DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

A CONSTRAINT PROGRAMMING BASED TRANSFORMATION APPROACH FOR A MULTI-OBJECTIVE AND MULTI-MODE RESOURCE INVESTMENT PROJECT SCHEDULING PROBLEM UNDER FUZZY-STOCHASTIC ENVIRONMENTS

by
Gizem ÇAKIR

October, 2021
İZMİR

A CONSTRAINT PROGRAMMING BASED TRANSFORMATION APPROACH FOR A MULTI-OBJECTIVE AND MULTI-MODE RESOURCE INVESTMENT PROJECT SCHEDULING PROBLEM UNDER FUZZY-STOCHASTIC ENVIRONMENTS

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree Master of Science in Industrial Engineering, Industrial Engineering Program

by
Gizem ÇAKIR

October, 2021
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"A CONSTRAINT PROGRAMMING BASED TRANSFORMATION APPROACH FOR A MULTI-OBJECTIVE AND MULTI-MODE RESOURCE INVESTMENT PROJECT SCHEDULING PROBLEM UNDER FUZZY-STOCHASTIC ENVIRONMENTS"** completed by **GİZEM ÇAKIR** under supervision of **ASSOC.PROF.DR. KEMAL SUBULAN** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

.....

Assoc.Prof.Dr. Kemal SUBULAN

Supervisor

.....

Prof. Dr. Şeyda TOPALOĞLU YILDIZ

(Jury Member)

.....

Assoc. Prof. Dr. Alper HAMZADAYI

(Jury Member)

Prof.Dr. Özgür ÖZÇELİK

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

First of all, I would like to thanks to my Master's thesis supervisor, Assoc. Prof. Dr. Kemal SUBULAN for his leading, motivation, academic support, correction, understanding and patience during the development of this thesis.

I would like to express my special thanks to my parents İlknur and Ünal ÇAKIR. I could not have succeeded without their encouragement, support, and endless love.

Finally, I would like to present my thanks to my dearest friend, Oğuzcan GÖKGÖZ for his encouragement, deep understanding, positive energy.

Gizem ÇAKIR

A CONSTRAINT PROGRAMMING BASED TRANSFORMATION APPROACH FOR A MULTI-OBJECTIVE AND MULTI-MODE RESOURCE INVESTMENT PROJECT SCHEDULING PROBLEM UNDER FUZZY-STOCHASTIC ENVIRONMENTS

ABSTRACT

In the first part of this thesis, a multi-mode resource-constrained discrete time-cost trade-off problem (MRC-DTCTP) is discussed under deterministic environments. To solve this problem, a modified integer linear programming model (ILP) is first proposed. Afterward, a constraint programming (CP) model is also developed to solve realistic size problem instances within reasonable computation time. Several search strategies are also employed to improve performance of the proposed CP model. In addition to a comprehensive experimental study on several small/medium/large scale benchmark instances, a real-life application of an international Enterprise Resource Planning (ERP) software company is also presented. The computational analysis has shown that the proposed CP model can generate superior results within a shorter computing time than the modified ILP model, especially for relaxed project deadlines and tighter human resource availabilities.

In the second part, a novel constraint programming-based transformation approach for solving a multi-objective and multi-mode, fuzzy-stochastic resource investment project scheduling problem (FS-MRIPSP) is proposed. In fact, this approach mainly depends on a bound & decomposition principle which divides it into crisp middle, lower and upper level problems. Furthermore, stochastic nature of the problem is taken into account via a multi-scenario based stochastic programming technique. Finally, a weighted additive fuzzy goal program (WAFGP) is embedded into the proposed approach to produce compromise fuzzy project schedules which trade-off between the expected values of project makespan and total resource costs. Computational results have shown that the proposed CP-based approach outperforms the similar ILP-based approach in terms of both solution quality and computational time.

Keywords: Multi-mode resource-constrained discrete time-cost tradeoff problem, multi-objective resource investment project scheduling problem, constraint programming, integer linear programming, fuzzy mathematical programming, scenario-based stochastic programming, ERP implementation projects

BULANIK - STOKASTİK ORTAMLARDA ÇOK AMAÇLI VE ÇOK MODLU BİR KAYNAK YATIRIMLI PROJE ÇİZELGELEME PROBLEMİ İÇİN KISIT PROGRAMLAMA TABANLI BİR DÖNÜŞTÜRME YAKLAŞIMI

ÖZ

Bu tezin ilk bölümünde, çok modlu, kaynak kısıtlı, kesikli zaman-maliyet ödünleşim problemi (ÇMKK-KZMÖP), belirlilik altında ele alınmıştır. Problemin çözümü için ilk olarak, yeniden düzenlenmiş doğrusal tamsayılı programlama (DTP) modeli önerilmiştir. Daha sonra, bir kısıt programlama (KP) modeli geliştirilerek, gerçek hayata uygun boyutlardaki problemlerin, makul süreler içerisinde çözümü hedeflenmiştir. Çeşitli arama stratejileri de uygulanarak, önerilen KP modelinin performansı arttırılmaya çalışılmıştır. Küçük/Orta/Büyük boyutlu test problemleri üzerinde gerçekleştirilen bir deneysel çalışmanın yanı sıra, uluslararası bir Kurumsal Kaynak Planlaması (KKP) yazılım firmasına ait bir gerçek hayat uygulamasına da yer verilmiştir. Hesaplamalı analizler göstermektedir ki, önerilen KP modeli ile özellikle gevşetilmiş teslim zamanı ve sıkı insan kaynağı kısıtları altında, DTP modeline göre, daha kısa süreler içerisinde, çok daha kaliteli çözümler elde edilebilmektedir.

Tezin ikinci bölümünde ise, çok amaçlı çok modlu bulanık-stokastik kaynak yatırımlı proje çizelgeleme probleminin (ÇAÇM-BS-KYPÇP) çözümü için, KP tabanlı yeni bir dönüştürme yaklaşımı önerilmiştir. Bu yaklaşım, temel olarak bir sınır & ayrıştırma prensibine dayanmaktadır. Böylece problem, belirlilik altındaki alt, üst ve orta seviyedeki problemlere ayrıştırılabilmektedir. Probleme ait rassallık türünde belirsizlik içeren bileşenler de senaryo tabanlı stokastik programlama tekniği kullanılarak ele alınmıştır. Proje tamamlanma zamanı ve toplam kaynak maliyeti gibi çelişen hedeflerin beklenen değerlerini eş zamanlı olarak ele alabilmek ve uzlaşık bulanık proje çizelgeleri üretebilmek amacıyla, KP temelli dönüştürme yaklaşımının içerisine, bir ağırlıklı toplamsal bulanık hedef programlama tekniği dâhil edilmiştir. Hesaplamalı sonuçlar göstermektedir ki, önerilen KP temelli dönüştürme yaklaşımı hem çözüm kalitesi hem de hesaplama süresi açısından, klasik DTP temelli yaklaşımlara göre çok daha iyi performans gösterebilmektedir.

Anahtar kelimeler: Çok modlu kaynak kısıtlı kesikli zaman-maliyet ödünleşim problemi, çok amaçlı kaynak yatırımlı proje çizelgeleme problemi, kısıt programlama, tamsayılı doğrusal programlama, bulanık matematiksel programlama, senaryo tabanlı stokastik programlama, ERP uyarlama projeleri

CONTENTS

Page

M.Sc THESIS EXAMINATION RESULT FORM ii
ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZ vi
LIST OF FIGURES xi
LIST OF TABLES xiii

CHAPTER ONE - INTRODUCTION 1

1.1 Background and Research Motivations 1
1.2 Contributions and Framework of the Thesis 4
1.3 Outline of the Thesis 6
1.4 Produced Publications Based on Research in the Scope of This Thesis 7

CHAPTER TWO - LITERATURE REVIEW 9

2.1 A Brief Literature Review on Resource-Constrained Discrete Time-Cost Tradeoff Problems 9
2.2 Literature Survey on Project Scheduling Problems under Mixed Fuzzy-Stochastic Uncertainties 17

CHAPTER THREE - CONSTRAINT PROGRAMMING APPROACH FOR A MULTI-MODE RESOURCE-CONSTRAINED DISCRETE TIME-COST TRADEOFF PROBLEM 23

3.1 Problem Description and Mathematical Formulations of the MRC-DTCTP 23
3.1.1 The Modified Integer Linear Programming (ILP) Model Formulation 25
3.1.2 The Proposed Constraint Programming (CP) Model Formulation 27
3.1.2.1 Variable and Value Ordering Strategies 30

3.2 Computational Study 32
3.2.1 Generating Benchmark Instances 32
3.2.2 Comparison of the Computational Results 35
3.3 A Case Study of POM Module Group Implementation Project of an ERP Software 43
3.3.1 Company Information 43
3.3.2 Data Description for the Real-Life Application 45
3.3.3 Comparison of the Computational Results and Scenario Analysis for Managerial Insights 53

CHAPTER FOUR - CONSTRAINT PROGRAMMING-BASED TRANSFORMATION APPROACH FOR A MULTI-OBJECTIVE & MULTI-MODE RESOURCE INVESTMENT PROJECT SCHEDULING PROBLEM UNDER MIXED FUZZY-STOCHASTIC ENVIRONMENTS 59

4.1 Problem Description and Mathematical Formulation of FS-MRIPSP 59
4.1.1 Integer Linear Programming (ILP) Model Formulation of the Multi-Objective FS-MRIPSP 61
4.1.2 Constraint Programming (CP) Model Formulation of the Multi-Objective FS-MRIPSP 64
4.2 The Proposed ILP & CP-Based Transformations with a Bound & Decomposition Principle 67
4.2.1 ILP-Based Transformation Approach with a Bound & Decomposition Principle 67
4.2.2 CP-Based Transformation Approach with a Bound & Decomposition Principle 73
4.3 Illustrative Example of Classical MRCPSP under Mixed Fuzzy & Stochastic Uncertainties 77
4.4 Weighted Additive Fuzzy Goal Program (WAFGP) for the Multi-Objective FS-MRIPSP 84
4.5 A Case Study of POM Module Group Project Implementation of an ERP Software 90

4.5.1 Data Description for Case Study of the FS-MRIPSP.....90
4.5.2 Fuzzy-Stochastic Multi-Objective Optimization Results of the Proposed ILP and CP-Based Transformation Approaches 101

CHAPTER FIVE - CONCLUSIONS & FUTURE RESEARCH DIRECTIONS 117

5.1 Summary of the Thesis and Concluding Remarks 117
5.2 Future Research Directions.....119

REFERENCES 121

APPENDICES.....139

LIST OF FIGURES

Page

Figure 3.1 Summary of the main characteristics/features of the MRC-DTCTP. 24
Figure 3.2 Sales, R&D and consulting network of the ERP company all over the world 44
Figure 3.3 Common steps of the project implementation methodology of the ERP software company 45
Figure 3.4 The AON project network diagram for the POM module group implementation process of an ERP software 49
Figure 3.5 Time-cost tradeoff curves for the POM module group implementation project of the ERP software product 58
Figure 4.1 Summary of characteristics/features of the FS-MRIPSP 60
Figure 4.2 AON project network diagram with fuzzy durations and the EFTs and LFTs 78
Figure 4.3 Scenario tree for stochastic capacities of the resources and resource requirements of the project activities 80
Figure 4.4 Comparison of membership functions for fuzzy expected project makespan provided by ILP & CP-based approaches 80
Figure 4.5 Gantt chart of a fuzzy project schedule under high resource capacities & low requirements (scenario#11) 82
Figure 4.6 Gantt chart of a fuzzy project schedule under low resource capacities & low requirements (scenario#1) 82
Figure 4.7 Gantt chart of a fuzzy project schedule under high resource capacities & high requirements (scenario#16) 82
Figure 4.8 Gantt chart of a fuzzy project schedule under low resource capacities & high requirements (scenario#6) 82
Figure 4.9 The renewable resource profiles under high capacity & low requirements (scenario#11) 83
Figure 4.10 The renewable resource profiles under low capacity & high requirements (scenario#6) 83

Figure 4.11 Membership functions of fuzzy goals in the middle level problem: a) Project makespan, b) Resource costs 85
Figure 4.12 Flowchart of the proposed bound & decomposition-based solution methodology for multi-objective FS-MRIPSP 89
Figure 4.13 Scenario tree generation for the stochastic renewable/non-renewable resource requirements 99
Figure 4.14 Gantt chart of a fuzzy compromise project schedule with minimum makespan under scenario#11 (w1 = 0.6 & w2 = 0.4) 112
Figure 4.15 Gantt chart of a fuzzy compromise project schedule with maximal makespan under scenario#1 (w1 = 0.6 & w2 = 0.4) 113
Figure 4.16 The resource profile of the process experts (human resources) for the optimistic case under scenario#11 (w1 = 0.6 & w2 = 0.4) 114
Figure 4.17 The resource profile of the ERP Company's consultants (human resources) for the most likely case under scenario#1 (w1 = 0.6 & w2 = 0.4) 115
Figure 4.18 Fuzzy availability levels of ERP company's consultants under different scenarios 116
Figure 4.19 Fuzzy availability levels of human resources (process experts) under different scenarios 116
Figure 4.20 Availability levels of project funds/budget under different scenarios ..116

LIST OF TABLES

Page

Table 1.1 Framework of this thesis 6
Table 2.1 Summary of the literature review on different variants of the resource-constrained discrete time-cost trade-off problems (RC-DTCTPs) 15
Table 2.2 Summary of the literature review on project scheduling problems under mixed fuzzy and stochastic uncertainties 21
Table 3.1 Mathematical nomenclature for the modified ILP model formulation 25
Table 3.2 Additional mathematical notation for the CP model formulation of the MRC-DTCTP 28
Table 3.3 Comparison of the results on different search strategies for randomly chosen benchmark instances 36
Table 3.4 Computational results of the proposed ILP and CP models for the small, medium, and large-sized benchmark instances 38
Table 3.5 The project data for the ERP software product's production and operations management (POM) module implementation process 47
Table 3.6 Human resources (in man-days) demanded by the project activities under different execution modes. 52
Table 3.7 Comparison of the optimization results provided by the proposed CP and ILP models under different project deadlines and human resources availabilities 55
Table 4.1 Mathematical nomenclature for the integer programming (ILP) formulation of the FS-MRIPSP 61
Table 4.2 Additional mathematical notation for the CP formulation of the FS-MRIPSP 64
Table 4.3 Stochastic project data with discrete probabilities: Resources' capacities and resource requirements of the activities 78
Table 4.4 Optimization details for ILP & CP models of the crisp equivalent lower, middle and upper problems 79
Table 4.5 The corresponding fuzzy payoff matrix 85

Table 4.6 Fuzzy data for the production and operations management (POM) module group implementation project of the ERP system 92
Table 4.7 Stochastic human resource requirements of the project activities under different execution modes 96
Table 4.8 Stochastic implementation costs ($) with discrete probabilities (%) under different execution modes 100
Table 4.9 Fuzzy payoff matrix provided by the ILP-based transformation approach 101
Table 4.10 Fuzzy payoff matrix provided by the CP-based transformation approach 101
Table 4.11 Details of single-objective optimization results obtained from the ILP-based approach 102
Table 4.12 Details of single-objective optimization results obtained from the CP-based approach 103
Table 4.13 Comparative study for testing different search strategies on the middle level problem of the CP-based transformation approach in the multi-objective optimization phase 105
Table 4.14 Comparison of the multi-objective optimization results of the proposed ILP and CP-based approaches with WAFGP.....107
Table 4.15 Fuzzy project makespan values under different scenarios ($w_1 = 0.6$ & $w_2 = 0.4$) 110

CHAPTER ONE
INTRODUCTION

1.1 Background and Research Motivations

Nowadays, enterprises' various products and services have increased rapidly from day to day because of the competitive environment, more customization, shorter product life cycles, and technological developments. For this reason, most of the corporate enterprises have turned into complex business areas where many projects are managed simultaneously. However, today's higher project investment costs, long implementation times and project scheduling difficulties have made it difficult for many enterprises to survive in a costly and competitive business environment. Therefore, it is crucial to generate effective and efficient project plans/schedules and trade-off optimization between total project time and costs is required to maximize the overall project benefits. Based on this motivation two different advanced project scheduling problems are taken into consideration in the scope of this thesis.

In the first part of this thesis, a multi-mode resource-constrained discrete time-cost trade-off problem (MRC-DTCTP) which combines the classical multi-mode resource-constrained project scheduling (MRCPSP), and the discrete time-cost trade-off problems (DTCTP) is discussed. Actually, both problems are very well-known project scheduling problems in the literature. Nevertheless, the DTCTP does not directly consider the limited renewable and/or nonrenewable project resources. Similarly, the classical MRCPSP also doesn't consider the trade-off between the project makespan and total resource usage costs. Thus, the integrated MRC-DTCTP can be seen as a more practical problem for real-life settings. The main objective of the MRC-DTCTP is to minimize total project cost consisting of direct and indirect costs of the renewable project resources under a pre-determined project deadline. In addition to the alternative execution modes, each project activity can also be realized under two execution methods, i.e., normal and crashing ways in the MRC-DTCTP. In other words, some project activities can be realized under a selected mode of the crashing way by an imposing additional direct cost to shorten durations. In this case, direct cost results

from using project resources during the crashing duration of these activities. On the other hand, project makespan value is closely related to the project indirect cost. For that reason, to select the most appropriate execution modes, it is also vital to decide on the execution methods for each project activity to balance the trade-off between these direct and indirect project costs. Because, in order to satisfy the targeted project deadline, the direct cost will increase, and indirect cost will decrease as the project makespan decreases. Due to these different problem characteristics, the MRC-DTCTP has a more complex structure than the classical MRCPSPs and DTCTPs. These additional problem features also make its solution much more difficult besides its NP-hard nature. Although many studies are available on the MRCPSP and DTCTP, there is still a lack of research in the literature on MRC-DTCTP. Furthermore, there is a scarcity of exact solution methods for the MRC-DTCTP despite some available heuristic and meta-heuristic based solution approaches in the literature. For this reason, development of exact solution methods is needed to solve MRC-DTCTPs which have some special characteristics/features.

In the second part of this thesis, a multi-objective and multi-mode fuzzy-stochastic resource investment project scheduling problem (FS-MRIPSP) is introduced firstly in the literature. Because, many project scheduling applications may contain different types of uncertainties such as fuzziness, stochasticity, dynamism etc. simultaneously in real-life settings. However, there is still a lack of studies in the literature on fuzzy-stochastic and/or fuzzy-random project scheduling problems which are composed of mixed fuzzy and stochastic project parameters/variables. Unlike the classical RCPSPs, capacities or availability levels of the project resources are also defined as decision variables instead of project parameters or input data as in RIPSPs or resource availability cost problem (RACP). On the other hand, most of the classical transformation approaches in fuzzy mathematical programming literature utilized standard fuzzy arithmetic operations or Zadeh's extension principle with fuzzy ranking techniques while converting a fuzzy optimization problem into its crisp equivalent form. However, some recent studies (Subulan, 2020; Sotoudeh-Anvari, 2020; Baykasoğlu & Subulan, 2017, 2019) have proven that standard fuzzy arithmetic-based transformation approaches may cause questionable results or unrealizable solutions for

many real-life engineering applications. For instance, several risky situations may arise in project scheduling and transportation applications under fully fuzzy environments (Baykasoğlu & Subulan; 2017; 2019; Subulan, 2020). Additionally, these standard fuzzy arithmetic-based transformation approaches are mainly developed depending upon integer linear programs (ILPs). In other words, most of the transformation approaches were mainly focused on integer linear programs (ILPs) with fuzzy parameters/variables in fuzzy mathematical programming literature. However, these ILP-based solution approaches may be inadequate for solving large-scaled combinatorial fuzzy optimization problems, like project scheduling under fuzzy-stochastic environments. On the other hand, ILP-based solution approaches are not able to handle large-sized combinatorial fuzzy optimization problems like resource constrained project scheduling problem (RCPSP) and its variant, i.e., resource investment project scheduling problem (RIPSP). Fortunately, many studies available in the literature have demonstrated that CP-based solution methodologies are able to produce promising results for deterministic project scheduling problems (Liess & Michelon, 2008; Berthold et al., 2010; Li & Womer 2015; Hauder et al., 2020; Kong & Dou, 2021). However, none of them handled such a fully uncertain project scheduling problem under a mixed fuzzy and stochastic environment. It should be mentioned here that the problem size and therefore, model size including the number of constraints and decision variables will increase significantly after realizing a transformation from fuzzy-stochastic version to crisp equivalent form. For this reason, it may be useful to develop a CP-based solution approach to convert the fuzzy-stochastic model into its crisp equivalent form. Considering this research gap, a novel CP-based transformation approach with a bound & decomposition principle is introduced in this thesis. In fact, some project parameters such as the usage costs of renewable resources, activity durations and therefore, their earliest/latest finishing times are considered as uncertain and stated by triangular fuzzy numbers. By the way, stochastic nature of some project parameters, i.e., resource requirements of the renewable/non-renewable resources are also considered by a multi-scenario based stochastic programming approach. In addition to these fuzzy and stochastic project parameters, completion times of the project activities are also defined as mixed fuzzy-stochastic decision variables because of the fuzzy activity durations and

aforementioned stochastic project parameters. Furthermore, availability levels of both renewable and non-renewable resources are also stated as mixed fuzzy-stochastic variables which represent fuzzy resource capacities under different scenarios. Therefore, the proposed transformation approach is also able to generate fuzzy project schedules and resource availabilities under different scenarios.

1.2 Contributions and Framework of the Thesis

The contributions of this thesis are divided into two parts for the MRC-DTCTP and FS-MRIPSP, respectively. Framework of this thesis is also given in Table 1.1.

For the MRC-DTCT problem in Chapter 3, the following contributions are made:

- A modified integer linear programming (ILP) model of the MRC-DTCTP is proposed based on the original integer non-linear programming (INLP) model of Wuliang and Schengen (2009) in order to obtain a more simplified and explicit mathematical model formulation with a linear objective function. Afterward, a new constraint programming (CP) model for the MRC-DTCTP is also developed by considering crashable execution modes for each project activity.
- In order to improve the performance of the proposed CP model, different search strategies are tested on several randomly generated benchmark instances. According to the extensive computational study, the most appropriate search strategies are identified and used to solve larger-scale problem instances. The performances of the proposed CP and ILP models are also compared based on these benchmark instances. To show validity and practicality of the proposed CP model, a real-life case study is also presented for an international ERP software company's POM module group implementation project. As well as the extensive experimental study on various benchmark instances, optimization results provided by both modified ILP and CP models are also compared based on this real-life application. The professional human resources of the ERP software

company and their knowledge/expertise are considered as the limited renewable project resources. For that reason, further what-if analysis is also conducted to visualize the time-cost trade-off curves under various human resource availabilities and project deadlines. Thus, some important managerial insights are obtained and offered to the company's ERP project managers.

To the best of our knowledge, there is no similar research in the literature that handles the similar MRC-DTCTP through a CP approach and compares its optimization results with an ILP approach.

For the FS-MRIPSP problem in Chapter 4, the following contributions are made:

- A fully uncertain, multi-objective and multi-mode RIPSP with fuzzy and stochastic parameters and fuzzy-stochastic decision variables are introduced. In fact, all of the project input data and outputs are stated as uncertain and several fuzzy project schedules with fuzzy availabilities of the renewable/non-renewable resources are generated under different scenarios.
- A new CP-based transformation approach with a bound & decomposition principle is developed to solve a real-life case study of POM module group implementation project of an international ERP software company under a reasonable computing time. In multi-objective optimization phase, different search strategies are tested to obtain better quality solutions within shorter run times. Computational results of the proposed CP-based solution approach are also compared to the results of a similar ILP-based method. It was proven that CP-based transformation approaches may provide superior results in terms of both solution quality and computational time under mixed fuzzy-stochastic environments.

To the best of our knowledge, there is no available study in the literature on a CP-based transformation approach for solving fuzzy or fuzzy-stochastic mathematical programming problems. Actually, this is the main contribution of this thesis to the fuzzy mathematical programming literature.

Table 1.1 Framework of this thesis

Chapter	Chapter 3	Chapter 4
Problem Definition	Multi- mode resource constrained discrete time-cost tradeoff problem under deterministic environment	Multi-objective & multi-mode resource investment project scheduling problem under mixed fuzzy-stochastic uncertainties
Solution Methodologies	Integer linear programming, Constraint programming	Integer linear programming-based transformation approach with a bound & decomposition principle, Constraint programming-based transformation approach with a bound & decomposition principle, Weighted additive fuzzy goal programming
Fuzzy parameters	N.A	EFT and LFT values, Activity durations, Unit usage cost of renewable resources
Stochastic parameters	N.A	Required amount of renewable/non-renewable resources
Uncertain variables	N.A	Availability levels of the renewable/ non-renewable resources (Fuzzy-Stochastic) All of the activity execution times (Fuzzy-Stochastic)
Objective Functions	Minimize total direct and indirect project costs	Minimize expected values of the project makespan and total usage costs of the project resources

1.3 Outline of the Thesis

This thesis consists of totally 5 chapters. First chapter is dedicated to presentation of the introduction, research motivations, goals and contributions of this thesis. The remaining chapters are organized as follows.

In Chapter 2, literature reviews on both resource-constrained discrete time-cost tradeoff problems (RC-DTCTPs) and project scheduling problems under mixed fuzzy-stochastic uncertainties are presented respectively.

In Chapter 3, the problem description and mathematical programming formulations of the MRC-DTCTP are given. In other words, its integer linear programming and constraint programming models are developed by assuming that all of the model parameters and decision variables are known in advance. Afterwards, an extensive computational study is presented based on several randomly generated small, medium, and large-sized benchmark instances. Comparisons of the computational results of the

CP and ILP models are also carried out. Then, the proposed CP and modified ILP models are also implemented to an international ERP software company's real-life POM module implementation project. In order to provide some managerial implications on project time-cost tradeoffs, some scenarios under different human resource availabilities and project deadlines are also considered.

In Chapter 4, problem definition and mathematical programming formulations of the FS-MRIPSP are given. Then, fundamentals of the proposed CP and ILP-based transformation approaches with a bound & decomposition principle are described. Afterward, the proposed approaches are first exemplified based on a basic example of a classical MRCPSP under fuzzy-stochastic environments. Thereafter, a WAFGP is also embedded in the proposed ILP and CP-based approaches so as to handle multi-objective nature of the FS-MRIPSP. Finally, the validity and practicality of the proposed CP-based transformation approach is also tested on a real-life ERP implementation project data and comparison of the computational results is carried out.

Chapter 5 presents the summary of this thesis, concluding remarks and future research directions.

1.4 Produced Publications Based on Research in the Scope of This Thesis

The following scientific research papers have generated throughout the preparation of this thesis. These papers are either published by proceeding books of some conferences or have the status of under review and publication in some international journals. These papers created were also assigned to certain thesis sections.

Chapter 1, Section 1.1-1.2; Chapter 2, Section 2.1; Chapter 3, Section 3.1-3.3; Chapter 5, Section 5.1-5.2:

- Çakır, G., Asılkefeli, C., Yıldız, S.T., & Subulan, K. (2020). Constraint programming approach for a multi-mode resource constrained discrete time-cost tradeoff problem. In *Abstract Book of IES'20 International Engineering Symposium Engineering Applications in Industry (Virtual)*, Turkey, 5-6 &10-

13 December, from http://ies.idu.edu.tr/wp-content/uploads/2021/06/IES_20-OzetMetinKitabi_v4-1.pdf.

- Çakır, G., Subulan, K., Topaloğlu Yildiz S., &. Asılkefeli, C. (2021). Constraint programming approach for a multi-mode resource constrained discrete time-cost tradeoff problem, (Submitted to Computers & Industrial Engineering Journal, Under review status).

Chapter 1, Section 1.1-1.2; Chapter 2, Section 2.2; Chapter 4, Section 4.1-4.5; Chapter 5, Section 5.1-5.2:

- Subulan, K., & Çakır, G. (2020). Constraint programming-based transformation approach for a multi-objective & multi-mode resource investment project scheduling problem under mixed fuzzy-stochastic uncertainties. In *Full Paper Book of IES'20 International Engineering Symposium Engineering Applications in Industry (Virtual)*, Turkey, 5-6 &10-13 December, from http://ies.idu.edu.tr/wp-content/uploads/2021/06/IES_20-OzetMetinKitabi_v4-1.pdf.
- Subulan K., & Çakır G. (2021). Constraint programming-based transformation approach for a multi-objective & multi-mode resource investment project scheduling problem under mixed fuzzy-stochastic uncertainties, (Submitted to Soft Computing Journal, Accepted and published).

CHAPTER TWO
LITERATURE REVIEW

2.1 A Brief Literature Review on Resource-Constrained Discrete Time-Cost Tradeoff Problems

This section presents a brief literature survey on the RC-DTCTPs. Indeed, there are many variants of the project scheduling problems in the literature. The classical resource-constrained project scheduling problem (RCPSP) is the most fundamental and one of the most widely studied project scheduling problem in the literature (Özdamar & Ulusoy, 1995; Brucker et al., 1999; Hartmann, 2001; Kolisch & Padman, 2001; Demeulemeester et al., 2002; Kassandra et al., 2018). However, the RCPSP has been redesigned many times to meet highly changing project planning and scheduling expectations. For instance, the multi-mode resource-constrained project scheduling problem (MRCPSP) has become one of the most extensively studied generalizations of this problem (Wang & Fang, 2011; 2012; Li & Zhang, 2013; Okada et al., 2014; Schnell & Hartl, 2017; Zamani, 2019). In the MRCPSPs, each activity can be performed in more than one way denoting modes and describing different combinations for activity durations and resource requirements. It is essential to select the most appropriate mode for each project activity to utilize the available project resources and time more effectively in this situation. Apart from these RCPSPs/MRCPSPs, discrete time–cost tradeoff problem (DTCTP), which was first introduced by Hindelang and Muth (1979), is another popular project scheduling problem where activity durations are discrete, a no-increasing function of the amount of a single nonrenewable resource dedicated to it (Wuliang & Chengen, 2009). In the DTCTPs, it is commonly assumed that the tradeoff follows a discrete non-increasing pattern, i.e., more resources can be allocated to expedite a project activity at a larger cost (Vanhoucke & Debels, 2007). Moreover, there are three different versions of the DTCTP: deadline, budget and time-cost curve problems. The deadline problem aims to minimize the total cost for satisfying a pre-specified deadline whereas the budget problem minimizes the project makespan under a budget limitation. On the other hand, the time–cost curve problem attempts to determine the entire time–cost profile over a

set of feasible project durations (Herroelen et al. 1999; Vanhoucke et al., 2002; Wuliang & Chengen, 2009). Besides, the MRC-DTCTP handled throughout this thesis combines these classical problems, i.e., the MRCPSP and DTCTP, and its solution is far more challenging due to its complex problem structure and NP-hardness.

An overview of the recent articles which focused on the DTCTP and MRC-DTCTP can be summarized as follows: Sonmez and Bettemir (2012) proposed a hybrid evolutionary algorithm that integrates genetic algorithm (GA), simulated annealing (SA) and quantum simulated annealing techniques (QSA) for solving the DTCTP in order to minimize project direct and indirect costs. Similarly, Aminbakhsh and Sonmez (2016) developed a discrete particle swarm optimization (DPSO) algorithm for solving larger-sized DTCTP instances. Liu et al. (2020) applied a discrete symbiotic organisms search (DSOS) algorithm to obtain high quality solutions for the DTCTP within reasonable computation time. They also incorporated the penalty and incentive costs into the objective function to take into account the difference between the project deadline and completion time. In order to cope with uncertainty in activity cost parameters, Hazır et al. (2011) proposed three alternative optimization models for a robust DTCTP including interval data. The first model was solved exactly by Benders decomposition procedure and the solutions of the remaining two models were provided approximately by the Tabu search algorithm. Said & Haouari (2015) also addressed the robust DTCTP with stochastic activity durations while minimizing project activity costs and linear tardiness costs. After that, they developed a hybrid solution methodology based on a mixed-integer programming model, single-variable global optimization and Monte-Carlo simulation approaches. Apart from these articles which mainly focused on the classical DTCTP, Vanhoucke et al. (2002) extended the DTCTP by incorporating time-switch constraints. These constraints assign specific starting times to the project activities and force them to be active and inactive throughout the specified time periods. They implemented an exact branch-and-bound procedure and a heuristic algorithm for solving the DTCTP with time-switch constraints. Vanhoucke (2005) also improved the branch-and-bound algorithm developed in the formerly study of Vanhoucke et al. (2002) which can produce superior results within shorter times. Furthermore, Vanhoucke & Debels (2007) also

concentrated on three versions of the DTCTP, i.e., the DTCTP with time-switch constraints, the DTCTP with work continuity constraints, and the DTCTP with net present value maximization. They applied a new meta-heuristic approach based on the Tabu search algorithm for the solution of these problems.

It should be emphasized here that all of these previously mentioned studies that mainly focused on different variants of the DTCTPs did not consider the capacity constraints of the project resources. Unfortunately, there is still a lack of literature that concentrates on the integration the RCPSPs and DTCTPs. Demeulemeester et al. (1996) conducted one of the pioneering studies on the DTCTP with the limited capacity of the non-renewable project resources. They described two algorithms based on dynamic programming logic to optimize three objectives i.e., minimize project duration under fixed resource availability, minimize total resource consumption under a pre-specified project deadline and produce effective time-resource profiles over the feasible project durations. Nader et al. (2011) developed a two-phase model for solving multi-mode resource-constrained time-cost tradeoff (MRCTCT) problem in which the effects of time-cost trade-off and resource-constrained allocations are considered simultaneously. In the first phase, they implemented a GA-based time-cost trade-off analysis to select the proper execution modes. After that, some lightweight analytical computations are performed in the second phase to achieve minimum project duration with the chosen modes and satisfy the resource capacity constraints. Finally, they provided a multi-attribute fitness function that the decision maker can change according to his/her project time-cost preferences under renewable resource limitations. Ghoddousi et al. (2013) proposed a non-domination-based genetic algorithm (NSGA-II) to solve the MRC-DTCTP and resource allocation and leveling problems (RLP) concurrently. They targeted to minimize three objectives, i.e., renewable resources movement deviation on the project makespan, project duration and total project cost composed of direct, indirect and penalty costs. Two-phase differential evolution (DE) algorithm was developed by Cheng and Tran (2014) for a multi-objective optimization of the time-cost tradeoff resource-constrained (TCTRC) project scheduling problem. Indeed, two sub-systems namely the TCT and resource-

constrained scheduling were integrated to simultaneously minimize the project makespan and project cost.

Unlike the previous studies on deterministic project scheduling environments, Li et al. (2020) addressed a multi-objective time-cost-robustness trade-off project scheduling problem under uncertainty of the activity durations. They implemented an epsilon-constraint method-based GA to generate Pareto schedule sets and handled three objectives instantaneously, i.e., minimizing the project makespan, total cost and maximizing the robustness. Cheng & Weng (2009) proposed a two-phase GA approach for solving the resource-constrained time-cost tradeoff (RCTCT) problem. This approach allows interruption and overlap of the project activities while minimizing the project's direct/indirect and the interruption costs. Menesi et al. (2013) proposed a constraint programming approach that can handle project deadline and resource capacity constraints more easily for solving larger sized RCTCTP instances ranging from 10 to 2000 activities. They targeted to minimize the sum of direct/indirect project costs, penalties, and incentive costs. For solution of the RC-DTCTPs, Sonmez et al. (2016) presented a critical sequence crashing heuristic (CSCH) algorithm and applied it to a realistic size construction project with limited renewable resources and project deadline constraints. According to their computational study, the CSCH outperforms the state-of-the-art methods of Menesi et al. (2013), Hegazy & Menesi (2012) and Chen & Weng (2009) in terms of both solution quality and computing time.

In contrast to the formerly discussed studies focusing on only the renewable project resources, Jing-wen & Hui-fang (2010) considered both of the renewable and non-renewable project resources while dealing with multi-resource MRC-DTCTP. They developed an improved GA-based solution approach to minimize total project cost under the contract duration. Indeed, total project cost involves the direct cost of renewable/nonrenewable resources, indirect cost, lateness penalty and earliness bonus.

Ahn & Erenguc (1998) introduced a multi-pass algorithm for the RCTCTP by considering multiple crashable modes. In that problem, the duration/cost of the project activities were specified by choosing the proper mode and the duration reduction

(crashing) can also be applicable for the chosen mode to minimize the sum of all activity costs and penalty cost. Similarly, Wuliang & Chengen (2009) also considered the crashing ways of the project activities by predefining the resource prices which cause additional direct costs while shortening their durations. In the solution phase, they developed an improved GA approach and verified its results by using an exact solution method based on the branch-and-bound algorithm. Afruzi et al. (2013) extended the MRC-DTCTP by incorporating per unit time usage cost and direct cost of the project resources whose values varied according to the chosen execution mode and method, i.e., normal/crashing ways. They developed a multi-objective meta-heuristic algorithm namely an adjusted fuzzy dominance genetic algorithm (AFDGA) to balance the total project cost and makespan objectives. In addition to minimize the typical project cost and time objectives, Afruzi et al. (2014) also presented a resource-constrained discrete time-cost-quality trade-off problem (DTCQTP) which aims to maximize project quality under resource and mode identity constraints. They employed a multi-objective imperialist competitive algorithm (MOICA) and compared its effectiveness with well-known algorithms such as NSGAII, PESAII-clustering; PESAII-grid based; and SPEA2. Xu et al. (2012) developed a multi-objective decision making model for solving a discrete time–cost–environment trade-off problem (DTCETP) in which total project duration was stated as a fuzzy variable. They utilized an expected value operator with an optimistic–pessimistic index in order to defuzzify the uncertain normal and crashing durations. Afterward, they applied a fuzzy adaptive hybrid GA approach to achieve compromise solutions that satisfy the weighted-sum objective function consisting of four different goals, i.e., minimizing the total project cost, duration, crashing cost, and environmental impact, respectively.

Finally, all of the studies mentioned above can be summarized and classified as in Table 2.1 according to the problem type and different characteristics/features (including budget/deadline limitations, renewable/non-renewable resource constraints, multiple activity modes, crashable modes, uncertain project parameters/variables and project goals) and solution methodology. It can be seen in Table 2.1 that most of the existing studies have concentrated on the heuristic/metaheuristic-based solution methods because of the NP-hard nature and

complex structure of the stated problems. However, good quality solutions can also be generated using exact solution approaches like CP (Menesi et al., 2013). For that reason, a CP approach is employed in this thesis for solving the present MRC-DTCTP. Unlike the study of Menesi et al. (2013), execution methods, i.e., normal and crashing ways and the relevant direct/indirect costs of the renewable project resources are also considered in this thesis. According to our extensive computational study and the real-life application given in the latter sections, promising results can also be obtained using the proposed CP approach for the realistic size problem instances. To the best of our knowledge, there is no similar research in the literature that handles the similar MRC-DTCTP through a CP approach and compares its optimization results with an ILP approach.

Table 2.1 Summary of the literature review on different variants of the resource-constrained discrete time-cost trade-off problems (RC-DTCTPs)

Article	Problem Characteristics									Solution Methodology
	Problem Type	Limitations		Resource Constraints		Multiple Modes	Crashable Modes	Uncertainty	Project Goals	
		Deadline	Budget	Renewable	Non-Renewable					
Liu et al. (2020)	DCOP					✓			Minimize Project Costs	DSOS (Discrete Symbiotic Organisms Search)
Aminbakhsh & Sonmez (2016)	DTCTP	✓				✓			Minimize Project Costs	DPSO (Discrete Particle Swarm Optimization)
Sonmez & Bettemir (2012)	DTCTP					✓			Minimize Project Costs	A Hybrid Strategy (GAs-Genetic Algorithm; SA-Simulated Annealing; QSA-Quantum Simulated Annealing Techniques)
Hazır et al. (2011)	DTCTP	✓				✓		✓	Minimize Project Costs	Exact approach and Heuristic Algorithms
Wuliang & Chengen (2009)	MRC-DTCPT	✓	✓	✓		✓	✓		Minimize Project Costs	GA (Genetic Algorithm)
Ghoddousi et al. (2013)	MRC-DTCRO			✓		✓			Minimize Project Makespan, Project Costs & Resources Moment Deviations	NSGA-II (Non-domination Based Genetic Algorithm)
Afruzi et al. (2013)	MRC-DTCTP			✓		✓	✓		Minimize Project Makespan & Project Costs	AFDGA (Adjusted Fuzzy Dominance Genetic Algorithm)
Demeulemeester et al.(1996)	DTCTP	✓			✓	✓			Minimize Project Makespan, Resource Usages & Project Costs	Dynamic Programing
Said & Haouari (2015)	DTCTP-T					✓		✓	Minimize Total Activity Costs & Tardiness	A Hybrid Approach Based on Mixed-Integer Programing & Single-Variable Global Optimization & Monte-Carlo Simulation
Li et al. (2020)	DTCRTP	✓	✓	✓		✓		✓	Minimize Project Makespan, Project Costs & Maximize Robutness	ECGA (The Epsilon-Constrained Method Based Genetic Algorithm)
Cheng & Tran (2014)	TCTRC			✓		✓			Minimize Project Makespan & Project Costs	Two-phase Differential Evaluation Algorithm
Jing- wen & Hui-fang (2010)	MRC-DTCTP			✓	✓	✓			Minimize Project Costs	An Improved GA

Table 2.1 Continues

Article	Problem Characteristics									Solution Methodology
	Problem Type	Limitations		Resource Constraints		Multiple Modes	Crashable Modes	Uncertainty	Project Goals	
		Deadline	Budget	Renewable	Non-Renewable					
Nader et al. (2011)	MRCTCT			✓		✓			Minimize Project Makespan & Project Costs	GA
Afruzi et al.(2014)	MIRC-DTCQTP			✓		✓	✓		Minimize Project Makespan, Project Costs & Maximize Project Qualitiy	MOICA (Multi-Objective Imperialist Competitive Algorithm)
Sonmez et al. (2016)	RCDTCTP	✓		✓		✓			Minimize Project Costs	CSCH (A Critical Sequence Crashing Heuristic)
Menesi et al. (2013)	RCTCTP	✓		✓		✓			Minimize Project Costs	Constraint Programming
Chen & Weng (2009)	RCTCTP	✓		✓		✓			Minimize Project Costs & Activity Interruption Costs	Two-Phase GA
Xu et al. (2012)	DTCETP	✓	✓	✓	✓	✓	✓	✓	Minimize Expected Project Cost, Expected Makespan, Crashing Costs & Enviromental Impacts	Fuzzy-Based Adaptive Hybrid GA
Vanhoucke et al. (2002)	DTCTPTSC	✓				✓			Minimize Project Costs	Branch and Bound algorithm & Heuristic Procedure
Vanhoucke (2005)	DTCTPTSC	✓				✓			Minimize Project Costs	A New Branch and Bound Algorithm
Vanhoucke & Debels (2007)	DTCTP	✓				✓				A New Meta-heuristic Procedure
Ahn & Erenguc(1998)	RCPSPMCM			✓		✓	✓		Minimize Project Costs & Penalty Costs	A Multi-Pass Heuristic Algorithm
The Present Study	MRC-DTCTP	✓		✓		✓	✓		Minimize Total Direct & Indirect Project Costs	Constraint Programing & Integer Programming

DCOP-Deadline Constrained Cost Optimization Problem; DTCTP-Discrete Time-Cost Tradeoff Problem; MRC-DTCPT-Multi-Mode Resource-Constrained Discrete Time-Cost Tradeoff Problem; MRC-DTCRO-Multi-Mode Resource-Constrained Discrete Time–Cost-Resource Optimization; DTCTP-T- Discrete Time-Cost Tradeoff Problem with Tardiness Objective; DTCRTP-Discrete Time-Cost-Robustness Tradeoff Problem; TCTRC-Time-Cost Tradeoff Resource Constraint; MIRC-DTCQTP- Mode Identity Resource-constrained Discrete Time–Cost–Quality Trade-off Problem; MRCDTCTP- Multi-Resource Constrained Discrete Time-Cost Tradeoff problem; MRCTCT-Multi-Mode Resource-Constrained Time-Cost Tradeoff; RCDTCTP- Resource-Constrained Discrete Time–Cost Tradeoff Problem; RCTCT-Resource-Constrained Time-Cost Tradeoff Problem; DTCETP- Discrete Time–Cost–Environment Tradeoff Problem; DTCTPTSC-Discrete Time-Cost Tradeoff Problem with Time Switch Constraints; RCPSPMCM- Resource-Constrained Project Scheduling Problem with Multiple Crashable Modes

2.2 Literature Survey on Project Scheduling Problems under Mixed Fuzzy-Stochastic Uncertainties

In this section, literature surveys on fuzzy-stochastic and fuzzy-random project scheduling problems are presented. In fact, available researches in this field can be mainly classified into three parts, i.e., fuzzy, stochastic and fuzzy-random project scheduling problems according to the types of uncertainty (Subulan 2019; Subulan, 2020). Most of the previous researches handled the uncertain project parameters/input data by making use of probability or fuzzy set theories. Specifically, they generally considered the uncertain project parameters as stochastic input data which were defined as random variables with known probability distributions and proposed several stochastic programming based solution techniques (Malcolm et al., 1959; Jørgensen, 1999; Ashtiani et al., 2011; Artigues et al., 2013; Li & Womer, 2015; Wang et al., 2015; Tseng & Ko, 2016; Chand et al., 2016; Chakrabortty et al., 2017; Tao et al., 2018; Uysal et al., 2018; Rostami et al., 2018; Bruni et al., 2018; Chen et al., 2018; Chen, et al., 2019; Rezaei et al., 2020; Sallam et al., 2021; Balouka & Cohen, 2021; Chen et al., 2021). On the other hand, some of the other remaining researches defined the uncertain project data as fuzzy parameters and developed fuzzy mathematical programming based solution approaches (Hapke et al., 1994; Hapke & Slowinski, 1996; Ozdamar & Alanya, 2001; Pan et al., 2001; Wang, 2002; Wang, 2004; Liu et al., 2007; Long & Ohsato, 2008; Bhaskar et al., 2011; Atli & Kahraman, 2012; Masmoudi & Hait, 2013; Huang et al., 2013; Atli & Kahraman, 2014; Knyazeva et al., 2015; Kaveh et al., 2016; Sajadi et al., 2017; Khalilzadeh et al., 2017; Yousefli, 2017; Zhang et al., 2018; Roghanian et al., 2018; Birjandi & Mousavi, 2019; Mahdavi et al., 2021). In fact, there are lots of studies either on stochastic or fuzzy project scheduling problems. In other words, many existing researches have focused on only single type of uncertainty, either fuzziness or randomness. Specifically, different uncertainty types were usually handled separately by most of the former preliminary studies. However, many real-life project scheduling applications may contain different types of uncertainties concurrently. Actually, some project inputs may involve only fuzziness or randomness whereas others may cover them simultaneously. Therefore, different kinds of uncertainties in project parameters/variables should be modelled and taken

into consideration jointly. As highlighted by Alipouri et al. (2020) that some project activities have been formerly realized in several times and therefore, there are sufficient historical data related to their durations, resource requirements etc. Nevertheless, other project activities may rarely or never be performed before and therefore, they may have no historical data. In this case, randomness and fuzziness arise instantaneously and the corresponding project scheduling problem can be named as fuzzy-stochastic. Unfortunately, there is still a lack of studies in the literature on fuzzy-stochastic or fuzzy-random project scheduling problems. Actually, its literature is in its burn-in phase.

