09	38
08-040A1600	AR001863	12/01/08	15	08-040A1550	AR001835	05/11/09	39
08-040A1540	AR110640	12/01/08	18	08-040A1550	AR001835	05/18/09	39
08-040A1540	AR110640	12/08/08	21	08-040A1550	AR001835	05/25/09	11
08-040A1540	AR001863	12/01/08	13	08-040A1550	AR001993	05/04/09	22
08-040A1540	AR001863	12/08/08	10	08-040A1550	AR001993	06/01/09	3
08-040A1540	AR110962X	12/22/08	30	08-040A1550	AR001993	06/08/09	16
08-040A1540	AR110962X	12/29/08	30	08-040A1550	AR001993	06/15/09	14
08-040A1540	AR110962X	01/05/09	26	08-040A1550	AR001993	06/22/09	15
08-040A1540	AR110962X	01/12/09	26	08-040A1550	AR001993	06/29/09	14
08-040A1540	AR110962X	01/19/09	4	08-040A1550	A44C.AC1	05/04/09	161
08-040A1540	AR001749	12/01/08	30	08-040A1550	A44C.AC1	05/11/09	109
08-040A1540	AR001749	12/08/08	43	08-040A1550	A44C.AC2	05/04/09	127
08-040A1540	AR001749	12/15/08	20	08-040A1550	A44C.AC2	05/11/09	127
08-040A1540	AR001835	12/08/08	1	08-040A1550	A44C.AC2	05/18/09	16
08-040A1540	AR001835	02/09/09	20	08-040A1550	A44C.W1	05/04/09	86
08-040A1540	AR001835	02/16/09	21	08-040A1550	A44C.W1	05/11/09	87
08-040A1540	AR001835	02/23/09	23	08-040A1550	A44C.W1	05/18/09	68
08-040A1540	AR001835	03/02/09	26	08-040A1550	A44C.W1	05/25/09	29
08-040A1540	AR001835	03/09/09	25	08-040A1550	A44C.W2	05/04/09	86
08-040A1540	AR001351X	12/01/08	16	08-040A1550	A44C.W2	05/11/09	87
08-040A1540	AR002162	12/01/08	41	08-040A1550	A44C.W2	05/18/09	68
08-040A1540	AR002162	12/08/08	37	08-040A1550	A44C.W2	05/25/09	29
08-040A1540	AR001993	12/08/08	3	08-040A1550	A44C.DD3	05/04/09	360
08-040A1540	AR001993	12/22/08	10	08-040A1570	AR001863	08/10/09	7
08-040A1540	AR001993	12/29/08	12	08-040A1570	AR001863	08/17/09	9
08-040A1540	AR001993	01/05/09	10	08-040A1570	AR110962X	06/15/09	34
08-040A1540	AR001993	01/12/09	15	08-040A1570	AR110962X	06/22/09	32
08-040A1540	AR001993	01/19/09	15	08-040A1570	AR110962X	06/29/09	15
08-040A1540	AR001993	02/02/09	7	08-040A1570	AR001749	06/15/09	18
08-040A1540	AR001993	02/09/09	21	08-040A1570	AR001749	06/22/09	22
08-040A1540	A44C.AC1	12/01/08	95	08-040A1570	AR001993	06/29/09	2
08-040A1540	A44C.AC1	12/08/08	117	08-040A1570	AR001993	07/06/09	19
08-040A1540	A44C.AC1	01/05/09	42	08-040A1570	AR001993	07/13/09	25
08-040A1540	A44C.AC1	01/12/09	16	08-040A1570	AR001993	08/03/09	1
08-040A1540	A44C.AC2	01/12/09	34	08-040A1570	AR001993	08/10/09	18
08-040A1540	A44C.AC2	01/19/09	98	08-040A1640	AR001749	12/07/09	35
08-040A1540	A44C.AC2	01/26/09	109	08-040A1640	AR001749	12/14/09	13
08-040A1540	A44C.AC2	02/02/09	29	08-040A1640	AR110962X	12/07/09	31
08-040A1540	A44C.W1	02/23/09	80	08-040A1640	AR110962X	12/14/09	31

08-040A1540	A44C.W1	03/02/09	99	08-040A1640	AR110962X	12/21/09	15
08-040A1540	A44C.W1	04/06/09	2	08-040A1640	AR001993	12/07/09	42
08-040A1540	A44C.W1	04/20/09	49	08-040A1640	AR001993	12/14/09	23
08-040A1540	A44C.W1	04/27/09	40	08-040A1640	AR001863	12/07/09	32
08-040A1540	A44C.W2	02/09/09	4	08-040A1640	AR001863	12/14/09	0
08-040A1540	A44C.W2	02/16/09	69	08-040A1640	AR001835	12/07/09	41
08-040A1540	A44C.W2	02/23/09	94	08-040A1640	AR001835	12/14/09	7
08-040A1540	A44C.W2	03/02/09	99	08-040A1640	AR109312	12/07/09	26
08-040A1540	A44C.W2	03/09/09	4	08-040A1640	AR109312	12/14/09	6
08-040A1540	A44C.DD3	12/01/08	360	08-040A1640	AR110640	12/07/09	32
08-040A1560	AR001993	02/09/09	2	08-040A1640	AR903358	12/07/09	45
