An Evolutionary Stochastic Discrete Time-Cost Trade-Off Method for Repetitive and Non- Repetitive Construction Projects

Bahaa Hussein

A Thesis

In the Department

of

Building, Civil, and Environmental Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of

Doctor of Philosophy (Building Engineering) at

Concordia University

Montreal, Quebec, Canada

July 2019

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Bahaa Hussein

Entitled: An Evolutionary Stochastic Discrete Time-Cost Trade-Off Method for Repetitive and Non- Repetitive Construction Projects

and submitted in partial fulfillment of the requirements for the degree of

Doctor Of Philosophy (Building Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

____ Chair
Dr. Abdel R. Sebak

____ External Examiner
Dr. Jeffery Hudson Rankin

____ External to Program
Dr. Amin Hammad

____ Examiner
Dr. Sang Hyeok Han

____ Examiner
Dr. Mazdak Nik-Bakht

____ Thesis Supervisor
Dr. Osama Moselhi

Approved by ____
Dr. Michelle Nokken, Graduate Program Director

September 13, 2019. ____
Dr. Amin Asif, Dean
Gina Cody School of Engineering & Computer Science

Abstract

An Evolutionary Stochastic Discrete Time-Cost Trade-Off Optimization Method for Repetitive and Non- Repetitive Construction Projects

Bahaa Hussein, Ph.D.

Concordia University, 2019

This thesis aims at developing a method for solving the scheduling problem of discrete Time-Cost Trade-off (TCT) in an uncertain context. The method determines the elite execution mode for each project activity to optimize for minimization of the overall project's cost and/or duration while satisfying a specified Joint Confidence Level (JCL) of both time and cost. In this thesis, each resource allocation to individual activities is referred to as mode, and each alternative solution is referred to as a chromosome. A new evolutionary method formulation is developed. The method is of two-folds, the first is an experimental module where generations of chromosomes are developed using a design of experiments and blocking techniques based on a novel approach of partitioning the project scheduling network. At each generation, a complete enumeration is performed for a selection of primary activities, and the elite modes are identified and carried forward to successive generations until all elite modes are identified to form the elements of the supreme chromosome solution. To provide flexibility and practicality, the developed method allows for the end-user interactive selection of execution modes. This arises from the bias of project managers in favouring

specific modes that may not be optimal. Hence, the second fold of the developed method is a random search module that quantifies the effect of changing a mode within a chromosome on the total project cost and duration under a targeted JCL of both time and cost. The method also accounts for penalties/ incentives as a function of time associated with the late/ early project completion depending on the contract provisions. The developed method is also extended for repetitive construction projects considering optimization of crew work continuity for typical and non-typical repetitive activities, i.e. those having identical work amounts and those having a deviation in their work amounts and, therefore, different cost and duration in different repetitive units. The supreme chromosome solution and the main effect plots provide the decision-maker a guideline for making well-informed implementation strategies. The performed computational results demonstrate not only the method benefits and accuracy but also its superiority over current methods for stochastic TCT optimization. The method has been fully coded using Google Apps Script, Google BigQuery SQL statements, and in-line JavaScript functions.

To whom her prayers lightened me throughout my life, my mother.
To the river of tenderness and role models of educators and parents, my dear father.
To the stars in my sky, my wife, son and daughter
I dedicate the fruit of this effort as a gratitude.

Acknowledgment

I would like to express my sincerest gratitude to my supervisor Dr. Osama Moselhi, for his guidance, support, and foremost his patience throughout the course of this study. His valuable advice and commitment far exceeded the call of duty.

I would like to thank my wife Sarah, and my children Zaid and Deena, for having been my source of strength and inspiration. Their encouragement was the reason for me to continue in this research, and their patience in putting up with the life of a student is deeply appreciated.

Finally, I would like to express my sincere thanks to my father Dr. Abdulsattar and grandfather Dr. Abdulmonem, for their continued love, advice, and support.