An overview of the recent articles which focused on modelling and solving hybrid fuzzy-random project scheduling problems can be summarized as follows: The fuzzy-stochastic project scheduling problem has been first initiated by Ke and Liu (2007) in the literature. They designed a hybrid intelligent algorithm based on GA for solving a project scheduling problem with mixed uncertainty of randomness and fuzziness. They supposed that activity durations will be random fuzzy variables and then, three different types of random fuzzy models, i.e., expected cost minimization, (α, β)-cost minimization and chance maximization models were proposed to satisfy several management needs. A similar project scheduling problem was also presented by Ke et al. (2012) in which the philosophy of fuzzy random programming was introduced depending upon different ranking criteria of fuzzy random variables. Ke and Ma (2014) also modelled the project time-cost trade-off problem with fuzzy random activity durations. Similar to their previous researches, they developed three types of models and designed an intelligent algorithm which combines fuzzy random simulations with GA to solve them. Huang et al. (2009) developed a hybrid intelligent algorithm based on GA and random fuzzy variables for solving an expected cost model of a software project scheduling problem with random fuzzy activity durations. A linearized MIP technique was developed by Nematian et al. (2010) based on the expected value of fuzzy-random variables, fuzzy inequality approaches and max-min operator of Bellman & Zadeh (1970) for solving a RCPSP with fuzzy-random activity durations, ready times and project deadline. A similar MIP-based transformation approach was also employed by Artykov and Atymtayeva (2015) by focusing on

fuzzy-random activity durations only. Xu and Zhang (2012) proposed a hybrid GA with a fuzzy logic controller to solve the multi-objective resource-constrained multiple project scheduling problem with fuzzy-random resource consumptions and due-dates. A new solution approach which hybridizes adaptive particle swarm optimization (PSO) with GA and a fuzzy-random simulation algorithm was developed by Gang et al. (2013) for a multi-project RCPSP including fuzzy-random activity durations and resource costs. Zhang (2014) designed a bi-level multi-objective PSO algorithm for solving a RCPSP with fuzzy-random activity durations, resource availabilities and tardiness penalty cost values. In a similar way, a combinatorial-priority-based hybrid PSO algorithm was presented by Xu and Feng (2014) for a multi-mode resource-constrained multiple project scheduling problem. The normal/crashed durations, resource consumptions, variable and crashing costs were also represented by fuzzy-random variables. Xu et al. (2015) developed a fuzzy random simulation-based bi-level global-local-neighbor PSO algorithm for a multiple decision maker project scheduling problem in a fuzzy-random environment. They aimed at maximizing the profit, minimizing the project makespan and project costs which correspond to the objectives of different project stakeholders. In order to formulate a robust optimization model of a preemption RCPSP with fuzzy-random activity durations and resource consumptions, a proactive strategy was proposed by Chen and Zhang (2016). Unlike the previously mentioned researches, they also took into account the activity splitting concept. A bi-level fuzzy random multiple objective model was developed by Zhang et al. (2019) for solving a multi-mode RCPSP in which project activities' costs, revenues and resource requirements were described as fuzzy-random variables. A self-adaptive differential evolution hyper-heuristic algorithm was developed by Alipouri et al. (2019) to minimize the expected makespan of a FS-RCPSP in which the activity durations were described as fuzzy-random variables. They also recommended that modelling other project parameters, i.e., the resource requirements of the activities as fuzzy-random variables may find a research merit in the future studies. Alipouri et al. (2020) also presented a resource flow network based mathematical model with fuzzy random variables to solve a fuzzy-stochastic RCPSP. They converted that original model to an LP with fuzzy variables/parameters by using the expected values of fuzzy-random variables. Afterward, that model was also transformed into a MILP model with

final crisp parameters/variables. To test performance of that model, several benchmark instances were also generated and solved by an optimization software. A hybrid interval programming and chance constrained optimization-based approach was developed by Subulan (2020) for solving a multi-objective and multi-mode RIPSP under mixed interval and stochastic uncertainties. This approach is also able to produce more reliable and risk-free compromise project schedules according to the risk attitude of the project manager.

Lastly, all of the previously discussed studies on fuzzy-stochastic project scheduling problems can be summarized and categorized as in Table 2.2 according to the problem type, problem characteristics (multi-objectivity, multiple execution modes, multiple projects, risk considerations etc.), uncertain parameters/variables, application study and the solution method. It was recognized in Table 2.2 that most of these existing studies have concentrated on classical RCPSPs under fuzzy-stochastic or fuzzy-random environments. Actually, there is a scarcity of researches on fuzzy, stochastic and fuzzy-stochastic RIPSPs or RACPs in the literature (Yamashita et al., 2007; Xiong et al., 2011; Xiong et al., 2014; Subulan, 2020). In other words, RIPSPs or RACPs were commonly discussed under deterministic environments (Drexl & Kimms, 2001; Hsu & Kim, 2005; Najafi & Niaki, 2006; Shadrokh & Kianfar, 2007; Ranjbar et al., 2008; Najafi & Azimi, 2009; Javanmard et al., 2017; Zhu et al., 2017; Merrikhi et al., 2018; Jedrzejowicz & Ratajczak-Ropel, 2019). Additionally, resource requirements of the project activities and capacities of the renewable and non-renewable resources were generally taken as deterministic parameters. Moreover, project schedules or project activities' starting and completion times which correspond to the decision variables took deterministic values in most of the previously mentioned studies. However, project schedules should also be defined as uncertain variables in such fully uncertain environments where all of the project parameters are stated as fuzzy, stochastic or fuzzy-stochastic data (Subulan, 2019; 2020; Baykasoğlu & Subulan, 2017; 2019).

Table 2.2 Summary of the literature review on project scheduling problems under mixed fuzzy and stochastic uncertainties

Article	Uncertainty	Problem type	Multi-objective	Multi-mode	Multiple projects	Risk attitude	Uncertain components						Application or case study	Solution methodology
							Durations	Resource requirement	Renewable resource availability	Non-renewable resource availability	Objective values (Makespan etc.)	Project schedules		
Ke & Liu (2007)	Random fuzzy	PSP					✓						N.A	GA-based hybrid intelligent algorithm
Ke et al. (2012)	Fuzzy random	PSP					✓						N.A	Fuzzy random simulations with GA
Ke & Ma (2014)	Fuzzy random	TCTP					✓						N.A	Fuzzy random simulations with GA
Huang et al. (2009)	Random fuzzy	PSP					✓						Software development	GA-based hybrid intelligent algorithm
Nematian et al. (2010)	Fuzzy random	RCPSP					✓				✓	✓	N.A	MILP-based transformation method
Artykov & Atymtayeva (2015)	Fuzzy random	RCPSP					✓				✓	✓	N.A	MILP-based transformation method
Xu & Zhang (2012)	Fuzzy random	rc-mPSP	✓		✓			✓					Water conservancy & hydropower construction	Hybrid GA with fuzzy logic controller
Gang et al. (2013)	Fuzzy random	RCPSP		✓	✓		✓						Industrial Equipment Installation	Fuzzy random simulation-based hybrid PSO & GA
Zhang (2014)	Fuzzy random	RCPSP	✓				✓	✓	✓				N.A	Bi-level multiple objective PSO algorithm

Table 2.2 Continues

Article	Uncertainty	Problem type	Multi-objective	Multi-mode	Multiple projects	Risk attitude	Uncertain components						Application or case study	Solution methodology
							Durations	Resource requirement	Renewable resource availability	Non-renewable resource availability	Objective values (Makespan etc.)	Project schedules		
Xu & Feng (2014)	Fuzzy random	RCPSP	✓	✓	✓		✓	✓					Hydropower Construction	Combinatorial priority-based hybrid PSO
Xu et al. (2015)	Fuzzy random	PSP	✓	✓			✓						Hydropower station construction	Fuzzy random simulation-based PSO
Chen & Zhang (2016)	Fuzzy random	PRCPSP					✓	✓					N.A	A transformation method with a proactive strategy
Zhang et al. (2019)	Fuzzy random	RCPSP	✓	✓				✓					Photovoltaic power generation plant	Multi-objective adaptive PSO
Alipouri et al. (2019)	Fuzzy stochastic	RCPSP					✓				✓	✓	N.A	Self-adaptive DE hyper-heuristic
Alipouri et al. (2020)	Fuzzy stochastic	RCPSP					✓				✓	✓	N.A	MILP-based transformation method
Subulan (2020)	Interval-stochastic	RIPSP	✓	✓		✓	✓	✓	✓	✓	✓	✓	ERP software implementation	Hybrid interval & chance-constrained programs
The present study	Fuzzy-stochastic	RIPSP	✓	✓			✓	✓	✓	✓	✓	✓	ERP software implementation	Constraint programming (CP)

N.A – Not Available, PSP – Project Scheduling Problem, TCTP – Project Time-Cost Trade-off Problem, RCPSP – Resource-Constrained Project Scheduling Problem, rc-mPSP – Resource-Constrained Multiple Project Scheduling Problem, PRCPSP – Preemption RCPSP, RIPSP – Resource Investment Project Scheduling Problem

CHAPTER THREE
CONSTRAINT PROGRAMMING APPROACH FOR A MULTI-MODE RESOURCE-CONSTRAINED DISCRETE TIME-COST TRADEOFF PROBLEM

3.1 Problem Description and Mathematical Formulations of the MRC-DTCTP

This chapter presents a multi-mode resource-constrained discrete time-cost tradeoff problem (MRC-DTCTP) reformulated based on the original discrete-time binary integer programming model of Wuliang and Schengen (2009). This preliminary implicit model has been modified to obtain a more simplified explicit ILP formulation with a linear objective function. In fact, Wuliang and Schengen (2009) developed an original model with nonlinear objective fuction which includes product of two decision variables. Indeed, two binary variables are multiplied while formulating the project direct cost. In this thesis, this objective function is reformulated in order to obtain a linear formula. The main aim of this problem is to generate an appropriate project schedule so that the project is completed within the pre-determined deadline while the total project cost is minimized. Besides the study of Wuliang and Schengen (2009), some other researchers also attempted to solve the same problem by developing heuristic/meta-heuristic algorithms due to its NP-hard nature (Jing- wen & Hui-fang, 2010; Nader et al., 2011; Afruzi et al., 2013). Apart from the existing research in the literature, a new CP model and a modified ILP model are proposed in this thesis to solve the MRC-DTCTP and makes their comparative computational analysis. To do this, mathematical formulations of these models are first given in this chapter. Then, their performances are also compared to each other by solving several benchmark instances in the latter section.

As in the classical MRCPSPs, a project which involves n project activities is demonstrated by an activity-on-node (AON) project network $G = (V, E)$ in which the arcs represent the precedence relations (E), whereas the nodes represent the project activities (V) for the MRC-DTCTP. Each project activity cannot be started before all its

predecessors P_j are finished. In that AON project network, the initial and terminating nodes are determined as dummy project activities. Moreover, one or more execution modes (M_j) that trade-off between the durations and resource consumptions have been identified by the project experts for each project activity. Apart from these execution modes, there are also two execution methods defined as normal and crashing ways for the project activities. The crashing way is used to shorten the activity duration in the selected mode by incurring additional direct costs when the project activities need to be speeded up to satisfy the target deadline. Two different types of costs are incurred when the project resources are consumed throughout the project duration. The first one is the direct costs based on the usage of project resources during the crashed duration of the project activities and is measured per unit time of the utilized resources. In other words, no direct costs are imposed when the project activities are executed in normal way. This type of project cost can also be considered as additional overtime costs. The second cost term is the indirect cost associated with the usage time of the resources throughout the entire project duration. Thus, the total project time or makespan is closely related to the project indirect costs. Moreover, since most knowledge intensive industries, i.e., software companies, utilize expertise human resources or workforce sources, this thesis focuses on only the renewable project resources with their relevant direct and indirect costs. In summary, the characteristics/features of the MRC-DTCTP can be displayed as in Figure 3.1.

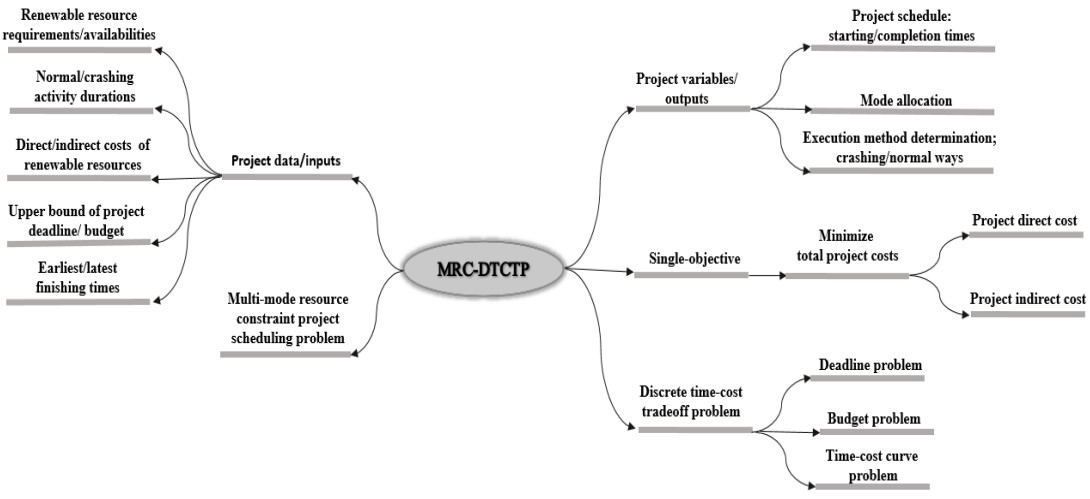


Figure 3.1 Summary of the main characteristics/features of the MRC-DTCTP.

In the next two sub-sections, the mathematical formulations of the proposed ILP and CP models are discussed in detail.

3.1.1 The Modified Integer Linear Programming (ILP) Model Formulation

The modified ILP model formulation of the MRC-DTCTP is presented in the following Equations 3.1-3.6 by using the mathematical nomenclature given in Table 3.1.

Table 3.1 Mathematical nomenclature for the modified ILP model formulation

Indices & sets	
$j \in V$	set of project activities to be scheduled
$j' \in V'$	set of dummy project activities used in the AON project network
$r \in R$	set of the renewable project resources
P_j	set of immediate predecessors of project activity j
$m \in M_j$	set of all available modes for project activity j
$m' \in M_j$	set of modes of crashing way for the project activity j
$t \in T$	set of discrete-time periods for the resource allocation to any project activity
Parameters	
c_r	direct cost per unit time of the renewable resource r
u_r	indirect cost per unit time of the renewable resource r
K_r	available capacity for the renewable resource r
d_{jm}	$\begin{cases} ed_{jm'} & \text{crashing duration of project activity } j \text{ that is performed under crashing mode } m' \\ nd_{jm} & \text{normal duration of project activity } j \text{ that is performed under normal mode } m \end{cases}$
k_{jmr}	amount of the required renewable resource r by project activity j under mode $m \in M_j$
T_{max}	project deadline
EFT_j	the earliest finishing time of the project activity j
LFT_j	the latest finishing time of the project activity j
Decision variable	
x_{jmt}	$\begin{cases} 1, & \text{if project activity } j \text{ is performed under mode } m \text{ and finished at time period } t \\ 0, & \text{otherwise} \end{cases}$

$$Minimize \sum_{j\in V} \sum_{m'\in M_j \& k_{jm'r}>0} \sum_{t\in T} \sum_{r\in R} x_{jm't} . ed_{jm'} . c_r + \sum_{r\in R} \sum_{m\in M_j} \sum_{t\in[EFT_J, LFT_J]} t . x_{Jmt} . u_r \quad (3.1)$$

Subject to;

$$\sum_{m,m'\in M_j} \sum_{t\in[EFT_j, LFT_j]} x_{jm(m')t} = 1 \quad \forall j \in V \quad (3.2)$$

$$\sum_{m,m' \in M_i} \sum_{t \in [EFT_i, LFT_i]} t.x_{im(m')t} \leq \sum_{m,m' \in M_j} \sum_{t \in [EFT_j, LFT_j]} \left(t - \{ed_{jm'}, nd_{jm}\}\right).x_{jm(m')t} \quad \forall j \in V, \forall i \in P_j \tag{3.3}$$

$$\sum_{j \in V \setminus \{V'\}} \sum_{m,m' \in M_j} k_{jm(m')r} \cdot \sum_{\tau \in [t, t+\{ed_{jm'}, nd_{jm}\}-1]} x_{jm(m')\tau} \leq K_r \quad \forall r \in R, \forall t \in T \tag{3.4}$$

$$\sum_{m,m' \in M_j} \sum_{t \in [EFT_j, LFT_j]} t.x_{Jm(m')t} \leq T_{max} \tag{3.5}$$

$$x_{jm(m')t} \in \{0,1\} \qquad \forall j \in V, \ \forall m, m' \in M_j, \forall t \in [EFT_j, LFT_j] \tag{3.6}$$

The objective function specified in Equation 3.1 aims to minimize the total project cost consisting of direct costs and indirect costs, respectively. It should be noted here that the project's indirect costs arise from using renewable resources for the project's duration. The indirect cost per unit time of the renewable resources is multiplied by the total project duration in order to calculate indirect costs. On the other hand, direct costs are imposed when some project activities are executed in the crashing way. Therefore, the direct cost per unit time of using the renewable resources is multiplied by the crashing duration of these activities to compute the total direct costs. According to the constraint set in Equation 3.2, each project activity should be performed under a single mode and completed in a single time period that takes place in the range of its earliest and latest finishing times. It should also be noted here that once a mode and execution method is chosen for any project activity, they cannot be changed during the activity execution phase. Additionally, it is assumed that the project activities must be realized in the selected modes without any interruption. The constraint set in Equation 3.3 ensures that all of the precedence relations among the project activities should be fullfilled. Indeed, any project activity can be started after all of its predecessors are completed. The capacity availabilities for the renewable project resources, i.e., human resources, are checked in the constraint set given by Equation 3.4. In detail, total usage of each renewable resource cannot be higher than its available capacity at each time period. The constraint set in Equation 3.5 restricts the total project duration to be no longer than the pre-specified project deadline. For that reason, there will be a conflict between the total project cost objective and this project time limitation. In other words, when a tight project deadline is

determined, the total project cost will increase. On the contrary, the total project cost can be reduced in case of soft deadlines. Finally, project activities' completion times are described as binary integer variable in Equation 3.6. Moreover, ILOG OPL codes/scripts of this modified ILP model are also given in Appendix-1.

3.1.2 The Proposed Constraint Programming (CP) Model Formulation

The CP is a powerful technique for solving large-sized combinatorial optimization problems in reasonable computing time (Bartak, 1999; Baptiste et al., 2012). It can be used as an alternative solution approach to ILP to solve real-life problems, especially in the scheduling field. The model size, i.e., the number of constraints and decision variables, does not increase excessively compared to the classical ILP-based mathematical programming approaches. In other words, the CP approach needs less computation time because of using fewer decision variables and constraints when compared to the ILP approach (Ozfirat et al., 2006). To simplify the application of CP approach, IBM ILOG CPLEX Optimization Studio contains a CP modeling tool that enables to solve large-scale scheduling problems efficiently. The CPLEX-CP solver engine can solve CP models and achieve good quality feasible solutions using two techniques: constraint propagation and constructive search. Firstly, initial constraint propagation eliminates some values from the domain of decision variables that will not satisfy all of the model constraints. In addition to this constraint propagation mechanism, constructive search is also used by the CP optimizer to monitor the search for finding a solution in the remaining part of the search space (IBM, 2017). Indeed, these constructive search and constraint propagation mechanisms are utilized simultaneously by the CP optimizer until a feasible solution is found. It should also be noted here that if the domain of any decision variable becomes empty during the iterations of these mechanisms, this leads to an unknown solution (Menesi et al., 2013; Baykasoğlu et al., 2020). The other main advantage of the CP approach for solving the scheduling problems is that no enumeration of time is needed. In other words, there is no need to use an additional time index while modeling scheduling problems via the CP approach. This advantage may lead to fewer decision variables and

constraints (Menesi et al., 2015). However, as a limitation or disadvantage of CP approach, it may fail to obtain an optimal solution when a lower bound does not exist (Lustig & Puget, 2001). In addition to the previously given ILP model, the CP model of the MRC-DTCP is also developed in this chapter as given in Equations 3.7-3.11 by using the mathematical nomenclature in Tables 3.1-3.2. As shown in Table 3.2, some additional decision variables unique to the CP modeling tool of ILOG OPL CPLEX Optimization Studio are defined. In greater detail, a time interval during performance of a project activity is represented by the interval variable X_j. Besides, Y_{jm} and $Y_{jm'}$ represent optional interval variables that enable selecting normal or crashing ways of different alternative modes. In other words, Y_{jm} and $Y_{jm'}$ are identified with an option "optional"; that is, one can decide not to consider them in the solution project schedule. These optional interval variables are used to determine the proper mode and execution method for a given project activity j.

Table 3.2 Additional mathematical notation for the CP model formulation of the MRC-DTCTP

Additional decision variables	
X_j	an interval of time during a project activity j is executed
Y_{jm} , $Y_{jm'}$	optional interval variables for project activity j when treated in mode m in the normal way or mode m' in the crashing way, respectively $(m, m' \in M_j)$

$$Minimize \sum_{j\in V} \sum_{m'\in M_j \,\&\, k_{jm'r}>0} \sum_{r\in R} presenceOf\,(Y_{jm'}).sizeOf\,(Y_{jm'}).c_r + \sum_{r\in R}(endOf(X_j)).u_r \quad (3.7)$$

Subject to;

$$alternative\left(X_j, \{Y_{jm(m')} \mid m, m' \in M_j\}\right) \quad \forall j \in V \quad (3.8)$$

$$endBeforeStart(X_i, X_j) \quad \forall j \in V, \forall i \in P_j \quad (3.9)$$

$$\sum_{j\in V} \sum_{m,m'\in M_j \,\&\, k_{jm(m')r}>0} pulse(Y_{jm(m')}, k_{jm(m')r}) \le K_r \quad \forall r \in R \quad (3.10)$$

$$endOf(X_J) \le T_{max} \quad J \in V' \quad (3.11)$$

In addition to these interval variables, the above CP model also includes some tuples and special functions like *endOf*, *alternative*, *endBeforeStart*, *sizeOf*, *pulse* and *presenceOf* etc. These functions can be described as follows. In Equation 3.7, the total project cost is aimed to be minimized as in the objective function of the ILP model in Equation 3.1. For calculating the project direct cost, *presenceOf* and *sizeOf* functions are utilized. Firstly, the *presenceOf* function is used, which returns "1" when a given interval $Y_{jm'} \mid \forall m' \in M_j$ is *present.* Otherwise, it returns "0" when the interval $Y_{jm'}\ \forall m' \in M_j$ is absent. Secondly, the *sizeOf* function returns the size of the variable $Y_{jm} \mid \forall m' \in M_j$ in terms of the crashed duration. Thus, the project direct cost is calculated by multiplying the direct cost per unit time usage of renewable resources and the crashing duration of the project activities. In the second part of Equation 3.7, the project indirect cost is calculated by using the *endOf* function. This function returns the end time of a given interval variable (X_J) for a dummy ending activity $J \in V'$. After that, this value is also multiplied by the indirect cost per unit time usage of the renewable resources to provide the total project indirect cost. According to the constraint set in Equation 3.8, *alternative* function can provide proper mode selection for each project activity. In addition, this constraint ensures that each project activity should be executed under only one mode and has to be completed at a single time period. The constraint, i.e., $alternative\left(X_j, \{Y_{jm(m')} \mid m, m' \in M_j\}\right)$ enables to model an exclusive alternative between $\{Y_{jm(m')} \mid \forall m, m' \in M_j\}$. It should also be noted here that when an interval variable arises in the solution, it is called as *present*; otherwise, it is named as *absent*. If an interval variable X_j is present, then exactly one interval variable $Y_{jmm'} \mid \forall m, m' \in M_j$ will also be present and X_j starts and finishes together with this selected mode. In fact, Equation 3.8 corresponds to the constraint sets in Equations 3.2 and 3.6 in the former ILP model formulation. According to the constraint set in Equation 3.9, a project activity cannot be started before its predecessors are completed. This constraint satisfies the precedence relations among the activities, and it also corresponds to the constraint set in Equation 3.3. The constraint function $endBeforeStart(X_i, X_j)$ ensures that the end of a given interval variable, i.e., X_i should be less than or equal to the start of its successor activity, i.e., the interval variable X_j. This

constraint provides the appropriate timing of the interval variables in the solution. Equation 3.10 displays the capacity restrictions of the renewable project resources where $pulse\left(Y_{jm(m')}, k_{jm(m')r}\right)$ is an elementary cumulative function and its value is equal to $k_{jm(m')r}$ during the interval in which the optional interval $Y_{jm(m')} \mid \forall m, m' \in M_j$ is *present*. In detail, *pulse* function increases or decreases the level of the resource utilization by the pre-specified amount, i.e., $, k_{jm(m')r}$ for the length of a given interval variable, i.e., $Y_{jm(m')} \mid \forall m, m' \in M_j$. This capacity constraint for the renewable project resources also corresponds to Equation 3.4 in the formerly presented ILP model formulation. Finally, Equation 3.11 restricts the project duration with the pre-determined deadline or threshold value to balance the project makespan and total cost. In Equation 3.11, the *endOf* function returns the end time of a given interval variable (X_J) for the dummy ending activity. This value should not be greater than the upper bound of the project deadline (T_{max}). This constraint set in Equation 3.11 also corresponds to Equation 3.5 in the modified ILP model formulation. Moreover, ILOG OPL codes/scripts of the proposed CP model are also given in Appendix-2. More details about these functions can also be accessed by using IBM OPL Language User's manual (2017).

3.1.2.1 Variable and Value Ordering Strategies

One of the essential advantages of the CP approach is that its efficiency can be improved by manipulating the search tree to be discovered. The CP approach tries to find a feasible solution by controlling the search tree using various standard search strategies. In addition, different variable and value ordering strategies can be specified by using search phases to improve the performance of the CP approach. In this context, decision-maker(s) can determine the search procedure which identifies the strategies for selecting the subsequent variable (variable ordering) and the value allocated to this variable (value ordering). In detail, selecting the right order of variables is aimed by using variable ordering heuristics.

On the other hand, value ordering is another part of the search strategy that determines the order of values allocated to these variables during the search process. According to the available studies in the literature, variable ordering strategies affect the search complexity, whereas the value ordering strategies may have a significant impact on the first solution time (Bartak, 2005; Mouhoub & Jafari, 2011; Baykasoğlu et al., 2017; Baykasoglu et al., 2020). Indeed, a search strategy may contain either a single criterion or multi-criteria search phase. The search phase consists of a variable and a value selector to describe which variable should be allocated to which value *(single criteria search phase)*. Alternatively, it may include the order of more than one search phase or variable *(multi-criteria search phase)*. It should be noted here that such strategies are specified dynamically since the order of variables and values are altered during the search. Indeed, the domains of the variables are not fixed, they are also varied at each iteration.

In this thesis, eight different search strategies, including specific variable and value ordering strategies, are implemented to solve larger-sized MRC-DTCTP instances to improve the performance of the proposed CP model. Additionally, specific functions used to develop these search strategies are also clarified as follows.

(1) *selectSmallest* is a function selecting variables or values with the smallest evaluation.
(2) *selectLargest* is a function selecting variables or values with the largest evaluation.
(3) *domainSize* is a function returning the number of possible values in a current variable domain.
(4) *varImpact* is a function evaluating the impact of a variable. The evaluation is associated with the average reductions in the search space observed for this variable. The variables that have greater values can be fixed throughout the search to provide more space reduction.
(5) *valueImpact* is a function selecting a value that has an impact on the variable instantiations.

(6) *valueSucessRate* is a function selecting a value with a success rate calculated as the difference between failures and instantiations and divided by the number of instantiations (IBM, 2017).

Finally, these variable and value ordering strategies given in Appendix-3 are also compared in the next section in order to determine the most effective and efficient strategies for solving large-sized MRC-DTCTP benchmark instances.

3.2 Computational Study

3.2.1 Generating Benchmark Instances

In order to examine the computational complexity of the MRC-DTCTP and test the performances of the proposed CP and ILP models, several benchmark instances are reproduced based on the classical MRCPSP data sets from the Library PSPLIB (http://www.om-db.wi.tum.de/psplib/getdata_mm.html). Only one problem instance was generated and solved formerly in the study of Wuliang and Schengen (2009) to demonstrate the validity and verification of their original INLP model. However, since the MRC-DTCTP has some new aspects, standard test instances of the classical MRCPSP cannot be utilized to compare the computational results of the proposed CP and ILP models. For this reason, small and medium-sized benchmark instances of the classical MRCPSP in PSPLIB are first modified according to the characteristics of the MRC-DTCTP. In detail, all data modifications are provided for the selected small and medium-sized test instances, i.e., J18, J20, and J30 as follows:

- Each project activity has three execution modes. Additionally, each activity can be realized under normal or crashing ways which represent the execution methods. However, these execution methods are not taken into account by the classical MRCPSP test instances. Therefore, the available data sets are modified by adding these normal and crashing ways.

- The classical MRCPSP test instances also include nonrenewable project resources. However, only renewable resources are considered in this MRC-DTCTP and therefore, nonrenewable resource (*NR*) data in the original test instances are omitted, and only the requirements/capacities of the renewable resources (*R*) are utilized with their original values (*NR*= [0]; *R*= [2]).
- Since the classical MRCPSP does not contain direct and indirect costs per unit of time, (c_r and u_r) their values are also specified for each renewable resource. For the sake of simplicity, a pre-specified rate between direct and indirect costs is assumed to be constant for each renewable resource and the value of this rate (Equation 3.12) is fixed to 3.5 in all of the benchmark instances. Furthermore, there is also a constant rate (Equation 3.13) set to 2.0 between the indirect costs of the resources.

$$\Omega = c_r/u_r \tag{3.12}$$

$$\mu = u_1/u_2 \tag{3.13}$$

- Similar to the study of Wuliang and Schengen (2009), all normal durations (nd_{jm}) for the project activities are lengthened so that they can be crashed. Their values are set to triplication of the original values in the classical MRCPSP test instances. In other words, the normal duration (nd_{jm}) of each activity is generated as in the following Equation 3.14.

$$nd_{jm} = (original\ activity\ duration) * 3 \tag{3.14}$$

- All project activities can be executed in normal or crashing ways. The crashing duration (ed_{jm}) of each activity is generated by using its normal way duration as in the following Equation 3.15.

$$ed_{jm} = ceil(nd_{jm} * 0.6) \tag{3.15}$$

- Upper bound on the project deadline (T_{max}) for each benchmark instance is generated by the following Equation 3.16. In this equation, the project due-date is taken from the classical MRCPSP test instances as well.

$$T_{max} = round(project\ due - date * 2.3) \tag{3.16}$$

On the other hand, since the larger sized classical MRCPCP test instances do not include multiple execution modes, some data sets like J60, J90 and J120 are taken from the Library PSPLIB (http://www.omdb.wi.tum.de/psplib/getdata_sm.html) and modified by adding execution methods only, i.e., normal and crashing ways. In other words, it is assumed that one execution mode is available for each execution method while generating the large-sized MRC-DTCTP benchmark instances. Thus, single-mode RC-DTCTP test problems with normal and crashing ways are reproduced based on these available data sets, namely J60, J90, and J120. Furthermore, these original benchmark instances are also modified as follows to meet the previously mentioned requirements or additional problem characteristics:

- All of the normal durations (nd_{jm}) for the project activities have the same value with the activity durations of the original benchmark instances.
- Except for some project activities whose normal duration value is equal to 1.0, all of the other project activities may be performed under the crashing way. The crashing duration (ed_{jm}) of each activity is generated as Equation 3.17:

$$ed_{jm} = round(nd_{jm} * 0.6) \tag{3.17}$$

- The project due-date data of the original MRCPSP benchmark instances are used as upper bounds on the project deadlines (T_{max}).
- The nonrenewable project resources (*NR*) in the original MRCPSP benchmark instances are also assumed as renewable ones while generating the large-sized MRC-DTCTP instances. In other words, since the stated MRC-DTCTP does not cover nonrenewable project resources, their original values are also used for additional renewable resources (*NR*= [0]; *R*= [4]).
- Similar to the formerly generated small and medium-sized benchmark instances, the constant rates for direct and indirect cost values, i.e., Ω and μ are set to 3.5 and 2.0, respectively for all of the large-sized problem instances (Equations 3.12 and 3.18).

$$\mu = u_1/u_3 = u_2/u_4 \tag{3.18}$$

3.2.2 Comparison of the Computational Results

In order to compare the computational results of the proposed ILP and CP models, the regenerated small, medium and large-sized benchmark instances are solved by using ILOG CPLEX optimization studio v12. 6. 1 on an Intel Core7 2GHz IBM PC. Before solving all of the benchmark instances, previously discussed search strategies in section *3.1.2.1* are applied to some randomly chosen problem instances to determine the most efficient variable and value ordering strategy. In detail, only one problem instance was chosen randomly from each data set of the J30, J60, J90, and J120 benchmark instances to evaluate the performance of these search strategies. As shown in Table 3.3, significant improvements on the objective function values can be achieved by performing strategies-2, 4, 6 and 8 within the pre-specified shorter run time limits.

It should also be noted here that shorter runtime limits are determined to analyze the impact of these strategies on the solution quality and computation time. In other words, these runtime limits are increased to perform a comprehensive experimental study on all of the benchmark instances. After evaluating these search strategies for solution quality, strategies-2, 4, 6, and 8 are also compared depending on when the best result is first found. As depicted in Table 3.3, strategy-2 achieved the best-found solutions in shorter computation times. In other words, search strategy-2 outperforms all of the other strategies in terms of both solution quality and computation time. For this reason, all of the remaining benchmark instances are just solved by using this strategy-2.

Table 3.3 Comparison of the results on different search strategies for randomly chosen benchmark instances

Data Set	Strategy No.	Objective value	First Solution Time* (sec.)	Run Time Limit
j3014_2	No Strategy	**1255.5**	265.717	5 min
	1	1462.5	259.68	
	2	**1255.5**	17.034	
	3	1462.5	282.124	
	4	**1255.5**	17.591	
	5	1546.5	53.422	
	6	**1255.5**	25.363	
	7	1483.5	298.501	
	8	**1255.5**	23.838	
j605_4	No Strategy	1081	893.415	15 min
	1	1209	130.132	
	2	**1042.5**	28.8	
	3	1209	130.409	
	4	**1042.5**	28.764	
	5	1209	131.457	
	6	**1042.5**	28.128	
	7	1209	132.12	
	8	**1042.5**	29.556	
j9037_7	No Strategy	1712.5	290.108	15 min
	1	2127	514.349	
	2	**1460.5**	422.484	
	3	2127	520.313	
	4	**1460.5**	422.708	
	5	2127	500.049	
	6	**1460.5**	428.852	
	7	2127	508.995	
	8	**1460.5**	428.488	
j12040_1	No Strategy	783	469.203	15 min
	1	2220	591.955	
	2	**657**	623.127	
	3	2220	590.247	
	4	**657**	624.345	
	5	2220	594.413	
	6	**657**	638.189	
	7	2220	601.413	
	8	**657**	629.692	

*The time best solution is first found.