08-040A1560	AR001993	02/16/09	23	08-040A1640	AR903358	12/14/09	45
08-040A1560	AR001993	02/23/09	19	08-040A1640	AR903358	12/21/09	45
08-040A1560	AR001993	03/02/09	17	08-040A1640	AR903358	12/28/09	45
08-040A1560	AR001993	03/09/09	15	08-040A1640	AR903358	01/04/10	42
08-040A1560	AR001993	03/16/09	13	08-040A1640	AR903358	01/11/10	4
08-040A1560	AR001993	03/23/09	11	08-040A1640	A44C.FR4	12/07/09	180
08-040A1560	AR001863	01/19/09	23	08-040A1640	A44C.FR4	12/14/09	180
08-040A1560	AR001863	01/26/09	2	08-040A1640	A44C.FR4	12/21/09	180
08-040A1560	AR110962X	01/19/09	22	08-040A1640	A44C.FR4	12/28/09	178
08-040A1560	AR110962X	01/26/09	28	08-040A1640	A44C.FR4	01/04/10	170
08-040A1560	AR110962X	02/02/09	29	08-040A1640	A44C.FR4	01/11/10	170
08-040A1560	AR110962X	02/09/09	29	08-040A1640	A44C.FR4	01/18/10	170
08-040A1560	AR110962X	02/16/09	18	08-040A1640	A44C.FR4	01/25/10	170
08-040A1560	AR001749	01/19/09	17	08-040A1640	A44C.FR4	02/01/10	170
08-040A1560	A44C.DD3	01/19/09	180	08-040A1640	A44C.FR4	02/08/10	180
08-040A1620	AR110640	03/30/09	9	08-040A1640	A44C.FR4	02/15/10	180
08-040A1620	AR110640	04/06/09	9	08-040A1640	A44C.FR4	02/22/10	180
08-040A1620	AR110640	04/13/09	9	08-040A1640	A44C.FR4	03/01/10	180
08-040A1620	AR110640	04/20/09	8	08-040A1640	A44C.FR4	03/08/10	180
08-040A1620	AR110640	04/27/09	9	08-040A1640	A44C.FR4	03/15/10	180
08-040A1620	AR001749	03/30/09	26	08-040A1640	A44C.FR4	03/22/10	180
08-040A1620	AR001749	04/06/09	14	08-040A1640	A44C.FR4	03/29/10	52
08-040A1620	AR110962X	03/30/09	21	08-040A1580	AR001863	08/24/09	17
08-040A1620	AR110962X	04/06/09	30	08-040A1580	AR110962X	08/24/09	36
08-040A1620	AR110962X	04/13/09	12	08-040A1580	AR110962X	08/31/09	35
08-040A1620	AR001993	03/30/09	18	08-040A1580	AR110962X	09/07/09	34
08-040A1620	AR001993	04/06/09	21	08-040A1580	AR110962X	09/14/09	24
08-040A1620	AR001993	04/13/09	14	08-040A1580	AR001351X	08/24/09	17
08-040A1620	AR001863	04/13/09	30	08-040A1580	AR001749	08/24/09	2
08-040A1620	AR001863	04/20/09	10	08-040A1580	AR001749	08/31/09	26
08-040A1620	AR001835	03/30/09	32	08-040A1580	AR001749	09/07/09	29
08-040A1620	AR001835	04/06/09	8	08-040A1580	AR001835	08/24/09	39
08-040A1620	AR109312	03/30/09	13	08-040A1580	AR001835	08/31/09	18
08-040A1620	AR109312	04/06/09	12	08-040A1580	AR001993	08/24/09	35

08-040A1620	AR109312	04/13/09	12	08-040A1580	AR001993	09/07/09	17
08-040A1620	AR109312	04/20/09	3	08-040A1580	AR001993	09/14/09	17
08-040A1620	A44C.FR4	03/30/09	180	08-040A1580	A44C.AC1	09/07/09	90
08-040A1620	A44C.FR4	04/06/09	180	08-040A1580	A44C.AC2	10/19/09	31
08-040A1620	A44C.FR4	04/13/09	180	08-040A1580	A44C.AC2	10/26/09	38
08-040A1620	A44C.FR4	04/20/09	180	08-040A1580	A44C.AC2	01/04/10	21
08-040A1620	A44C.FR4	04/27/09	180	08-040A1580	A44C.W1	02/01/10	35
08-040A1620	A44C.FR4	05/04/09	180	08-040A1580	A44C.W1	02/08/10	55
08-040A1620	A44C.FR4	05/11/09	180	08-040A1580	A44C.W2	02/08/10	23
08-040A1620	A44C.FR4	05/18/09	180	08-040A1580	A44C.W2	03/01/10	67
08-040A1620	A44C.FR4	05/25/09	180	08-040A1580	A44C.DD3	08/24/09	180
08-040A1620	A44C.FR4	06/01/09	180	08-040A1590	AR001863	03/08/10	4
08-040A1620	A44C.FR4	06/08/09	180	08-040A1590	AR110962X	03/08/10	4
08-040A1620	A44C.FR4	06/15/09	180	08-040A1590	AR001351X	03/08/10	16
08-040A1620	A44C.FR4	06/22/09	180	08-040A1590	AR001993	03/08/10	9