Table of Contents

LIST OF FIGURES .... **X**

LIST OF TABLES .... **XIV**

CHAPTER 1 : INTRODUCTION **1**

1.1. GENERAL 1

1.2. PROBLEM STATEMENT 4

1.3. RESEARCH SCOPE AND OBJECTIVES 7

1.4. THESIS ORGANIZATION 8

CHAPTER 2 : LITERATURE REVIEW **10**

2.1. INTRODUCTION 10

2.2. WHAT IS A PROJECT? 10

2.3. WHAT IS RISK? 13

2.4. PROBABILITY THEORY 14

2.5. TIME-COST TRADE-OFF IN CONSTRUCTION PROJECTS 15

2.6. DISCRETE-EVENT OPTIMIZATION APPROACHES 18

2.7. REPETITIVE CONSTRUCTION PROJECTS OPTIMIZATION 26

2.8. EXISTING FORECASTING METHODS 30

2.8.1. Deterministic forecasting methods *30*

2.8.2. Probabilistic forecasting methods *33*

2.9. RISK ANALYSIS 39

2.10. PROJECT COSTS 45

2.11. DESIGN OF EXPERIMENTS 48

2.12. FINDINGS OF THE LITERATURE REVIEW 53

CHAPTER 3 : THEORETICAL BACKGROUND **55**

3.1. INTRODUCTION 55

3.2. THEORETICAL CRITICAL PATH METHOD 55

3.3. MODELLING TYPE OF COSTS 59

3.4. MODELLING UNCERTAINTY 63

3.5. JOINT COST AND SCHEDULE CONFIDENCE LEVEL 70

3.6. SENSITIVITY ANALYSIS 76
3.7. TIME-COST TRADE-OFF ANALYSIS 78

CHAPTER 4 : IMPLEMENTATION OF THE DEVELOPED METHOD ON TRADITIONAL NON-REPETITIVE CLASS PROJECTS 83

4.1. INTRODUCTION 83
4.2. CLOUD APPLICATION USED IN THE ESDTCT METHOD 86
4.3. DEFINITION OF TERMS USED IN THE ESDTCT METHOD 89
4.4. IMPLEMENTATION OF DESIGN OF EXPERIMENTS 91
4.5. FUNDAMENTALS OF THE PARTITIONING PROCESS 93
4.6. FORMULATION OF THE BI-OBJECTIVE FITNESS FUNCTION 99
4.7. COMPUTATIONAL PROCEDURE AND FORMULATION 103
4.7.1. Evolutionary experiment enumeration module: $ESDTCT^{Exp}$ 106
4.7.2. Complete Random experiments module: $ESDTCT^{Rand}$ 111
4.8. NUMERICAL EXAMPLES 114
4.8.1. Example 1: Testing $ESDTCT^{Exp}$ module on a small size project under uncertain environment. 114
4.8.2. Example 2: Testing $ESDTCT^{Rand}$ module on a small size project under uncertain environment. 124
4.8.3. Example 3: Testing ESDTCT under certain environment. 128
4.8.4. Example 4: Testing ESDTCT on a large size project under certain environment. 133
4.8.5. Example 5: Testing ESDTCT on a large size project under uncertain environment. 142
4.8.6. Example 6: Testing with discrete risk events. 152

CHAPTER 5 : IMPLEMENTATION OF THE DEVELOPED METHOD ON REPETITIVE CLASS PROJECTS 157

5.1. INTRODUCTION 157
5.2. COMPUTATIONAL PROCEDURE 159

5.3. CALCULATIONS FOR CREW WORK INTERRUPTION TIME. 161
5.4. FORMULATION OF THE TRI-OBJECTIVE FITNESS FUNCTION 164
5.5. OPTIMIZATION FOR CREW WORK CONTINUITY. 165
5.6. FORMULATION OF THE RP-ESDTCT^{Exp} 169
5.7. NUMERICAL EXAMPLES OF REPETITIVE PROJECTS 176
5.7.1. Example 7: Testing RP-ESDTCT^{Exp} module under certain environment...176
5.7.2. Example 8: Testing RP-ESDTCT^{Exp} module under uncertain environment. 185
5.7.3. Example 9: Testing including penalty costs 192

CHAPTER 6 : CONCLUSIONS, CONTRIBUTIONS AND FUTURE WORK. 194

6.1. CONCLUSIONS 194
6.2. CONTRIBUTIONS 196
6.3. LIMITATIONS OF THE DEVELOPED METHOD 198
6.4. DIRECTIONS FOR FURTHER RESEARCH 200