The optimization details of the proposed ILP and CP models are also reported for all of the small, medium and large-sized benchmark instances in Table 3.4. In detail, the total number of constraints and decision variables, the best-found objective values, the relevant computation times and optimality gaps are included for both proposed CP and ILP models. Additionally, lower bounds of the test problems are also listed for the ILP model to

calculate the optimality gap (%) of the best-found solutions. These optimality gap values show the percentage of deviation between the best-found objective value and the lower bound. On the other hand, since the CP approach could not provide the lower bound, optimality gaps for the proposed CP model are calculated based on the lower bound of the modified ILP model as formulated in Equation 3.19.

$$Optimality\ Gap\% = \frac{Best\ objective\ provided\ by\ ILP\ or\ CP\ -\ lower\ bound\ obtained\ from\ ILP}{The\ best\ objective\ value\ of\ ILP\ or\ CP} \times 100 \quad (3.19)$$

For instance, the best objective value found by the proposed CP model is equal to 1770 for j305_3 problem instance. Using the above Equation 3.19, the optimality gap of the CP model is calculated as 13.986% for this benchmark instance. Similarly, the optimality gap of the ILP model is computed as 15.980% for the same test problem. In addition to these optimization details, the first solution times of the proposed CP model, indicating the first discovery time of the best solution, are also reported in Table 3.4. This is a very important quantitative metric when evaluating the performance of the developed search strategies and comparing the results of the proposed CP and ILP models. Additionally, Gap(*Obj*)% and Gap(*Time*)% are also utilized to compare the computational results of the proposed CP and ILP models in terms of the solution quality and computing times, respectively. In detail, Gap(*Obj*)% denotes the percentage of deviation between the best objective values found by the proposed ILP and CP models as formulated by Equation 3.20.

$$Gap(Obj)\% = \frac{Best\ \ objective\ found\ by\ ILP\ \ -\ The\ best\ objective\ found\ by\ CP}{The\ best\ objective\ found\ by\ ILP} \times 100 \quad (3.20)$$

As seen in Table 3.4, Gap(*Obj*)% is calculated as 2.318% for the j305_3 problem instance. Finally, Gap(*Time*)% demonstrates the percentage of deviation between the solution times of the proposed ILP and CP models. According to Equation 3.21, Gap(*Time*)% is computed as 30.814% for the j301_6 benchmark instance.

$$\text{Gap(Time)}\% = \frac{Solution\ time\ of\ the\ ILP\ \ -\ Solution\ time\ of\ the\ CP}{Solution\ time\ of\ the\ ILP} \times 100 \quad (3.21)$$

Table 3.4 Computational results of the proposed ILP and CP models for the small, medium, and large-sized benchmark instances

Number of Project Activities	Data Set No	ILP						CP						Gap(obj) %	Gap(time) %
		Number of Constraints	Number of Decision Variables	The Best-found Objective Value	Solution Time (sec.)	Optimality Gap%	Lower Bound	Number of Constraints	Number of Decision Variables	The Best-found Objective Value	Solution Time (sec.)	First Solution Time (sec.)	Optimality Gap%		
J18	j1822_4	893	49825	616.5	43.015	0	616.500	62	130	616.5	17.580	1.738	0	0	59.131
	j1850_2	935	52375	427	6.953	0	427	62	130	427	5.546	2.405	0	0	20.236
	j1854_9	935	52423	883.5	90.797	0	883.500	62	130	883.5	35.396	17.128	0	0	61.016
	j1861_9	935	52405	924	200.969	0	924	62	130	924	93.947	6.884	0	0	53.253
	j1861_10	929	51985	1393.5	600	13.961	1198.950	62	130	1362	117.291	8.860	0	2.260	80.452
J20	j2030_2	1133	70417	630	600	7.944	579.952	68	142	630	320.218	1.245	0	0	46.630
	j2030_4	1001	61573	805.5	600	12.097	708.059	68	142	795	148.796	106.723	0	1.304	75.201
	j2038_4	1097	67915	1198.5	600	7.796	1105.070	68	142	1198.5	110.641	48.765	0	0	81.560
	j2038_5	1019	62731	910.5	600	8.264	835.257	68	142	910.5	98.801	12.015	0	0	83.533
	j2038_7	1013	62431	1312.5	600	10.165	1179.080	68	142	1312.5	230.700	4.640	0	0	61.550
	j2046_3	1025	63211	762	600	25.491	567.755	68	142	730.5	328.450	49.564	0	4.134	45.258
	j2062_1	1037	63985	1144.5	600	10.195	1027.820	68	142	1134	145.339	45.554	0	0.917	75.777
	j2062_2	947	58057	940.5	242.140	0	940.500	68	142	940.5	179.254	74.494	0	0	25.971
	j2062_3	1091	67501	1110	287.734	0	1110	68	142	1110	137.669	19.952	0	0	52.154
	j2062_4	1043	64435	892.5	83.891	0	892.500	68	142	892.5	45.855	18.774	0	0	45.340
J30	j301_1	1461	131077	469.5	25.172	0	469.500	96	210	469.5	8.385	1.947	0	0	66.689
	j301_2	1605	144967	532.5	203.984	0	532.500	96	210	532.5	33.081	14.633	0	0	83.783
	j301_5	1515	136399	525	186.422	0	525	96	210	525	49.629	14.154	0	0	73.378
	j301_6	1551	139795	600	42	0	600	96	210	600	29.058	4.132	0	0	30.814
	j305_3	1557	140353	1812	600	15.980	1522.450	96	210	1770	600	248.955	13.986	2.318	0
	j306_1	1527	137485	1135.5	600	26.380	835.955	96	210	1125	600	91.014	25.693	0.925	0
	j306_4	1467	131737	967.5	600	14.542	826.803	96	210	925.5	600	154.016	10.664	4.341	0
	j306_7	1545	139183	774	600	11.971	681.344	96	210	732	56.356	45.730	0	5.426	90.607
	j309_10	1449	130093	406	385.672	0	406	96	210	406	59.55	51.441	0	0	84.559
	j3013_8	1371	122581	1072.5	600	17.595	883.794	96	210	1062	600	7.093	16.78	0.979	0
	j3014_1	1593	143767	720	321.468	0	720	96	210	720	37.023	14.104	0	0	88.483
	j3014_2	1497	134629	1318.5	600	17.354	1089.69	96	210	1234.5	600	478.768	11.73	6.371	0
	j3014_7	1461	131125	814.5	600	19.337	657	96	210	772.5	600	287.108	14.951	5.157	0
	j3014_9	1563	140953	730.5	600	15.811	615	96	210	678	101.686	58.937	0	7.187	83.052
	j3014_10	1449	130045	804	600	11.18	714.112	96	210	783	451.332	171.618	0	2.612	24.778
	j3030_10	1473	132463	1018.5	600	8.264	934.333	96	210	1008	600	128.463	7.308	1.031	0
	j3038_10	1527	132463	1087.5	600	15.905	914.529	96	210	1045.5	600	8.265	12.527	3.862	0

Table 3.4 Continues

Number of Project Activities	Data Set No	ILP						CP						Gap(obj) %	Gap(time) %
		Number of Constraints	Number of Decision Variables	The Best-found Objective Value	Solution Time (sec.)	Optimality Gap%	Lower Bound	Number of Constraints	Number of Decision Variables	The Best-found Objective Value	Solution Time (sec.)	First Solution Time (sec.)	Optimality Gap%		
J30	j3038_3	1617	146215	976.5	600	20.667	774.69	96	210	955.5	600	55.364	18.923	2.151	0
	j3038_5	1515	136303	1870.5	600	24.229	1417.29	96	210	1765.5	600	142.422	19.723	5.613	0
	j3038_7	1533	138181	867	600	17.307	716.947	96	210	846	600	63.576	15.254	2.422	0
	j3046_6	1581	142693	1014	600	20.416	806.977	96	210	951	600	24.789	15.144	6.213	0
	j3053_3	1509	135781	1519.5	600	11.152	1350.04	96	210	1498.5	600	45.024	9.907	1.382	0
	j3054_10	1563	141031	1491	600	6.055	1400.72	96	210	1480.5	600	107.201	5.389	0.704	0
	j3061_2	1611	145639	1171.5	600	20.389	932.643	96	210	1129.5	600	41.862	17.429	3.585	0
	j3062_10	1473	132367	568.5	203.718	0	568.5	96	210	568.5	71.416	65.028	0	0	64.944
J60	j605_1	1296	35269	938.5	3600	38.976	572.711	163	184	931.5	3600	3498.01	38.517	0.746	0
	j605_2	1540	42815	1485	3600	41.335	871.17	163	184	1429	3600	1129.13	39.036	3.771	0
	j605_3	1508	41843	1267.5	3600	40.47	754.538	163	184	1141.5	3600	2410.07	33.899	9.941	0
	j605_4	1328	36269	1000.5	3600	41.955	580.745	163	184	983	3600	3212.29	40.921	1.749	0
	j605_5	1560	43433	1464.5	3600	52.647	693.49	163	184	1328	3600	2390.62	47.779	9.321	0
	j605_6	1412	38859	723.5	3600	34.721	472.295	163	184	657	3600	1878	28.113	9.191	0
	j606_7	1424	39237	363.5	18.766	0	363.5	163	184	363.5	7.522	1.672	0	0	59.917
	j605_8	1452	40099	964	3600	47.626	504.881	163	184	803	3600	1039.59	37.126	16.701	0
	j605_9	1396	38341	502.5	1916.08	0	502.5	163	184	502.5	16.095	3.912	0	0	99.16
	j605_10	1475	40837	887.5	3600	17.522	731.99	163	184	891	3600	181.717	17.846	-0.394	0
J90	j9037_1	2367	95577	2135	3600	54.335	974.945	294	274	1722	3600	322.958	43.383	19.344	0
	j9037_2	2287	91893	2384.5	3600	62.767	887.818	294	274	1789.5	3600	1967.46	50.387	24.953	0
	j9037_3	2187	87263	2030.5	3600	65.538	699.748	294	274	1155.5	3600	2141.08	39.442	43.093	0
	j9038_4	2315	93183	531	82.859	0	531	294	274	531	36.59	35.986	0	0	55.841
	j9037_5	2267	90969	2687	3600	65.47	927.819	294	274	1955.5	3600	666.2	52.553	27.224	0
	j9037_6	2535	103295	3271	3600	64.388	1164.86	294	274	2137	3600	902.107	45.491	34.668	0
	j9037_7	2231	89307	2626	3600	65.287	911.567	294	274	1460.5	3600	425.09	37.585	44.383	0
	j9037_8	2383	96315	3019	3600	66.456	1012.68	294	274	2042.5	3600	62.968	50.42	32.345	0
	j9038_9	2435	98697	561	621.828	0	561	294	274	561	41.062	39.015	0	0	93.397
	j9038_10	2283	91685	648	23.594	0	648	294	274	648	26.466	0.916	0	0	-12.173
J120	j12040_1	2739	146065	1056	3600	54.547	479.984	350	364	615	3600	1726.96	21.954	41.761	0
	j12040_2	3107	168499	771	3600	29.961	540	350	364	540	886.021	884.147	0	29.961	75.388
	j12040_3	2927	157515	522	1009.45	0	522	350	364	522	225.633	204.428	0	0	77.648
	j12040_4	3123	169449	672	155.953	0	672	350	364	672	77.153	0.53	0	0	50.528

Table 3.4 Continues

Number of Project Activities	Data Set No	ILP						CP						Gap(obj) %	Gap(time) %
		Number of Constraints	Number of Decision Variables	The Best-found Objective Value	Solution Time (sec.)	Optimality Gap%	Lower Bound	Number of Constraints	Number of Decision Variables	The Best-found Objective Value	Solution Time (sec.)	First Solution Time (sec.)	Optimality Gap%		
J120	j12040_5	2883	154819	606	696.641	0	606	350	364	606	1.635	63.923	0	0	99.765
	j12041_6	2964	157517	896	3600	30.696	620.963	350	364	787.5	3600	1031.34	21.148	12.109	0
	j12041_7	2840	149955	1099	3600	42.404	632.976	350	364	833	3600	441.523	24.012	24.204	0
	j12041_8	3216	172867	1206	3600	32.05	819.473	350	364	1031	3600	596.154	20.517	14.511	0
	j12041_9	2996	159467	1417	3600	40.265	846.444	350	364	1123	3600	444.411	24.627	20.748	0
	j12041_10	2944	156263	1401.5	3600	49.047	714.113	350	364	873	3600	207.11	18.2	37.71	0

It should also be emphasized here that the small and medium-sized benchmark instances such as J18, J20 and J30 are run under a 10-minutes run time limit, whereas the larger-sized problem instances like J60, J90, and J120 are run under a one-hour runtime limit. According to Table 3.4, it is seen that the model size, i.e., the number of decision variables and constraints, will increase rapidly when the modified ILP approach is employed. This is due to the usage of time index, i.e., *t*, to define the decision variables and constraints in the development of the ILP model. Actually, this is one of the main disadvantages of the classical ILP approaches, and it may cause no feasible solution for the present MRC-DTCTP. Fortunately, the proposed CP model has relatively smaller number of decision variables and constraints as shown in Table 3.4. Consequently, since the CP model dimension will not increase excessively, it can be concluded that it is a more appropriate solution approach than the modified ILP approach for solving the MRC-DTCTP instances. In contrast, the modified ILP model might cause substantial increases in the model size, making the solution of the stated problem much more difficult to solve.

For the small-sized problem instances like J18 and J20, the proposed CP model can find optimal solutions for all benchmark instances, whereas the modified ILP model has found optimal solutions for 7 out of 15 test problems. Besides, the modified ILP model runs in average 45.3% slower than the CP model for these 7 test problems optimally solved by both ILP and CP models. Moreover, the proposed CP model has 1.07% better objective values than the modified ILP model for the remaining 8 out of 15 problem instances which are not solved optimally by the ILP model. Although the same objective values with the CP model have been found for some of these test problems, i.e., j2030_2, j2038_4, j2038_5, etc., their solutions cannot be stated as “optimal” since their solution processes are not terminated within the pre-specified runtime limit. Actually, after examining the optimal solutions found by the CP model, it can be concluded that the modified ILP model has already achieved optimal solutions for these test problems. However, if the proposed CP model could not find the optimal solutions, it is not possible to conclude that these results found by the modified ILP model are optimal or not. According to Table 3.4, the modified ILP model has just found the optimal solutions for 7 out of a total 25 random

test instances for the medium-sized benchmark instances, i.e., J30. Fortunately, in addition to these 7 benchmark instances solved optimally, the proposed CP model has also found the optimal solutions of 3 additional test problems. In other words, totally 10 out of 25 test instances are solved optimally by the proposed CP model. Furthermore, it should be noted here that the proposed CP model achieved these optimal solutions 70.379% faster than the modified ILP model. The proposed CP model has provided 3.46% better solutions than the ILP model for the remaining 18 test instances in solution quality (or the best-found objective values).

For the large-sized benchmark instances, i.e., J60, J90, and J120, 30 test problems are chosen randomly from the Library PSPLIB and modified according to the current RC-DTCTP. As mentioned previously, these test problems include only one execution mode for normal and crashing ways. According to Table 3.4, both ILP and CP models have found optimal solutions for 2 out of 10 test instances of J60. It is clear in Table 3.4 that the proposed CP model achieved these optimal solutions in average %79.538 faster than the modified ILP model.

Furthermore, the proposed CP model outperforms the modified ILP model for 7 out of these 10 test problems in solution quality. However, the modified ILP model has found a better objective value for only one problem instance, i.e., j605_10. It should also be emphasized here that the proposed CP model has achieved %6.378 better objective values than the ILP model for all of the remaining 8 problem instances. When results of the larger-sized benchmark instances (J90 and J120) are evaluated, it is recognized that the difference between the performances of the proposed CP and ILP models will become more apparent. For instance, the proposed CP and ILP models have found optimal solutions for 3 out of 10 randomly chosen J90 test instances. However, it should be mentioned here that the CP model accomplished these optimal solutions 45.688% faster than the modified ILP model.

Furthermore, the proposed CP model has found 32.287% superior objective values than the ILP model for the remaining 7 test instances of J90. Finally, similar to the computational results of J60 and J90 benchmark instances, the proposed CP model also outperforms the modified ILP model for the larger-sized problem instances of J120 in terms of both computational times and the best-found objective values. Indeed, the proposed CP model achieved the optimal solutions for 4 out of 10 test problems, while the modified ILP model found the optimal solutions for 3 test problems under a one-hour runtime limit. It should also be noted here that these optimal solutions are accomplished by the proposed CP model 75.98% faster than the modified ILP model. Moreover, the best objective values found by the proposed CP model are 25.858% better than the ILP model's results for the remaining 7 test problems.

3.3 A Case Study of POM Module Group Implementation Project of an ERP Software

In order to show the practicality and applicability of the proposed CP model, a case study for an implementation project of the production-and-operations management (POM) module group of an ERP software company is also presented in this section. In addition to the previously comprehensive computational study on several benchmark instances, the modified ILP and the proposed CP models are verified on this real-life application.

3.3.1 Company Information

An international ERP software company provides several comprehensive services, including software sales & development, consultancy, maintenance, education, project management to its customer firms worldwide. It has many R&D centers, over 200 professional ERP consultants, and many solution partners in different countries such as Turkey, Germany, India, United Arab Emirates, South Korea, Switzerland, and United States. This company offers several innovative industry-specific ERP solutions in different sectors in 16 languages in 32 countries with more than 1000 clients and more

than 30000 concurrent active users. It also supports comprehensive services including software sales & development, consultancy, maintenance, education, project management. Besides, it holds many quality certificates like ISO/IEC 27001 information security management, ISO/IEC15504 software process improvement & capability determination (SPICE), and ISO 9001:2015 (Subulan, 2020). Figure 3.2 displays its sales, R&D and consulting network all over the world.



Figure 3.2 Sales, R&D and consulting network of the ERP company all over the world (IAS Inc., 2021)

After the sale of the ERP software product of this company, it is necessary to organize a comprehensive project schedule and activity timetables by creating a work breakdown structure (WBS) and setting different project stages during its implementation projects. In other words, a detailed project implementation plan with activity time schedules is generally needed for this software company's ERP system module-integration process after the sales agreement with each customer firm, because business process requirements and expectations of each customer firm may change by their industry type, firm structure, and size. On the other hand, certain common project implementation stages can also be defined for many customer firms of different sizes and structures. Therefore, this ERP software company constituted some common steps for implementing its ERP projects, as shown in Figure 3.3.

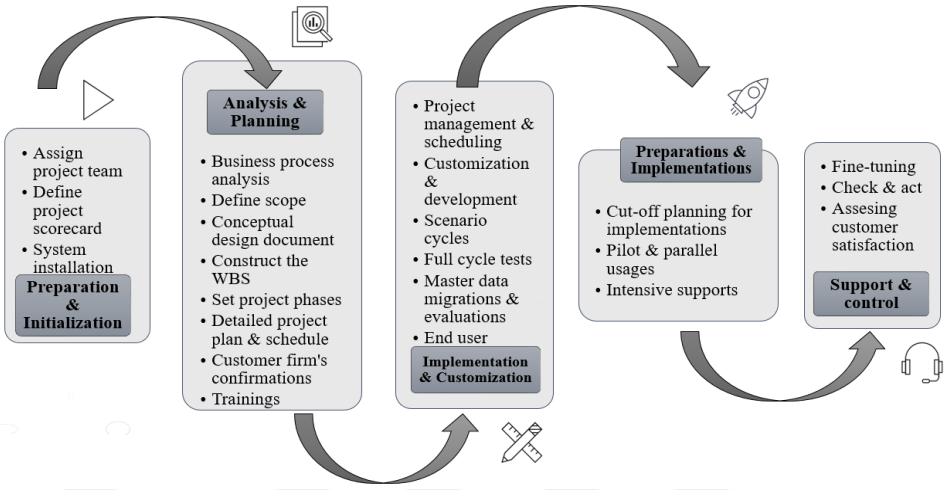


Figure 3.3 Common steps of the project implementation methodology of the ERP software company (IAS Inc., 2021)

3.3.2 Data Description for the Real-Life Application

In the scope of this application study, we have just concentrated on this software company's POM module group implementation project belonging to one of its medium-size customer firms in the textile industry. ERP product of this company has many modules. Its POM module group also contains several modules such as basic core data and system implementation (BAS), bill-of-materials (BOM) management, production routing (ROU), inventory management and control (INV), production planning and control (PRD), material requirements planning (MRP), quality management – statistical process control (QLT) and finally capacity planning and production scheduling (CAP). Indeed, the implementation process for all these modules into the customer firm’s business system requires totally 44 project activities, as given in Table 3.5. This table also includes the immediate predecessors and durations of these activities for each mode under normal and crashing ways. Since this case is handled in a fully deterministic environment, normal and crashing durations of the activities under different execution modes also take deterministic values. Each project activity has a different number of execution modes. Moreover, these project activities have two types of processing durations for the normal

and crashing ways. In detail, some project activities may be executed under crashing way to satisfy a pre-specified project deadline by the project managers. In this case, crashing durations are considered for the relevant project activities operated under crashing way. Besides, some modes of these project activities cannot be executed in a crashing way, as obviously seen in Table 3.5. It should also be noted here that these activity durations may vary not only according to the ERP software product's development level but also to the industrial scale, sector, production process complexity, and information system infrastructure of the customer firm.

In Table 3.5, EFT and LFT values calculated using the forward and backward pass procedures are also presented for each project activity by considering both the crashing and normal durations. Additionally, the precedence relations among these project activities are also determined by the ERP consultants/company experts and indicated by an activity-on-node (AON) project network as in Figure 3.4.

As seen in Table 3.5, some common project activities are needed jointly by different POM modules to design and adapt the support tables, data collection, analysis of the business processes of the customer firm, designing some output reports and documents, etc. In other words, these common activities are usually carried out for each POM module in a similar way. For this reason, they can be called the milestones of the module integration process in the ERP implementation projects (Subulan, 2020). Apart from these common project activities, some other module-specific activities that should be performed for only a specific system module are also required to maintain the project implementation process. For example, project activity 8.3, described as "*Define capacity groups & dispatching rules for scheduling*" is only related to the POM group's capacity planning & production scheduling module (CAP). As mentioned before, all of these project activities in the POM module integration process are not independent of each other, and therefore, precedence relationships and interactions among them should be considered while generating the project schedules.

Table 3.5 The project data for the ERP software product's production and operations management (POM) module implementation process

Implementation Process of the POM Module Group (Activity Descriptions)	Immediate Predecessors	Normal Duration (Day)			Crashing Duration (Day)			Earliest Finish Time (EFT)	Latest Finish Time (LFT)
		Mode-1	Mode-2	Mode-3	Mode-1	Mode-2	Mode-3		
1. BAS (Basic Core Data) Module									
1.1 Define material types, codes and records	-	3	4	7	2	3	5	2	206
1.2 Specify material supply categories	1.1	1	3	4	-	2	3	3	209
1.3 Fictionalize the variant structures	1.1	1	3	-	-	2	-	3	209
1.4 Determine relationships between materials and storage system	1.1	4	5	7	3	4	5	5	209
1.5 Transfer of the basic core data	1.2, 1.3, 1.4	1	3	5	-	2	4	6	210
2. BOM (Bill-of-Material) Module									
2.1. Analyze BOM structures and define alternative BOMs	1.5	4	5	7	3	4	5	9	221
2.2 Design and adapt support tables for BOM & analyze variant structures	1.5	5	8	9	4	6	7	10	221
2.3 Specify quantities and stock keeping units (SKUs)	2.1, 2.2	3	4	-	2	3	-	12	223
2.4 Transfer of the BOM data via electronic data interchange (EDI)	2.3	3	4	5	2	3	4	14	225
3. ROU – Production Routing Module									
3.1 Analyze manufacturing processes & evaluate production routing structures	1.5	7	9	10	5	7	8	11	215
3.2 Specify types of operations and routing alternatives	3.1	3	4	-	2	3	-	13	217
3.3 Design and adapt support tables for the ROU module	3.2	7	-	-	5	-	-	18	222
3.4 Generate variant dependent operation codes	3.2	3	4	5	2	3	4	15	222
3.5 Data transfer for the operations, work-centers and cost-centers	3.3, 3.4	3	4	5	2	3	4	20	224
3.6 Provide material network diagrams (Operation process charts)	3.5	1	2	3	-	1	2	21	225
4. INV – Inventory Management Module									
4.1 Analyze inventory movements and prepare stock documents	1.5	4	5	7	3	4	5	9	257
4.2 Determine lot-sizing policies and analyze the barcode system	1.5	5	7	8	4	5	6	10	257
4.3 Design and adapt support tables for INV module	4.1, 4.2	7	8	10	5	6	8	15	262
4.4 Integrate sales, procurement, production and inventory modules	4.3, 5.4	9	10	-	7	8	-	39	269
4.5 Specify costing methods and perform the stock valuation for accounting	4.3	3	-	-	2	-	-	17	269
4.6 Integrate activity-based costing, accounting and inventory modules	4.4, 4.5	5	7	-	4	5	-	43	273
5. PRD – Production Planning & Control Module									
5.1 Analyze production process, collect manufacturing and process data	2.4, 3.6	7	8	9	5	6	7	26	230
5.2 Design and adapt support tables for production planning module	5.1	7	-	-	5	-	-	31	235
5.3 Fictionalize the production planning & work order types and structures	5.2	1	2	3	-	1	2	32	237
5.4 Analyze the relationships between work orders & inventory movements	5.2	1	3	4	-	2	3	32	237
5.5 Define shift schedules and associate them with the company’s work calendar	5.2	1	2	3	-	1	2	32	237

Table 3.5 Continues

Implementation Process of the POM Module Group (Activity Descriptions)	Immediate Predecessors	Normal Duration (Day)			Crashing Duration (Day)			Earliest Finish Time (EFT)	Latest Finish Time (LFT)
		Mode-1	Mode-2	Mode-3	Mode-1	Mode-2	Mode-3		
5.6 Analyze contract manufacturing operations and residual products	5.2	3	4	5	2	3	4	33	237
5.7 Design output reports and documents for production planning & control	5.3, 5.4, 5.5, 5.6	5	8	-	4	6	-	37	241
6. QLT – Process Quality Control Module									
6.1 Analyze the operational quality control process of the company	5.7	3	4	-	2	3	-	39	264
6.2 Design and adapt support tables for process quality control module	6.1	5	7	8	4	5	6	43	268
6.3 Define quality plans: quality characteristics, sampling methods, center lines, specification limits, defect classes with their priorities	6.2	4	5	7	3	4	5	45	271
6.4 Data collection from the control/test operations in manufacturing	6.2	3	4	5	2	3	4	45	271
6.5 Design output reports: quality control charts & process capability analysis	6.3, 6.4	3	4	-	2	3	-	48	273
7. MRP – Materials Requirement Planning Module									
7.1 Determine basic MRP parameters: Timing, procedures, order quantity/production lot-sizing methods for production and procurement etc.	5.7	3	4	5	2	3	4	39	243
7.2 Specify MRP scenario types and structures	5.7	1	3	-	-	2	-	38	243
7.3 Design and adapt support tables for MRP module	7.1,7.2	5	7	8	4	5	6	43	247
7.4 Implementation tests for validation and verification of the MRP outputs	7.3	4	5	6	3	4	5	46	250
7.5 Design output reports which can identify relationships between customers' sales orders & alternative MRP solutions	7.4	4	5	7	3	4	5	49	253
8. CAP – Capacity planning & Production Scheduling Module									
8.1 Analyze the long-short term production capacities & production lot-sizing	7.5	7	8	10	5	6	8	54	258
8.2 Design and adapt support tables for CAP module	8.1	5	-	-	4	-	-	58	262
8.3 Define capacity groups & dispatching rules for scheduling	8.1	1	3	4	-	2	3	55	262
8.4 Specify capacity types & resource (labor, machinery equipment) planning	8.1	3	4	5	2	3	4	56	262
8.5 Develop heuristic algorithms via an integrated development environment embedded in the ERP software for scheduling work orders	8.2, 8.3, 8.4	10	13	16	8	10	12	66	270
8.6 Design output reports and documents: resource utilization charts for labor and machinery equipment and Gantt charts for the work order operations	8.5	4	7	-	3	5	-	59	273

For instance, after the integration process of the basic core data (BAS), bill of material (BOM), and production routing (ROU) modules, more advanced consecutive modules, i.e., material requirement planning (MRP) and production planning (PRD) can be embedded in the business system of the customer firm as shown in Figure 3.4. Similarly, the capacity planning and production scheduling (CAP) module should be integrated after implementing the MRP and PRD modules.

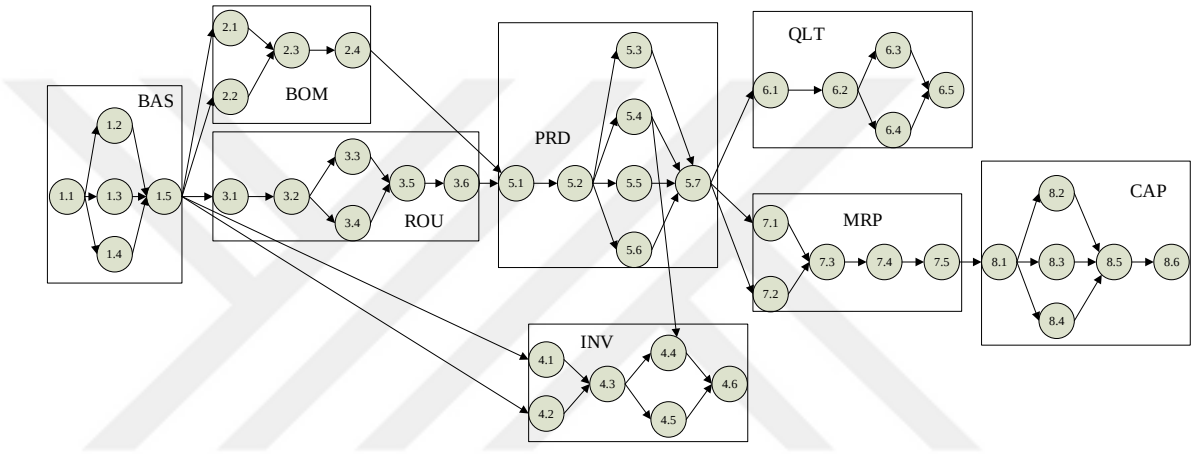


Figure 3.4 The AON project network diagram for the POM module group implementation process of an ERP software

As mentioned before, all of the project sources are considered renewable resources, along with this case study. The human resources of this ERP software company, i.e., ERP project implementation consultants and process experts of the customer firm, are considered renewable project resources. They may have various abilities/capabilities, roles, and responsibilities throughout the project implementation stage. In detail, professional human resources and their knowledge/expertise, which can be seen as fundemantal sources of an ERP software company, play a crucial role in the implementation stage of a successful ERP project (Tadinen, 2005; Plaza, 2016; Subulan, 2020). Additionally, experiences and observations of the process experts in the customer firm's specific business/manufacturing process significantly impact the ERP project implementation phase. The business processes of the customer firm can be improved, and innovative ERP solutions can be developed with the help of the professionality of these

ERP consultants and process experts. For this reason, they should support each other, and cooperation and collaboration among them will have a major effect on satisfying the customer firm's needs and expectations throughout the project implementation phases. Based on this fact, the human resource requirements of each project activity are determined in terms of the labor force (in man-days) and presented in Table 3.6. According to this table, the human resource requirements of the project activities for the ERP consultants are greater than the amounts of demanded process experts. Furthermore, there is a tradeoff between the human resource requirements and activity durations (see Tables 3.5-3.6). In detail, fewer amounts of human resources are demanded in case of longer activity durations. For this purpose, different execution modes are taken into consideration for each activity. Opposite to these execution modes, it is assumed that the human resource requirements of the project activities are not changed according to different execution methods. To put it another way, the amounts of human resource requirements have fixed values for the normal and crashing ways. This assumption was also made in the original study of Wuliang and Schengen (2009). It is a realistic assumption in knowledge-intensive industries like implementation projects of software companies, because additional direct costs per day of these human resources will be imposed if the crashing way is used for some project activities. These direct costs may vary according to the abilities and working performances of the ERP consultants and process experts. In other words, these human resources may increase their performances by working hard, and therefore they may crash the duration of some project activities. Extra direct costs will arise in this situation since some of these project activities are executed in a crashing way.

In this case study, direct and indirect costs per day for ERP consultants of the software company are determined as $120 and $90 by the project managers, respectively. Similarly, the direct and indirect costs per day of the process experts in the customer firm are also taken as $60 and $45, respectively. In order to gain some managerial insights, different availability or capacity levels are determined by the project managers for these human resources. In detail, three different cases, i.e., [40 35], [42 28], and [44 23], are considered

for the capacity levels of the ERP consultants and process experts. For instance, the minimum and maximum availability levels are taken as 23 and 35 man-days for the process experts of the customer firm. Similarly, the minimum and maximum availability levels are defined as 40 and 44 man-days for the ERP consultants of the software company. Finally, different project deadlines (T_{max}) whose values are ranged between 95 and 125 days are identified by the project managers to observe the tradeoff between completion time and total project costs. While determining these minimum and maximum values for the project deadlines, the earliest finishing times (EFT) and the latest finishing times (LFT) of the project activities are considered.

Determination of these project deadlines is challenging for the ERP project managers since the limits on the project completion time may affect selecting the most appropriate execution mode and method. In the next Section 3.3.3, extensive computational results of the proposed CP and ILP models are presented by considering 21 different scenarios in which different resource availability levels and project deadlines are taken into account.

Table 3.6 Human resources (in man-days) demanded by the project activities under different execution modes.

Project Activities	ERP Supplier's Consultants (in man-days)			Process Experts in Customer (in man-days)		
	Mode-1	Mode-2	Mode-3	Mode-1	Mode-2	Mode-3
1.1	15	9	6	5	3	2
1.2	10	7	5	3	2	1
1.3	6	4	-	2	1	-
1.4	18	15	10	6	5	3
1.5	20	16	12	7	5	4
2.1	20	18	15	8	6	4
2.2	25	22	18	3	2	1
2.3	12	10	-	4	3	-
2.4	8	5	3	2	1	0
3.1	30	21	18	28	25	20
3.2	12	8	-	10	7	-
3.3	24	-	-	8	-	-
3.4	15	12	9	10	7	5
3.5	12	9	6	8	6	4
3.6	3	2	1	1	0	0
4.1	25	22	18	12	10	8
4.2	20	18	16	15	10	8
4.3	30	25	20	8	6	5
4.4	40	35	-	10	7	-
4.5	15	-	-	10	-	-
4.6	20	16	-	7	5	-
5.1	40	35	30	30	25	20
5.2	21	-	-	8	-	-
5.3	8	6	4	7	5	3
5.4	6	4	3	5	4	2
5.5	3	2	1	2	1	0
5.6	6	4	3	4	3	2
5.7	14	10	-	5	3	-
6.1	12	8	-	12	9	-
6.2	28	24	20	8	6	4
6.3	24	20	16	18	16	13
6.4	20	16	12	25	20	16
6.5	16	10	-	6	4	-
7.1	25	20	15	10	8	6
7.2	9	6	-	5	3	-
7.3	21	18	14	7	5	3
7.4	24	21	18	12	10	8
7.5	18	15	12	6	4	3
8.1	30	28	25	20	16	11
8.2	40	-	-	8	-	-
8.3	12	10	7	4	3	2
8.4	15	12	10	10	7	5
8.5	50	45	40	0	0	0
8.6	32	28	-	7	5	-

3.3.3 Comparison of the Computational Results and Scenario Analysis for Managerial Insights

The validity and applicability of the proposed CP and ILP models are also tested based on this ERP project data given in Sections 3.1.1-3.1.2. Indeed, both proposed ILP and CP models are solved with these data presented in Figure 3.4 and Tables 3.5-3.6 by using ILOG CPLEX optimization studio V12.6.1 on an Intel Corei7 2 GHz IBM PC. The detailed computational results of the proposed CP and ILP models are presented in Table 3.7. The solutions of these models are obtained for each scenario within one hour. As mentioned earlier, 21 scenarios are produced based on the combinations of three different availability levels of the human resources (R_1=40 & R_2=35; R_1=42 & R_2=28; R_1=44 & R_2=23) and seven project deadline values (95, 100, 105, 110, 115, 120 and 125). The search-strategy-2, which provided superior results (chosen as the best strategy in Section 3.1.2.1), is applied again to improve the performance of the proposed CP model. Similar to the formerly performed experimental study on various benchmark instances, the number of constraints and decision variables can be reduced significantly when the proposed CP model is applied to solve this real-life ERP project scheduling problem (see Table 3.7). However, the dimension of the modified ILP model increases substantially, which requires more computational effort. The main reason is that the usage of the time index in the ILP model causes rapid increases in the number of decision variables and constraints. Fortunately, the proposed CP model achieved optimal solutions for 6 out of 21 scenarios within less than 324 seconds. The modified ILP model could not find optimal solutions for these scenarios under the one-hour runtime limit, and it also has a very high average optimality gap value, 26.85%. The modified ILP model, on the other hand, can find feasible solutions in tight project deadline values like 95 and 100 days. The proposed CP model could not find any feasible solution for scenarios #1 and #8, with tight deadline limits (95 days) and highly restricted human resources (R_1=40 & R_2=35 and R_1=42 & R_2=28). In line with this, it was also highlighted by Menesi et al. (2013) that the results of the CPLEX-CP solver are very sensitive to strict deadline constraints. It means that the CP approach can achieve better solutions in case of relaxed project deadlines. In detail,

the modified ILP model provides feasible solutions with total project cost values of $29055 and $27165 for scenarios#1 and #8, respectively. On the other hand, when the project deadline is set to 95 days and the availability levels of the human resources are taken as (R_1=40 & R_2=35 and R_1=42 & R_2=28), the proposed CP model fails to produce any feasible solution. Fortunately, if the availability level of the ERP consultants is increased up to R_1=44, the proposed CP model can find a feasible solution with a 25.434% optimality gap despite defining a 95 days project deadline. It should also be noted here that some critical ERP project activities which affect the project makespan will be executed in a crashing way in case of these tight project deadlines, i.e., 95 and 100 days. It is also obviously seen in Table 3.7 that the availability levels of the ERP consultants have a larger impact on the total project cost when compared to the process experts of the customer firm. If very low values are set to the availability level of ERP consultants and project deadlines, it is impossible to find any feasible solution by the proposed CP model.

In scenarios #13 and #14, where the human resource availabilities are set to R_1=42 & R_2=28, the total project cost objective does not change even if the project deadline is increased from 120 to 125 days. In other words, if the project deadline is extended by more than 120 days, it will not affect the total project cost objective. In other words, the same total project cost value, i.e., $12600, is yielded by the proposed CP model for both of these scenarios. However, as seen in Table 3.7 that the project deadline has a crucial influence on the total project cost for all other scenarios since total project cost values decrease when the project deadline is increased in all of the other scenarios. In other words, as the project deadline is increased up to 125 days, most of the project activities will be executed under the normal way, and therefore total direct cost decreases, whereas the total indirect cost increases at the same time. Because, as mentioned previously, direct project costs are resulted from crashing some project activities, whereas the indirect project costs are mainly associated with the project makespan, which is directly affected by the project deadline.

Table 3.7 Comparison of the optimization results provided by the proposed CP and ILP models under different project deadlines and human resources availabilities

Scenario No	Human Resource Availability [R1 R2]	Project Deadline (T_{max})	ILP						CP						Gap(obj) %	Gap(time) %
			Number of Constraints	Number of Decision Variables	The Best-found Objective Value	Solution Time (sec.)	Optimality Gap%	Lower bound	Number of Constraints	Number of Decision Variables	The Best-found Objective Value	First Solution Time (sec.)	Solution Time (sec.)	Optimality Gap%		
1	[40 35]	95	657	75211	29055	3600	23.3	22283.8	114	302	NA	NA	3600	NA	NA	NA
2		100	657	75211	24540	3600	27.55	17778.8	114	302	25380	3412.63	3600	29.950	-3.423	0
3		105	657	75211	20655	3600	29.16	14631	114	302	19395	1771.25	3600	24.563	6.100	0
4		110	657	75211	16560	3600	23.77	12623.6	114	302	18030	1825.75	3600	29.986	-8.877	0
5		115	657	75211	16455	3600	31	11353.6	114	302	14355	1601.96	3600	20.908	12.762	0
6		120	657	75211	14460	3600	26.24	10666.2	114	302	13020*	111.797	913.868	0	9.959	74.615
7		125	657	75211	13755	3600	22.9	10605	114	302	12705*	7.578	253.065	0	7.634	92.970
8	[42 28]	95	657	75211	27165	3600	25.06	20357.8	114	302	NA	NA	3600	NA	NA	NA
9		100	657	75211	25590	3600	37.9	15892.6	114	302	35880	3359.05	3600	55.706	-40.211	0
10		105	657	75211	21075	3600	38.03	13060.9	114	302	20865	320.014	3600	37.403	0.996	0
11		110	657	75211	16770	3600	31.13	11550	114	302	15930	56.526	3600	27.495	5.009	0
12		115	657	75211	14805	3600	26.95	10815	114	302	13935	207.628	3600	22.390	5.876	0
13		120	657	75211	13020	3600	17.86	10694.6	114	302	12600*	14.922	216.627	0	3.226	93.983
14		125	657	75211	13335	3600	19.25	10767.6	114	302	12600*	54.079	235.409	0	5.512	93.461
15	[44 23]	95	657	75211	29265	3600	27.04	21351.9	114	302	28635	2737.17	3600	25.434	2.153	0
16		100	657	75211	23070	3600	27.97	16618.1	114	302	23910	2443.78	3600	30.497	-3.641	0
17		105	657	75211	20655	3600	34.63	13501.7	114	302	20025	241.608	3600	32.576	3.050	0
18		110	657	75211	18450	3600	37.09	11606	114	302	16560	1252.57	3600	29.915	10.244	0
19		115	657	75211	15300	3600	26.19	11293.3	114	302	14355	3456.16	3600	21.328	6.176	0
20		120	657	75211	13830	3600	17.91	11353.2	114	302	13020*	130.309	218.18	0	5.857	93.939
21		125	657	75211	13020	3600	12.98	11330.4	114	302	12705*	60.111	106.954	0	2.419	97.029

*Optimal total project cost ; R1- ERP company's consultants ; R2- Process experts in customer firm; First solution time – when the best solution is found

When the performances of the proposed CP and ILP models are evaluated according to different availability levels of the human resources, it also is noticed that the proposed CP model found optimal solutions for 2 out of a total 7 scenarios in the case of R_1=40 & R_2=35. In contrast, no optimal solution is found by the modified ILP model for all of these scenarios. However, as mentioned earlier, the proposed CP model could not reach any feasible solution when the project deadline is set to 95 days in scenario #1, where the human resource availabilities are again set to R_1=40 & R_2=35. Fortunately, the proposed CP model has found 4.026% better total project cost values than the modified ILP model for the remaining 6 out of a total of 7 scenarios. Furthermore, it should be noted here that the proposed CP model achieved these solutions 27.93% faster than the modified ILP model for all of these six scenarios.

Similarly, the proposed CP model could not reach any feasible solution when the project deadline is set to 95 days, and the human resource availabilities are taken as R_1=42 & R_2=28, respectively. Even though the proposed CP model has found 40.211% worse total project cost value than the modified ILP model when the project deadline limit is set to 100 days, it can find optimal solutions for 2 out of seven scenarios. On the other hand, the modified ILP model could not provide an optimal solution for all seven scenarios. Fortunately, the proposed CP model accomplished 4.124% better total project cost values than the modified ILP model for scenarios #10, #11, #12, #13, and #14. Besides, the proposed CP model also achieved these solutions 37.488% faster than the modified ILP model for all of these five scenarios.

When the human resource availabilities are set to R_1=44 & R_2=23, the proposed CP model finds a feasible solution under a 95-day project deadline. Furthermore, while the proposed CP model found the optimal solution for 2 out of 7 scenarios again, the modified ILP model could not achieve an optimal solution for these scenarios. The proposed CP model finds optimal solutions under all human resource availability levels in case of relaxed project deadlines (120 and 125 days). Moreover, the proposed CP model has found 3.751% better project cost values and achieved these solutions 27.28% faster than the modified ILP model for all of these seven scenarios.

In summary, the handled MRC-DTCTP throughout this thesis has strategic importance for the ERP project managers and other project stakeholders as it enables the construction of the time-cost tradeoff curves to gain managerial insights. The success of the ERP implementation projects is closely related to determining the right balance between total project cost and completion time. For this reason, the time-cost tradeoff curves for this ERP implementation project application are also drawn in Figure 3.5 under different human resource availability levels and project deadlines. According to Figure 3.5, as the project deadline gradually lengthens, total project costs can be reduced under the same human resource availability levels. In detail, direct project costs will decrease, and accordingly, the indirect project costs will increase for the larger project deadline since the usage of crashing way will decrease, and the project makespan will be longer in this case. It is also noteworthy here that direct human resource costs per day are relatively higher than indirect costs. Therefore, the total project cost will generally decrease in case of longer project deadlines. For instance, when the total project deadline is lengthened from 95 to 125 days under the pre-specified human resource availabilities, i.e., R_1=44 & R_2=23, the total project cost can be decreased from $28.635 to $12.705 by using the proposed CP model. Similarly, the total project cost can also be reduced from $29.265 to $13.020 when the modified ILP model is utilized. As shown in Figure 3.5 the proposed CP model can achieve relatively higher cost reductions since it yields an optimal solution for the relaxed project deadlines (see scenario #21).

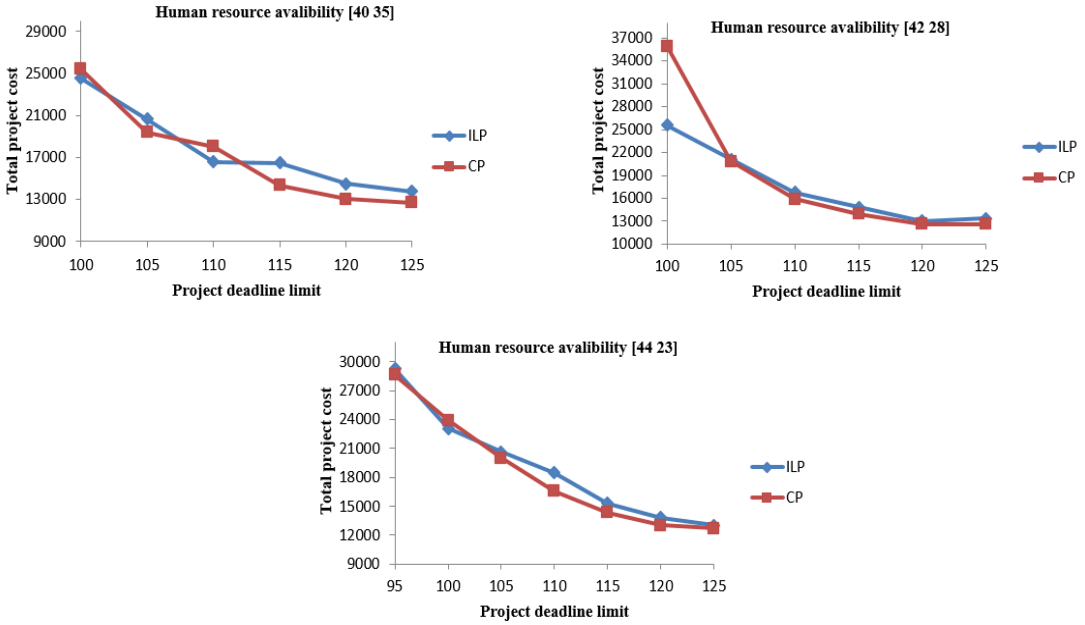


Figure 3.5 Time-cost tradeoff curves for the POM module group implementation project of the ERP software product

In a nutshell, the proposed CP model can be a useful approach under relaxed project deadlines for the realistic size of MRC-DTCTP in terms of computational time and solution quality. On the other hand, the modified ILP model can also provide acceptable solutions for the present ERP project case study under tighter project deadlines. It was also highlighted by Wuliang & Chengen (2009) that the present MRC-DTCTP with tight deadlines could be converted to the classical MRCPSP when all of the project activities are executed in a crashing way. However, as mentioned before, it will be a challenging task to obtain good quality feasible solutions by using the modified ILP model within reasonable computing times for the larger-sized problem instances. Contrary to the modified ILP model, the proposed CP model can overcome these difficulties and generate acceptable project schedules that satisfy the ERP project manager's expectations in realistic size problem instances. In addition, ERP project managers of the software company and the customer firm's project directors can make better decisions under limited human resources by balancing total project duration and total project cost via the proposed CP model.

CHAPTER FOUR
CONSTRAINT PROGRAMMING-BASED TRANSFORMATION APPROACH FOR A MULTI-OBJECTIVE & MULTI-MODE RESOURCE INVESTMENT PROJECT SCHEDULING PROBLEM UNDER MIXED FUZZY-STOCHASTIC ENVIRONMENTS

4.1 Problem Description and Mathematical Formulation of FS-MRIPSP

This chapter presents a multi-objective FS-MRIPSP which is reformulated based on the discrete-time binary integer programming model of the traditional RCPSP by Pritsker et al. (1969). In fact, that preliminary model is extended in this thesis by incorporating mixed fuzzy and stochastic types of uncertainties into a multi-objective and multi-mode RIPSP. Unlike the classical RCPSPs, capacities or availability levels of the project resources are defined as decision variables instead of project parameters or input data as in RIPSPs or resource availability cost problem (RACP). Recently, Subulan (2020) has already proposed another variant of that original model by taking into consideration mixed stochastic and interval types of uncertainties. In fact, a hybrid chance-constrained optimization and interval programming based solution approach was developed in order to handle mixed interval and stochastic types of uncertainties concurrently. It was also highlighted by Subulan (2020) that fuzzy version of the examined problem can also be studied in the future since epistemic uncertainty of some project parameters can be better reflected by using fuzzy set theory. Because, fuzzy numbers may provide more information on the uncertain project data instead of using interval/grey numbers. Because, interval numbers just take into account the lower and upper bounds on the imprecise project parameters. Thus, the membership function values of several possible outcomes are ignored unfortunately. For that reason, some of the project parameters which have epistemic uncertainties are represented by triangular fuzzy numbers instead of interval valued numbers in this thesis. Moreover, from the theoretical and methodological perspectives, instead of using a chance-constrained stochastic optimization method, this thesis employs a multi-scenario based stochastic programming approach to cope with the random project parameters and avoid the risk of constraint violation. In detail, renewable/non-renewable resource

requirements (human resources and project budget) of the project activities are assumed to be stochastic parameters since they may have randomness in nature. Because of the dynamic cash incoming and outgoing flows of the companies, fund requirements of the projects may be stochastic. Therefore, values of these random project parameters are represented by discrete scenarios with the probability of each outcome. On the other hand, availability levels or capacities of both renewable & non-renewable resources are defined as fuzzy-stochastic decision variables and therefore, it is targeted to determine their ideal fuzzy values under each scenario. Because of these structural modifications, the examined problem, namely FS-MRIPSP will have a multi-objective nature since effective and efficient resource utilization may be as important as traditional project makespan objective. Due to the ambiguity/vagueness in the activity durations, their earliest/latest finishing times can be seen as the other critical issue when modelling and solving real-life MRIPSPs under uncertainty. Actually, project managers/experts may not know certain values of these project parameters in advance, but they may easily specify their pessimistic, most likely and optimistic values according to their knowledge and expertise. For that reason, instead of using probability theory, fuzzy sets and possibility theory can be utilized to handle these uncertain project parameters. In summary, main characteristics of the FS-MRIPSPS can be outlined as in Figure 4.1.

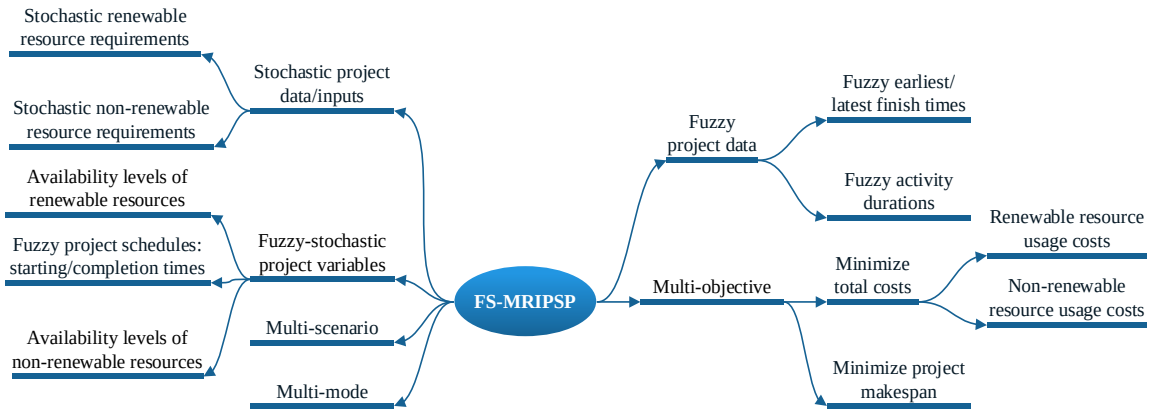


Figure 4.1 Summary of characteristics/features of the FS-MRIPSP

4.1.1 Integer Linear Programming (ILP) Model Formulation of the Multi-Objective FS-MRIPSP

In light of these facts, ILP model formulation of the FS-MRIPSP is first presented in Equations 4.1-4.8 by using the mathematical nomenclature in Table 4.1. Actually, the following formulation corresponds to a scenario-based stochastic & possibilistic programming model of the multi-objective FS-MRIPSP.

Table 4.1 Mathematical nomenclature for the integer programming (ILP) formulation of the FS-MRIPSP

Indices & sets	
$j \in J$	set of the project activities to be scheduled
$r \in R$	set of the renewable project resources (Human resources)
$n \in N$	set of the non-renewable project resources (Project funds or budget)
P_j	set of immediate predecessors of the activity j
$q \in Q_j$	set of available execution modes for the activity j
$\tilde{t}$	fuzzy time period for resource allocation to a project activity
$s \in S$	set of potential scenarios
Fuzzy project parameters	
$\widetilde{EFT}_j$	fuzzy earliest finishing time of the activity j
$\widetilde{LFT}_j$	fuzzy latest finishing time of the activity j
$\tilde{d}_{jq}$	fuzzy duration of the activity j under mode q
$\tilde{C}_r$	fuzzy unit usage cost of renewable resource type r
Stochastic project parameters	
kr_{jqrs}	required amount of renewable resource type r on activity j under mode q in scenario s
kn_{jqns}	required amount of non-renewable resource type n on activity j under mode q in scenario s
π_s	occurrence probability of scenario s
Fuzzy-stochastic decision variables	
$\widetilde{KR}_{rs}$	availability level of renewable resource type r throughout the project duration in scenario s
$\widetilde{KN}_{ns}$	availability level of non-renewable resource type n throughout the project duration in scenario s
$x_{jq\tilde{t}s}$ $\begin{cases}1\\0\end{cases}$	if activity j is executed in mode q and completed in fuzzy time period $\tilde{t}$ in scenario s; otherwise

$$Minimize\ EV(Project\ Makespan)\ \ \tilde{Z}_1 \cong \sum_{\tilde{t} \in \widetilde{EFT}_J}^{\widetilde{LFT}_J} \sum_{s=1}^{S} \pi_s . \tilde{t} . x_{J1\tilde{t}s} \tag{4.1}$$

$$Minimize\ EV(Total\ Resource\ Costs)\ \ \tilde{Z}_2 \cong \sum_{n=1}^{N} \sum_{s=1}^{S} \pi_s . \widetilde{KN}_{ns} + \sum_{r=1}^{R} \sum_{s=1}^{S} \pi_s . \widetilde{KR}_{rs} . \tilde{C}_r \tag{4.2}$$

The model constraints of the FS-MRIPSP are formulated as in the following Equations 4.3-4.8.

Subject to;

$$\sum_{q=1}^{Q_j} \sum_{\tilde{t} \cong \widetilde{EFT}_j}^{\widetilde{LFT}_j} x_{jq\tilde{t}s} = 1 \quad \forall j \in J, \forall s \in S \tag{4.3}$$

$$\sum_{q=1}^{Q_i} \sum_{\tilde{t} \cong \widetilde{EFT}_i}^{\widetilde{LFT}_i} \tilde{t}.x_{iq\tilde{t}s} \leq \sum_{q=1}^{Q_j} \sum_{\tilde{t} \cong \widetilde{EFT}_j}^{\widetilde{LFT}_j} (\tilde{t} - \tilde{d}_{jq}).x_{jq\tilde{t}s} \quad \forall j \in J, \forall i \in P_j, \forall s \in S \tag{4.4}$$

$$\sum_{j=2}^{J-1} \sum_{q=1}^{Q_j} kr_{jqrs} . \sum_{\tilde{\tau} \cong \tilde{t}}^{\tilde{t}+\tilde{d}_{jq}-1} x_{jq\tilde{\tau}s} \leq \widetilde{KR}_{rs} \quad \forall r \in R, \forall \tilde{t} \in T, \forall s \in S \tag{4.5}$$

$$\sum_{j=2}^{J-1} \sum_{q=1}^{Q_j} kn_{jqns} . \sum_{\tilde{t} \cong \widetilde{EFT}_j}^{\widetilde{LFT}_j} x_{jq\tilde{t}s} \leq \widetilde{KN}_{ns} \quad \forall n \in N, \forall s \in S \tag{4.6}$$

$$x_{jq\tilde{t}s} \in \{0,1\} \quad \forall j \in J, q \in Q_j, \tilde{t} \in \{\widetilde{EFT}_j, \widetilde{LFT}_j\}, \forall s \in S \tag{4.7}$$

$$\widetilde{KR}_{rs}, \widetilde{KN}_{ns} \geq 0 \quad and \quad integer \quad \forall r \in R, \forall n \in N, \forall s \in S \tag{4.8}$$

The first goal in Equation 4.1 aims to reduce expected value of the uncertain project makespan which is related to the fuzzy completion time of the dummy ending activity. In the second goal, expected value of the total cost resulting from the usage of both renewable and non-renewable resources is minimized in Equation 4.2. It should be highlighted here that since the non-renewable resource is defined in terms of the monetary value (project funds or budget), availability level or usage amount of this resource type is considered only in Equation 4.2. On the other hand, fuzzy usage costs of the renewable resources are multiplied by their usage amounts/availability levels to obtain a similar monetary value. In most of the real-life project scheduling cases, project managers desire to decrease usage costs of the renewable/non-renewable project resources as well as minimizing the project makespan. However, minimizing the project makespan may need to utilize more resources which cause higher total cost. On the other hand, usage of less amount of resources may cause longer project makespan. For that reason, a compromise solution should be found by taking account the trade-off between project makespan and total cost. Actually, these are the most frequently encountered conflicting objectives in RIPSP applications. The constraint set in Equation 4.3 ensures that each activity can be executed under a single mode in

each scenario. Moreover, each activity should be completed at a fuzzy time which is ranged between its fuzzy EFTs and LFTs. In other words, only one execution mode and fuzzy time can be assigned to each project activity under each scenario. According to the constraint set in Equation 4.4, precedence relations between the project activities should be satisfied under each scenario. Actually, fuzzy durations and project activities' starting and completion times make this precedence constraint much more complicated when compared to its deterministic version. Because, fuzzy completion time of an activity may intersect with fuzzy starting times of its successors. Because of these overlapped fuzzy numbers, precedence relations between the project activities may not be fulfilled explicitly. This problematic issue has already mentioned by Subulan (2019 & 2020) and defined as one of the risky situations in fully uncertain project scheduling applications. Actually, standard fuzzy arithmetic based solution approaches may cause these risky situations. This problematic issue can be overcome by making use of the proposed bound & decomposition based solution approach without using standard fuzzy arithmetic operations and fuzzy ranking techniques directly. Constraint set in Equation 4.5 is related to the capacity limitations of the renewable project resources. Apart from the classical RCPSPs, availability/capacity levels for different types of renewable resources are defined as fuzzy-stochastic decision variables. For that reason, with the help of the second objective function given in Equation 4.2, this constraint set also provides minimum renewable resource utilization. According to Equation 4.5, the capacity/availability level of each renewable resource type cannot be exceeded at each fuzzy time period under each scenario. Similarly, capacity restrictions on the non-renewable project resources are formulated in Equation 4.6. In other words, availability level of each non-renewable resource cannot be surpassed throughout the project duration in each scenario. It should be noted here that when the standard fuzzy arithmetic based solution approaches are implemented, capacity unavailability risks may also arise for these renewable/non-renewable project resources during the project implementation phases (Subulan, 2019 & 2020). Furthermore, by adding the scenario index ($s \in S$) to the relevant fuzzy-stochastic decision variables on the resource availability levels, stochastic nature of the project parameters/inputs can be simply reflected to the project schedules (fuzzy completion times of the activities). More specifically, the required

availability level of each project resource can be acquired under different scenarios. However, it should be mentioned here that such a scenario-based stochastic programming approach may result increases in the problem dimension due to the addition of this scenario index ($s \in S$) to the all model variables. Actually, this is the main disadvantage of this approach. In Equation 4.7, fuzzy completion times of the project activities are described as binary integer variables. The non-preemption of the project activities can also be guaranteed by means of Equation 4.3 and 4.7. Finally, fuzzy-stochastic availability levels of the renewable/non-renewable project resources are also defined as general integer variables in Equation 4.8.

4.1.2 Constraint Programming (CP) Model Formulation of the Multi-Objective FS-MRIPSP

Similarly, CP model formulation is given by Equations 4.9-4.14 by using the mathematical nomenclature in Tables 4.1-4.2. Indeed, CP model of the FS-MRIPSP is denoted by making use of the Optimization Programming Language (OPL) (IBM, 2017). As shown in Table 4.2, some additional special variables are also used in the OPL. Actually, the first interval variable represents an interval of time during a project activity execution under different scenarios. On the other hand, the second one is an optional interval variable in which one can determine not to consider them in the solution project schedule. In other words, we need to define this optional interval variable due to the alternative/different modes for executing a given project activity. The mode selection is generally represented by an exclusive alternative between intervals. The interval variable $\tilde{Y}_{jqs}$ is defined with an option "*optional"*; that is, one can decide not to consider them in the solution project schedule. Actually, these interval variables are provided to express a fuzzy project schedule in the OPL (Okubo et al., 2015).

Table 4.2 Additional mathematical notation for the CP formulation of the FS-MRIPSP

Additional interval type of fuzzy-stochastic decision variables	
$\tilde{X}_{js}$	an interval of fuzzy time during a project activity *j* execution under scenario *s*
$\tilde{Y}_{jqs}$	an optional interval variable for activity *j* when it is processed within an interval of fuzzy time under mode *q* in scenario *s*

$$Minimize\ EV(Project\ Makespan)\ \ \tilde{Z}_1 \cong \sum_{s=1}^{S} \pi_s . endOf\left(\tilde{X}_{Js}\right) \tag{4.9}$$

$$Minimize\ EV(Total\ Resource\ Costs)\ \ \tilde{Z}_2 \cong \sum_{n=1}^{N}\sum_{s=1}^{S} \pi_s . \widetilde{KN}_{ns} + \sum_{r=1}^{R}\sum_{s=1}^{S} \pi_s . \widetilde{KR}_{rs} . \tilde{C}_r \tag{4.10}$$

Subject to;

$$alternative\left(\tilde{X}_{js}, \{\tilde{Y}_{jqs} \mid \forall q \in Q_j\}\right)\ \ \forall j \in J, \forall s \in S \tag{4.11}$$

$$endBeforeStart\left(\tilde{X}_{is}, \tilde{X}_{js}\right)\ \ \forall i \in P_j, \forall j \in J, \forall s \in S \tag{4.12}$$

$$\sum_{j \in J} \sum_{q \in Q_j \mid kr_{jqrs} > 0} pulse\left(\tilde{Y}_{jqs}, kr_{jqrs}\right) \leq \widetilde{KR}_{rs}\ \ \forall r \in R, \forall s \in S \tag{4.13}$$

$$\sum_{j \in J} \sum_{q \in Q_j \mid kn_{jqns} > 0} kn_{jqns} . presenceOf\left(\tilde{Y}_{jqs}\right) \leq \widetilde{KN}_{ns}\ \ \forall n \in N, \forall s \in S \tag{4.14}$$

It should be emphasized that all of the previous CP models for project scheduling problems usually used these interval type variables under deterministic environments (Liess & Michelon, 2008; Berthold et al., 2010; Liu & Song, 2011; Okubo et al., 2015). In other words, deterministic time intervals were commonly defined in the available literature. Unlike the former studies, these interval variables are described under fuzziness throughout this thesis. In fact, fuzzy time intervals are described because of the fuzzy activity durations. Therefore, the proposed CP model is also able to generate fuzzy project schedules as in the previously formulated ILP model. Apart from these additional interval variables, the proposed CP model also utilizes some tuples and special functions like *endOf*, *alternative*, *endBeforeStart*, *pulse* and *presenceOf*. A brief description of each function will be given in this section. For more details, please refer to IBM OPL Language User's Manual (2017). Firstly, function *endOf* returns the fuzzy end time of a given interval variable $\left(\tilde{X}_{Js}\right)$ for dummy ending activity. Thus, objective function in Equation 4.9 minimizes the expected value of project makespan and correspond to the objective function given in Equation 4.1. The second objective function in Equation 4.10 which aims to minimize expected value of total resource usage costs remains the same as in Equation 4.2. In Equation 4.11, *alternative* function is utilized for mode selection of each project activity. Let $\tilde{X}_{js}$ and $\tilde{Y}_{jqs} \mid \forall q \in Q_j$ be

fuzzy intervals. The constraint, i.e., $alternative\left(\tilde{X}_{js}, \{\tilde{Y}_{jqs} \mid \forall q \in Q_j\}\right)$ enables to model an exclusive alternative between $\{\tilde{Y}_{jqs} \mid \forall q \in Q_j\}$. It should also be emphasized here that when a fuzzy interval variable is used in the solution schedule, it is named as *present*, otherwise it is called *absent*. If a fuzzy interval, i.e., $\tilde{X}_{js}$ is present, then exactly one fuzzy interval $\tilde{Y}_{jqs} \mid \forall q \in Q_j$ is present and $\tilde{X}_{js}$ begins and ends together with this selected mode. Therefore, constraint in Equation 4.11 implies that each project activity is processed under only one mode and it is executed only once under each scenario. Actually, this constraint corresponds to the constraint sets in Equations 4.3 and 4.7 in the previous ILP formulation. The constraint function $endBeforeStart\left(\tilde{X}_{is}, \tilde{X}_{js}\right)$ states that the end of a given fuzzy interval variable, i.e., $\tilde{X}_{is}$ is less than or equal to the start of a given fuzzy interval variable $\tilde{X}_{js}$. Thus, constraint 4.12 guarantees the precedence relations among the project activities and corresponds to the constraint set in Equation 4.4. Constraint 4.13 represents the capacity restrictions for all of the renewable resources; where $pulse\left(\tilde{Y}_{jqs}, kr_{jqrs}\right)$ is an elementary cumulative function and its value is equal to kr_{jqrs} during fuzzy time interval in which optional interval $\tilde{Y}_{jqs}\ \forall q \in Q_j$ is *present*. In detail, a fuzzy interval increases the cumulated resource usage at its start time and decreases it when it releases this resource at its end time by making use of *pulse* function. In other words, *pulse* function increases or decreases the level of the resource utilization by a given amount, i.e., kr_{jqrs} for the length of a given fuzzy interval variable $\tilde{Y}_{jqs}\ \forall q \in Q_j$ under each scenario. This constraint set in Equation 4.13 also corresponds to Equation 4.5 in the proposed ILP formulation. Unlike this renewable resource capacity restriction, *presenceOf* function is utilized for satisfying the capacity limitations of non-renewable resources in Equation 4.14. Here, *pulse* function should not be employed since the capacity constraints of non-renewable resources should not be satisfied for the length of a given fuzzy interval. Instead, capacity levels of the non-renewable resources should not be exceeded throughout the entire project duration in each scenario. To do this, the function *presenceOf* is used which returns "1" when a given fuzzy interval $\tilde{Y}_{jqs}\ \forall q \in Q_j$ is *present*. Otherwise, it returns "0" when fuzzy interval $\tilde{Y}_{jqs} \forall q \in Q_j$ is *absent*. More specifically, *presenceOf* function returns the Boolean expression that represents the presence statue of a fuzzy interval variable, i.e., $\tilde{Y}_{jqs} \forall q \in Q_j$. If activity

j is executed under mode q in scenario s, fuzzy interval $\tilde{Y}_{jqs}$ must be *present* and its value is taken as "1"; otherwise, this function returns the value of "0". Constraint set in Equation 4.14 also corresponds to Equation 4.6 in the ILP model.

4.2 The Proposed ILP & CP-Based Transformations with a Bound & Decomposition Principle

In this section, ILP & CP-based transformation approaches with a bound & decomposition principle are presented in order to convert the FS-MRIPSP into its crisp equivalent form. As mentioned previously, in order to avoid the direct usage of standard fuzzy arithmetic operations and fuzzy ranking techniques (Sotoudeh-Anvari, 2020; Subulan, 2020), it is preferred to implement a bound & decomposition principle of Jayalakshmi & Pandian (2012). Because, this approach does not apply fuzzy arithmetic operations directly and it does not require the usage of any fuzzy ranking function and additional auxiliary fuzzy variables. It mainly depends on the decomposition of any given fuzzy mathematical programming problem (FMPP) into three or four crisp problems with additional bounded variables and constraints while dealing with triangular and trapezoidal fuzzy numbers, respectively (Baykasoğlu & Subulan, 2015). Since this approach divides the FMPP into three or four crisp equivalent problems, model dimension, i.e., the number of decision variables and constraints do not increase excessively because of this decomposition stage. In fact, this is one of its important advantages for solving such a combinatorial fuzzy optimization problem, i.e., FS-MRIPSP. On the other hand, standard fuzzy arithmetic based transformation approaches may result in considerable increases in the model size which makes the solution of the problem much more difficult. The details of this approach can be summarized through the following sections.

4.2.1 ILP-Based Transformation Approach with a Bound & Decomposition Principle

When the aforementioned fuzzy project parameters and variables are represented by triangular fuzzy numbers, the previous ILP formulation given by Equations 4.1-4.8

can be reformulated as in the following Equations 4.15-4.22. Indeed, additional indices, i.e., *l*, *m* and *u* are added to all fuzzy parameters and decision variables so as to represent the core, left and right margins of the triangular fuzzy numbers, respectively.

$$Minimize\ (Z1_l, Z1_m, Z1_u) = \sum_{(t_l,t_m,t_u)=(EFT_{Jl},EFT_{Jm},EFT_{Ju})}^{(LFT_{Jl},LFT_{Jm},LFT_{Ju})} \sum_{s=1}^{S} \pi_s \,.\, (t_l, t_m, t_u).\, x_{J1(t_l,t_m,t_u)s} \tag{4.15}$$

$$Minimize\ (Z2_l, Z2_m, Z2_u) = \sum_{n=1}^{N}\sum_{s=1}^{S} \pi_s.\, (KN_{nsl}, KN_{nsm}, KN_{nsu}) + \sum_{r=1}^{R}\sum_{s=1}^{S} \pi_s \,.\, (KR_{rsl}, KR_{rsm}, KR_{rsu}).\, (C_{rl}, C_{rm}, C_{ru}) \tag{4.16}$$

Subject to;

$$\sum_{q=1}^{Q_j} \sum_{(t_l,t_m,t_u)=(EFT_{jl},EFT_{jm},EFT_{ju})}^{(LFT_{jl},LFT_{jm},LFT_{ju})} x_{jq(t_l,t_m,t_u)s} = 1 \quad \forall j \in J, \forall s \in S \tag{4.17}$$

$$\sum_{q=1}^{Q_i} \sum_{(t_l,t_m,t_u)=(EFT_{il},EFT_{im},EFT_{iu})}^{(LFT_{il},LFT_{im},LFT_{iu})} (t_l, t_m, t_u).\, x_{iq(t_l,t_m,t_u)s} \leq \sum_{q=1}^{Q_j} \sum_{(t_l,t_m,t_u)=(EFT_{jl},EFT_{jm},EFT_{ju})}^{(LFT_{jl},LFT_{jm},LFT_{ju})} \{(t_l, t_m, t_u) \ominus (d_{jql}, d_{jqm}, d_{jqu})\}.\, x_{jq(t_l,t_m,t_u)s} \quad \forall j \in J, \forall i \in P_j, \forall s \in S \tag{4.18}$$

$$\sum_{j=2}^{J-1}\sum_{q=1}^{Q_j} kr_{jqrs} \,.\, \sum_{(\tau_l,\tau_m,\tau_u)=(t_l,t_m,t_u)}^{(t_l,t_m,t_u)\oplus(d_{jql},d_{jqm},d_{jqu})-1} x_{jq(\tau_l,\tau_m,\tau_u)s} \leq (KR_{rsl}, KR_{rsm}, KR_{rsu}) \quad \forall r \in R, \forall (t_l, t_m, t_u) \in T, \forall s \in S \tag{4.19}$$

$$\sum_{j=2}^{J-1}\sum_{q=1}^{Q_j} kn_{jqns} \,.\, \sum_{(t_l,t_m,t_u)=(EFT_{jl},EFT_{jm},EFT_{ju})}^{(LFT_{jl},LFT_{jm},LFT_{ju})} x_{jq(t_l,t_m,t_u)s} \leq (KN_{nsl}, KN_{nsm}, KN_{nsu}) \quad \forall n \in N, \forall s \in S \tag{4.20}$$

$$x_{jq(t_l,t_m,t_u)s} \in \{0,1\} \quad \forall j \in J, q \in Q_j, \forall s \in S, (t_l, t_m, t_u) \in \{(EFT_{jl}, EFT_{jm}, EFT_{ju}), (LFT_{jl}, LFT_{jm}, LFT_{ju})\} \tag{4.21}$$

$$(KR_{rsl}, KR_{rsm}, KR_{rsu}), (KN_{nsl}, KN_{nsm}, KN_{nsu}) \geq 0 \ \ and \ \ integer \quad \forall r \in R, \forall n \in N, \forall s \in S \tag{4.22}$$

By applying the bound & decomposition principle of Jayalakshmi & Pandian (2012), the above fuzzy-stochastic model is first decomposed into the middle level, upper level and lower level problems (Decomposition stage). Afterward, these crisp equivalent problems should be solved separately and fuzzy efficient solutions of the FS-MRIPSP can be generated by utilizing their solutions as bounded variables and constraints. It should be mentioned here that middle level problem in which at least one decision variable of the middle level problem exists into the constraints is solved at first. The crisp equivalent form of the middle level problem can be formulated as in the following Equations 4.23-4.30.

$$Minimize \quad Z1_m = \sum_{t_m=EFT_{Jm}}^{LFT_{Jm}} \sum_{s=1}^{S} \pi_s . t_m . x_{J1t_ms} \tag{4.23}$$

$$Minimize \quad Z2_m = \sum_{n=1}^{N} \sum_{s=1}^{S} \pi_s . KN_{nsm} + \sum_{r=1}^{R} \sum_{s=1}^{S} \pi_s . KR_{rsm} . C_{rm} \tag{4.24}$$

Subject to;

$$\sum_{q=1}^{Q_j} \sum_{t_m=EFT_{jm}}^{LFT_{jm}} x_{jqt_ms} = 1 \quad \forall j \in J, \forall s \in S \tag{4.25}$$

$$\sum_{q=1}^{Q_i} \sum_{t_m=EFT_{im}}^{LFT_{im}} t_m . x_{iqt_ms} \leq \sum_{q=1}^{Q_j} \sum_{t_m=EFT_{jm}}^{LFT_{jm}} \{t_m - d_{jqm}\} . x_{jqt_ms} \quad \forall j \in J, \forall i \in P_j, \forall s \in S \tag{4.26}$$

$$\sum_{j=2}^{J-1} \sum_{q=1}^{Q_j} kr_{jqrs} . \sum_{\tau_m=t_m}^{t_m+d_{jqm}-1} x_{jq\tau_ms} \leq KR_{rsm} \quad \forall r \in R, \forall t_m \in T, \forall s \in S \tag{4.27}$$

$$\sum_{j=2}^{J-1} \sum_{q=1}^{Q_j} kn_{jqns} . \sum_{t_m=EFT_{jm}}^{LFT_{jm}} x_{jqt_ms} \leq KN_{nsm} \quad \forall n \in N, \forall s \in S \tag{4.28}$$

$$x_{jqt_ms} \in \{0,1\} \quad \forall j \in J, q \in Q_j, \forall s \in S, t_m \in \{EFT_{jm}, LFT_{jm}\} \tag{4.29}$$

$$KR_{rsm}, KN_{nsm} \geq 0 \quad and \quad integer \quad \forall r \in R, \forall n \in N, \forall s \in S \tag{4.30}$$

After generating the solution of the above middle level problem, the upper level problem in which at least one decision variable of this upper level problem occurs into

the constraints and should not take place in the middle level problem is formulated as in Equations 4.31-4.43. Before solving this upper level problem, values of the decision variables are replaced with the values provided by the middle level problem. In other words, outputs of the middle level problem are used as input data while formulating the bound constraints of the upper level problem (see Equations 4.39-4.41). Additionally, this upper level problem should also satisfy the bounded constraints in Equations 4.33-4.34 which are related to the core and right margins of fuzzy objective values.

$$Minimize \quad Z1_u = \sum_{t_u=EFT_{Ju}}^{LFT_{Ju}} \sum_{s=1}^{S} \pi_s . t_u . x_{J1t_us} \tag{4.31}$$

$$Minimize \quad Z2_u = \sum_{n=1}^{N} \sum_{s=1}^{S} \pi_s . KN_{nsu} + \sum_{r=1}^{R} \sum_{s=1}^{S} \pi_s . KR_{rsu} . C_{ru} \tag{4.32}$$

Subject to;

$$Z1_u \geq Z1_m^* \tag{4.33}$$

$$Z2_u \geq Z2_m^* \tag{4.34}$$

$$\sum_{q=1}^{Q_j} \sum_{t_u=EFT_{ju}}^{LFT_{ju}} x_{jqt_us} = 1 \quad \forall j \in J, \forall s \in S \tag{4.35}$$

$$\sum_{q=1}^{Q_i} \sum_{t_u=EFT_{iu}}^{LFT_{iu}} t_u . x_{iqt_us} \leq \sum_{q=1}^{Q_j} \sum_{t_u=EFT_{ju}}^{LFT_{ju}} \{t_u - d_{jqu}\} . x_{jmt_us} \quad \forall j \in J, \forall i \in P_j, \forall s \in S \tag{4.36}$$

$$\sum_{j=2}^{J-1} \sum_{q=1}^{Q_j} kr_{jqrs} . \sum_{\tau_u=t_u}^{t_u+d_{jqu}-1} x_{jq\tau_us} \leq KR_{rsu} \quad \forall r \in R, \forall t_u \in T, \forall s \in S \tag{4.37}$$

$$\sum_{j=2}^{J-1} \sum_{q=1}^{Q_j} kn_{jqns} . \sum_{t_u=EFT_{ju}}^{LFT_{ju}} x_{jqt_us} \leq KN_{nsu} \quad \forall n \in N, \forall s \in S \tag{4.38}$$

$$t_u . x_{jqt_us} \geq t_m^* . x_{jq^*t_m^*s} \quad \forall j \in J, \forall s \in S, q = q^* \tag{4.39}$$

$$KR_{rsu} \geq KR_{rsm}^* \quad \forall r \in R, \forall s \in S \tag{4.40}$$

$$KN_{nsu} \geq KN^{*}_{nsm} \quad \forall n \in N, \forall s \in S \tag{4.41}$$

$$x_{jqt_us} \in \{0,1\} \quad \forall j \in J, q \in Q_j, \forall s \in S, t_u \in \{EFT_{ju}, LFT_{ju}\} \tag{4.42}$$

$$KR_{rsu}, KN_{nsu} \geq 0 \quad and \quad integer \quad \forall r \in R, \forall n \in N, \forall s \in S \tag{4.43}$$

According to constraint sets in Equations 4.33-4.34, objective function values of the upper level problem should be greater than the objective values of the middle level problem. Actually, these logical constraints enable to form of a triangular fuzzy objective value for the stated FS-MRIPSP. Constraint set in Equation 4.39 ensures that right margins of the project activities' fuzzy completion times should be greater than the core or most likely values of these fuzzy completion times under each scenario. Indeed, the project activities' completion times which are obtained from the upper level problem should be larger than their most likely values produced by the middle level problem. In a similar manner, bound constraints are also formulated for the availability levels of both renewable and non-renewable project resources as in Equations 4.40-4.41. Actually, these are the bound constraints added to the upper level problem. After solving the middle and upper level problems, lower level problem in which at least one decision variable of this lower level problem occurs into the constraints and should not take place in both of the middle and upper level problems is formulated in Equations 4.44-4.57.

$$Minimize \quad Z1_l = \sum_{t_l=EFT_{Jl}}^{LFT_{Jl}} \sum_{s=1}^{S} \pi_s . t_l . x_{J1t_ls} \tag{4.44}$$

$$Minimize \quad Z2_l = \sum_{n=1}^{N} \sum_{s=1}^{S} \pi_s . KN_{nsl} + \sum_{r=1}^{R} \sum_{s=1}^{S} \pi_s . KR_{rsl} . C_{rl} \tag{4.45}$$

Subject to;

$$Z1_l \leq Z1^{*}_m \tag{4.46}$$

$$Z2_l \leq Z2^{*}_m \tag{4.47}$$

$$\sum_{q=1}^{Q_j} \sum_{t_l=EFT_{jl}}^{LFT_{jl}} x_{jqt_ls} = 1 \quad \forall j \in J, \forall s \in S \tag{4.48}$$

$$\sum_{q=1}^{Q_i}\sum_{t_l=EFT_{il}}^{LFT_{il}} t_l.x_{iqt_ls} \le \sum_{q=1}^{Q_j}\sum_{t_l=EFT_{jl}}^{LFT_{jl}} \{t_l - d_{jql}\}.x_{jmt_ls} \quad \forall j \in J, \forall i \in P_j, \forall s \in S \tag{4.49}$$

$$\sum_{j=2}^{J-1}\sum_{q=1}^{Q_j} kr_{jqrs} . \sum_{\tau_l=t_l}^{t_l+d_{jql}-1} x_{jq\tau_ls} \le KR_{rsl} \quad \forall r \in R, \forall t_l \in T, \forall s \in S \tag{4.50}$$

$$\sum_{j=2}^{J-1}\sum_{q=1}^{Q_j} kn_{jqns} . \sum_{t_l=EFT_{jl}}^{LFT_{jl}} x_{jqt_ls} \le KN_{nsl} \quad \forall n \in N, \forall s \in S \tag{4.51}$$

$$t_l.x_{jqt_ls} \le t_m^*.x_{jq^*t_m^*s} \quad \forall j \in J, \forall s \in S, q = q^* \tag{4.52}$$

$$\sum_{q\neq q^*}^{Q_j}\sum_{t_l=EFT_{jl}}^{LFT_{jl}} t_l.x_{jqt_ls} = 0 \quad \forall j \in J, \forall s \in S \tag{4.53}$$

$$KR_{rsl} \le KR_{rsm}^* \quad \forall r \in R, \forall s \in S \tag{4.54}$$

$$KN_{nsl} \le KN_{nsm}^* \quad \forall n \in N, \forall s \in S \tag{4.55}$$

$$x_{jqt_ls} \in \{0,1\} \quad \forall j \in J, q \in Q_j, \forall s \in S, t_l \in \{EFT_{jl}, LFT_{jl}\} \tag{4.56}$$

$$KR_{rsl}, KN_{nsl} \ge 0 \quad and \quad integer \quad \forall r \in R, \forall n \in N, \forall s \in S \tag{4.57}$$

In a similar way, before solving the lower level problem, values of the decision variables are replaced with the obtained solutions from the middle and upper level problems. The lower level problem should also satisfy the bound constraints given in Equations 4.46-4.47 and 4.52-4.55. Constraint sets in Equations 4.44-4.45 guarantee that objective function values of the lower level problem should be equal or less than the objective values of the middle level problem. In fact, these logical bound constraints make sure that fuzzy objective values will take a form of triangular fuzzy numbers. Constraint set in Equation 4.52 satisfies that left margins of the project activities' fuzzy completion times should be lower than the core or most likely values of these fuzzy completion times under each scenario. It should be highlighted here that the same execution mode $(q^* \in Q_j)$ which is provided by the solution middle level problem should be utilized in these bound constraints (Equations 4.39 and 4.52) of the upper and lower level problems, respectively. Because, a project activity should be

completed at a fuzzy time with the same execution mode under each scenario. In other words, different execution modes cannot be selected for the same project activity under the same scenario in the lower, middle, and upper level problems. Indeed, constraint set in Equation 4.39 of the upper level problem can meet this statement by oneself. On the other hand, an additional logical constraint in Equation 4.53 is also required to satisfy this condition in the lower level problem. Because, on the contrary to the constraint set in Equation 4.39, Equation 4.52 does not guarantee the selection of the same execution mode in the middle level problem by oneself. By formulating an additional logical constraint as in Equation 4.53, a different execution mode which is not found by the solution of middle level problem cannot be chosen for any project activity under the same scenario. Fortunately, this logical restriction can be automatically satisfied and does not need by the CP model of the lower level problem. In Equations 4.54-4.55, bound constraints are also formulated so as to obtain the optimistic values of the resources' availability levels. When these middle, upper and lower level problems are solved sequentially, their separate solutions are brought together to generate fuzzy efficient solutions, i.e., project schedules, $x_{jq(t_l,t_m,t_u)s}$ and resources' availability levels, $(KR_{rsl}, KR_{rsm}, KR_{rsu})$, $(KN_{nsl}, KN_{nsm}, KN_{nsu})$ under each scenario with the expected values of fuzzy project makespan, $(Z1_l, Z1_m, Z1_u)$ and total resource usage costs $(Z2_l, Z2_m, Z2_u)$.