REFERENCES 201

APPENDIX A: ESDTCT METHOD SCRIPT 215

List of Figures

Figure 1.1 Typical discrete time-cost relationship of an activity. 3
Figure 1.2 Typical discrete time-cost relationship of an activity with uncertainty in time and cost. 3
Figure 2.1 Project's Triple Constraints. 11
Figure 2.2 Project's Success Criteria. 12
Figure 2.3 General process steps for Meta-heuristic optimization. 20
Figure 2.4 Process steps for Genetic Algorithm optimization. 24
Figure 2.5 Risks affecting construction projects. 42
Figure 2.6 Cost and Schedule Interrelationships. Source: Smart, C.B., 2007 NASA Cost Symposium. 46
Figure 2.7 Types of project costs 47
Figure 2.8 Factors and Responses. 48
Figure 3.1 Type of logic relationships between the project activities. 58
Figure 3.2 Uncertainty modelling of variable and indirect costs. 61
Figure 3.3 Modelling penalty and bonus cost for the schedule-driven project. ... 63
Figure 3.4 Popular probability distribution functions PDFs. 63
Figure 3.5 Normal probability distribution functions. 64
Figure 3.6 Triangular probability distribution functions. 65
Figure 3.7 Trigen probability distribution functions. 66
Figure 3.8 Double Triangular probability distribution functions. 67
Figure 3.9 Beta-PERT probability distribution functions. 67
Figure 3.10 Uniform probability distribution functions. 70
Figure 3.11 Joint Cost and Schedule Confidence Level (JCL) process steps. ... 70
Figure 3.12 A simple schedule network illustration. 71
Figure 3.13 Assignment of FC, VC and IC costs on project activities. 71
Figure 3.14 Risk-adjusted schedule. 73
Figure 3.15 Uncertainty assignment to the activity durations and costs. 74

Figure 3.16. JCL Scatterplot. 75
Figure 3.17. The marginal probability distribution and marginal risk 76
Figure 3.18 Time – Cost trade-off process steps. 78
Figure 3.19 Relationships between the normal and crashed time and cost. 80
Figure 3.20. Time – cost trade-off relationship. 81
Figure 4.1. Illustration of siloed cost and schedule risk and trade-off assessments. 84
Figure 4.2. Illustration of proposed integrated process. 85
Figure 4.3. Flow chart for the Integration of the developed ESDTCT modules. 86
Figure 4.4. ESDTCT cloud network diagram. 88
Figure 4.5. Immediate reverse dominator tree. 91
Figure 4.6. Definition and structure of terms. 91
Figure 4.7 Main Effect Plot. 93
Figure 4.8 Illustration for the project network partitioning. 96
Figure 4.9 Illustrative example for the project network partitioning at each generation. 97
Figure 4.10 Illustrative example for mode combinations – chromosomes of each generation. 98
Figure 4.11. Typical JCL scatter plot and the frontier curve showing the different optimization objectives. 100
Figure 4.12. Nadir and Ideal vector points for the frontier curve. 102
Figure 4.13. Normalized frontier curve. 103
Figure 4.14. Flow chart of $ESDTCT^{EXP}$ method. 105
Figure 4.15. Flow chart of $ESDTCT^{Rand}$ method. 106
Figure 4.16 Example 1: Network configuration 116
Figure 4.17 Example 1: Network configuration showing primary activities at each generation in the ESDTCT method. 116

Figure 4.18 Example 1: ESDTCT supreme chromosome frontier curve results verses NA-ACO method optimal chromosome reported by (Kalhor et al. 2011). 120
Figure 4.19 Example 1: ESDTCT^{Exp} optimal chromosome for different JCL and different optimization objectives........ 123
Figure 4.20 Example 1: base case 50% JCL frontier curve showing cost and schedule contingency calculations. 123
Figure 4.21 Example 1: ESDTCT^{Exp} optimal chromosome for different JCL and different optimization objectives........ 124
Figure 4.22 Example 2: Main effect plot for response vectors 127
Figure 4.23 Example 2: Tornado chart for activity relative importance. 127
Figure 4.24 Example 3: Main effect plot for response vectors for the objective of cost minimization. 131
Figure 4.25 Example 3: Main effect plot for response vectors for the objective of schedule minimization. 132
Figure 4.26 Example 3: Main effect plot for response vectors for the objective of joint cost and schedule minimization. 132
Figure 4.27 Example 4: Network configuration showing primary activities at each generation in the ESDTCT method........ 136
Figure 4.28 Example 4: Main effect plot for response vectors 140
Figure 4.29 Example 4: Tornado chart for activity relative importance towards trade-off. 141
Figure 4.30 Example 5: Main effect plot for response vectors 150
Figure 4.31 Example 5: Tornado chart for activity relative importance towards trade-off. 151
Figure 4.32. Numerical example 6: Risk adjusted network configuration. 154
Figure 4.33 Example 6: Network configuration showing primary activities at each generation in the ESDTCT method........ 154
Figure 4.34 Example 6: Main effect plot for response vectors 155