4.2.2 CP-Based Transformation Approach with a Bound & Decomposition Principle

Similar to the ILP formulations given in Equations 4.15-4.22, the previously formulated CP model in Equations 4.9-4.14 can be reformulated as in the following Equations 4.58-4.63 when dealing with fuzzy project parameters and variables. Actually, an additional index is incorporated into this formulation to represent the core, left and right margins of the triangular fuzzy numbers.

$$Minimize \quad (Z1_l, Z1_m, Z1_u) = \sum_{s=1}^{S} \pi_s . endOf(X_{Jsl}, X_{Jsm}, X_{Jsu}) \tag{4.58}$$

$$Minimize\ (Z2_l, Z2_m, Z2_u) = \sum_{n=1}^{N}\sum_{s=1}^{S} \pi_s.(KN_{nsl}, KN_{nsm}, KN_{nsu}) + \sum_{r=1}^{R}\sum_{s=1}^{S} \pi_s.(KR_{rsl}, KR_{rsm}, KR_{rsu}).(C_{rl}, C_{rm}, C_{ru}) \quad (4.59)$$

Subject to;

$$alternative\left((X_{jsl}, X_{jsm}, X_{jsu}), \{(Y_{jqsl}, Y_{jqsm,} Y_{jqsu}) |\ q \in Q_j\}\right) \quad \forall j \in J, \forall s \in S \quad (4.60)$$

$$endBeforeStart\left((X_{isl}, X_{ism}, X_{isu}), (X_{jsl}, X_{jsm}, X_{jsu})\right) \quad \forall i \in P_j, \forall j \in J, \forall s \in S \quad (4.61)$$

$$\sum_{j\in J}\sum_{q\in Q_j \mid kr_{jqrs}>0} pulse\left((Y_{jqsl}, Y_{jqsm,} Y_{jqsu}), kr_{jqrs}\right) \leq (KR_{rsl}, KR_{rsm}, KR_{rsu}) \quad \forall r \in R, \forall s \in S \quad (4.62)$$

$$\sum_{j\in J}\sum_{q\in Q_j \mid kn_{jqns}>0} kn_{jqns}.presenceOf\left((Y_{jqsl}, Y_{jqsm,} Y_{jqsu})\right) \leq (KN_{nsl}, KN_{nsm}, KN_{nsu}) \quad \forall n \in N, \forall s \in S \quad (4.63)$$

Similarly, CP model of the crisp equivalent middle level problem can be formulated as in the following Equations 4.64-4.70. It should be noted here that only the core or most likely values of the fuzzy activity durations, completions times and availability levels are utilized in this middle level problem formulation.

$$Minimize \quad Z1_m = \sum_{s=1}^{S} \pi_s.endOf(X_{Jsm}) \quad (4.64)$$

$$Minimize \quad Z2_m = \sum_{n=1}^{N}\sum_{s=1}^{S} \pi_s.KN_{nsm} + \sum_{r=1}^{R}\sum_{s=1}^{S} \pi_s.KR_{rsm}.C_{rm} \quad (4.65)$$

Subject to;

$$endOf\left(X_{Jsm}\right) \leq LFT_{Jm} \quad \forall s \in S \quad (4.66)$$

$$alternative(X_{jsm}, \{Y_{jqsm} \mid q \in Q_j\}) \quad \forall j \in J, \forall s \in S \quad (4.67)$$

$$endBeforeStart(X_{ism}, X_{jsm}) \quad \forall i \in P_j, \forall j \in J, \forall s \in S \quad (4.68)$$

$$\sum_{j\in J}\sum_{q\in Q_j \mid kr_{jqrs}>0} pulse\left(Y_{jqsm}, kr_{jqrs}\right) \leq KR_{rsm} \quad \forall r \in R, \forall s \in S \quad (4.69)$$

$$\sum_{j\in J}\sum_{q\in Q_j \mid kn_{jqns}>0} kn_{jqns}.presenceOf(Y_{jqsm}) \le KN_{nsm} \quad \forall n \in N, \forall s \in S \tag{4.70}$$

It should be emphasized here that completion time of the dummy ending activity is controlled by making use of the constraint set in Equation 4.66 to satisfy that end of the finishing time interval should be equal or lower than the previously computed latest finish time under each scenario. Therefore, this CP model formulation will also be consistent with the ILP model which uses the earliest and latest finishing time information. After solving this crisp middle level problem by making use of the constraint programming solver of IBM ILOG OPL Optimization Studio, CP model formulation of the crisp equivalent upper level problem can be constructed as in Equations 4.71-4.82. Only the pessimistic values or right margins of fuzzy activity durations, completion times and availability levels are involved in this formulation.

$$Minimize \quad Z1_u = \sum_{s=1}^{S} \pi_s.endOf(X_{Jsu}) \tag{4.71}$$

$$Minimize \quad Z2_u = \sum_{n=1}^{N}\sum_{s=1}^{S} \pi_s.KN_{nsu} + \sum_{r=1}^{R}\sum_{s=1}^{S} \pi_s.KR_{rsu}.C_{ru} \tag{4.72}$$

Subject to;

$$Z1_u \ge Z1_m^* \tag{4.73}$$

$$endOf(X_{Jsu}) \le LFT_{Ju} \quad \forall s \in S \tag{4.74}$$

$$Z2_u \ge Z2_m^* \tag{4.75}$$

$$alternative(X_{jsu}, \{Y_{jqsu} \mid q = q*\}) \quad \forall j \in J, \forall s \in S \tag{4.76}$$

$$endBeforeStart(X_{isu}, X_{jsu}) \quad \forall i \in P_j, \forall j \in J, \forall s \in S \tag{4.77}$$

$$\sum_{j\in J}\sum_{q\in Q_j \mid kr_{jqrs}>0} pulse(Y_{jqsu}, kr_{jqrs}) \le KR_{rsu} \quad \forall r \in R, \forall s \in S \tag{4.78}$$

$$\sum_{j\in J}\sum_{q\in Q_j \mid kn_{jqns}>0} kn_{jqns}.presenceOf(Y_{jqsu}) \le KN_{nsu} \quad \forall n \in N, \forall s \in S \tag{4.79}$$

$$endOf(Y_{jqsu} \mid q = q*) \ge endOf(Y_{jqsm}^*) \quad \forall j \in J, \forall s \in S \tag{4.80}$$

$$KR_{rsu} \geq KR^{*}_{rsm} \quad \forall r \in R, \forall s \in S \tag{4.81}$$

$$KN_{nsu} \geq KN^{*}_{nsm} \; \forall n \in N, \forall s \in S \tag{4.82}$$

The above CP model also includes similar bound constraints in Equations 4.73-4.65 and 4.80-4.82. Actually, these bound constraints related to the fuzzy objective values are the same with the previous ILP formation of the upper level problem in Equations 4.33-4.34. Apart from the ILP formulation, *endOf* function is utilized in some bound constraints of the CP model to satisfy that the right margins of the fuzzy completion times should be greater than their most likely or core values (see Equation 4.80). Moreover, pessimistic completion time of the dummy ending activity should also be equal or lower than the right margin of its fuzzy latest finishing time (see Equation 4.74). It is clear in Equations 4.76 and 4.80 that the same execution mode which is previously specified by the solution of the middle level problem must be selected for each project activity under each scenario. Similarly, CP model formulation of the crisp equivalent lower level problem is presented in Equations 4.83-4.94.

$$Minimize \quad Z1_l = \sum_{s=1}^{S} \pi_s . endOf(X_{Jsl}) \tag{4.83}$$

$$Minimize \quad Z2_l = \sum_{n=1}^{N} \sum_{s=1}^{S} \pi_s . KN_{nsl} + \sum_{r=1}^{R} \sum_{s=1}^{S} \pi_s . KR_{rsl} . C_{rl} \tag{4.84}$$

Subject to;

$$Z1_l \leq Z1^{*}_{m} \tag{4.85}$$

$$Z2_l \leq Z2^{*}_{m} \tag{4.86}$$

$$endOf(X_{Jsl}) \leq LFT_{Jl} \quad \forall s \in S \tag{4.87}$$

$$alternative(X_{jsl}, \{Y_{jqsl} \mid \; q = q*\}) \quad \forall j \in J, \forall s \in S \tag{4.88}$$

$$endBeforeStart(X_{isl}, X_{jsl}) \quad \forall i \in P_j, \forall j \in J, \forall s \in S \tag{4.89}$$

$$\sum_{j \in J} \sum_{q \in Q_j \mid kr_{jqrs} > 0} pulse(Y_{jqsl}, kr_{jqrs}) \leq KR_{rsl} \quad \forall r \in R, \forall s \in S \tag{4.90}$$

$$\sum_{j\in J}\sum_{q\in Q_j \mid kn_{jqns}>0} kn_{jqns}\,.\,presenceOf\left(Y_{jqsl}\right) \le KN_{nsl} \quad \forall n \in N, \forall s \in S \tag{4.91}$$

$$endOf(Y_{jqsl} \mid\ q = q*) \ \le endOf\left(Y^*_{jqsm}\right) \quad \forall j \in J, \forall s \in S \tag{4.92}$$

$$KR_{rsl} \le KR^*_{rsm} \quad \forall r \in R, \forall s \in S \tag{4.93}$$

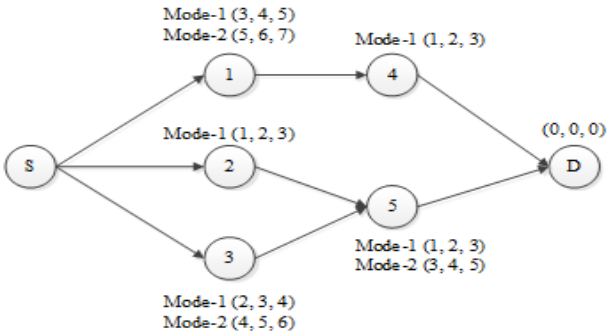
$$KN_{nsl} \le KN^*_{nsm} \quad \forall n \in N, \forall s \in S \tag{4.94}$$

In the above formulation, only the optimistic values or left margins of the fuzzy activity durations, completion times and availability levels are taken into account. After solving the middle, upper and lower level problems sequentially, their solutions are aggregated to produce fuzzy efficient solutions related to (i) project schedules i.e., $\left(X_{jsl}, X_{jsm}, X_{jsu}\right)$ and $\left(Y_{jqsl}, Y_{jqsm}, Y_{jqsu}\right)$, (ii) resources' availability levels, i.e., $(KR_{rsl}, KR_{rsm}, KR_{rsu})$ and $(KN_{nsl}, KN_{nsm}, KN_{nsu})$ under each scenario with the expected values of the fuzzy objectives $(Z1_l, Z1_m, Z1_u)$ and $(Z2_l, Z2_m, Z2_u)$.

4.3 Illustrative Example of Classical MRCPSP under Mixed Fuzzy & Stochastic Uncertainties

This section presents a basic example of the classical MRCPSP with a single makespan criterion in order to show validity and applicability of the proposed ILP & CP-based transformation approaches. Actually, this example problem is derived based on a deterministic problem instance from the available literature (Kolisch & Drexl, 1997). Indeed, that original deterministic problem is extended by including the following fuzzy and stochastic project data. The project is composed of 5 non-dummy activities with 2 modes and they require uncertain amounts of monotype renewable and non-renewable resources whose capacities are also uncertain. The activity-on-node (AON) project network diagram is shown in Figure 4.2 with fuzzy activity durations under distinct modes above/below each node. The fuzzy EFTs & LFTs of the project activities are also calculated and presented in the same figure. The start time of the project is certain and equal to $t_0 = (0, 0, 0)$. The length of the project planning horizon is computed as 24 discrete-time periods by summing maximum

activity durations that are chosen from among all the execution modes. In contrast to the fuzzy activity durations, resource requirements of the project activities and the resources' capacities are considered as stochastic project parameters since they may have randomness in nature. In Table 4.3, stochastic data related to the resources' capacities and the resource requirements of the project activities are given with the pre-specified discrete probabilities under each mode. Actually, these project parameters are defined as discrete random variables with known probabilities. In detail, it is assumed that both renewable and non-renewable resources may have low or high capacity levels with a discrete probability during the project realization phase.



Activity	Fuzzy EFT	Fuzzy LFT
S	(0, 0, 0)	(10, 13, 16)
1	(3, 4, 5)	(13, 17, 21)
2	(1, 2, 3)	(13, 17, 21)
3	(2, 3, 4)	(13, 17, 21)
4	(4, 6, 8)	(14, 19, 24)
5	(3, 5, 7)	(14, 19, 24)
D	(4, 6, 8)	(14, 19, 24)

Figure 4.2 AON project network diagram with fuzzy durations and the EFTs and LFTs

Similarly, project activities may require low or high amounts of the resources with respect to the dynamically changing conditions. Based on this random data, 16 different scenarios with equal joint probabilities are generated as depicted in Figure 4.3.

Table 4.3 Stochastic project data with discrete probabilities: Resources' capacities and resource requirements of the activities

Project Activities	**Renewable Resource**				**Non-renewable Resource**			
	Low (Pr. 50%)		**High (Pr. 50%)**		**Low (Pr. 50%)**		**High (Pr. 50%)**	
	Mode-1	**Mode-2**	**Mode-1**	**Mode-2**	**Mode-1**	**Mode-2**	**Mode-1**	**Mode-2**
S	0	*	0	*	0	*	0	*
1	2	1	3	3	2	0.5	3	1
2	1	*	2	*	0	*	1	*
3	3	1	4	2	3	1	4	2
4	1	*	2	*	0	*	1	*
5	2	1	3	2	2	0.5	3	1
D	0	*	0	*	0	*	0	*
Random resource capacities	4		6		8		10	

(*) Not available, (Pr.) Scenario probability

When the previously formulated ILP & CP models of the crisp equivalent lower, middle and upper level problems are run under these fuzzy & stochastic data through ILOG CPLEX optimization studio version 12.10 on an Intel Core i7 2.4 GHz IBM PC, optimization results are obtained as given in Table 4.4. The fuzzy expected objective values, i.e., project makespan provided by both ILP and CP-based transformation approaches are also demonstrated in Figure 4.4.

Table 4.4 Optimization details for ILP & CP models of the crisp equivalent lower, middle and upper problems

Optimization results	**ILP-based transformation**			**CP-based transformation**		
	Lower Level	**Middle Level**	**Upper Level**	**Lower Level**	**Middle Level**	**Upper Level**
Model class		PILP			CP	
Total integer variables	3106	4098	5090	272	272	272
Total constraints	722	577	770	2169	276	2169
Total non-zero elements	21186	28113	41474	*	*	*
Total solver iterations	21	27	-	*	*	*
Solutions found	4	3	1	1	7	1
Optimality gap (%)	-	-	-	-	-	-
CPU time (sec.)	0.03	0.20	0.19	0.07	6.62	0.95
Fuzzy expected makespan (objective value)	(5.812, 8.4375, 11.0625)			(5.875, 8.4375, 11.0)		

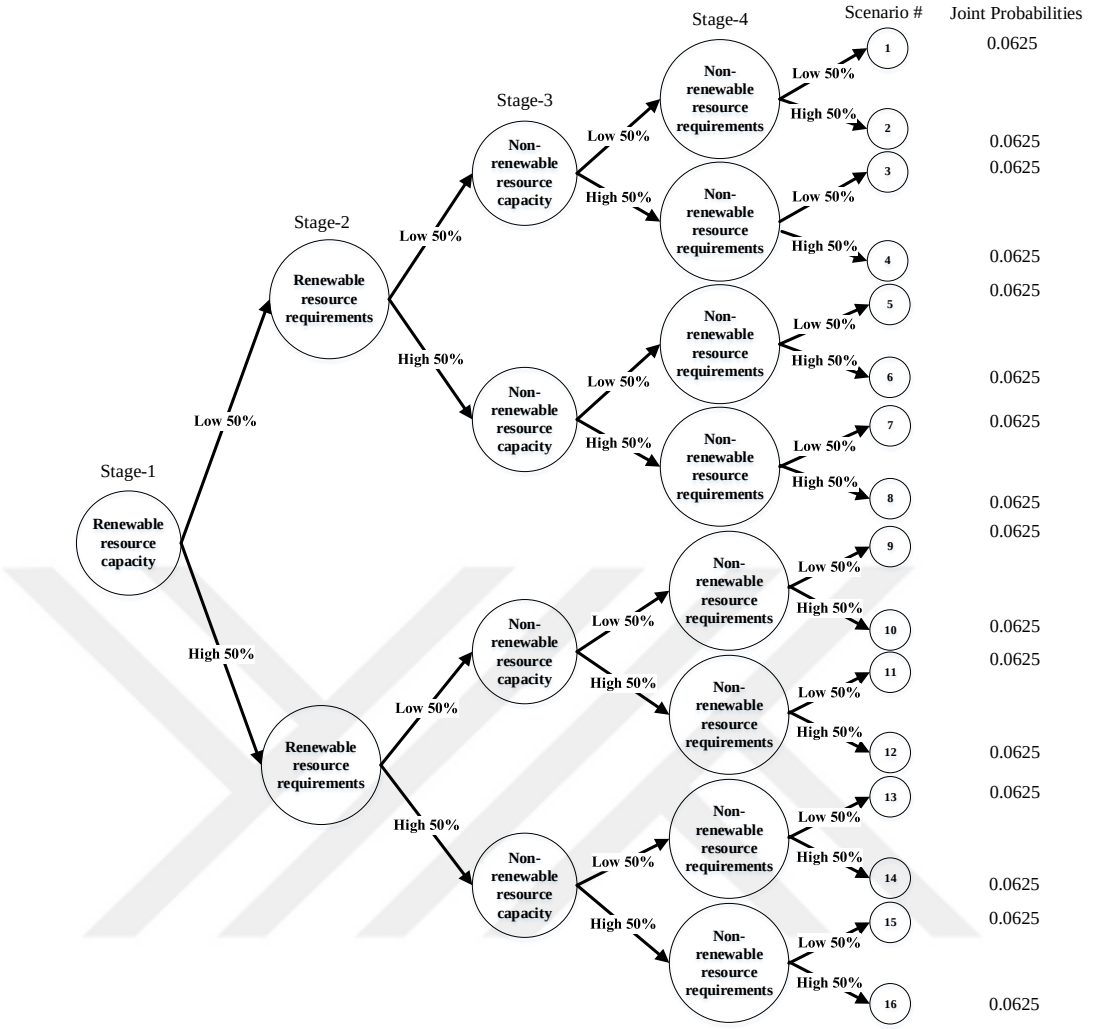


Figure 4.3 Scenario tree for stochastic capacities of the resources and resource requirements of the project activities

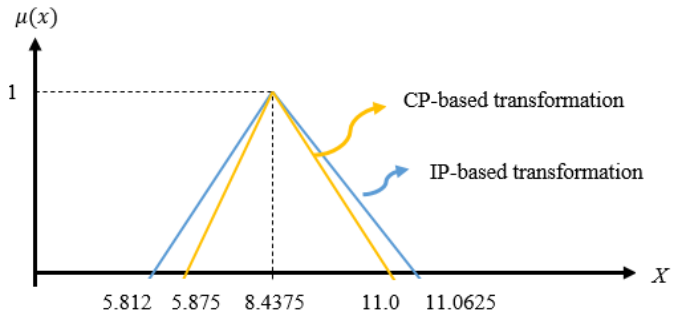


Figure 4.4 Comparison of membership functions for fuzzy expected project makespan provided by ILP & CP-based approaches

When the results of the ILP and CP-based transformation approaches are compared, it is recognized that they found the same most likely or core value (8.4375) for the expected project makespan objective. This means that the ILP and CP models of the middle level problem generated the same objective value. However, the left and right margins of the fuzzy expected project makespan objective are differentiated as clearly seen in Figure 4.4. This is because of different alternative optimal solutions may found by solving the middle level problem via ILP and CP-based transformation approaches. In detail, solution of the middle level problem by ILP and CP-based approaches leads to different alternative solutions with the same objective value, i.e., 8.4375. Although these alternative optimal solutions have the same objective value, they include different mode selections for some of the project activities. However, as mentioned previously in Sections 4.2.1 and 4.2.2, any given project activity should be executed under the same mode for all of the middle, lower and upper level problems. In other words, selected modes by solving the lower and upper level problems have to be the same with the chosen modes by solution of the middle level problem. For this reason, alternative optimal solutions of the middle level problem with different mode selections may result different left and right margins for the fuzzy expected project makespan. Fortunately, when we compared these fuzzy expected project makespan values, i.e., (5.812, 8.4375, 11.0625) and (5.875, 8.4375, 11.0) by using any fuzzy ranking technique, it is clear that none of them is superior or inferior to each other. In other words, both fuzzy expected project makespan values can be equally accepted by the project manager(s). For this reason, it can be concluded that there is no significant differences between fuzzy efficient solutions found by the ILP and CP-based transformation approaches. Furthermore, these fuzzy efficient solutions are also found within similar computational times. Details of the fuzzy completion times obtained from the proposed ILP and CP-based transformation approaches are also presented in Appendix-4. It is obviously seen in Appendix-4 that except scenario#7, the same execution modes are selected under all of the scenarios and the same fuzzy project makespan are obtained. However, different modes, i.e., 1 versus 2, are selected by the proposed ILP and CP-based approaches for activity#3 under scenario#7. Therefore, different fuzzy project makespan values, i.e., (7, 11, 15) and (8, 11, 14) are generated for scenario#7. Actually, this difference in scenario#7 causes aforementioned variation

in the fuzzy expected project makespan values, i.e., (5.812, 8.4375, 11.0625) and (5.875, 8.4375, 11.0). Lastly, some of the generated fuzzy project schedules under different scenarios can be visualized by the following Gantt charts as in Figures 4.5-4.8.

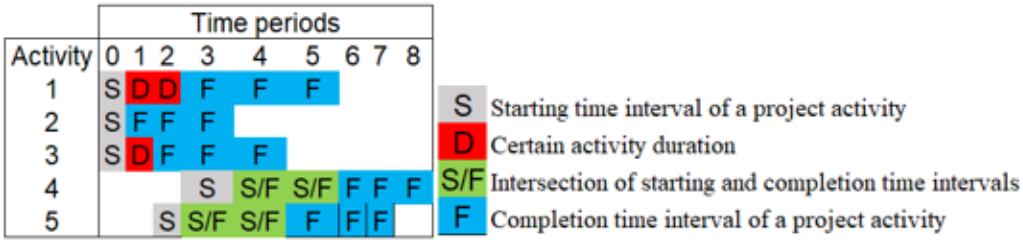


Figure 4.5 Gantt chart of a fuzzy project schedule under high resource capacities & low requirements (scenario#11)

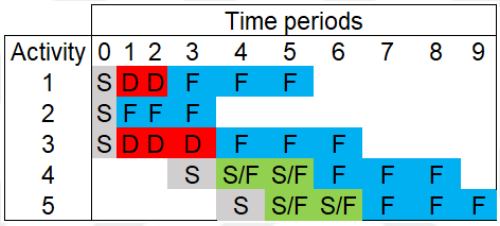


Figure 4.6 Gantt chart of a fuzzy project schedule under low resource capacities & low requirements (scenario#1)

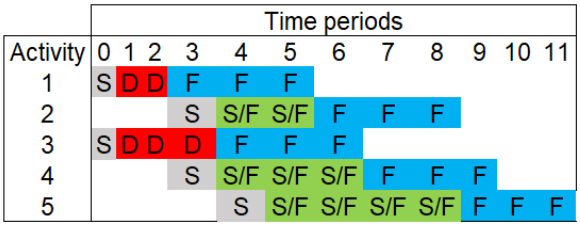


Figure 4.7 Gantt chart of a fuzzy project schedule under high resource capacities & high requirements (scenario#16)

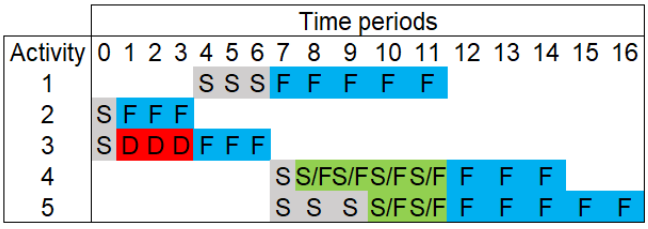


Figure 4.8 Gantt chart of a fuzzy project schedule under low resource capacities & high requirements (scenario#6)

It is clearly seen in Figure 4.5-4.8 that the project makespan will be influenced by stochastic resources' capacities and the amounts of resource requirements. In fact, fuzzy expected project makespan will be lengthier in case of the lower resource capacities and the higher resource requirements of the project activities (see Figure 4.8 and scenario#6). Some of the corresponding renewable resource profiles for the generated solutions under different scenarios#6 and #11 are also demonstrated in the following Figures. 4.9-4.10.

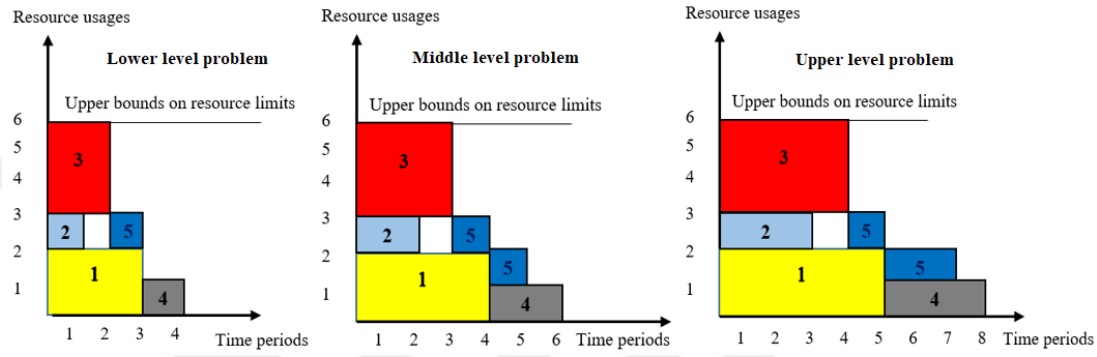


Figure 4.9 The renewable resource profiles under high capacity & low requirements (scenario#11)

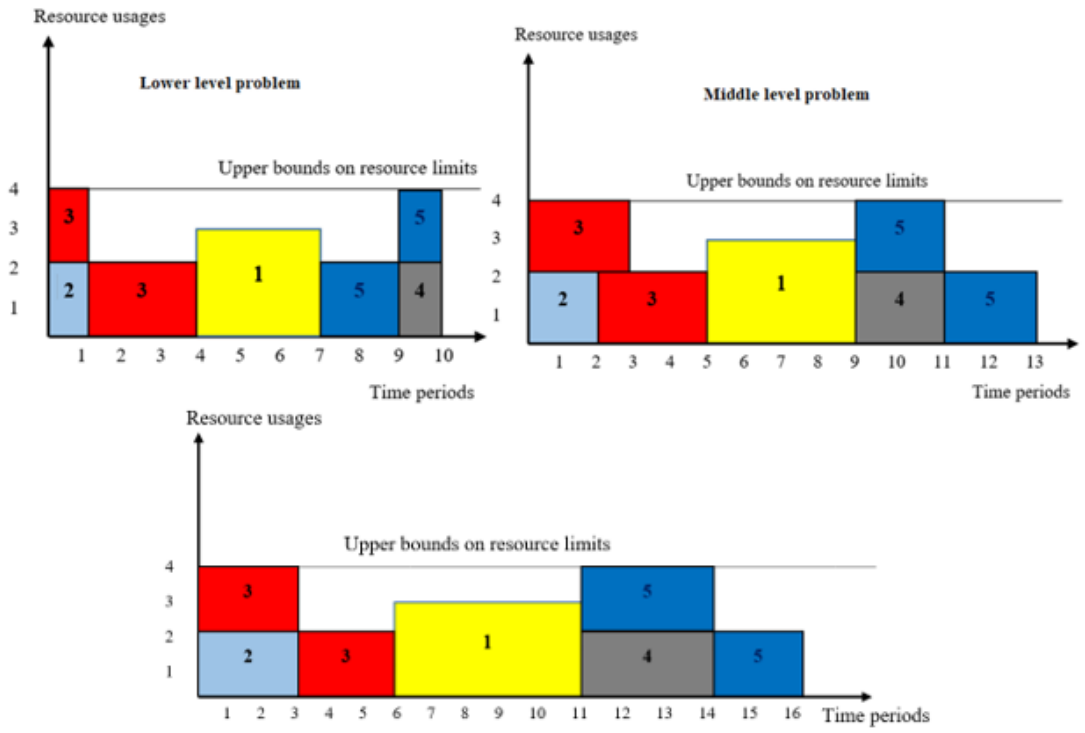


Figure 4.10 The renewable resource profiles under low capacity & high requirements (scenario#6)

4.4 Weighted Additive Fuzzy Goal Program (WAFGP) for the Multi-Objective FS-MRIPSP

To cope with the contradictory objectives instantaneously (i.e., minimizing the project makespan and total resource costs) and produce compromise fuzzy project schedules to the project manager(s), a weighted additive fuzzy goal programming method is applied in this thesis. This method integrates weighted membership functions of fuzzy goals whose relative importance degrees or weights may have different values. Indeed, a single weighted linear utility function is utilized by this method to express the overall preference of the decision maker(s) for aggregating the relative importance of these fuzzy goals. Actually, this utility function is computed by multiplying the membership function of each fuzzy goal with their importance weights. Then, the provided results are added together to provide maximal satisfaction of the fuzzy goals (Amid et al., 2009; Baykasoğlu & Subulan, 2016). In this thesis, we also modified this classical WAFGP method by associating it with the previously mentioned bound & decomposition principle in Sections 4.2.1 and 4.2.2. Because traditional WAFGP method is just able to generate crisp compromise solutions under uncertainty of the aspiration levels of the multiple objectives. However, we desire to obtain fuzzy efficient solutions (or fuzzy project schedules) which trade-off between the project makespan and total resource usage costs. To do this, fuzzy payoff matrix should be first constructed after solving the previously formulated ILP or CP models of the single-objective middle, lower and upper level problems independently. Firstly, ILP or CP model of the middle level problem should be solved for each goal separately. Subsequently, bound constraints should be added to the ILP or CP models of the lower and upper level problems for each objective according to the solution of middle level problem. Then, these lower and upper level problems should also be solved for each objective individually. Thereafter, a fuzzy payoff matrix as shown in Table 4.5 can be obtained from the single-objective optimization results of the middle, lower and upper level problems.

Table 4.5 The corresponding fuzzy payoff matrix

Objectives	Z1	Z2
Z1	$(Z1_l^*, Z1_m^*, Z1_u^*)$	$(Z2_l, Z2_m, Z2_u)$
Z2	$(Z1_l, Z1_m, Z1_u)$	$(Z2_l^*, Z2_m^*, Z2_u^*)$

Fuzzy values in the main diagonal of this matrix, i.e., $(Z1_l^*, Z1_m^*, Z1_u^*)$ and $(Z2_l^*, Z2_m^*, Z2_u^*)$, denote the individual best solutions of the middle, lower and upper level problems for each goal. Thus, these values can be used as benchmarks to determine the ideal/reference points. In addition, the remaining values within this matrix, i.e., $(Z1_l, Z1_m, Z1_u)$ and $(Z2_l, Z2_m, Z2_u)$ can be used as lower-upper bounds on the fuzzy goals. By using this fuzzy payoff matrix, aspiration level and max-min limits of each objective function can be determined under uncertainty. On the other hand, project managers can also decide these reference values, lower and upper limits of the goals based on their experiences and preferences. Fortunately, determination of these values by using such a fuzzy payoff matrix guarantees feasibility of each goal in the solution space. After generating this fuzzy payoff matrix, membership function of each fuzzy goal should be constructed by using these reference values and max-min limits. In detail, membership function values represent the satisfaction levels of the goals (Baykasoğlu & Subulan, 2016). For instance, membership functions of these fuzzy goals are constructed as in Equations 4.95-4.96 for the middle level problem and they are also demonstrated in Figure 4.11.

$$\mu_1^m \leq \frac{Z2_m - ZComp_{makespan}^m}{Z2_m - Z1_m^*} \tag{4.95}$$

$$\mu_2^m \leq \frac{Z1_m - ZComp_{cost}^m}{Z1_m - Z2_m^*} \tag{4.96}$$

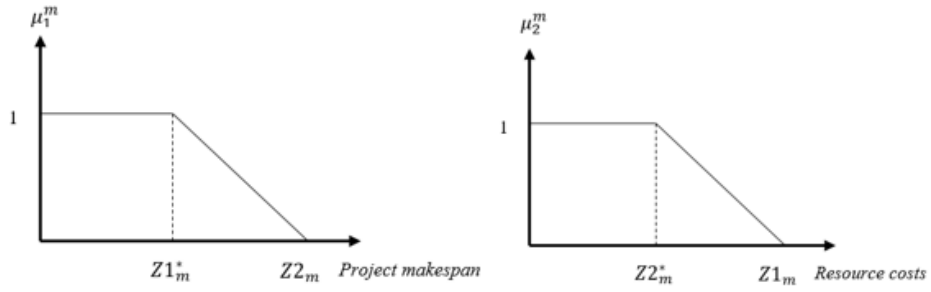


Figure 4.11 Membership functions of fuzzy goals in the middle level problem: a) Project makespan, b) Resource costs

In Equations 4.95-4.96, $ZComp^{m}_{makespan}$ and $ZComp^{m}_{cost}$ represent the core or most likely values of fuzzy goals in the compromise solution that will be generated by the *WAFGP* method. Afterward, the relative importance weights of these objectives, i.e., w_1 and w_2 should be specified according to their priorities in such a way that higher weights are assigned to the objectives with higher priority levels. Lastly, ILP and CP models of the crisp equivalent middle level problem can be reformulated as in Equations 4.97-4.98 for the multi-objective FS-MRIPSP.

$$Maximize \;\; w_1.\mu_1^m + w_2.\mu_2^m \tag{4.97}$$

Subject to;

Constraint sets related to the membership functions given in Equations 4.95-4.96.

Constraint sets in Equations 4.23-4.30 for the ILP model or Equations 4.64-4.70 for the CP model.

$$0 \leq \mu_1^m, \mu_2^m \leq 1 \tag{4.98}$$

In Equation 4.97, it is intended to maximize the summation of weighted satisfaction degrees of the fuzzy goals. The lower/upper bounds on these satisfaction levels or membership function values are also taken into consideration in Equation 4.98. After solving the above *WAFGP* model for the middle level problem, bounds on the fuzzy-stochastic decision variables (or project schedules and resources' availability levels) and the objective values can be easily provided to reformulate the ILP or CP models of the lower and upper level problems. By using *WAFGP* method, ILP and CP models of crisp equivalent upper level problem are modified in Equations 4.99-4.109.

$$Maximize \;\; w_1.\mu_1^u + w_2.\mu_2^u \tag{4.99}$$

Subject to;

$$\mu_1^u \leq \frac{Z2_u - ZComp^u_{makespan}}{Z2_u - Z1_u^*} \tag{4.100}$$

$$\mu_2^u \leq \frac{Z1_u - ZComp^u_{cost}}{Z1_u - Z2_u^*} \tag{4.101}$$

Constraint sets in Equations 4.31-4.32, 4.35-4.38 and 4.42-4.43 for ILP model. Constraint sets in Equations 4.71-4.72 and 4.76-4.79 for the CP model.

$$ZComp^{u}_{makespan} \geq ZComp^{m}_{makespan} \tag{4.102}$$

$$ZComp^{u}_{cost} \geq ZComp^{m}_{cost} \tag{4.103}$$

$$t_u.x_{jqt_us} \geq t(comp)^{*}_{m}.x(comp)_{jq^{*}t^{*}_{m}s} \quad \forall j \in J, \forall s \in S, q = q(comp)^{*}\ for\ the\ ILP\ model \tag{4.104}$$

$$alternative(X_{jsu}, \{Y_{jqsu} \mid\ q = q(comp)^{*}\}) \quad \forall j \in J, \forall s \in S\ for\ the\ CP\ model \tag{4.105}$$

$$endOf(Y_{jqsu} \mid\ q = q(comp)^{*}) \geq endOf\left(Y(comp)_{jq^{*}sm}\right) \quad \forall j \in J, \forall s \in S\ for\ the\ CP\ model \tag{4.106}$$

$$KR_{rsu} \geq KR(comp)^{*}_{rsm} \quad \forall r \in R, \forall s \in S\ for\ both\ \ ILP\ \&\ CP\ models \tag{4.107}$$

$$KN_{nsu} \geq KN(comp)^{*}_{nsm} \quad \forall n \in N, \forall s \in S\ for\ both\ \ ILP\ \&\ CP\ models \tag{4.108}$$

$$0 \leq \mu^{u}_{1}, \mu^{u}_{2} \leq 1 \tag{4.109}$$

Membership functions of fuzzy goals, i.e., μ^{u}_{1} and μ^{u}_{2} are first constructed in Equations 4.100-4.101 for the upper level problem. In the above formulation, it should be emphasized here that Equation 4.104 should be used instead of Equations 4.105-4.106 when the ILP model of the upper level problem is solved. In the opposite case, Equation 4.104 should be removed and Equations 4.105-4.106 should be actively used while solving the CP model. Equations 4.107-4.108 should be added to both of the ILP & CP models of the multi-objective upper level problem for the availability levels of the project resources. As mentioned previously, Equations 4.102-4.103 ensure that the right margins or pessimistic values of the fuzzy objectives in the produced compromise solution have to be greater than their core or most likely values provided by the solution of middle level problem. Equations 4.104-4.106 guarantee that the same execution modes chosen in the middle level problem should be selected again in the upper level problem. Moreover, the right margins (or pessimistic values) of the fuzzy completion times of the activities have to be greater than their core or most likely values in the compromise solutions. In a similar way, ILP and CP models of the crisp equivalent lower level problem can be revised as in Equations 4.110-4.121.

$$Maximize \quad w_1.\mu_1^l + w_2.\mu_2^l \tag{4.110}$$

Subject to;

$$\mu_1^l \leq \frac{Z2_l - ZComp_{makespan}^l}{Z2_l - Z1_l^*} \tag{4.111}$$

$$\mu_2^l \leq \frac{Z1_l - ZComp_{cost}^l}{Z1_l - Z2_l^*} \tag{4.112}$$

Constraint sets in Equations 4.44-4.45, 4.48-4.51 and 4.56-4.57 for the ILP model. Constraint sets in Equations 4.83-4.84 and 4.89-4.91 for the CP model.

$$ZComp_{makespan}^l \leq ZComp_{makespan}^m \tag{4.113}$$

$$ZComp_{cost}^l \leq ZComp_{cost}^m \tag{4.114}$$

$$t_l.x_{jqt_ls} \leq t(comp)_m^*.x(comp)_{jq^*t_m^*s} \quad \forall j \in J, \forall s \in S, q = q(comp)^* \; for \; the \; ILP \; model \tag{4.115}$$

$$\sum_{q \neq q(comp)^*}^{Q_j} \sum_{t_l = EFT_{jl}}^{LFT_{jl}} t_l.x_{jqt_ls} = 0 \quad \forall j \in J, \forall s \in S \; for \; the \; ILP \; model \tag{4.116}$$

$$alternative(X_{jsl}, \{Y_{jqsl} \mid \; q = q(comp)^*\}) \qquad \forall j \in J, \forall s \in S \; for \; the \; CP \; model \tag{4.117}$$

$$endOf(Y_{jqsl} \mid \; q = q(comp)^*) \leq endOf\big(Y(comp)_{jq^*sm}\big) \quad \forall j \in J, \forall s \in S \; for \; the \; CP \; model \tag{4.118}$$

$$KR_{rsl} \leq KR(comp)_{rsm}^* \quad \forall r \in R, \forall s \in S \; for \; both \; ILP \; \& \; CP \tag{4.119}$$

$$KN_{nsl} \leq KN(comp)_{nsm}^* \quad \forall n \in N, \forall s \in S \; for \; both \; ILP \; \& \; CP \; models \tag{4.120}$$

$$0 \leq \mu_1^l, \mu_2^l \leq 1 \tag{4.121}$$

In Equations 4.111-4.112, membership functions of the lower level problem, i.e., μ_1^l and μ_2^l are constructed based on the fuzzy payoff matrix given in Table 4.5. Then, the original constraints of the ILP or CP models are incorporated into the *WAFGP*. According to the Equations. 4.113-4.114, the left margins or optimistic values of fuzzy objectives in the produced compromise solutions have to be less than their core or most likely values. If the ILP model is used, Equations 4.115-4.116 should be included

instead of Equations 4.117-4.118. Equations 4.115-4.118 maintain that the same execution modes chosen in the middle level problem should also be selected again in the lower level problem and the left margins (or optimistic values) of the fuzzy completion times of the activities have to be less than their core or most likely values in the compromise solutions. After solving ILP or CP models of the middle, lower and upper level problems using the modified *WAFGP* method, several fuzzy compromise project schedules can be derived. Summary of the proposed ILP & CP-based transformation approaches with a bound & decomposition principle can be depicted by a flow chart given in Figure 4.12. As clearly seen in Figure 4.12, the proposed transformation approach can be realized on both ILP and CP models and is composed of two subsequent solution phases, namely single and multi-objective optimization.

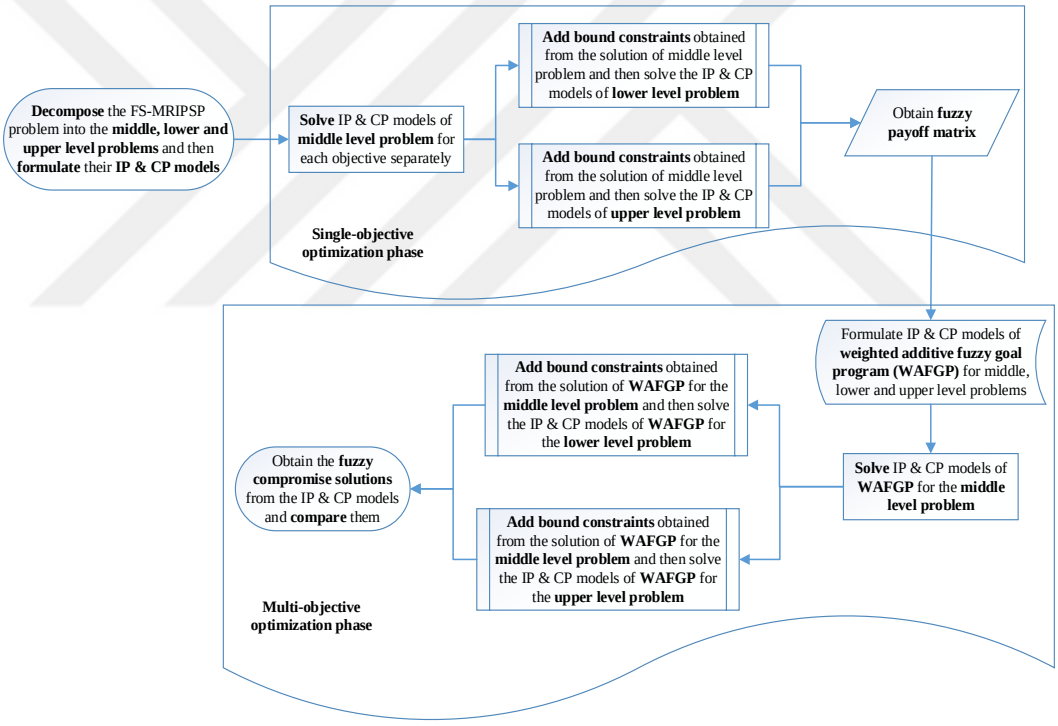


Figure 4.12 Flowchart of the proposed bound & decomposition-based solution methodology for multi-objective FS-MRIPSP

4.5 A Case Study of POM Module Group Project Implementation of an ERP Software

In order to verify the proposed CP-based transformation approach and display its validity and applicability, an application study is presented for a production-and-operations management (POM) module implementation process of an international ERP software company in Turkey. Actually, this case study is very similar with the previously mentioned application in Section 3.3 of Chapter 3. On the other hand, it should be noted here that although the previous application has only deterministic project data, this case study has many fuzzy and stochastic project parameters.

4.5.1 Data Description for Case Study of the FS-MRIPSP

As in the previous application in Chapter 3, this case study also concentrated on the POM module group project implementation of this ERP software product so as to elaborate main project activities related to the module integration processes. Recently, a comprehensive study was also performed by Subulan (2020) for integration process of all of these ERP modules under mixed interval & stochastic types of uncertainties. However, this present case study has just concentrated on the POM module group project implementation of the same ERP software product. To do this, totally 44-activities are first described as given in Table 4.6 for implementation process of each module in the POM group. Thereafter, precedence relations among these project activities and the integration processes of different modules are specified by the ERP consultants and the activity-on-node (AON) project network has been already drawn in Figure 3.4. The immediate predecessors of each project activity have already presented in Table 4.6. As shown in this table, we also defined multiple execution modes for each project activity since the activity durations may be differentiated with respect to the amounts of allocated project resources. By the way, it should be emphasized here that human resources are defined as the renewable project resources throughout this case study. Particularly, different types of human resources like ERP software company's consultants and process experts in the customer firm are specified as the renewable project resources. Because, professional human resources and knowledge/expertise of the consultants are the primary sources of an ERP software

company and play a critical role in effective ERP project implementation (Motwani et al., 2002; Tadinen, 2005; Plaza, 2016; Subulan, 2020). Moreover, experienced process experts in the customer firm are as vital as the success of ERP consultants. Because these experts support the ERP consultants for improving the firm's business processes and developing innovative ERP solutions. For that reason, ERP consultants should be in cooperation and collaboration with the customer firm's process experts. Therefore, effective, and efficient utilization of human resources may have significant impact on the success of ERP projects. On the other hand, these professionals have different abilities, tasks, and responsibilities during the project realization.

As clearly seen in Table 4.6, activity durations (in days) are represented by triangular fuzzy numbers since they are forecasted by the ERP consultants based on their knowledge and expertise. Indeed, optimistic, most likely and pessimistic values of the activity durations are determined by the project team consisting of ERP consultants and process experts.

Table 4.6 Fuzzy data for the production and operations management (POM) module group implementation project of the ERP system

Implementation Process of the POM Module Group (Activity Descriptions)	Immediate Predecessors	Fuzzy Duration (Day)			Fuzzy Earliest Finish Time (EFT)	Fuzzy Latest Finish Time (LFT)
		Mode-1	Mode-2	Mode-3		
1. BAS (Basic Core Data) Module						
1.1 Define material types, codes and records	-	(1,2,3)	(2,3,4)	(4,5,6)	(1,2,3)	(102,134,163)
1.2 Specify material supply categories	1.1	(0,1,1)	(1,2,2)	(2,3,4)	(1,3,4)	(104,137,167)
1.3 Fictionalize the variant structures	1.1	(0,1,2)	(1,2,3)	-	(1,3,5)	(104,137,167)
1.4 Determine relationships between materials and storage system	1.1	(2,3,4)	(2,4,5)	(3,5,6)	(3,5,7)	(104,137,167)
1.5 Transfer of the basic core data	1.2, 1.3, 1.4	(0,1,2)	(1,2,3)	(3,4,5)	(3,6,9)	(104,138,169)
2. BOM (Bill-of-Material) Module						
2.1. Analyze BOM structures and define alternative BOMs	1.5	(2,3,4)	(3,4,5)	(4,5,6)	(5,9,13)	(112,150,182)
2.2 Design and adapt support tables for BOM & analyze variant structures	1.5	(3,4,5)	(4,6,6)	(5,6,7)	(6,10,14)	(112,150,182)
2.3 Specify quantities and stock keeping units (SKUs)	2.1, 2.2	(1,2,2)	(2,3,4)	-	(7,12,16)	(113,152,184)
2.4 Transfer of the BOM data via electronic data interchange (EDI)	2.3	(1,2,3)	(2,3,5)	(3,4,5)	(8,14,19)	(114,154,187)
3. ROU - Production Routing Module						
3.1 Analyze manufacturing processes & evaluate production routing structures	1.5	(4,5,6)	(6,7,7)	(7,8,10)	(7,11,15)	(108,143,175)
3.2 Specify types of operations and routing alternatives	3.1	(1,2,2)	(1,2,3)	-	(8,13,17)	(109,145,177)
3.3 Design and adapt support tables for the ROU module	3.2	(4,5,6)	-	-	(12,18,23)	(113,150,183)
3.4 Generate variant dependent operation codes	3.2	(1,2,3)	(2,3,3)	(2,4,5)	(9,15,20)	(113,150,183)
3.5 Data transfer for the operations, work-centers, and cost-centers	3.3, 3.4	(1,3,3)	(2,3,4)	(3,4,4)	(13,21,26)	(114,153,186)
3.6 Provide material network diagrams (Operation process charts)	3.5	(0,1,1)	(1,2,2)	(1,2,3)	(13,22,27)	(114,154,187)
4. INV – Inventory Management Module						
4.1 Analyze inventory movements and prepare stock documents	1.5	(2,3,4)	(2,4,5)	(3,5,5)	(5,9,13)	(137,186,228)
4.2 Determine lot-sizing policies and analyze the barcode system	1.5	(3,4,5)	(3,5,6)	(5,6,7)	(6,10,14)	(137,186,228)
4.3 Design and adapt support tables for INV module	4.1, 4.2	(4,5,6)	(5,6,7)	(7,8,9)	(10,15,20)	(141,191,234)
4.4 Integrate sales, procurement, production, and inventory modules	4.3, 5.4	(6,8,9)	(7,8,10)	-	(26,41,51)	(147,199,243)
4.5 Specify costing methods and perform stock valuation for accounting	4.3	(1,2,3)	-	-	11,17,23)	(147,199,243)
4.6 Integrate activity-based costing, accounting, and inventory modules	4.4, 4.5	(3,4,5)	(4,5,6)	-	(29,45,56)	(150,203,248)
5. PRD - Production Planning & Control Module						
5.1 Analyze production process, collect manufacturing and process data	2.4, 3.6	(3,5,6)	(4,6,7)	(5,7,8)	(16,27,33)	(117,159,193)
5.2 Design and adapt support tables for production planning module	5.1	(4,5,7)	-	-	(20,32,40)	(121,164,200)
5.3 Fictionalize the production planning & work order types and structures	5.2	(0,1,1)	(1,2,2)	(1,2,3)	(20,33,41)	(122,166,202)
5.4 Analyze the relationships between work orders & inventory movements	5.2	(0,1,2)	(1,2,3)	(2,3,4)	(20,33,42)	(122,166,202)
5.5 Define shift schedules and associate them with the company's work calendar	5.2	(0,1,1)	(0,1,2)	(1,2,2)	(20,33,41)	(122,166,202)
5.6 Analyze contract manufacturing operations and residual products	5.2	(1,2,2)	(1,2,3)	(3,4,4)	(21,34,42)	(122,166,202)

Table 4.6 Continues

Implementation Process of the POM Module Group (Activity Descriptions)	Immediate Predecessors	Fuzzy Duration (Day)			Fuzzy Earliest Finish Time (EFT)	Fuzzy Latest Finish Time (LFT)
		Mode-1	Mode-2	Mode-3		
5. PRD - Production Planning & Control Module						
5.7 Design output reports and documents for production planning & control	5.3, 5.4, 5.5, 5.6	(3,4,5)	(5,6,7)	-	(24,38,47)	(125,170,207)
6. QLT – Process Quality Control Module						
6.1 Analyze the operational quality control process of the company	5.7	(1,2,2)	(2,3,4)	-	(25,40,49)	(144,194,237)
6.2 Design and adapt support tables for process quality control module	6.1	(3,4,5)	(3,5,6)	(4,6,7)	(28,44,54)	(147,198,242)
6.3 Define quality plans: quality characteristics, sampling methods, center lines, specification limits, defect classes with their priorities	6.2	(2,3,4)	(3,4,5)	(4,5,6)	(30,47,58)	(149,201,246)
6.4 Data collection from the control/test operations in manufacturing	6.2	(1,2,3)	(2,3,4)	(3,4,5)	(29,46,57)	(149,201,246)
6.5 Design output reports: quality control charts & process capability analysis	6.3, 6.4	(1,2,2)	(2,3,4)	-	(31,49,60)	(150,203,248)
7. MRP – Materials Requirement Planning Module						
7.1 Determine basic MRP parameters: Timing, procedures, order quantity/production lot-sizing methods for production and procurement etc.	5.7	(2,3,3)	(2,3,4)	(3,4,5)	(26,41,50)	(127,173,210)
7.2 Specify MRP scenario types and structures	5.7	(0,1,1)	(1,2,3)	-	(24,39,48)	(127,173,210)
7.3 Design and adapt support tables for MRP module	7.1, 7.2	(3,4,5)	(4,5,6)	(5,6,7)	(29,45,55)	(130,177,215)
7.4 Implementation tests for validation and verification of the MRP outputs	7.3	(2,3,3)	(3,4,4)	(3,4,5)	(31,48,58)	(132,180,218)
7.5 Design output reports which can identify relationships between customers' sales orders & alternative MRP solutions	7.4	(2,3,5)	(3,4,6)	(3,5,6)	(33,51,63)	(134,183,223)
8. CAP – Capacity planning & Production Scheduling Module						
8.1 Analyze the long-short term production capacities & production lot-sizing	7.5	(4,5,7)	(5,6,8)	(6,8,9)	(37,56,70)	(138,188,230)
8.2 Design and adapt support tables for CAP module						
8.3 Define capacity groups & dispatching rules for scheduling	8.1	(3,4,5)	-	-	(40,60,75)	(141,192,235)
8.4 Specify capacity types & resource (labor, machinery equipment) planning	8.1	(0,1,1)	(1,2,3)	(2,3,4)	(37,57,71)	(141,192,235)
8.5 Develop heuristic algorithms via an integrated development environment embedded in the ERP software for scheduling work orders	8.1	(1,2,3)	(2,3,4)	(3,4,5)	(38,58,73)	(141,192,235)
	8.2, 8.3,8.4	(7,8,9)	(9,10,12)	(10,12,15)		
8.6 Design output reports and documents: resource utilization charts for labor and machinery equipment and Gantt charts for the work order operations	8.5	(2,3,4)	(3,5,6)	-	(47,68,84) (49,71,88)	(148,200,244) (150,203,248)

Unfortunately, it is a challenging task to identify deterministic values for these durations due to the highly changing and dynamic project environments. Because, activity durations may vary with respect to the industrial scale, sector and size of the customer firm. Moreover, customer firm's information system infrastructure and its product's development level may also have significant effects on these durations. Furthermore, dynamic situations such as the availability/reliability of the human resources, unanticipated or extra expectations of the customer firm restrict the project managers to define deterministic activity durations. Briefly, fuzzy activity durations under different execution modes are defined as in Table 4.6 to cope with these uncertain conditions. Thus, the earliest and latest finish times of the project activities are computed as fuzzy numbers and given in Table 4.6. Actually, their values are determined by using the forward and backward pass calculations.

Before applying the proposed ILP and CP-based transformation approaches, these fuzzy earliest and latest finishing times should be computed for each project activity. In a nutshell, all the fuzzy data are presented in Table 4.6 for the POM module group implementation project of the ERP system. On contrary to the fuzzy activity durations, human resource requirements which are described in terms of the labor force (in man-days) and the implementation costs ($) of the project activities are defined as stochastic project parameters with pre-specified discrete probabilities. Because, ERP project managers announced that these parameters may have randomness in nature. According to the customer firm's unforeseen expectations or its additional business process requirements, human resources demanded by the project activities may change in random fashion. Moreover, efficiency and reliability factors of the human resources may influence the standard durations and resource requirements of the project activities. Actually, effective working hours and human resource utilizations may be affected by these factors. As highlighted by Subulan (2020), inexperienced ERP consultants may cause inefficient operations/processes and therefore, the longer activity durations and the higher human resource requirements may arise. Additionally, the day offs, medical leaves and shirking may result in ineffective and unreliable human resources. Moreover, since the ERP consultants are responsible for various projects of different customer firms, their effective and efficient working hours

in a specific firm may reduce because of the over workload. This may result reductions in both efficiency and reliability of the human resources. Thus, utilizations of the human resources may drop and the project makespan gets longer. For all of these reasons, human resources requirements of the project activities are described as discrete random variables and given in Table 4.7. To do this, different scenario probabilities are defined by the ERP project team for the pessimistic, most likely, and optimistic values of these human resource requirements. As obviously seen in Table 4.7, the higher amounts of human resources are demanded in case of shorter activity durations. However, the less amount of human resources are needed under some modes whose activity durations get longer. It should also be emphasized here that probabilistic human resource requirements are described for the process experts by considering the standardized ERP solutions and implementation-oriented innovative ERP solutions. Because, standardized ERP solutions may not be suitable for some of the customer firms' business processes and therefore, additional implementations and adaptations may be required for more customized ERP solutions.

As clearly seen in Tables 4.6-4.7 that the same number of execution modes are not valid for all the project activities. In other words, project activities may have different numbers of execution modes since some activity durations cannot be decreased even if the more human resources are allocated to them. Moreover, human resource requirements for the ERP consultants are relatively larger than the process experts of the customer firm. It should be emphasized here that availability levels of different types of human resources are not taken as constant deterministic parameters (or input data) since the ERP company also desires to determine their ideal values. In other words, while producing the ERP project schedules, it is also targeted to decide on the availability levels of the human resources. Unfortunately, deterministic values cannot be determined for these resource availability levels since all of the project data are considered as uncertain. In other words, it is a challenging task to specify certain values for these availability levels in case of the stochastic human resource requirements of the project activities. Thus, it is intended to determine their pessimistic, most likely and optimistic values under different scenarios.

Table 4.7 Stochastic human resource requirements of the project activities under different execution modes

Project Activities	ERP Supplier's Consultants (in man-days)									Process Experts in Customer (in man-days)					
	Optimistic (Scenario probability 30%)			Most Likely (Scenario probability 40%)			Pessimistic (Scenario probability 30%)			Standardized ERP solutions (Scenario probability 40%)			(Implementation oriented) (Scenario probability 60%)		
	Mode-1	Mode-2	Mode-3	Mode-1	Mode-2	Mode-3	Mode-1	Mode-2	Mode-3	Mode-1	Mode-2	Mode-3	Mode-1	Mode-2	Mode-3
1.1	12	8	4	15	9	6	20	15	12	3	2	1	5	3	2
1.2	8	5	3	10	7	5	15	12	10	2	1	0	3	2	1
1.3	4	3	-	6	4	-	10	8	-	1	0	-	2	1	-
1.4	15	12	9	18	15	10	22	18	13	4	2	1	6	5	3
1.5	18	14	10	20	16	12	25	20	15	5	3	2	7	5	4
2.1	16	15	12	20	18	15	24	20	18	6	4	2	8	6	4
2.2	20	18	15	25	22	18	30	26	22	1	0	0	2	1	0
2.3	9	7	-	12	10	-	16	13	-	4	3	-	4	3	-
2.4	6	4	2	8	5	3	10	7	5	1	0	0	2	1	0
3.1	25	18	15	30	21	18	36	27	24	10	7	6	28	25	20
3.2	10	7	-	12	8	-	18	12	-	8	5	-	10	7	-
3.3	20	-	-	24	-	-	30	-	-	5	-	-	8	-	-
3.4	13	10	8	15	12	9	20	16	13	7	4	3	10	7	5
3.5	10	8	5	12	9	6	15	12	10	6	3	2	8	6	4
3.6	2	1	0	3	2	1	5	4	2	1	0	0	1	0	0
4.1	20	18	15	25	22	18	32	26	23	6	5	4	12	10	8
4.2	17	15	12	20	18	16	25	22	20	7	6	4	15	10	8
4.3	24	20	17	30	25	20	34	29	25	4	3	2	8	6	5
4.4	33	30	-	40	35	-	45	40	-	5	2	-	10	7	-
4.5	12	-	-	15	-	-	20	-	-	4	-	-	10	-	-
4.6	16	12	-	20	16	-	25	20	-	4	3	-	7	5	-
5.1	32	28	25	40	35	30	45	40	35	10	8	6	30	25	20
5.2	18	-	-	21	-	-	26	-	-	5	-	-	8	-	-
5.3	6	4	3	8	6	4	10	8	6	4	3	2	7	5	3
5.4	5	3	2	6	4	3	8	6	4	3	2	1	5	4	2

Table 4.7 Continues

Project Activities	ERP Supplier's Consultants (in man-days)									Process Experts in Customer (in man-days)					
	Optimistic (Scenario probability 30%)			Most Likely (Scenario probability 40%)			Pessimistic (Scenario probability 30%)			Standardized ERP solutions (Scenario probability 40%)			(Implementation oriented) (Scenario probability 60%)		
	Mode-1	Mode-2	Mode-3	Mode-1	Mode-2	Mode-3	Mode-1	Mode-2	Mode-3	Mode-1	Mode-2	Mode-3	Mode-1	Mode-2	Mode-3
5.5	2	1	0	3	2	1	5	4	2	1	0	0	2	1	0
5.6	4	3	2	6	4	3	8	6	4	3	2	1	4	3	2
5.7	10	7	-	14	10	-	18	13	-	4	2	-	5	3	-
6.1	10	6	-	12	8	-	15	10	-	10	8	-	12	9	-
6.2	22	19	16	28	24	20	31	26	23	6	4	3	8	6	4
6.3	20	17	12	24	20	16	27	23	20	13	12	10	18	16	13
6.4	16	13	10	20	16	12	22	18	16	20	17	14	25	20	16
6.5	13	8	-	16	10	-	19	14	-	4	3	-	6	4	-
7.1	20	17	12	25	20	15	30	25	20	8	6	4	10	8	6
7.2	7	5	-	9	6	-	13	10	-	4	2	-	5	3	-
7.3	16	14	11	21	18	14	26	21	18	5	3	2	7	5	3
7.4	20	17	14	24	21	18	28	25	20	9	7	5	12	10	8
7.5	15	12	9	18	15	12	20	18	16	4	3	2	6	4	3
8.1	25	22	19	30	28	25	36	30	28	15	12	8	20	16	11
8.2	34	-	-	40	-	-	45	-	-	5	-	-	8	-	-
8.3	10	8	5	12	10	7	15	12	8	2	1	1	4	3	2
8.4	11	9	7	15	12	10	20	16	12	7	5	3	10	7	5
8.5	40	36	32	50	45	40	60	50	45	0	0	0	0	0	0
8.6	27	24	-	32	28	-	36	32	-	5	3	-	7	5	-

In other words, availability levels of the human resources are also specified as fuzzy-stochastic decision variables (or project outputs). Therefore, the examined project scheduling problem will become a multi-objective and multi-mode resource investment project scheduling (MRIPSP) where the total usage costs of the human resources is aimed to minimize the project makespan. Actually, effective usage of the human resources in ERP project implementations is as important as minimum project makespan for both of the software company and customer firms. Thus, it is a crucial issue to handle these contradictory goals all together.

In addition to these renewable human resources, project fund or budget of the customer firm is taken into consideration as the non-renewable project resource since its availability is limited throughout the project planning horizon. Similarly, its availability level is also defined as a decision variable since total implementation cost of different POM modules is also targeted to minimize in addition to the total human resource costs. Moreover, based on the former financial indicators and accounting reports of the customer firm, availability level of the project fund cannot determine precisely because of the dynamic incoming/outgoing cash flows along with the project planning horizon. For that reason, it is also defined as a fuzzy-stochastic decision variable and expected to take fuzzy monetary values under various scenarios.

Finally, stochastic implementation costs of different POM module integration operations (or project's activities) under different execution modes are given in Table 4.8. In a similar manner, ERP consultants evaluated these implementation cost values under different scenarios with discrete occurrence probabilities. All the uncertain project data presented in Tables 4.6-4.8 are determined by the ERP consultants and process experts based on their past experiences in the similar customer firms and industries. As mentioned previously, these ERP project data may change completely from one customer firm to another with respect to its industrial scale and information technology infrastructure. Based on these random data related to the human resource requirements and implementation costs of the project activities, 18-different scenarios with unequal joint probabilities are derived as shown in Figure 4.13. In the next section, details of the single

and multi-objective optimization results of the proposed ILP & CP-based transformation approaches are also presented and compared to each other for this case study.

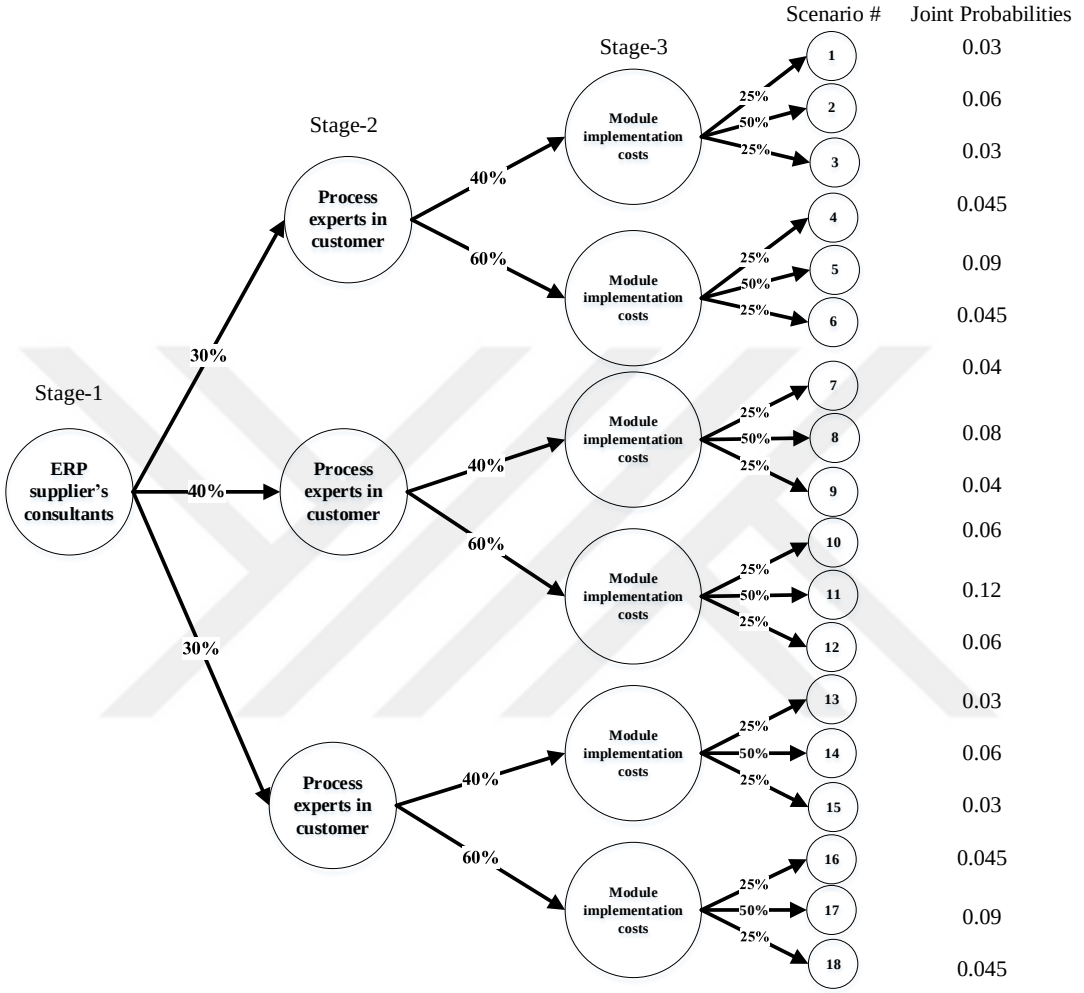


Figure 4.13 Scenario tree generation for the stochastic renewable/non-renewable resource requirements

Table 4.8 Stochastic implementation costs ($) with discrete probabilities (%) under different execution modes

Activities	Optimistic Scenario (Probability 25%)			Most Likely Scenario (Probability 50%)			Pessimistic Scenario (Probability 25%)		
	Mode-1	Mode-2	Mode-3	Mode-1	Mode-2	Mode-3	Mode-1	Mode-2	Mode-3
1.1	160	120	95	200	150	120	230	175	140
1.2	65	50	40	80	60	50	92	70	60
1.3	55	40	-	70	50	-	81	60	-
1.4	80	65	55	100	80	70	115	95	80
1.5	160	130	95	200	160	120	230	185	140
2.1	200	160	145	250	200	180	290	230	210
2.2	400	330	300	500	420	380	575	485	430
2.3	240	220	-	300	280	-	345	320	-
2.4	280	240	210	350	300	270	405	345	310
3.1	145	120	95	180	150	120	210	175	140
3.2	65	60	-	80	75	-	95	85	-
3.3	260	-	-	320	-	-	370	-	-
3.4	90	75	65	110	95	80	125	110	95
3.5	275	245	240	340	310	300	390	360	345
3.6	65	55	50	80	70	65	90	80	75
4.1	200	175	160	250	220	200	290	255	230
4.2	145	135	120	180	165	150	210	190	175
4.3	175	160	145	220	200	185	250	230	210
4.4	320	300	-	400	380	-	460	430	-
4.5	300	-	-	380	-	-	440	-	-
4.6	330	310	-	420	390	-	490	450	-
5.1	400	360	320	500	450	400	575	520	460
5.2	330	-	-	420	-	-	485	-	-
5.3	200	175	150	250	220	190	290	250	220
5.4	145	120	95	180	150	120	210	170	140
5.5	65	48	40	80	60	50	95	70	60
5.6	55	40	35	70	50	40	80	60	45
5.7	360	320	-	450	400	-	520	460	-
6.1	145	120	-	180	150	-	215	175	-
6.2	255	240	225	320	300	280	385	345	320
6.3	240	225	200	300	280	250	360	320	280
6.4	185	160	145	230	200	180	270	230	205
6.5	240	225	-	300	280	-	360	320	-
7.1	280	240	215	350	300	270	420	345	310
7.2	80	65	-	100	80	-	120	90	-
7.3	400	360	320	500	450	400	600	520	460
7.4	200	175	160	250	220	200	300	255	230
7.5	375	345	325	470	430	410	565	495	470
8.1	480	400	360	600	500	450	700	575	520
8.2	420	-	-	520	-	-	620	-	-
8.3	120	95	80	150	120	100	180	140	115
8.4	80	65	55	100	80	70	120	95	80
8.5	2000	1600	1440	2500	2000	1800	3000	2300	2100
8.6	480	400	-	600	500	-	700	575	-

4.5.2 Fuzzy-Stochastic Multi-Objective Optimization Results of the Proposed ILP and CP-Based Transformation Approaches

In the single objective optimization phase, the middle, lower and upper level problems which were previously formulated in Sections 4.2.1 and 4.2.2 are first solved for each objective function independently by making use of an optimization suite, i.e., ILOG CPLEX optimization studio v12.10 on an Intel Corei7 2 GHz IBM PC. In other words, single-objective optimization results of these sub-problems are provided by both ILP & CP-based transformation approaches. Then, the generated results are aggregated to construct the following fuzzy payoff matrices in Tables 4.9-4.10. The details of the single-objective optimization process are also presented for each transformation approach in Tables 4.11-4.12. As clearly seen in Tables 4.9-4.10, the same optimal project makespan value, i.e., 71 is achieved by both ILP and CP-based transformation approaches for the middle level problem. However, since the produced solutions represent the alternative optimal solutions of this middle level problem, different objective values, i.e., 89.76 versus 92 and 49.815 versus 50 are acquired for the upper and lower level problems, respectively. Moreover, it should be noted here that optimization results of the project makespan objective are produced within reasonable computation times by both ILP and CP-based transformation approaches (see Tables 4.11-4.12).

Table 4.9 Fuzzy payoff matrix provided by the ILP-based transformation approach

Objectives	Project makespan $(\tilde{Z}_1)$	Total resource costs $(\tilde{Z}_2)$
$\tilde{Z}_1$	(49.815, 71, 89.76)	(20238, 22011, 24157)
$\tilde{Z}_2$	(150, 203, 248)	(13370.87, 13943.8, 14516.8)*

Table 4.10 Fuzzy payoff matrix provided by the CP-based transformation approach

Objectives	Project makespan $(\tilde{Z}_1)$	Total resource costs $(\tilde{Z}_2)$
$\tilde{Z}_1$	(50, 71, 92)	(20164, 21937, 23710)
$\tilde{Z}_2$	(105.25, 134.66, 171.53)	(13523.75, 14120.75, 14700)*

Actually, optimal value of the fuzzy expected project makespan can be achieved within relatively shorter computation times by the CP-based approach when compared to the ILP-based one. Although optimal project makespan value can be easily achieved in very shorter computation times, it is more difficult to obtain appropriate solutions

for the total resource cost objective. Fortunately, optimal solution of the middle level problem with total resource cost objective is accomplished by the ILP-based approach under 2-hours runtime limit. However, solution of this middle level problem with total resource cost objective is terminated after 2-hours later for the CP-based approach. In other words, CP model of the middle level problem are solved under 2-hours runtime limit and just a feasible solution with 1.26% optimality gap can be obtained. Based on these facts, it can be concluded that inclusion of the total resource cost objective which mainly uses the availability levels of the renewable and non-renewable project resources makes the solution of the stated problem much more difficult. Fortunately, it is clearly seen in Table 4.12 that optimal solutions for the lower and upper level problems can be easily achieved by the CP-based approach in shorter runtimes after adding the bound constraints which are obtained from the solution of middle level problem. In other words, CP-based transformation approach outperforms the ILP-based approach in terms of the computational time for both project makespan and total resource cost objectives. According to Tables 4.11-4.12, although the ILP-based transformation approach contains relatively large number of constraints and integer-binary variables, it leads to a better solution with a minimum fuzzy expected project makespan value of (49.815, 71, 89.76).

Table 4.11 Details of single-objective optimization results obtained from the ILP-based approach

	ILP-based transformation					
	Project makespan			Total resource cost		
	Lower	Middle	Upper	Lower	Middle	Upper
Model class	Pure integer linear programming (PILP)					
# of constraints	9110	9307	11810	9110	9307	11810
# of binary variables	361206	499068	609012	361206	499123	609012
# of integer variables	54	54	54	54	54	54
#of non-zero elements	10484691	18155767	26759967	10484691	18162895	26759967
# of solver iterations	0	674	2662	497996	3277467	83530
CPU time (sec.)	15.22	29.05	36.3	264.7	5835.6	100.8
Optimality gap (%)	0	0	0	0	0	0

Similarly, ILP-based transformation approach also outperforms the CP-based approach for the total resource cost objective in terms of solution quality. Indeed, it provides a superior solution with a minimum fuzzy expected total resource cost value of (13370.87, 13943.8, 14516.8). Briefly, ILP-based transformation approach leads to

better quality solutions for both objective functions in the single-objective optimization process. For all of these reasons, we utilized the single objective optimization results (or fuzzy payoff matrix in Table 4.9) of the ILP-based transformation approach to construct the membership functions in multi-objective optimization phase of the proposed solution methodology (See Figure 4.12). Again, it should also be emphasized here that CP-based transformation approach is superior in terms of computing time in the single-objective optimization phase. Moreover, the number of constraints and binary-integer decision variables can be significantly decreased by utilizing such a CP-based transformation approach (see Table 4.11-4.12).

Table 4.12 Details of single-objective optimization results obtained from the CP-based approach

	CP-based transformation					
	Project makespan			Total resource cost		
	Lower	Middle	Upper	Lower	Middle	Upper
# of constraints	18766	2003	18766	19072	2022	19072
# of variables	2970	2970	2970	2970	2970	2970
Memory usage	67907371	91444846	68907828	89211885	170872757	104904112
# of choice points	102	102	102	2960005	190135655	3431674
# of solutions	1	1	1	7	406	409
# of branches	20106	3474	20214	3716931	271461618	4747263
# of fails	72	72	72	750232	81527165	1042838
# of intervals	2916	2916	2916	2916	2916	2916
CPU time (sec.)	0.78	0.57	1.2	15.5	7200*	20.12
Optimality gap (%)	0	0	0	0	1.26	0

In the multi-objective optimization phase, membership functions for the aspiration levels of the fuzzy objectives are first constructed by using the max-min limits given in Table 4.9. Then, ILP and CP models of the middle, lower and upper level problems are modified by using WAFGP method so as to obtain the compromise solutions which trade-off between project makespan and total resource cost objectives. Similar with the single-objective phase, ILP and CP models including WAFGP method are first solved for the middle level problem with different importance weights of the objectives. ILOG OPL codes/scripts of these ILP and CP models of the middle level problem are also given in Appendices-5, 6 and 7 for both single and multi-objective phases. Unfortunately, ILP-based transformation approach cannot produce any feasible solution for some objective weights ($w_1 = 0.2$ & $w_2 = 0.8$; $w_1 = 0.4$ & $w_2 = 0.6$ and $w_1 = 0.6$ & $w_2 = 0.4$) within 2-hours runtime limit. It should be highlighted here that any feasible solution cannot be generated for these objective weights even

for 12-hours runtime limit. Actually, ILP-based transformation approach can find a feasible solution when the importance weight of the project makespan objective is increased up to 80% and the weight of the total resource cost objective decreased to 20% only ($w_1 = 0.8$ & $w_2 = 0.2$). Fortunately, several feasible solutions can be achieved by the CP-based approach for different objective weights within just 10-minutes runtime limit without applying any search strategy (see Table 4.13). In order to improve performance of the proposed CP-based approach, we also applied some search strategies in the literature for solution of the middle level problem (Young et al., 2017; Baykasoğlu et al., 2020; IBM, 2017). IBM ILOG CPLEX codes of these search strategies are also presented in Appendix-8. As shown in Table 4.13, significant improvements on the objective function values, i.e., Z1 and Z2 can be achieved by making use of these search strategies. On the other hand, it should be noted here that different search strategies may produce promising results for different objective weights. For instance, search strategies-2, 10, 11 and 12 generate better results for the objective weights of $w_1 = 0.2$ & $w_2 = 0.8$ and $w_1 = 0.6$ & $w_2 = 0.4$ whereas the best solutions are obtained by the search strategies-1, 14 and 15 for the objective weight of $w_1 = 0.4$ & $w_2 = 0.6$. Additionally, the best objective function values can be accomplished by the search strategy-12 for the objective weight of $w_1 = 0.8$ & $w_2 = 0.2$. While selecting the best search strategies to apply, we took into account the objective function with the highest weight. For example, only the values of project makespan, i.e., Z1 were considered when choosing the best search strategies for its highest objective weights ($w_1 = 0.6$ & $w_1 = 0.8$). After determining the best search strategies for different objective weights, the middle level problem was resolved with these weights under 2-hours runtime limit. Then, the produced solutions are used to formulate bound constraints in the lower and upper level problems. After solving the CP models including the WAFGP method of these lower and upper level problems, feasible fuzzy-stochastic solutions which trade-off between makespan and total resource cost objectives are obtained as given in Table 4.14 under 2-hours runtime limit.

Table 4.13 Comparative study for testing different search strategies on the middle level problem of the CP-based transformation approach in the multi-objective optimization phase

	No strategy		Search Strategy-1		Search Strategy-2		Search Strategy-3		Search Strategy-4		Search Strategy-5		Search Strategy-6		Search Strategy-7		Search Strategy-8	
Weights	Z1	Z2	Z1	Z2	Z1	Z2	Z1	Z2	Z1	Z2	Z1	Z2	Z1	Z2	Z1	Z2	Z1	Z2
20-80%	104.08	16935	99.96	16703	99.41	16914	99.96	16703	99.41	16914	99.96	16703	99.41	16914	99.96	16703	99.41	16914
40-60%	95.42	17184	92.92	17181	96.43	17061	92.92	17181	95.77	17090	92.92	17181	96.43	17061	92.92	17181	96.43	17061
60-40%	88.82	17293	79.92	18722	85.81	17517	79.92	18722	85.81	17517	79.92	18722	85.89	17523	79.92	18722	85.89	17524
80-20%	71.55	19472	71,13	19748	74.79	18721	71.13	19748	74.79	18721	71.13	19748	74.79	18721	71.13	19745	74.79	18721

In addition to fuzzy objective function values of these compromise solutions, the individual membership function values and the weighted additive membership values (total satisfaction degrees) for each sub-problem, i.e., middle, lower and upper are also involved in Table 4.14. Furthermore, optimization details such as the number of variables and constraints, memory usages, solver iterations etc. for both ILP and CP-based transformation approaches are included at the end of this table.

When results of the ILP and CP-based approaches are compared, it is recognized that ILP-based approach cannot generate any feasible solution for most of the objective weights under 2-hours runtime limit. As mentioned before, any feasible solution cannot also be produced by this ILP-based approach even for 12-hours runtime limit. Actually, this ILP-based approach can just find a feasible solution for the objective weights of $w_1 = 0.8$ & $w_2 = 0.2$. However, it is clearly seen in Table 4.14 that CP-based approach has found a superior result, i.e., (50.19, 71.14, 92.02) for the fuzzy expected project makespan objective when compared to the result of ILP-based approach (52.55, 72.73, 94.71). When the generated compromise solutions for these objective weights $(w_1 = 0.8 \ \& \ w_2 = 0.2)$ are compared to each other, it can be concluded that CP-based approach is able to provide better results for the project makespan objective within 2-hours runtime limit. In a nutshell, when the project managers declared that the total resource cost objective is relatively much more important than the project makespan objective, any compromise solution cannot be found by the ILP-based approach in the multi-objective optimization phase of the proposed methodology. In order to overcome this problematic issue, CP-based transformation approaches can be very useful to find several compromise solutions easily within reasonable computation times. Moreover, it is clearly seen in Table 4.14 that the model size including the numbers of decision variables and constraints can be significantly reduced by making use of such a CP-based transformation approach. This computational case study has shown that CP-based transformation approaches can be very useful for solving realistic size fuzzy-stochastic combinatorial optimization problems within reasonable computation times.

Table 4.14 Comparison of the multi-objective optimization results of the proposed ILP and CP-based approaches with WAFGP

			Different importance weights of the objectives							
			$(w_1 = 0.2 \ \& \ w_2 = 0.8)$		$(w_1 = 0.4 \ \& \ w_2 = 0.6)$		$(w_1 = 0.6 \ \& \ w_2 = 0.4)$		$(w_1 = 0.8 \ \& \ w_2 = 0.2)$	
			ILP-based approach	CP-based approach	ILP-based approach	CP-based approach	ILP-based approach	CP-based approach	ILP-based approach	CP-based approach
Expected project makespan $(\tilde{Z}_1)$			N.A	(79.31, 96.68, 118.1)	N.A	(71.72, 96.86, 119.5)	N.A	(56.53, 78.82, 101.83)	(52.55, 72.73, 94.71)	(50.19, 71.14, 92.02)
Total expected resource costs $(\tilde{Z}_2)$			N.A	(14382, 16726, 17813)	N.A	(14694, 16841, 17947)	N.A	(15781, 18333, 19579)	(15693, 16626, 17516)	(17061, 19403, 20670)
Membership function values	$(\tilde{Z}_1)$	μ_1^l	N.A	0.70564	N.A	0.7813	N.A	0.93292	0.9727	0.99621
		μ_1^m	N.A	0.80542	N.A	0.80405	N.A	0.94076	0.98689	0.99898
		μ_1^u	N.A	0.8209	N.A	0.81196	N.A	0.92372	0.96869	0.98572
	$(\tilde{Z}_2)$	μ_2^l	N.A	0.85275	N.A	0.80726	N.A	0.6491	0.66184	0.46268
		μ_2^m	N.A	0.65511	N.A	0.64087	N.A	0.45597	0.66753	0.32332
		μ_2^u	N.A	0.65806	N.A	0.64415	N.A	0.47489	0.68886	0.3617
Objective values	$w_1.\mu_1^l + w_2.\mu_2^l$		N.A	0.823329	N.A	0.796877	N.A	0.819393	0.861869	0.889501
	$w_1.\mu_1^m + w_2.\mu_2^m$		N.A	0.68517	N.A	0.706145	N.A	0.746843	0.915937	0.863846
	$w_1.\mu_1^u + w_2.\mu_2^u$		N.A	0.690626	N.A	0.711274	N.A	0.744189	0.882161	0.860914
Optimization details - Integer programming (ILP) based transformation approach										
# of constraints			9314		9314		9314		9314	
# of binary variables			499068		499068		499068		499068	
# of integer variables			499126		499126		499126		499126	
# of non-zero elements			18162958		18162958		18162958		18162958	
# of solver iterations			3694325		2491670		2516118		11559640	
CPU time (sec.)			7200		7200		7200		7200	

Table 4.14 Continues

	Different importance weights of the objectives			
	$(w_1 = 0.2 \ \& \ w_2 = 0.8)$	$(w_1 = 0.4 \ \& \ w_2 = 0.6)$	$(w_1 = 0.6 \ \& \ w_2 = 0.4)$	$(w_1 = 0.8 \ \& \ w_2 = 0.2)$
	CP-based approach	CP-based approach	CP-based approach	CP-based approach
Optimization details - Constraint programming (CP) based transformation approach				
# of constraints	2010	2010	2010	2010
# of variables	2970	2970	2970	2970
Memory usage	225480647	365670844	389611783	377936601
# of choice points	55004808	33985640	50917696	53226080
# of solutions	550	497	542	275
# of branches	79083997	49650532	72967423	74927284
# of fails	24057622	15605265	21999704	21637045
# of intervals	2916	2916	2916	2916
CPU time (sec.)	7200	7200	7200	7200

7200* - 2-hours runtime limit

Lastly, when we evaluated the results of CP-based approach on their own merits, it can be concluded that except the solution for the objective weights of $w_1 = 0.4$ & $w_2 = 0.6$, all of the generated results with other weights can be stated as compromise solutions since they are not dominated by each other. In other words, when the project makespan gets longer, total resource cost value will decrease. In the opposite case, total resource cost objective will deteriorate in case of the shorter project makespan values. However, most likely and pessimistic values (96.86 versus 16841 and 119.5 versus 17947) of the two objectives for the objective weights of $w_1 = 0.4$ & $w_2 = 0.6$ are dominated by the other solution whose project makespan and total resource cost values are equal to 96.68 versus 16726 and 118.1 versus 17813, respectively. On the other hand, it should also be highlighted here that optimistic values of these solutions (71.72 versus 14694 & 79.31 versus 14382) are not dominated by each other (see in Table 4.14). The dominance relations among these solutions can be more accurately evaluated after obtaining the optimal results. It should also be noted here that the presented solutions in Table 4.14 are not optimal since both of the ILP and CP-based approaches were terminated at 2-hours runtime limit.

Finally, ERP company's managers and consultants have announced that all of these solutions presented in Table 4.14 are applicable and therefore, they can be realized while generating appropriate project schedules under such a fully uncertain environment. Moreover, they also reached agreement with the customer firm's managers to apply one of these solutions. Indeed, they achieved a consensus on the compromise solution with the objective weights of $w_1 = 0.6$ & $w_2 = 0.4$. Because, they thought that project makespan objective is a little more important than the resource costs. For that reason, they have chosen that solution with the objective weights of $w_1 = 0.6$ & $w_2 = 0.4$ because it is more practical to implement. Thus, some of the managerial results of this solution is visualized in the remainder part of this chapter.

Firstly, the fuzzy project makespan values under different scenarios are presented in Table 4.15. According to this table, scenarios#1 & 11 have the minimum and maximum project makespan values, respectively. For this reason, Gantt charts and

some of the resource profiles related to only these scenarios were drawn in Figures 4.14-4.17. Details of the fuzzy completion times of the project activities with mode selections are also given in Appendices-9 and 10. With the help of these completion times and fuzzy durations of the activities, their starting time intervals are determined and displayed in Figures 4.14 and 4.15. It should be noted here that starting and completion time intervals may intersect with each other for most of the project activities. In other words, a project activity may be initialized or completed at this intersection time points. Actually, these intersections are resulted from fully uncertain nature of the examined problem.

Table 4.15 Fuzzy project makespan values under different scenarios ($w_1 = 0.6$ & $w_2 = 0.4$)

Scenario	Fuzzy project makespan	Scenario	Fuzzy project makespan
1	(61, 85, 108)	10	(60, 83, 108)
2	(57, 79, 104)	11	(50, 71, 92)
3	(59, 83, 105)	12	(57, 79, 103)
4	(53, 75, 98)	13	(56, 79, 102)
5	(58, 81, 104)	14	(56, 78, 102)
6	(59, 81, 105)	15	(53, 75, 98)
7	(57, 78, 99)	16	(55, 78, 100)
8	(55, 76, 98)	17	(59, 82, 105)
9	(61, 83, 108)	18	(60, 84, 107)

After drawing the fuzzy compromise project schedules under scenarios#1 & 11, some resource profiles of the renewable project sources, i.e., human resources are displayed in Figures 4.16-4.17. In detail, resource profile of the process experts in customer firm are shown in Figure 4.16 by considering just optimistic case under scenario#11. Actually, this resource profile is obtained from the solution of lower level problem only. In a similar way, resource profiles of this type of human resource can also be visualized for most likely and pessimistic cases by using the solutions of middle and upper level problems. In Figure 4.17, resource profile of ERP Company's consultants is depicted by considering just most likely case under scenario#1. Similarly, optimistic and pessimistic resource profiles of this kind of human resource can also be drawn by using the solutions of lower and upper level problems.

Ultimately, availability levels of the renewable and non-renewable project resources are demonstrated in Figure 4.18-4.20. In Figure 4.18, fuzzy availability levels of the ERP consultants are shown under different scenarios. Similarly, fuzzy

availability levels of the process experts are also displayed in Figure 4.19. According to Figure 4.18-4.19, middle and upper level problems gave the same results for the human resources. In other words, availability levels of the human resources have the same values in most likely and pessimistic cases. In Figure 4.20, availability levels of non-renewable resource, i.e., project budget needed under different scenarios are portrayed.

It should be noted here that crisp project funds/budget are provided under each scenario despite of defining them as fuzzy variables. In contrast to the human resource costs, availability levels of this project budget are not multiplied by a cost term while calculating the total resource costs. Actually, this is the main reason why crisp project budget values instead of fuzzy ones are obtained under each scenario.