Figure 4.35 Example 6: Tornado chart for activity relative importance towards trade-off. 156
Figure 5.1 Typical and Non-typical activities in repetitive projects. 158
Figure 5.2 Repetitive project network logic. 161
Figure 5.3 Iterative shift of start dates to reduce crew interruption times. 163
Figure 5.4. Normalized frontier surface. 165
Figure 5.5. Flow chart of proposed crew work interruption time optimization. 168
Figure 5.6. Flow chart of RP-ESDTCT^{EXP} method. 175
Figure 5.7 Three-span concrete bridge illustration (Selinger 1980). 177
Figure 5.8 Example 7: Network configuration. 179

List of Tables

Table 2.1 Capabilities and limitations of existing optimization methods for repetitive class projects. 27
Table 2.2 Example of 24 full factorial experimental design. 51
Table 4.1 Example 1: Data input (adapted from Eshtehardian et al. 2009) 118
Table 4.2 Example 1: ESDTCTExp supreme chromosome results at each generation (for a JCL = 50%) 119
Table 4.3 Example 1: ESDTCT optimal chromosome for different JCL and different optimization objectives. 122
Table 4.4 Optimal results for certain TCTP adapted from Kalhor et al. (2011). 129
Table 4.5 Example 3: ESDTCTExp supreme chromosome results at each generation for cost minimization. 130
Table 4.6 Example 3: ESDTCTExp supreme chromosome results at each generation for schedule minimization. 130
Table 4.7 Example 3: ESDTCTExp supreme chromosome results at each generation for joint cost and schedule minimization. 131
Table 4.8 Example 4: Data input activities 1 to 46) (adapted from Bettemir 2009) 137
Table 4.9 Example 4: Data input activities 53 to 63) (adapted from Bettemir 2009) 138
Table 4.10 Example 4: ESDTCTExp supreme chromosome results at each generation under certain environment. 139
Table 4.11 Time and cost optimal solution using various meta-heuristic optimization methods (reported by Bettemir 2009). 142
Table 4.12 Example 5: Data input 145
Table 4.13 Example 5: ESDTCTExp supreme chromosome results at each generation under uncertain environment. 149
Table 4.14 Numerical example 6: Data input for discrete risk events. 154

Table 4.15 Example 6: ESDTCTExp supreme chromosome results at each generation (for a JCL = 50%) 155
Table 5.1 Example 7: Data input (adapted from El-Rayes (1997)) 180
Table 5.2 Example 7: RP-ESDTCTEXP supreme chromosome different optimization objectives under certain environment. 182
Table 5.3 Example 7: Comparative results for the schedule minimization under certain environment. 183
Table 5.4 Example 7: Comparative results for the cost minimization under certain environment. 184
Table 5.5 Example 8: Uncertainty multiplier factors for cost and productivity factors. 188
Table 5.6 Example 8: Uncertain Data input. 189
Table 5.7 Example 8: RP-ESDTCTEXP supreme chromosome different optimization objectives under Uncertain environment. 190
Table 5.8 Example 8: Impact of different JCL %. 191
Table 5.9 Example 9: Impact of different penalty schemes. 193

List of Acronyms

AACE	Association for the Advancement of Cost Engineering
ACO	Ant Colony Optimization algorithm
API	Application Program Interface
CCR	Continuous Conditional Risks
CDF	Cumulative Density Function
CI	Criticality Index
CII	Construction Industry Institute
CPM	Critical Path Method
CRA	Cost Risk Analysis
CRI	Cruciality Index
CSRA	Cost and Schedule Risk Analysis
DCR	Discrete Conditional Risks
DOE	Design of experiments
DYNASTRAT	Dynamic-Strategy
EFT	Early Finish Time
ESDTCT	Evolutionary Stochastic Discrete Time-Cost Trade-Off Method
$ESDTCT^{Exp}$	Evolutionary Stochastic Discrete Time-Cost Trade-Off Method – Experiment enumeration module
$ESDTCT^{Rand}$	Evolutionary Stochastic Discrete Time-Cost Trade-Off Method – Random search module
EST	Early Start Time
EVM	Earned Value Method
FC	Fixed Cost
FMPD	The Fuzzy Modus Ponens Deduction technique
GA	Genetic Algorithm optimization method
GAP	Google Apps Script
GERT	Graphical Evaluation and Review Technique
GS	Google sheets

IC	Indirect cost
ICr	Indirect cost rate
IRDT	Immediate reverse dominator tree
JCL	Joint Confidence Level
LFT	Late Finish time
LOB	Line Of Balance
LSD	Latin Square Design
LST	Late Start Time
MCS	Monte Carlo simulation
MOGA	multi-objective genetic algorithm
MSAT	Multi-Simulation Analysis Technique
MUD	Model for Uncertainty Determination
NA-ACO	non-dominated archiving ant colony approach
NP-Hard	non-deterministic polynomial-time hard
PDF	Probability Distribution Function
PERT	Program Evaluation and Review Technique
PMI	Project Management Institute
PNET	Probabilistic Network Evaluation Technique
r	Pearson's coefficient
RA	Risk Analysis
RCBD	Randomized Complete Block Design
RFL	Resource Flow Logic
RP-ESDTCT^{Exp}	Repetitive Project Evolutionary Stochastic Discrete Time-Cost Trade-Off Method – Experiment enumeration module
SA	Simulated Anneal optimization method
SMS	Summary Master Schedule
SI	Significance Index
SRA	Schedule Risk Analysis
SS	Start to Start relationship

SSI	Schedule Sensitivity Index
TDC	Time Dependent Cost
TF	Total Float
TIC	Time Independent Cost
UDF	User-Defined-Functions (a JavaScript function)
VERT	Venture Evaluation and Review Technique
VC	Variable Cost
VCr	Variable Cost rate
WBS	Work Breakdown Structure
WFL	Work Flow Logic

CHAPTER 1 : Introduction

1.1. General

On construction projects, the time-cost trade-off problems (TCTP) are often practiced for compressing the project duration and/or reducing expenses. This is often achieved by utilizing alternative resources referred to here as modes. The pool of alternative modes depends mostly on the project topology, flavour, and knowledge of the contractor. The alternative selections of modes represent discrete strategies that require the project manager’s decision. Figure 1.1 presents an illustration of discrete modes of an earthwork activity, where three different excavation modes are available; the first is using jackhammers, the second is utilizing rock breakers and the third is using more sophisticated equipment like surface miners. In general, the more expensive mode leads to a shorter duration. Accordingly, the direct cost of those activities increases, whereas the reduction of the activity’s duration contributes to less variable cost and the reduction in the overall project’s duration reduces the project indirect cost.. Compressing the project duration is often achieved by utilizing alternative resources for critical path activities such as more productive equipment and higher-skilled labour. Conversely, cost reduction is achieved by selecting cheap resources that may lead to longer durations. Furthermore, under an uncertain environment, those strategies need to account for varying levels of uncertainty in time and cost. Some modes are riskier than others and have a broad distribution. For example, modes involving innovative technology may be riskier than traditional modes of execution.

Figure 1.2 presents an illustration for the discrete modes showing the uncertainty of the cost and time parameters for each mode. The uncertainty of the cost and time parameters have been represented by a probability distribution function (PDF), as indicated by historical data. When the overlaps between the PDFs of modes are significant, the problem becomes hard to evaluate using deterministic methods and, therefore, necessitates the consideration of uncertainty and leads to the so-called stochastic time-cost trade-off problem (STCTP). Simulation techniques are powerful to solve such problems; however, a procedure is required to guide the analysis towards finding the optimal non-dominated solution.

Discrete risk events are possible diversions from the original plan due to unfavourable events or conditions that become an intrinsic part of a project. Such risks have a probability of occurrence, when occur may impact the overall project's duration and cost. Conversely, opportunities are favourable events that may have a positive impact on the overall project's duration and cost. Simulation techniques have been used in the assessment of those events in the so-called joint confidence level (JCL) risk analysis.

Construction projects are often classified as non-repetitive; however, those which include repetitive sections or units of work are classified as repetitive. Examples of repetitive construction projects are high-rise buildings, housing projects, highways, pipeline networks, and bridges. Scheduling of such projects needs to consider the crew work continuity constraints to allow for crew movement and minimize the interruptions and idle times. Repeated activities in the different units of a project can be further classified as typical on non-typical. Typical activities have the same scope of work while non-typical have variation in the scope of work.