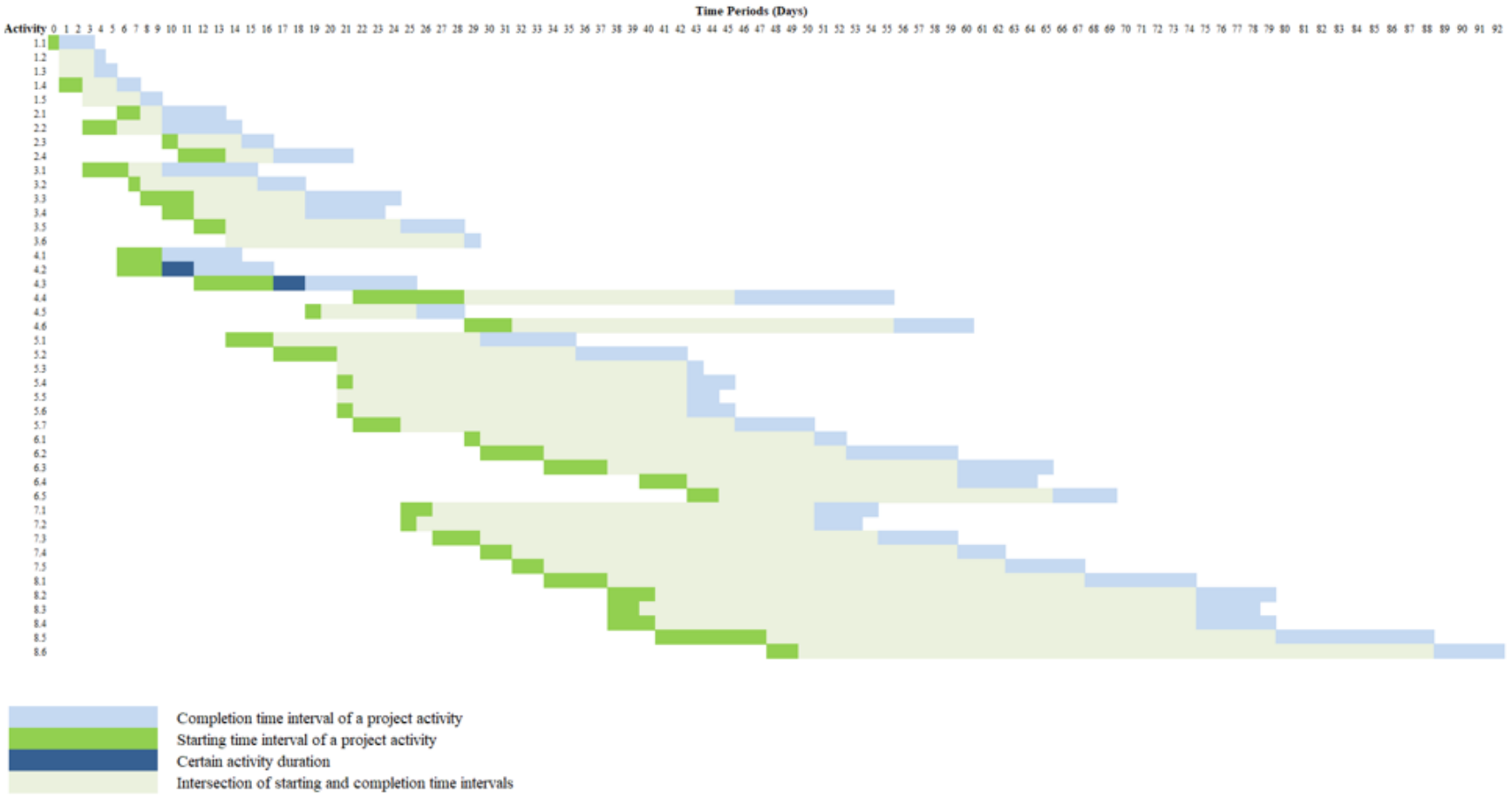


Figure 4.14 Gantt chart of a fuzzy compromise project schedule with minimum makespan under scenario#11 (w1 = 0.6 & w2 = 0.4)

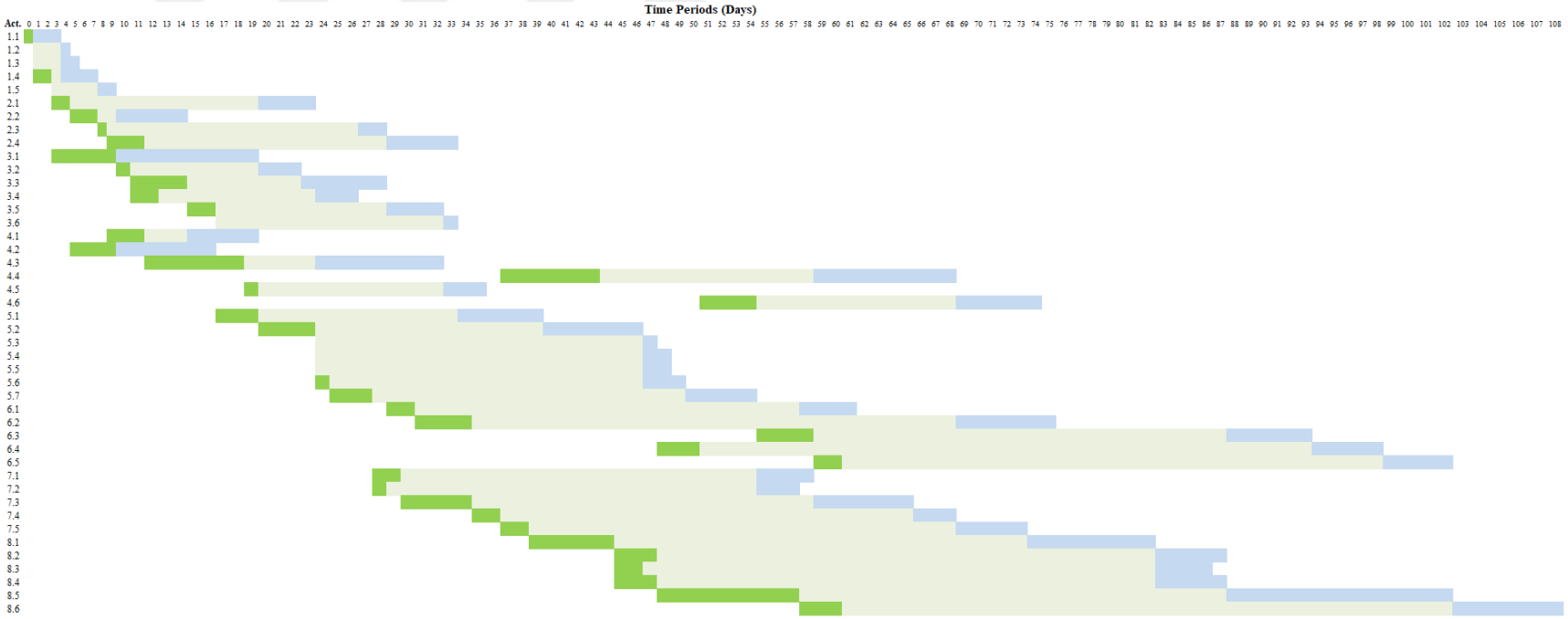


Figure 4.15 Gantt chart of a fuzzy compromise project schedule with maximal makespan under scenario#1 (w1 = 0.6 & w2 = 0.4)

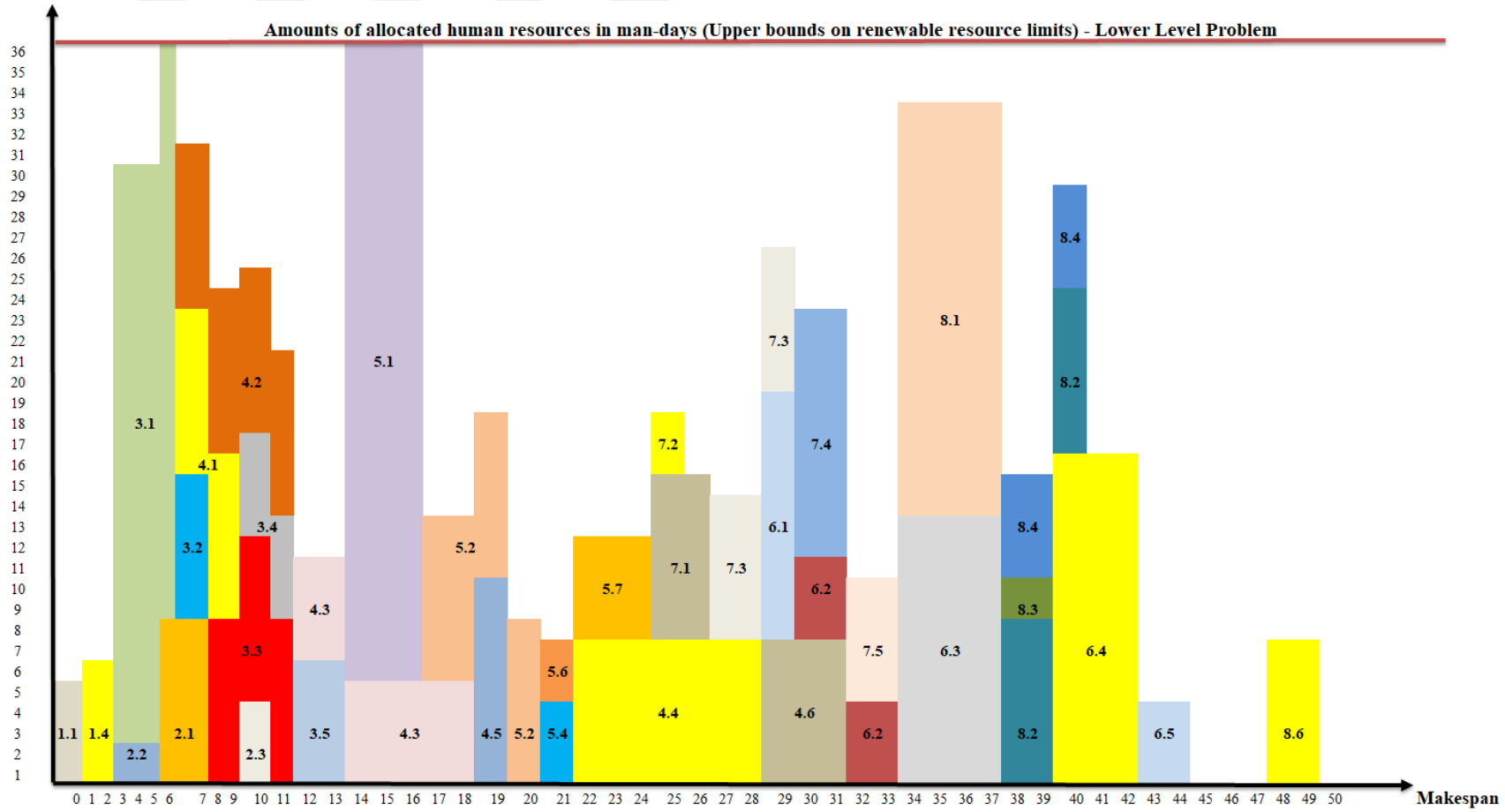


Figure 4.16 The resource profile of the process experts (human resources) for the optimistic case under scenario#11 (w1 = 0.6 & w2 = 0.4)

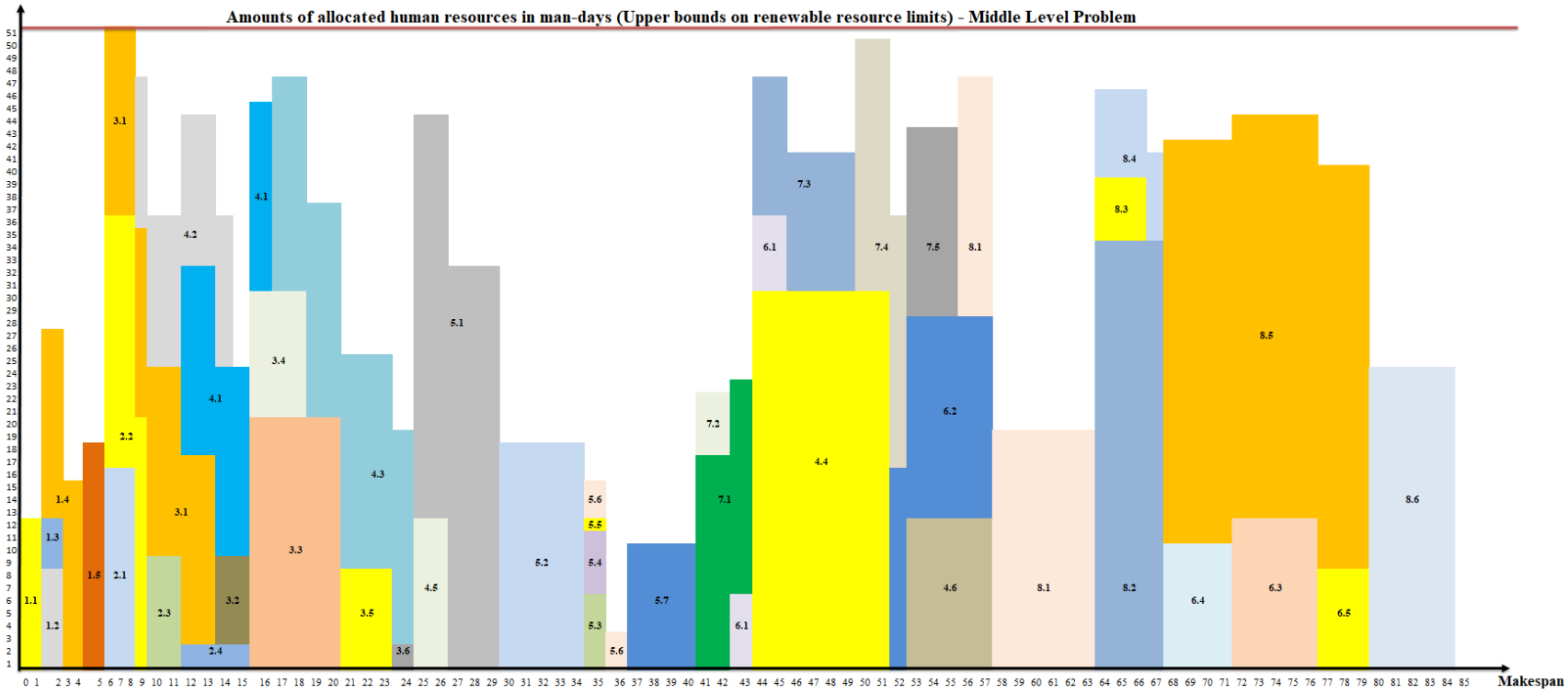


Figure 4.17 The resource profile of the ERP Company's consultants (human resources) for the most likely case under scenario#1 (w1 = 0.6 & w2 = 0.4)

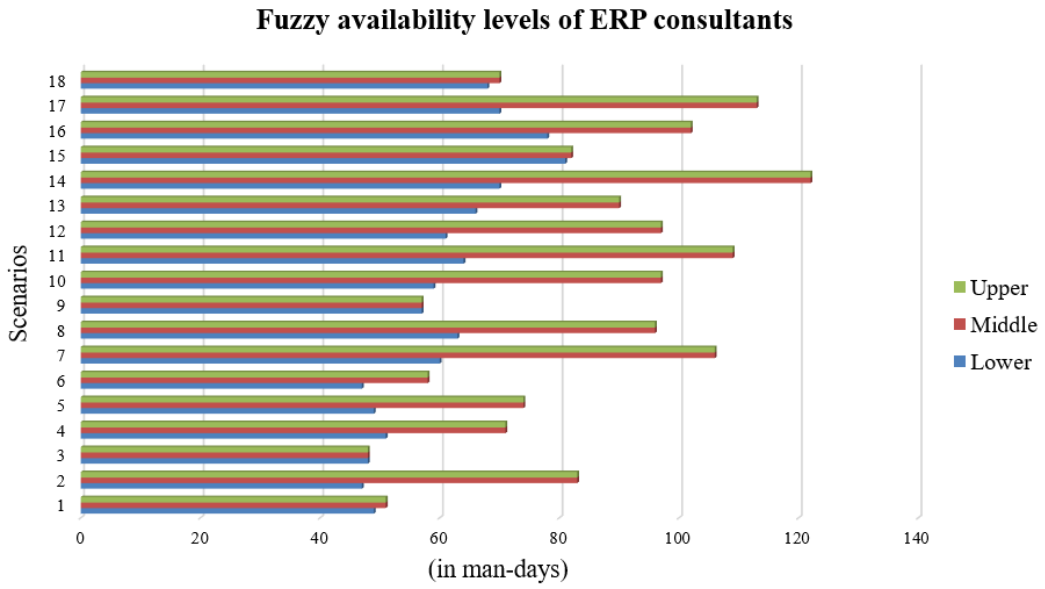


Figure 4.18 Fuzzy availability levels of ERP company's consultants under different scenarios

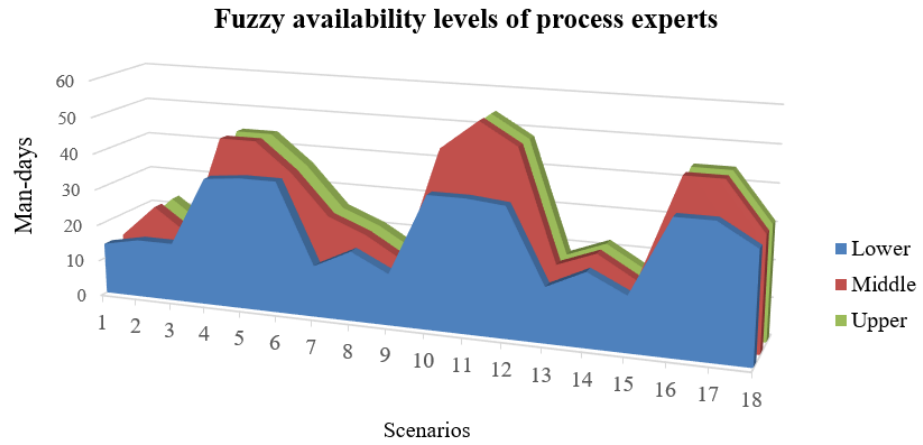


Figure 4.19 Fuzzy availability levels of human resources (process experts) under different scenarios

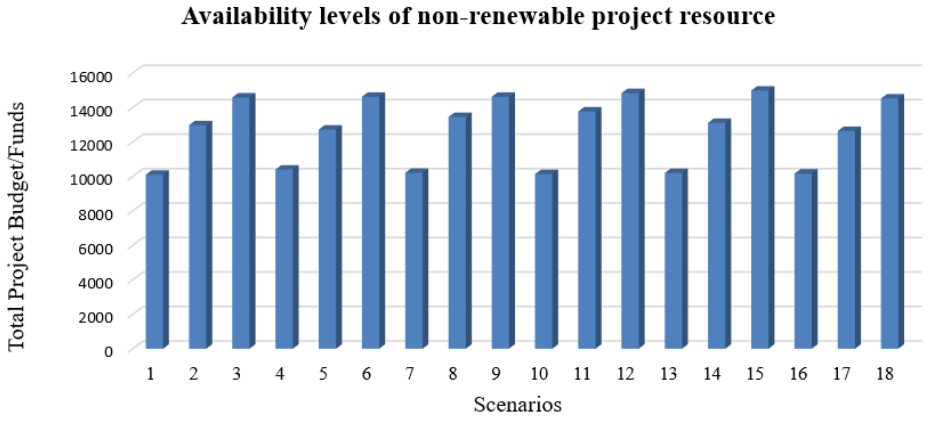


Figure 4.20 Availability levels of project funds/budget under different scenarios

CHAPTER FIVE
CONCLUSIONS & FUTURE RESEARCH DIRECTIONS

5.1 Summary of the Thesis and Concluding Remarks

Nowadays, effective and efficient usage of the limited resources (renewable/non-renewable) in knowledge-intensive projects is closely related to the optimal scheduling of project activities. For this reason, two advanced project scheduling problems are addressed in this thesis.

Firstly, MRC-DTCTP under fully deterministic environment is presented in Chapter 3. This problem aims to minimize total project cost consisting of direct and indirect costs of the renewable project resources while satisfying the constraints related to the precedence relations among the activities, maximum resource capacities, and project deadlines. It is also intended to determine each project activity's starting and completion times under the selected mode and execution method, i.e., normal and crashing ways. The first feature which distinguishes this problem from the classic MRCPSP is that it allows the project activities to be executed under the crashing way to shorten the activity duration with a selected mode. The second feature is that this problem also covers direct and indirect cost resulting from the usage of the renewable project resources. Indeed, balancing the trade-off between these direct and indirect project costs is targeted under a predetermined project deadline. Thus, the most appropriate execution modes and methods, i.e., crashing or normal ways, can be determined for each project activity. Based on these facts, a modified ILP model provided by revising the original integer non-linear programming (INLP) model of Wuliang and Schengen (2009) is first presented in Chapter 3. However, since the NP-hard nature of the MRC-DTCTP, the computational effort for the optimal solution of the modified ILP model may increase considerably. Since its model dimension, i.e., the number of decision variables and constraints increases rapidly for the medium and large-sized problem instances. For this reason, the CP model of this problem is also proposed to cope with the realistic size problems and make further comparisons. The performances of the proposed CP and ILP models are compared based on several

benchmark instances by performing a comprehensive experimental study. Moreover, several search strategies are employed for variable and value ordering to improve the performance of the proposed CP model. The computational experiments have shown that the proposed CP model can generate superior results for the medium and large-sized benchmark instances in solution quality and computation time. The validity and applicability of the proposed CP and ILP models have also been tested on a real-life application for the POM module group implementation process of an international ERP software company. The time-cost trade-off curves, which display total project costs against different project deadlines, are also drawn under different scenarios consisting of several human resource availabilities and project deadlines. According to the computational results, the proposed CP model outperforms the modified ILP model in most of these scenarios. In detail, the proposed CP model can produce superior results, especially for the more relaxed project deadlines and tighter human resource availabilities. Furthermore, it is also inferred that most the project activities should be executed under the crashing way in case of tighter project deadlines and highly restricted human resources.

In Chapter 4, a novel CP-based transformation approach with a bound & decomposition principle is introduced for solving a multi-objective and multi-mode RIPSP under mixed fuzzy-stochastic environments. Actually, all of the project input data are represented by either fuzzy or stochastic parameters for better reflection of the real world applications. Because of the fully uncertain nature of the project data, it is also aimed to generate fuzzy-stochastic outputs which correspond to project schedules and availability levels of the renewable/non-renewable project resources. Due the NP-complete nature of the FS-MRIPSP, it is a challenging task to produce optimal project schedules within reasonable computation times even under fully deterministic environments. Therefore, development of a CP-based transformation approach is intended within the scope of this thesis instead of using a classical ILP-based one. It should also be noted here that most of the transformation approaches in fuzzy mathematical programming literature usually utilize an ILP-based approach with fuzzy arithmetic operations and fuzzy ranking techniques. In recent years, it was also proven that these approaches may cause questionable results due to some problematic issues

resulting from the usage of standard fuzzy arithmetic. Furthermore, these traditionally ILP-based transformation approaches may also be inadequate while solving large-size combinatorial optimization problems under such a mixed fuzzy-stochastic uncertainties. Based on this motivation, the main aim of this thesis is to develop a CP-based transformation approach with a bound & decomposition principle which does not require fuzzy arithmetic and ranking operators directly. The proposed approach is also able to produce fuzzy project schedules and fuzzy resource availabilities under different scenarios. In order to cope with the multi-objective nature of the problem, the proposed transformation approach is also integrated with WAFGP method. The validity and practicality of the proposed approach is tested on a real-world application to a production-and-operations management (POM) module implementation process of an international Enterprise Resource Planning (ERP) software company. The generated results by the proposed CP-based transformation approach are also compared to a similar ILP-based approach. The computational results have shown that the proposed CP-based approach outperforms the ILP-based one in terms of both solution quality and computing times.

5.2 Future Research Directions

Suggestions for future researches based on the examined problems in Chapters 3 and 4 are specified as follows:

- The MRC-DTCTP discussed in Chapter 3 can be extended by considering the non-renewable project resources like total project budget limitations. Moreover, the proposed CP and ILP models of the MRC-DTCTP can also be modified by including multiple project scheduling cases for complex project organizations. In other words, development of the CP and ILP models and a heuristic or metaheuristic-based solution approach for a multi-mode resource-constrained discrete time-cost trade-off multiple project scheduling problem can be scheduled as future work. Furthermore, the MRC-DTCTP with fuzzy and stochastic project data/inputs can also be handled in future works since

many project scheduling applications may cover different types of uncertainties simultaneously.

- Development of a matheuristic solution methodology which hybridizes a meta-heuristic algorithm with the proposed CP models in Chapter 4 can be scheduled as future work to solve the larger scaled problem instances within shorter runtimes. Moreover, fuzzy-random variables can also be employed to describe some project parameters/input data for better reflection of the hybrid uncertainties in the future studies.

REFERENCES

Afruzi, E. N., Najafi, A. A., Roghanian, E., & Mazinani, M. (2014). A multi-objective imperialist competitive algorithm for solving discrete time, cost and quality trade-off problems with mode-identity and resource-constrained situations. *Computers & Operations Research*, *50*, 80-96.

Afruzi, E. N., Roghanian, E., Najafi, A. A., & Mazinani, M. (2013). A multi-mode resource-constrained discrete time–cost tradeoff problem solving using an adjusted fuzzy dominance genetic algorithm. *Scientia Iranica*, *20*(3), 931-944.

Ahn, T., & Erenguc, S. S. (1998). The resource constrained project scheduling problem with multiple crashable modes: a heuristic procedure. *European Journal of Operational Research*, *107*(2), 250-259.

Alipouri, Y., Sebt, M. H., Ardeshir, A., & Chan, W. T. (2019). Solving the FS-RCPSP with hyper-heuristics: A policy-driven approach. *Journal of the Operational Research Society*, *70*(3), 403-419.

Alipouri, Y., Sebt, M. H., Ardeshir, A., & Zarandi, M. H. F. (2020). A mixed-integer linear programming model for solving fuzzy stochastic resource constrained project scheduling problem. *Operational Research*, *20*(1), 197-217.

Amid, A., Ghodsypour, S. H., & O'Brien, C. (2009). A weighted additive fuzzy multiobjective model for the supplier selection problem under price breaks in a supply chain. *International Journal of Production Economics*, *121*(2), 323-332.

Aminbakhsh, S., & Sonmez, R. (2016). Discrete particle swarm optimization method for the large-scale discrete time–cost trade-off problem. *Expert Systems with Applications*, *51*, 177-185.

Artigues, C., Leus, R., & Nobibon, F. T. (2013). Robust optimization for resource-constrained project scheduling with uncertain activity durations. *Flexible Services and Manufacturing Journal*, *25*(1), 175-205.

Artykov, D., & Atymtayeva, L. (2015). A fuzzy linear programming approach for resource-constrained project scheduling. *Advanced Engineering Technology and Application*, *4*(3), 47-52.

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.

Atli, O., & Kahraman, C. (2012). Fuzzy resource-constrained project scheduling using taboo search algorithm. *International Journal of Intelligent Systems*, *27*(10), 873-907.

Atli, O., & Kahraman, C. (2014). Resource-constrained project scheduling problem with multiple execution modes and fuzzy/crisp activity durations. *Journal of Intelligent & Fuzzy Systems*, *26*(4), 2001-2020.

Balouka, N., & Cohen, I. (2021). A robust optimization approach for the multi-mode resource-constrained project scheduling problem. *European Journal of Operational Research*, *291*(2), 457-470.

Baptiste, P., Le Pape, C., & Nuijten, W. (2012). *Constraint-based scheduling: applying constraint programming to scheduling problems* (Vol. 39). US: Springer Science & Business Media.

Bartak, R. (1999). Constraint programming: In pursuit of the holy grail. *Proceedings of the Week of Doctoral Students*, *4*, 555-564.

Bartak, R. (2005). *Constraint propagation and backtracking-based search*. Retrieved August 17, 2021, from https://www.math.unipd.it/~frossi/cp-school/CPschool05notes.pdf

Baykasoğlu, A., & Subulan, K. (2015). An analysis of fully fuzzy linear programming with fuzzy decision variables through logistics network design problem. *Knowledge-Based Systems, 90,* 165-184.

Baykasoğlu, A., & Subulan, K. (2016). A multi-objective sustainable load planning model for intermodal transportation networks with a real-life application. *Transportation Research Part E: Logistics and Transportation Review, 95,* 207-247.

Baykasoğlu, A., & Subulan, K. (2017). Constrained fuzzy arithmetic approach to fuzzy transportation problems with fuzzy decision variables. *Expert Systems with Applications, 81,* 193-222.

Baykasoğlu, A., & Subulan, K. (2019). A direct solution approach based on constrained fuzzy arithmetic and metaheuristic for fuzzy transportation problems. *Soft Computing, 23*(5), 1667-1698.

Baykasoğlu, A., Dudaklı, N., Şenol, M. E., & Kömürcü, F. (2020). Mathematical programming approach to productivity improvement in wind turbine-blade manufacturing through a case study. *Engineering with Computers,* 1-14.

Baykasoğlu, A., Topaloğlu, Ş., & Şenyüzlüler, F. (2017). Manufacturing cell formation with flexible processing capabilities and worker assignment: Comparison of constraint programming and integer programming approaches. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 231*(9), 1-15.

Bellman, R. E., & Zadeh, L. A. (1970). Decision-Making in a fuzzy environment. *Management Science, 17*(4), 141-164.

Berthold, T., Heinz, S., Lubbecke, M. E., Möhring, R. H., & Schulz, J. (2010). A Constraint Integer Programming Approach for Resource-Constrained Project Scheduling. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems: 7th International Conference*. (vol. 6140) (313-317). Berlin: Springer.

Bhaskar, T., Pal, M. N., & Pal, A. K. (2011). A heuristic method for RCPSP with fuzzy activity times. *European Journal of Operational Research*, *208*(1), 57-66.

Birjandi, A., & Mousavi, S. M. (2019). Fuzzy resource-constrained project scheduling with multiple routes: A heuristic solution. *Automation in Construction*, *100*, 84-102.

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.

Bruni, M. E., Pugliese, L. D. P., Beraldi, P., & Guerriero, F. (2018). A computational study of exact approaches for the adjustable robust resource-constrained project scheduling problem. *Computers & Operations Research*, *99*, 178-190.

Chakrabortty, R. K., Sarker, R. A., & Essam, D. L. (2017). Resource constrained project scheduling with uncertain activity durations. *Computers & Industrial Engineering*, *112*, 537-550.

Chand, S., Singh, H. K., & Ray, T. (2016, July). Finding robust solutions for resource constrained project scheduling problems involving uncertainties. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, 225-232.

Chen, H., Ding, G., Qin, S., & Zhang, J. (2021). A hyper-heuristic based ensemble genetic programming approach for stochastic resource constrained project scheduling problem. *Expert Systems with Applications*, *167*, 114174.

Chen, H., Ding, G., Zhang, J., & Qin, S. (2019). Research on priority rules for the stochastic resource constrained multi-project scheduling problem with new project arrival. *Computers & Industrial Engineering*, *137*, 106060.

Chen, L., & Zhang, Z. (2016). Preemption resource-constrained project scheduling problems with fuzzy random duration and resource availabilities. *Journal of Industrial and Production Engineering*, *33*(6), 373-382.

Chen, P. H., & Weng, H. (2009). A two-phase GA model for resource-constrained project scheduling. *Automation in Construction*, *18*(4), 485-498.

Chen, Z., Demeulemeester, E., Bai, S., & Guo, Y. (2018). Efficient priority rules for the stochastic resource-constrained project scheduling problem. *European Journal of Operational Research*, *270*(3), 957-967.

Cheng, M. Y., & Tran, D. H. (2014). Two-phase differential evolution for the multiobjective optimization of time–cost tradeoffs in resource-constrained construction projects. *IEEE Transactions on Engineering Management*, *61*(3), 450-461.

Demeulemeester, E. L., & Herroelen, W.S. (2002). *Project scheduling: a research handbook.* Boston: Kluwer Academic Publishers.

Demeulemeester, E. L., Herroelen, W. S., & Elmaghraby, S. E. (1996). Optimal procedures for the discrete time/cost trade-off problem in project networks. *European Journal of Operational Research*, *88*(1), 50-68.

Drexl, A., & Kimms, A. (2001). Optimization guided lower and upper bounds for the resource investment problem. *Journal of the Operational Research Society*, *52*(3), 340-351.

Gang, J., Xu, J., & Xu, Y. (2013). Multiproject resources allocation model under fuzzy random environment and its application to industrial equipment installation engineering. *Journal of Applied Mathematics*, Article ID 818731, 1-19.

Ghoddousi, P., Eshtehardian, E., Jooybanpour, S., & Javanmardi, A. (2013). Multi-mode resource-constrained discrete time–cost-resource optimization in project scheduling using non-dominated sorting genetic algorithm. *Automation in construction*, *30*, 216-227.

Hapke, M., & Slowinski, R. (1996). Fuzzy priority heuristics for project scheduling. *Fuzzy sets and systems*, *83*(3), 291-299.

Hapke, M., Jaszkiewicz, A., & Slowinski, R. (1994). Fuzzy project scheduling system for software development. *Fuzzy Sets and Systems*, *67*(1), 101-117.

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

Hauder, V. A., Beham, A., Raggl, S., Parragh, S. N., & Affenzeller, M. (2020). Resource-constrained multi-project scheduling with activity and time flexibility. *Computers & Industrial Engineering*, *150*, 106857.

Hazır, Ö., Erel, E., & Günalay, Y. (2011). Robust optimization models for the discrete time/cost trade-off problem. *International Journal of Production Economics*, *130*(1), 87-95.

Hegazy, T., & Menesi, W. (2012). Heuristic method for satisfying both deadlines and resource constraints. *Journal of Construction Engineering and Management*, *138*(6), 688-696.

Herroelen, W., & De Reyck, B. (1999). Phase transitions in project scheduling. *Journal of Operational Research Society*, *50*, 148–156.

Hindelang, T. J., & Muth, J. F. (1979). A dynamic programming algorithm for decision CPM networks. *Operations Research*, *27*(2), 225-241.

Hsu, C. C., & Kim, D. S. (2005). A new heuristic for the multi-mode resource investment problem. *Journal of the Operational Research Society*, *56*(4), 406-413.

Huang, W., Ding, L., Wen, B., & Cao, B. (2009, May). Project scheduling problem for software development with random fuzzy activity duration times. In *International Symposium on Neural Networks*, 60-69. Berlin: Springer.

Huang, W., Oh, S. K., & Pedrycz, W. (2013). A fuzzy time-dependent project scheduling problem. *Information Sciences*, *246*, 100-114.

IAS Inc. Industrial Application Software, CANIAS ERP. (n.d.). Retrieved March 2, 2021, from https://www.canias40.com/tr.

IBM (2017). IBM ILOG CPLEX optimization studio Version 12 Release 8 user's manual.

Javanmard, S., Afshar-Nadjafi, B., & Niaki, S. T. A. (2017). Preemptive multi-skilled resource investment project scheduling problem: Mathematical modelling and solution approaches. *Computers & Chemical Engineering*, *96*, 55-68.

Jayalakshmi, M., & Pandian, P. (2012). A new method for finding an optimal fuzzy solution for fully fuzzy linear programming problems. *International Journal of Engineering Research and Applications*, *2*(4), 247-254.

Jedrzejowicz, P., &Ratajczak-Ropel, E. (2019). Experimental evaluation of a-teams solving resource availability cost problem. *Intelligent Decision Technologies*, 213-223.

Jing-wen, Z., & Hui-fang, S. (2010, November). Multi-resource constrained discrete time/cost trade-off problem and its improved genetic algorithm. In *2010 International Conference on Management Science & Engineering 17th Annual Conference Proceedings*, 123-128.

Jørgensen, T. (1999). *Project scheduling as a stochastic dynamic decision problem*. PhD Thesis, Norwegian University of Science and Technology, Trondheim, Norway.

Kassandra, T., & Suhartono, D. (2018). Resource-constrained project scheduling problem using firefly algorithm. *Procedia Computer Science*, *135*, 534-543.

Kaveh, A., Khanzadi, M., & Alipour, M. (2016). Fuzzy resource constraint project scheduling problem using CBO and CSS algorithms. *International Journal of Civil Engineering*, *14*(5), 325-337.

Ke, H., & Liu, B. (2007). Project scheduling problem with mixed uncertainty of randomness and fuzziness. *European Journal of Operational Research*, *183*(1), 135-147.

Ke, H., Ma, W., & Ma, J. (2012). Solving project scheduling problem with the philosophy of fuzzy random programming. *Fuzzy Optimization and Decision Making*, *11*(3), 269-284.

Khalilzadeh, M., Shakeri, H., Gholami, H., & Amini, L. (2017). A heuristic algorithm for project scheduling with fuzzy parameters. *Procedia Computer Science*, *121*, 63-71.

Knyazeva, M., Bozhenyuk, A., & Rozenberg, I. (2015). Resource-constrained project scheduling approach under fuzzy conditions. *Procedia Computer Science*, *77*, 56-64.

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

Kolisch, R., & Padman, R. (2001). An integrated survey of deterministic project scheduling. *Omega*, *29*(3), 249-272.

Kong, F., & Dou, D. (2021). Resource-Constrained Project Scheduling Problem under Multiple Time Constraints. *Journal of Construction Engineering and Management*, *147*(2), 04020170.

Li, H., & Womer, N. K. (2015). Solving stochastic resource-constrained project scheduling problems by closed-loop approximate dynamic programming. *European Journal of Operational Research*, *246*(1), 20-33.

Li, H., & Zhang, H. (2013). Ant colony optimization-based multi-mode scheduling under renewable and nonrenewable resource constraints. *Automation in construction*, *35*, 431-438.

Li, X., He, Z., Wang, N., & Vanhoucke, M. (2020). Multimode time-cost-robustness trade-off project scheduling problem under uncertainty. *Journal of Combinatorial Optimization*. Advance online publication. https://doi.org/10.1007/s10878-020-00636-7

Liess, O., & Michelon, P. (2008). A constraint programming approach for the resource-constrained project scheduling problem. *Annals of Operations Research*, *157*(1), 25-36.

Liu, D., Li, H., Wang, H., Qi, C., & Rose, T. (2020). Discrete symbiotic organisms search method for solving large-scale time-cost trade-off problem in construction scheduling. *Expert Systems with Applications*, *148*, 113230.

Liu, S. X., & Song, J. H. (2011). Combination of constraint programming and mathematical programming for solving resources-constrained project-scheduling problems. *Control Theory & Applications, 28*(8), 1113-1120.

Liu, S., Yung, K. L., & Ip, W. H. (2007). Genetic local search for resource-constrained project scheduling under uncertainty. *International Journal of Information and Management Sciences*, *18*(4), 347.

Long, L. D., & Ohsato, A. (2008). Fuzzy critical chain method for project scheduling under resource constraints and uncertainty. *International Journal of Project Management*, *26*(6), 688-698.

Lustig, I. J., & Puget, J. F. (2001). Program does not equal program: Constraint programming and its relationship to mathematical programming. *Interfaces, 31*(6), 29-53.

Mahdavi, A., Shirazi, B., & Rezaeian, J. (2021). Toward a scalable type-2 fuzzy model for resource-constrained project scheduling problem. *Applied Soft Computing*, *100*, 106988.

Malcolm, D. G., Rosenbloom, J.M., Clark, C.E., & Fazar, W. (1959). Application of a technique for research and development program evaluation. *Operations Research*, *7*, 646–669.

Masmoudi, M., & Haït, A. (2013). Project scheduling under uncertainty using fuzzy modelling and solving techniques. *Engineering Applications of Artificial Intelligence*, *26*(1), 135-149.

Menesi, W., & Hegazy, T. (2015). Multimode resource-constrained scheduling and leveling for practical-size projects. *Journal of Management in Engineering*, *31*(6), 04014092.

Menesi, W., Golzarpoor, B., & Hegazy, T. (2013). Fast and near-optimum schedule optimization for large-scale projects. *Journal of Construction Engineering and Management*, *139*(9), 1117-1124.

Merrikhi, E., Najafi, A.B., & Shahsavar, A. (2018). Project resource investment problem under progress payment model. *Journal of Industrial and Systems Engineering*, *11*(3), 84–101.

Mizrak Ozfirat, P., Edis, E., & Ozkarahan, I. (2006). Mathematical modeling approach for course timetabling problem of Dokuz Eylul University Industrial Engineering Department. *Proceedings of the International Conference on Modeling and Simulation*, *7*(2), 853-857.

Motwani, J., Mirchandani, D., Madan, M., & Gunasekaran, A. (2002). Successful implementation of ERP projects: evidence from two case studies. *International Journal of Production Economics*, *75*(1-2), 83-96.

Mouhoub, M., & Jafari, B. (2011, July). Heuristic techniques for variable and value ordering in csps. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, 457-464.

Nader, A. S., Roghanian, E., & Aghassi, H. (2011). A multi-mode resource-constrained optimization of time-cost trade-off problems in project scheduling

using a genetic algorithm. *Journal of Optimization in Industrial Engineering 8*, 55-64.

Najafi, A. A., & Niaki, S. T. A. (2006). A genetic algorithm for resource investment problem with discounted cash flows. *Applied Mathematics and Computation*, *183*(2), 1057-1070.

Najafi, A.A., & Azimi, F. (2009). A priority rule-based heuristic for resource investment project scheduling problem with discounted cash flows and tardiness penalties. *Mathematical Problems in Engineering*, *2009*, https://doi.org/10.1155/2009/106425.

Nematian, J., Eshghi, K., & Eshragh-Jahromi, A. (2010). A resource-constrained project scheduling problem with fuzzy random duration. *Journal of Uncertain Systems*, *4*(2), 123-132.

Okada, I., Takahashi, K., Zhang, W., Zhang, X., Yang, H., & Fujimura, S. (2014). A genetic algorithm with local search using activity list characteristics for solving resource-constrained project scheduling problem with multiple modes. *IEEJ Transactions on Electrical and Electronic Engineering*, *9*(2), 190-199.

Okubo, H., Miyamoto, T., Yoshida, S., Mori, K., Kitamura, S., & Izui, Y. (2015). Project scheduling under partially renewable resources and resource consumption during setup operations. *Computers & Industrial Engineering*, *83*, 91-99.

Özdamar, L., & Alanya, E. (2001). Uncertainty modelling in software development projects (with case study). *Annals of Operations Research*, *102*(1), 157-178.

Özdamar, L., & Ulusoy, G. (1995). A survey on the resource-constrained project scheduling problem. *IIE transactions*, *27*(5), 574-586.

Pan, H., Willis, R.J., & Yeh, C.H. (2001). Resource constrained project scheduling with fuzziness. In N. Mastorakis (Ed.). *Advances in fuzzy systems and evolutionary computation*, (1st ed.) (173-179). Danvers: World Scientific and Engineering Society Press.

Plaza, M. (2016). Balancing the costs of human resources on an ERP project. *Omega, 59*, 171-183.

Pritsker, A. A. B., Waiters, L. J., & Wolfe, P. M. (1969). Multiproject scheduling with limited resources: A zero-one programming approach. *Management Science, 16*(1), 93-108.

Ranjbar, M., Kianfar, F., & Shadrokh, S. (2008). Solving the resource availability cost problem in project scheduling by path relinking and genetic algorithm. *Applied Mathematics and Computation, 196*(2), 879-888.

Rezaei, F., Najafi, A. A., & Ramezanian, R. (2020). Mean-conditional value at risk model for the stochastic project scheduling problem. *Computers & Industrial Engineering, 142*, 106356.

Roghanian, E., Alipour, M., & Rezaei, M. (2018). An improved fuzzy critical chain approach in order to face uncertainty in project scheduling. *International Journal of Construction Management, 18*(1), 1-13.

Rostami, S., Creemers, S., & Leus, R. (2018). New strategies for stochastic resource-constrained project scheduling. *Journal of Scheduling, 21*(3), 349-365.

Said, S. S., & Haouari, M. (2015). A hybrid simulation-optimization approach for the robust Discrete Time/Cost Trade-off Problem. *Applied Mathematics and Computation, 259*, 628-636.

Sajadi, S. M., Azimi, P., Ghamginzadeh, A., & Rahimzadeh, A. (2017). A new fuzzy multi-objective multi-mode resource-constrained project scheduling model. *International Journal of Mathematics in Operational Research*, *11*(1), 45-66.

Sallam, K. M., Chakrabortty, R. K., & Ryan, M. J. (2021). A reinforcement learning based multi-method approach for stochastic resource constrained project scheduling problems. *Expert Systems with Applications*, *169*, 114479.

Schnell, A., & Hartl, R. F. (2017). On the generalization of constraint programming and boolean satisfiability solving techniques to schedule a resource-constrained project consisting of multi-mode jobs. *Operations Research Perspectives*, *4*, 1-11.

Shadrokh, S., & Kianfar, F. (2007). A genetic algorithm for resource investment project scheduling problem, tardiness permitted with penalty. *European Journal of Operational Research*, *181*(1), 86-101.

Sonmez, R., & Bettemir, Ö. H. (2012). A hybrid genetic algorithm for the discrete time–cost trade-off problem. *Expert Systems with Applications*, *39*(13), 11428-11434.

Sonmez, R., Iranagh, M. A., & Uysal, F. (2016). Critical sequence crashing heuristic for resource-constrained discrete time–cost trade-off problem. *Journal of Construction Engineering and Management*, *142*(3), 04015090.

Sotoudeh-Anvari, A. (2020). A critical review on theoretical drawbacks and mathematical incorrect assumptions in fuzzy OR methods: Review from 2010 to 2020. *Applied Soft Computing*, *93*, 106354.

Subulan, K. (2019). An interval programming based approach for fully uncertain resource-constrained project scheduling problem considering project manager's

attitude toward risk. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, *25*(4), 481-497.

Subulan, K. (2020). An interval-stochastic programming based approach for a fully uncertain multi-objective and multi-mode resource investment project scheduling problem with an application to ERP project implementation. *Expert Systems with Applications*, *149*, 113189.

Tadinen, H. (2005). *Human resources management aspects of enterprise resource planning (ERP) systems projects*. Master Thesis, Swedish School of Economics and Business Administration, Helsinki.

Tao, S., & Dong, Z. S. (2018). Multi-mode resource-constrained project scheduling problem with alternative project structures. *Computers & Industrial Engineering*, *125*, 333-347.

Tseng, C. C., & Ko, P. W. (2016). Measuring schedule uncertainty for a stochastic resource-constrained project using scenario-based approach with utility-entropy decision model. *Journal of Industrial and Production Engineering*, *33*(8), 558-567.

Vanhoucke, M. (2005). New computational results for the discrete time/cost trade-off problem with time-switch constraints. *European Journal of Operational Research*, *165*(2), 359-374.

Vanhoucke, M., & Debels, D. (2007). The discrete time/cost trade-off problem: extensions and heuristic procedures. *Journal of Scheduling*, *10*(4), 311-326.

Vanhoucke, M., Demeulemeester, E., & Herroelen, W. (2002). Discrete time/cost trade-offs in project scheduling with time-switch constraints. *Journal of the Operational Research Society*, *53*(7), 741-751.

Wang, J. (2002). A fuzzy project scheduling approach to minimize schedule risk for product development. *Fuzzy Sets and Systems*, *127*(2), 99-116.

Wang, J. (2004). A fuzzy robust scheduling approach for product development projects. *European Journal of Operational Research*, *152*(1), 180-194.

Wang, L., & Fang, C. (2011). An effective shuffled frog-leaping algorithm for multi-mode resource constrained project scheduling problem. *Information Sciences*, *181*(20), 4804-4822.

Wang, L., Huang, H., & Ke, H. (2015). Chance-Constrained model for RCPSP with uncertain durations. *Journal of Uncertainty Analysis and Applications*, *12*(3), 1–10.

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., Liu, J., & Abbass, H. A. (2011, June). An evolutionary multi-objective scenario-based approach for stochastic resource investment project scheduling. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, 2767-2774.

Xiong, J., Liu, J., Chen, Y., & Abbass, H. A. (2013). A knowledge-based evolutionary multiobjective approach for stochastic extended resource investment project scheduling problems. *IEEE Transactions on Evolutionary Computation*, *18*(5), 742-763.

Xu, J., & Feng, C. (2014). Multimode resource-constrained multiple project scheduling problem under fuzzy random environment and its application to a large scale hydropower construction project. *The Scientific World Journal*, Article ID 463692, 1-20.

Xu, J., & Zhang, Z. (2012). A fuzzy random resource-constrained scheduling model with multiple projects and its application to a working procedure in a large-scale water conservancy and hydropower construction project. *Journal of Scheduling*, *15*(2), 253-272.

Xu, J., Ma, Y., & Xu, Z. (2015). A bilevel model for project scheduling in a fuzzy random environment. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *45*(10), 1322-1335.

Xu, J., Zheng, H., Zeng, Z., Wu, S., & Shen, M. (2012). Discrete time–cost–environment trade-off problem for large-scale construction systems with multiple modes under fuzzy uncertainty and its application to Jinping-II Hydroelectric Project. *International Journal of Project Management*, *30*(8), 950-966.

Yamashita, D. S., Armentano, V. A., & Laguna, M. (2007). Robust optimization models for project scheduling with resource availability cost. *Journal of Scheduling*, *10*(1), 67-76.

Young, K. D., Feydy, T., & Schutt, A. (2017, August). Constraint programming applied to the multi-skill project scheduling problem. In *International Conference on Principles and Practice of Constraint Programming*, 308-317.

Yousefli, A. (2017). A fuzzy ant colony approach to fully fuzzy resource constrained project scheduling problem. *Industrial Engineering & Management Systems*, *16*(3), 307-315.

Zamani, R. (2013). An evolutionary search procedure for optimizing time–cost performance of projects under multiple renewable resource constraints. *Computers & Industrial Engineering*, *66*(2), 451-460.

Zhang, Q., Zhou, J., Wang, K., & Pantelous, A. A. (2018). An effective solution approach to fuzzy programming with application to project scheduling. *International Journal of Fuzzy Systems*, *20*(8), 2383-2398.

Zhang, Z. (2014, July). A MODM bi-level model with fuzzy random coefficients for resource-constrained project scheduling problems. In *2014 Seventh International Joint Conference on Computational Sciences and Optimization*, 666-669.

Zhang, Z., Liu, M., & Song, X. (2019). A bi-level fuzzy random model for multi-mode resource-constrained project scheduling problem of photovoltaic power plant. *Journal of Renewable and Sustainable Energy*, 11(3), 1-15.

Zhu, X., Ruiz, R., Li, S., & Li, X. (2017). An effective heuristic for project scheduling with resource availability cost. *European Journal of Operational Research*, *257*(3), 746-762.

APPENDICES

APPENDIX-1: IBM ILOG CPLEX code of the modified ILP model for the MRC-DTCTP

```
//ILP
int NbTasks=...;
int NbRs=...;
int NbModes=...;
int tperiod=...;
int CapR[1..NbRs]=...;
float c[1..NbRs]=...;
float u[1..NbRs]=...;
int Deadline=...;
range task=0..NbTasks;
range time=0..tperiod;
range mode=1..NbModes;
range resource=1..NbRs;
{int} succs[task]=...;
int d[task][mode]=...;
float eft[task]=...;
float lft[task]=...;
int k[task][mode][resource]=...;
dvar boolean x[task][mode][time];
dexpr float uc [r in resource]=(sum(m in mode, t in time:t>=eft[NbTasks]&&t<=lft[NbTasks])t*x[NbTasks][m][t])*u[r];
dexpr float ProjectCost_Fc= sum(j in task, m in mode,t in time,r in
resource:m>(NbModes/2)&&k[j][m][r]>0)(x[j][m][t]*d[j][m]*c[r])+sum(r in resource)uc[r];
minimize ProjectCost_Fc;
subject to {
forall(j in task)sum(m in mode,t in time:t>=eft[j] && t<=lft[j]) x[j][m][t]==1;
forall(i in task,j in succs[i])sum(m in mode,t in time:t>=eft[i] && t<=lft[i])t*x[i][m][t]<=sum(m in mode,t in
time:t>=eft[j]&&t<=lft[j])(t-d[j][m])*x[j][m][t];
forall(r in resource,t in time)sum(j in task,m in mode:j>=1&&j<=NbTasks-1)k[j][m][r]*sum(s in time:s>=t&&s<=(t+d[j][m]-
1))x[j][m][s]<=CapR[r];
sum(m in mode, t in time:t>=eft[NbTasks]&&t<=lft[NbTasks])t*x[NbTasks][m][t]<=Deadline;
};
```

APPENDIX-2: IBM ILOG CPLEX code of the proposed CP model for the MRC-DTCTP

```
//CP
using CP;
int NbTasks=...;
int NbRs=...;
int CapR[1..NbRs]=...;
float c[1..NbRs]=...;
float u[1..NbRs]=...;
int Deadline=...;
tuple Task {key int id;{int} succs; }
{Task} Tasks=...;
tuple Mode{
int mid;
int tid;
int nd;
int dmdR [1..NbRs];
}
{Mode} Modes=...;
dvar interval task [t in Tasks];
dvar interval mode [m in Modes] optional size m.nd;
execute{cp.param.timeLimit=10;
}
cumulFunction rUsage [r in 1..NbRs]=
sum (m in Modes:m.dmdR[r]>0) pulse(mode[m], m.dmdR[r]);
dexpr float uc [r in 1..NbRs]=(max(t in Tasks) endOf(task[t]))*u[r];
dexpr float ProjectCost_Fc=
sum( t in Tasks,m in Modes,r in 1..NbRs:m.tid==t.id && m.mid==2
&&m.dmdR[r]>0)(presenceOf(mode[m]))*(sizeOf(mode[m]))*c[r]+sum(r in 1..NbRs)uc[r];
minimize ProjectCost_Fc;
subject to {
forall (t in Tasks)alternative(task[t], all(m in Modes: m.tid==t.id) mode[m]);
forall (r in 1..NbRs)rUsage[r]<= CapR[r];
forall(t1 in Tasks, t2id in t1.succs)endBeforeStart(task[t1], task[<t2id>]);
max(t in Tasks) endOf(task[t])<= Deadline;
};
```

APPENDIX-3: Details of the variable and value ordering strategies implemented to improve the performance of the CP model

Strategy No.	Variable and Value Ordering Strategy (Search Phases)	Explanations

1	execute{
	varf=cp.factory;
	vaphase1=f.searchPhase(X_j, f.selectSmallest(f.domainSize()),
	f.selectSmallest(f.valueSuccessRate()));
	varphase2=f.searchPhase($Y_{jm(m')}$,
	f.selectSmallest(f.domainSize()),
	f.selectSmallest(f.valueSuccessRate()));
	cp.setSearchPhases(phase1,phase2); }
		After the variables with the smallest domain size are selected, the smallest success rate value is selected from the value set and assigned to X_j variable in phase-1. The variable ordering in search phase-1 schedules project activities before any other decision is applied. Similarly, value assignment is made for the $Y_{jm(m')}$ decision variable in phase-2.

2	execute{
	varf=cp.factory;
	varphase1=f.searchPhase(X_j,f.selectSmallest(f.domainSize()),
	f.selectSmallest(f.valueSuccessRate()));
	varphase2=f.searchPhase($Y_{jm(m')}$,
	f.selectSmallest(f.domainSize()),
	f.selectSmallest(f.valueSuccessRate()));
	cp.setSearchPhases(phase2,phase1); }
		After the variables with the smallest domain size are selected, the smallest success rate value is selected from the value set and assigned to the $Y_{jm(m')}$variable in phase-2. The variable ordering in search phase-2 schedules project activities before any other decision is applied. Similarly, value assignment is made for X_j decision variable in phase-1.

3	execute{
	varf=cp.factory;
	varphase1=f.searchPhase(X_j, f.selectLargest(f.Impact()),
	f.selectLargest(f.valueSuccessRate()));
	varphase2=f.searchPhase($Y_{jm(m')}$,f.selectLargest(f.Impact()),
	f.selectLargest(f.valueSuccessRate()));
	cp.setSearchPhases(phase1,phase2); }
		After the variables with the largest impact on search tree are selected, the largest success rate value is selected from the value set and assigned to X_j variable in phase-1. The variable ordering in search phase-1 schedules project activities before any other decision is applied. Similarly, value assignment is made for $Y_{jm(m')}$ decision variable in phase-2.

4	execute{
	varf=cp.factory;
	varphase1=f.searchPhase(X_j,f.selectLargest(f.Impact()),
	f.selectLargest(f.valueSuccessRate()));
	varphase2=f.searchPhase($Y_{jm(m')}$f.selectLargest(f.Impact()),
	f.selectLargest(f.valueSuccessRate()));
	cp.setSearchPhases(phase2,phase1); }
		After the variables with the largest impact on search tree are selected, the largest success rate value is selected from the value set and assigned to $Y_{jm(m')}$ variable in phase-2. The variable ordering in search phase-2 schedules project activities before any other decision is applied. Similarly, value assignment is made for X_j decision variable in phase-1.

APPENDIX-3: Continues

Strategy No.	Variable and Value Ordering Strategy (Search Phases)	Explanations

5	execute{
	varf=cp.factory;
	varphase1=f.searchPhase(X_j,f.selectLargest(f.Impact()),
	f.selectSmallest(f.valueSuccessRate()));
	varphase2=f.searchPhase($Y_{jm(m')}$,f.selectLargest(f.Impact()),
	f.selectSmallest(f.valueSuccessRate()));
	cp.setSearchPhases(phase1,phase2); }
		After the variables with the largest impact on search tree are selected, the smallest success rate value is selected from the value set and assigned to X_j variable in phase-1. The variable ordering in search phase-1 schedules project activities before any other decision is applied. Similarly, value assignment is made for $Y_{jm(m')}$ decision variable in phase-2.

6	execute{
	varf=cp.factory;
	varphase1=f.searchPhase(X_j,f.selectLargest(f.Impact()),
	f.selectSmallest(f.valueSuccessRate()));
	varphase2=f.searchPhase($Y_{jm(m')}$,f.selectLargest(f.Impact()),
	f.selectSmallest(f.valueSuccessRate()));
	cp.setSearchPhases(phase2,phase1); }
		After the variables with the largest impact on search tree are selected, the smallest success rate value is selected from the value set and assigned to $Y_{jm(m')}$ variable in phase-2. The variable ordering in search phase-2 schedules project activities before any other decision is applied. Similarly, value assignment is made for X_j decision variable in phase-1.

7	execute{
	varf=cp.factory;
	varphase1=f.searchPhase(X_j,f.selectSmallest(f.domainSize()),
	f.selectSmallest(f.valueImpact()));
	varphase2=f.searchPhase($Y_{jm(m')}$,f.selectSmallest(f.domainSize()),
	f.selectSmallest(f.valueImpact()));
	cp.setSearchPhases(phase1,phase2); }
		After the variables with the smallest domain size are selected, the smallest impact value is selected from the value set and assigned to X_j variable in phase-1. The variable ordering in search phase-1 schedules project activities before any other decision is applied. Similarly, value assignment is made for $Y_{jm(m')}$ decision variable in phase-2.

8	execute{
	varf cp.factory;
	varphase1=f.searchPhase(X_j,f.selectSmallest(f.domainSize()),
	f.selectSmallest(f.valueImpact()));
	varphase2=f.searchPhase($Y_{jm(m')}$,f.selectSmallest(f.domainSize()),
	f.selectSmallest(f.valueImpact()));
	cp.setSearchPhases(phase2,phase1); }
		After the variables with the smallest domain size are selected, the smallest impact value is selected from the value set and assigned to $Y_{jm(m')}$ variable in phase-2. The variable ordering in search phase-2 schedules project activities before any other decision is applied. Similarly, value assignment is made for X_j decision variable in phase-1.

APPENDIX-4: Fuzzy project schedules under different scenarios for the illustrative example

Scenario	ILP-based approach			CP-based approach		
	Activity	Mode	Fuzzy completion time	Activity	Mode	Fuzzy completion time
1	1	1	(3,4,5)	1	1	(3,4,5)
	2	1	(2,2,4)	2	1	(1,2,3)
	3	2	(4,5,6)	3	2	(4,5,6)
	4	1	(5,6,8)	4	1	(4,6,8)
	5	1	(5,7,9)	5	1	(5,7,9)
2	1	2	(5,6,7)	1	2	(5,6,7)
	2	1	(1,2,4)	2	1	(1,2,3)
	3	2	(4,5,6)	3	2	(4,5,6)
	4	1	(6,8,10)	4	1	(6,8,10)
	5	1	(5,7,9)	5	1	(5,7,9)
3	1	1	(3,4,5)	1	1	(3,4,5)
	2	1	(2,2,4)	2	1	(1,2,3)
	3	2	(4,5,6)	3	2	(4,5,6)
	4	1	(5,6,8)	4	1	(4,6,8)
	5	1	(5,7,9)	5	1	(5,7,9)
4	1	1	(3,4,5)	1	1	(3,4,5)
	2	1	(1,2,4)	2	1	(1,2,3)
	3	2	(4,5,6)	3	2	(4,5,6)
	4	1	(5,6,8)	4	1	(4,6,8)
	5	1	(5,7,9)	5	1	(5,7,9)
5	1	1	(3,4,5)	1	1	(3,4,5)
	2	1	(4,9,12)	2	1	(4,6,8)
	3	1	(6,7,9)	3	1	(6,9,12)
	4	1	(4,9,12)	4	1	(4,6,8)
	5	1	(7,11,15)	5	1	(7,11,15)
6	1	1	(7,9,11)	1	1	(7,9,11)
	2	1	(1,4,4)	2	1	(1,2,3)
	3	2	(4,5,6)	3	2	(4,5,6)
	4	1	(10,13,14)	4	1	(8,11,14)
	5	2	(10,13,16)	5	2	(10,13,16)
7	1	1	(3,4,5)	1	1	(3,4,5)
	2	1	(4,9,12)	2	1	(4,6,8)
	3	1	(6,7,9)	3	2	(7,9,11)
	4	1	(4,9,12)	4	1	(5,8,11)
	5	1	(7,11,15)	5	1	(8,11,14)
8	1	1	(3,4,5)	1	1	(3,4,5)
	2	1	(4,6,8)	2	1	(5,8,11)
	3	2	(7,9,11)	3	2	(7,9,11)
	4	1	(7,8,11)	4	1	(4,6,8)
	5	1	(8,11,14)	5	1	(8,11,14)
9	1	1	(3,4,5)	1	1	(3,4,5)
	2	1	(2,3,4)	2	1	(1,2,3)
	3	1	(2,3,4)	3	1	(2,3,4)
	4	1	(4,6,8)	4	1	(4,6,8)
	5	1	(4,5,7)	5	1	(3,5,7)
10	1	2	(5,6,7)	1	2	(5,6,7)
	2	1	(1,3,4)	2	1	(1,2,3)
	3	1	(2,3,4)	3	1	(2,3,4)
	4	1	(6,8,10)	4	1	(6,8,10)
	5	2	(5,7,9)	5	2	(5,7,9)

APPENDIX-4: Continues

Scenario	ILP-based approach			CP-based approach		
	Activity	Mode	Fuzzy completion time	Activity	Mode	Fuzzy completion time
11	1	1	(3,4,5)	1	1	(3,4,5)
	2	1	(2,3,4)	2	1	(1,2,3)
	3	1	(2,3,4)	3	1	(2,3,4)
	4	1	(4,6,8)	4	1	(4,6,8)
	5	1	(3,5,7)	5	1	(3,5,7)
12	1	1	(3,4,5)	1	1	(3,4,5)
	2	1	(3,3,4)	2	1	(1,2,3)
	3	2	(4,5,6)	3	2	(4,5,6)
	4	1	(5,6,8)	4	1	(4,6,8)
	5	1	(5,7,9)	5	1	(5,7,9)
13	1	1	(4,6,8)	1	1	(3,4,5)
	2	1	(1,2,3)	2	1	(4,6,8)
	3	2	(4,5,6)	3	2	(4,5,6)
	4	1	(5,8,11)	4	1	(4,6,8)
	5	1	(5,7,9)	5	1	(5,8,11)
14	1	2	(5,6,7)	1	2	(5,6,7)
	2	1	(1,7,9)	2	1	(5,7,9)
	3	2	(5,5,6)	3	2	(4,5,6)
	4	1	(6,8,10)	4	1	(6,8,10)
	5	1	(6,9,12)	5	1	(6,9,12)
15	1	1	(3,4,5)	1	1	(3,4,5)
	2	1	(4,6,8)	2	1	(4,6,8)
	3	2	(4,5,6)	3	2	(4,5,6)
	4	1	(5,6,9)	4	1	(4,6,8)
	5	1	(5,8,11)	5	1	(5,8,11)
16	1	1	(3,4,5)	1	1	(3,4,5)
	2	1	(4,6,8)	2	1	(4,6,8)
	3	2	(4,5,6)	3	2	(4,5,6)
	4	1	(5,6,9)	4	1	(4,6,8)
	5	1	(5,8,11)	5	1	(5,8,11)

APPENDIX-5: IBM ILOG CPLEX code of the ILP model for single objective middle level FS-MRIPSP

```
//MIDDLE LEVEL-Z1
int NbTasks= ...;
int NbRRsrcs= ...;
int NbNRsrcs= ...;
int NbPeriods= ...;
int NbModes=...;
int NbScenarios=...;
range RRsrcs=1..NbRRsrcs;
range NRsrcs=1..NbNRsrcs;
range Periods=0..NbPeriods;
range Tasks=0..NbTasks;
range Modes=1..NbModes;
range Scenarios=1..NbScenarios;
{int}succs[Tasks]=...;
int eft[Tasks]=...;
int lft[Tasks]=...;
int d[Tasks][Modes]=...;
float pi[1..NbScenarios]=...;
float kk[Tasks][Modes][RRsrcs][Scenarios]=...;
float nkk[Tasks][Modes][NRsrcs][Scenarios]=...;
int C[RRsrcs]=...;
dvar boolean x[Tasks][Modes][Periods][Scenarios];
dvar int RCapacity[RRsrcs][Scenarios];
dvar int NCapacity[NRsrcs][Scenarios];
dvar float+ z2;
execute{cplex.tilim=7200;
}
//MODEL(ILP)
minimize sum(j in Tasks:j==NbTasks,m in Modes,a in Scenarios,t in
Periods:t>=eft[j] &&t<=lft[j])pi[a]*t*x[j][m][t][a];
subject to {
z2==sum(r in RRsrcs, a in Scenarios)pi[a]*RCapacity[r][a]*C[r]+sum(n in NRsrcs,
a in Scenarios)pi[a]*NCapacity[n][a];
forall(j in Tasks,a in Scenarios)
sum(m in Modes,t in Periods:t>=eft[j] &&t<=lft[j])x[j][m][t][a]==1;
forall(i in Tasks,j in succs[i], a in Scenarios)
sum(m in Modes,t in Periods:t>=eft[i] &&t<=lft[i])t*x[i][m][t][a]<=sum(m in
Modes,t in Periods:t>=eft[j] &&t<=lft[j])
(t-d[j][m])*x[j][m][t][a];
forall(r in RRsrcs,t in Periods,a in Scenarios)
sum(j in Tasks,m in Modes:j>=1&&j<=NbTasks-1)kk[j][m][r][a]*sum(s in
Periods:s>=t&&s<=(t+d[j][m]-1))
x[j][m][s][a]<=RCapacity[r][a];
forall(n in NRsrcs,a in Scenarios)
sum(j in Tasks,m in Modes:j>=1&&j<=NbTasks-1)nkk[j][m][n][a]*sum(t in
Periods:t>=eft[j] &&t<=lft[j])
x[j][m][t][a]<=NCapacity[n][a];
};
```

APPENDIX-5: Continues

```
//MIDDLE LEVEL-Z2
int NbTasks= ...;
int NbRRsrcs= ...;
int NbNRsrcs= ...;
int NbPeriods= ...;
int NbModes=...;
int NbScenarios=...;
range RRsrcs=1..NbRRsrcs;
range NRsrcs=1..NbNRsrcs;
range Periods=0..NbPeriods;
range Tasks=0..NbTasks;
range Modes=1..NbModes;
range Scenarios=1..NbScenarios;
{int}succs[Tasks]=...;
int eft[Tasks]=...;
int lft[Tasks]=...;
int d[Tasks][Modes]=...;
float pi[1..NbScenarios]=...;
float kk[Tasks][Modes][RRsrcs][Scenarios]=...;
float nkk[Tasks][Modes][NRsrcs][Scenarios]=...;
int C[RRsrcs]=...;
dvar boolean x[Tasks][Modes][Periods][Scenarios];
dvar int RCapacity[RRsrcs][Scenarios];
dvar int NCapacity[NRsrcs][Scenarios];
dvar float+ z1;
execute{cplex.tilim=7200;
}
//MODEL(ILP)
minimize sum(r in RRsrcs, a in Scenarios)pi[a]*RCapacity[r][a]*C[r]+sum(n in
NRsrcs, a in Scenarios)pi[a]*NCapacity[n][a];
subject to {
z1==sum(j in Tasks:j==NbTasks,m in Modes,a in Scenarios,t in Periods:t>=eft[j]
&&t<=lft[j])pi[a]*t*x[j][m][t][a];
forall(j in Tasks,a in Scenarios)
sum(m in Modes,t in Periods:t>=eft[j] &&t<=lft[j])x[j][m][t][a]==1;
forall(i in Tasks,j in succs[i], a in Scenarios)
sum(m in Modes,t in Periods:t>=eft[i] &&t<=lft[i])t*x[i][m][t][a]<=sum(m in
Modes,t in Periods:t>=eft[j] &&t<=lft[j])
(t-d[j][m])*x[j][m][t][a];
forall(r in RRsrcs,t in Periods,a in Scenarios)
sum(j in Tasks,m in Modes:j>=1&&j<=NbTasks-1)kk[j][m][r][a]*sum(s in
Periods:s>=t&&s<=(t+d[j][m]-1))
x[j][m][s][a]<=RCapacity[r][a];
forall(n in NRsrcs,a in Scenarios)
sum(j in Tasks,m in Modes:j>=1&&j<=NbTasks-1)nkk[j][m][n][a]*sum(t in
Periods:t>=eft[j] &&t<=lft[j])
x[j][m][t][a]<=NCapacity[n][a];
};
```

APPENDIX-6: IBM ILOG CPLEX code of the CP model for single objective middle level FS-MRIPSP

```
//MIDDLE LEVEL-Z1
using CP;
int NbRs1s=...;
int NbRs2s=...;
int NbS=...;
range scenarios=1..NbS;
range NbRs1IDs=1..NbRs1s;
range NbRs2IDs=1..NbRs2s;
int C[NbRs1IDs]=...;
float pi[1..NbS]=...;
tuple Task {key int id;{int} succs;}
{Task} Tasks=...;
tuple Mode {int sid; int taskId; int id; int pt; int dmdR1[NbRs1IDs]; float
dmdR2[NbRs2IDs];}
{Mode} Modes=...;
dvar interval task[t in Tasks][a in 1..NbS ];
dvar interval mode[m in Modes] optional size m.pt;
dvar int CapRs1[NbRs1IDs][scenarios];
dvar int CapRs2[NbRs2IDs][scenarios];
dexpr float z2=sum (r in NbRs1IDs, a in scenarios) pi[a]*CapRs1[r][a]*C[r]+sum
(r in NbRs2IDs, a in scenarios)pi[a]*CapRs2[r][a];
execute{cp.param.timeLimit=7200;
}
//MODEL(CP)
minimize sum (a in scenarios) pi[a]*(max(t in Tasks) endOf(task[t][a]));
subject to {
z2==sum (r in NbRs1IDs, a in scenarios) pi[a]*CapRs1[r][a]*C[r]+sum (r in
NbRs2IDs, a in scenarios)pi[a]*CapRs2[r][a];
forall(t in Tasks, a in scenarios)
  alternative(task[t][a], all(m in Modes: m.taskId==t.id && m.sid==a) mode[m]);
forall(t1 in Tasks, t2id in t1.succs, a in scenarios)
  endBeforeStart(task[t1][a],task[<t2id>][a]);
forall(r in NbRs1IDs, a in scenarios)
  sum(m in Modes:m.dmdR1[r]>0 && m.sid==a) pulse(mode[m],
m.dmdR1[r])<=CapRs1[r][a];
forall(r in NbRs2IDs, a in scenarios)
  sum(m in Modes:m.dmdR2[r]>0 && m.sid==a)
m.dmdR2[r]*presenceOf(mode[m])<=CapRs2[r][a];
};
```

APPENDIX-6: Continues

```
//MIDDLE LEVEL-Z2
using CP;
int NbRs1s=...;
int NbRs2s=...;
int NbS=...;
range scenarios=1..NbS;
range NbRs1IDs=1..NbRs1s;
range NbRs2IDs=1..NbRs2s;
int C[NbRs1IDs]=...;
float pi[1..NbS]=...;
tuple Task {key int id;{int} succs;}
{Task} Tasks=...;
tuple Mode {int sid; int taskId; int id; int pt; int dmdR1[NbRs1IDs]; float
dmdR2[NbRs2IDs];}
{Mode} Modes=...;
dvar interval task[t in Tasks][a in 1..NbS ];
dvar interval mode[m in Modes] optional size m.pt;
dvar int CapRs1[NbRs1IDs][scenarios];
dvar int CapRs2[NbRs2IDs][scenarios];
dexpr float z1=sum (a in scenarios) pi[a]*(max(t in Tasks) endOf(task[t][a]));
execute{cp.param.timeLimit=7200;
}
//MODEL(CP)
minimize sum (r in NbRs1IDs, a in scenarios) pi[a]*CapRs1[r][a]*C[r]+sum (r in
NbRs2IDs, a in scenarios)pi[a]*CapRs2[r][a];
subject to {
z1==sum (a in scenarios) pi[a]*(max(t in Tasks) endOf(task[t][a]));
forall(a in scenarios,t in Tasks: t.id==45) endOf(task[t][a])<=203;
forall(t in Tasks, a in scenarios)
  alternative(task[t][a], all(m in Modes: m.taskId==t.id && m.sid==a) mode[m]);
forall(t1 in Tasks, t2id in t1.succs, a in scenarios)
  endBeforeStart(task[t1][a],task[<t2id>][a]);
forall(r in NbRs1IDs, a in scenarios)
  sum(m in Modes:m.dmdR1[r]>0 && m.sid==a) pulse(mode[m],
m.dmdR1[r])<=CapRs1[r][a];
forall(r in NbRs2IDs, a in scenarios)
  sum(m in Modes:m.dmdR2[r]>0 && m.sid==a)
m.dmdR2[r]*presenceOf(mode[m])<=CapRs2[r][a];
};
```

APPENDIX-7: IBM ILOG CPLEX code of the ILP and CP models for multi-objective middle level FS-MRIPSP

```
//MIDDLE LEVEL-COMPROMISE
int NbTasks= ...;
int NbRRsrcs= ...;
int NbNRsrcs= ...;
int NbPeriods= ...;
int NbModes=...;
int NbScenarios=...;
range RRsrcs=1..NbRRsrcs;
range NRsrcs=1..NbNRsrcs;
range Periods=0..NbPeriods;
range Tasks=0..NbTasks;
range Modes=1..NbModes;
range Scenarios=1..NbScenarios;
{int}succs[Tasks]=...;
int eft[Tasks]=...;
int lft[Tasks]=...;
int d[Tasks][Modes]=...;
float pi[1..NbScenarios]=...;
float kk[Tasks][Modes][RRsrcs][Scenarios]=...;
float nkk[Tasks][Modes][NRsrcs][Scenarios]=...;
int C[RRsrcs]=...;
dvar boolean x[Tasks][Modes][Periods][Scenarios];
dvar int RCapacity[RRsrcs][Scenarios];
dvar int NCapacity[NRsrcs][Scenarios];
dvar float+ z1;
dvar float+ z2;
dvar float+ MM1;
dvar float+ MM2;
execute{cplex.tilim=7200;
}
//MODEL(ILP)
maximize 0.60*MM1+0.40*MM2;
subject to {
MM1>=0;
MM1<=1;
MM2>=0;
MM2<=1;
MM1<=(203-z1)/(203-71);
MM2<=(22011-z2)/(22011-13943.875);
z1==sum(j in Tasks:j==NbTasks,m in Modes,a in Scenarios,t in Periods:t>=eft[j]
&&t<=lft[j])pi[a]*t*x[j][m][t][a];
z2==sum(r in RRsrcs, a in Scenarios)pi[a]*RCapacity[r][a]*C[r]+sum(n in NRsrcs,
a in Scenarios)pi[a]*NCapacity[n][a];
forall(j in Tasks,a in Scenarios)
sum(m in Modes,t in Periods:t>=eft[j] &&t<=lft[j])x[j][m][t][a]==1;
forall(i in Tasks,j in succs[i], a in Scenarios)
sum(m in Modes,t in Periods:t>=eft[i] &&t<=lft[i])t*x[i][m][t][a]<=sum(m in
Modes,t in Periods:t>=eft[j] &&t<=lft[j])
(t-d[j][m])*x[j][m][t][a];
forall(r in RRsrcs,t in Periods,a in Scenarios)
sum(j in Tasks,m in Modes:j>=1&&j<=NbTasks-1)kk[j][m][r][a]*sum(s in
Periods:s>=t&&s<=(t+d[j][m]-1))
x[j][m][s][a]<=RCapacity[r][a];
forall(n in NRsrcs,a in Scenarios)
sum(j in Tasks,m in Modes:j>=1&&j<=NbTasks-1)nkk[j][m][n][a]*sum(t in
Periods:t>=eft[j] &&t<=lft[j])
x[j][m][t][a]<=NCapacity[n][a];
};
```

APPENDIX-7: Continues

```
//MIDDLE LEVEL-COMPROMISE
using CP;
int NbRs1s=...;
int NbRs2s=...;
int NbS=...;
range scenarios=1..NbS;
range NbRs1IDs=1..NbRs1s;
range NbRs2IDs=1..NbRs2s;
int C[NbRs1IDs]=...;
float pi[1..NbS]=...;
tuple Task {key int id;{int} succs;}
{Task} Tasks=...;
tuple Mode {int sid; int taskId; int id; int pt; int dmdR1[NbRs1IDs]; float
dmdR2[NbRs2IDs];}
{Mode} Modes=...;
dvar interval task[t in Tasks][a in 1..NbS ];
dvar interval mode[m in Modes] optional size m.pt;
dvar int CapRs1[NbRs1IDs][scenarios];
dvar int CapRs2[NbRs2IDs][scenarios];
dexpr float z1=sum (a in scenarios) pi[a]*(max(t in Tasks) endOf(task[t][a]));
dexpr float z2=sum (r in NbRs1IDs, a in scenarios) pi[a]*CapRs1[r][a]*C[r]+sum
(r in NbRs2IDs, a in scenarios)pi[a]*CapRs2[r][a];
dexpr float MM1=(203-z1)/(203-71);
dexpr float MM2=(22011-z2)/(22011-13943.875);
execute{cp.param.timeLimit=7200;
}
//MODEL(CP)
maximize 0.60*MM1+0.40*MM2;
subject to {
MM1>=0;
MM1<=1;
MM2>=0;
MM2<=1;
MM1<=(203-z1)/(203-71);
MM2<=(22011-z2)/(22011-13943.875);
z1==sum (a in scenarios) pi[a]*(max(t in Tasks) endOf(task[t][a]));
z2==sum (r in NbRs1IDs, a in scenarios) pi[a]*CapRs1[r][a]*C[r]+sum (r in
NbRs2IDs, a in scenarios)pi[a]*CapRs2[r][a];
forall(t in Tasks, a in scenarios)
  alternative(task[t][a], all(m in Modes: m.taskId==t.id && m.sid==a) mode[m]);
forall(t1 in Tasks, t2id in t1.succs, a in scenarios)
  endBeforeStart(task[t1][a],task[<t2id>][a]);
forall(r in NbRs1IDs, a in scenarios)
  sum(m in Modes:m.dmdR1[r]>0 && m.sid==a) pulse(mode[m],
m.dmdR1[r])<=CapRs1[r][a];
forall(r in NbRs2IDs, a in scenarios)
  sum(m in Modes:m.dmdR2[r]>0 && m.sid==a)
m.dmdR2[r]*presenceOf(mode[m])<=CapRs2[r][a];
};
```

APPENDIX-8: IBM ILOG CPLEX code of the search strategies into the CP-based transformation approach for FS-MRIPSP

```
execute{
// Search strategy-1;
var f = cp.factory;
var phase1 = f.searchPhase(task, f.selectLargest(f.Impact()),
f.selectLargest(f.valueSuccessRate()));
var f = cp.factory;
var phase2 = f.searchPhase(mode, f.selectLargest(f.Impact()),
f.selectLargest(f.valueSuccessRate()));
cp.setSearchPhases(phase2,phase1);}
// Search strategy-2;
var f = cp.factory;
var phase1 = f.searchPhase(task, f.selectLargest(f.Impact()),
f.selectLargest(f.valueSuccessRate()));
var f = cp.factory;
var phase2 = f.searchPhase(mode, f.selectLargest(f.Impact()),
f.selectLargest(f.valueSuccessRate()));
cp.setSearchPhases(phase1,phase2);}
// Search strategy-3;
var f = cp.factory;
var phase1=
f.searchPhase(task,f.selectSmallest(f.domainSize()),f.selectSmallest(f.valueSucc
essRate()));
var f = cp.factory;
var phase2
=f.searchPhase(mode,f.selectSmallest(f.domainSize()),f.selectSmallest(f.valueSuc
cessRate()));
cp.setSearchPhases(phase1,phase2);}
// Search strategy-4;
var f = cp.factory;
var phase1=
f.searchPhase(task,f.selectSmallest(f.domainSize()),f.selectSmallest(f.valueSucc
essRate()));
var f = cp.factory;
var phase2
=f.searchPhase(mode,f.selectSmallest(f.domainSize()),f.selectSmallest(f.valueSuc
cessRate()));
cp.setSearchPhases(phase2,phase1);}
// Search strategy-5;
var f = cp.factory;
var phase1
=f.searchPhase(task,f.selectLargest(f.Impact()),f.selectSmallest(f.valueImpact()
));
var f = cp.factory;
var phase2 =
f.searchPhase(mode,f.selectLargest(f.Impact()),f.selectSmallest(f.valueImpact())
);
cp.setSearchPhases(phase1,phase2);}
// Search strategy-6;
var f = cp.factory;
var phase1
=f.searchPhase(task,f.selectLargest(f.Impact()),f.selectSmallest(f.valueImpact()
));
var f = cp.factory;
var phase2 =
f.searchPhase(mode,f.selectLargest(f.Impact()),f.selectSmallest(f.valueImpact())
);
cp.setSearchPhases(phase2,phase1);}
```

APPENDIX-8: Continues

```
// Search strategy-7;
var f = cp.factory;
var phase1 =
f.searchPhase(task,f.selectSmallest(f.domainSize()),f.selectSmallest(f.valueImpa
ct()));
var f = cp.factory;
var phase2=
f.searchPhase(mode,f.selectSmallest(f.domainSize()),f.selectSmallest(f.valueImpa
ct()));
cp.setSearchPhases(phase1,phase2);}
// Search strategy-8;
var f = cp.factory;
var phase1 =
f.searchPhase(task,f.selectSmallest(f.domainSize()),f.selectSmallest(f.valueImpa
ct()));
var f = cp.factory;
var phase2=
f.searchPhase(mode,f.selectSmallest(f.domainSize()),f.selectSmallest(f.valueImpa
ct()));
cp.setSearchPhases(phase2,phase1);}
```

APPENDIX-9: Compromise fuzzy project schedules under scenario#11 via the proposed CP-based transformation approach for the real-life case study

Activity	Mode	Fuzzy completion times	Activity	Mode	Fuzzy completion times
1.1	1	(1, 2, 3)	**5.2**	1	(21, 32, 42)
1.2	1	(1, 3, 4)	**5.3**	1	(21, 33, 43)
1.3	1	(1, 3, 5)	**5.4**	2	(22, 34, 45)
1.4	1	(3, 5, 7)	**5.5**	2	(21, 33, 44)
1.5	1	(3, 6, 9)	**5.6**	2	(22, 34, 45)
2.1	1	(8, 9, 13)	**5.7**	1	(25, 38, 50)
2.2	1	(6, 10, 14)	**6.1**	1	(30, 40, 52)
2.3	1	(11, 12, 16)	**6.2**	3	(34, 46, 59)
2.4	3	(14, 16, 21)	**6.3**	3	(38, 51, 65)
3.1	1	(7, 11, 15)	**6.4**	3	(43, 50, 64)
3.2	2	(8, 13, 18)	**6.5**	2	(45, 54, 69)
3.3	1	(12, 18, 24)	**7.1**	2	(27, 41, 54)
3.4	3	(12, 17, 23)	**7.2**	2	(26, 40, 53)
3.5	2	(14, 21, 28)	**7.3**	1	(30, 45, 59)
3.6	1	(14, 22, 29)	**7.4**	1	(32, 48, 62)
4.1	3	(10, 11, 14)	**7.5**	1	(34, 51, 67)
4.2	3	(12, 12, 16)	**8.1**	1	(38, 56, 74)
4.3	3	(19, 20, 25)	**8.2**	1	(41, 60, 79)
4.4	2	(29, 42, 55)	**8.3**	3	(40, 59, 78)
4.5	1	(20, 22, 28)	**8.4**	3	(41, 60, 79)
4.6	1	(32, 46, 60)	**8.5**	1	(48, 68, 88)
5.1	1	(17, 27, 35)	**8.6**	1	**(50, 71, 92)**

APPENDIX-10: Compromise fuzzy project schedules under scenario#1 via the proposed CP-based transformation approach for the real-life case study

Activity	Mode	Fuzzy completion times	Activity	Mode	Fuzzy completion times
1.1	1	(1, 2, 3)	**5.2**	1	(24, 35, 46)
1.2	1	(1, 3, 4)	**5.3**	1	(24, 36, 47)
1.3	1	(1, 3, 5)	**5.4**	1	(24, 36, 48)
1.4	1	(3, 5, 7)	**5.5**	2	(24, 36, 48)
1.5	1	(3, 6, 9)	**5.6**	2	(25, 37, 49)
2.1	1	(5, 9, 23)	**5.7**	1	(28, 41, 54)
2.2	1	(8, 10, 14)	**6.1**	2	(31, 46, 61)
2.3	1	(9, 12, 28)	**6.2**	3	(35, 58, 75)
2.4	3	(12, 16, 33)	**6.3**	3	(59, 77, 93)
3.1	3	(10, 14, 19)	**6.4**	3	(51, 72, 98)
3.2	2	(11, 16, 22)	**6.5**	2	(61, 80, 102)
3.3	1	(15, 21, 28)	**7.1**	2	(30, 44, 58)
3.4	2	(13, 19, 26)	**7.2**	2	(29, 43, 57)
3.5	2	(17, 24, 32)	**7.3**	3	(35, 50, 65)
3.6	1	(17, 25, 33)	**7.4**	1	(37, 53, 68)
4.1	3	(12, 17, 19)	**7.5**	1	(39, 56, 73)
4.2	3	(10, 15, 16)	**8.1**	3	(45, 64, 82)
4.3	3	(19, 25, 32)	**8.2**	1	(48, 68, 87)
4.4	2	(44, 52, 68)	**8.3**	3	(47, 67, 86)
4.5	1	(20, 27, 35)	**8.4**	3	(48, 68, 87)
4.6	2	(55, 58, 74)	**8.5**	3	(58, 80, 102)
5.1	1	(20, 30, 39)	**8.6**	2	**(61, 85, 108)**