In this thesis, we analyze the TCTP, crew work continuity, uncertainties, and risks simultaneously to identify the most optimum solution for a specified joint confidence level of time and cost.

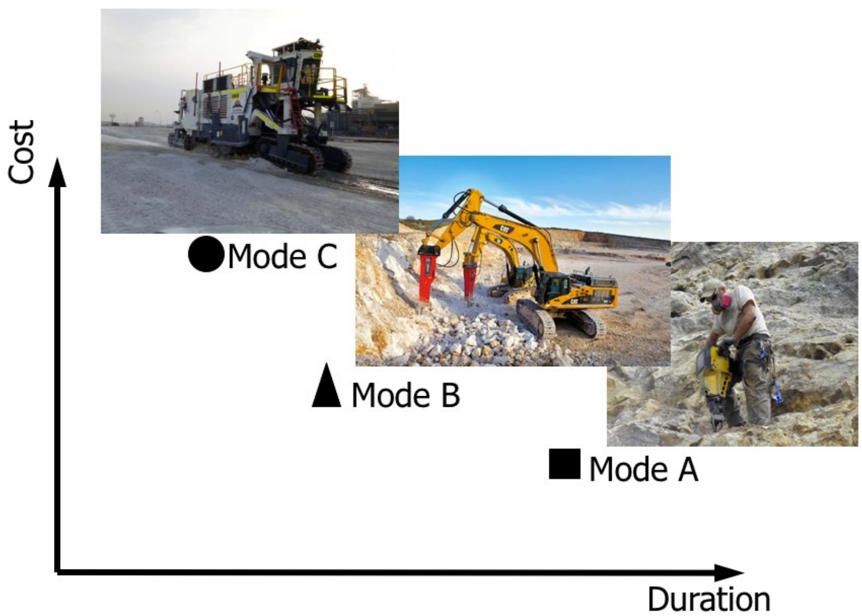


Figure 1.1 Typical discrete time-cost relationship of an activity.

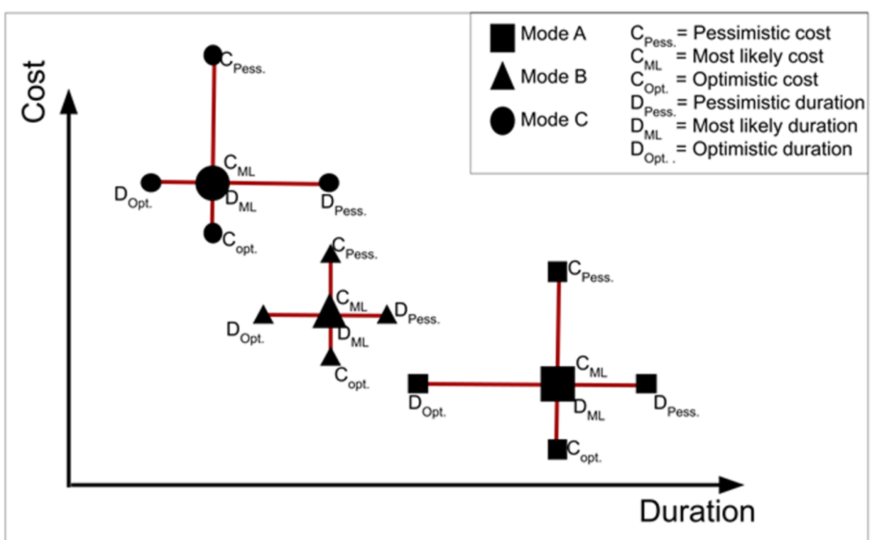


Figure 1.2 Typical discrete time-cost relationship of an activity with uncertainty in time and cost.

1.2. Problem Statement

TCTPs on construction projects is an essential aspect of the decision-making process; however, solving for this problem can be quite cumbersome, especially on large and complex projects. Solving the TCTP has been intensively researched since the 1960s; however, most of these methods are deterministic and consider a single cost and duration values for the activities involved. As well, these methods generally assume a linear relationship between time and cost and cannot adequately solve discrete time-cost trade-off problem (DTCTP) since the relationship between discrete modes may not be modelled. Solving the DTCTP has been categorized as a non-deterministic polynomial-time hard (NP-Hard) (De et al. 1995). The difficulty arises since, for each of the project activities, many modes may be available for execution using different resources such as crew skills, crew sizes or advanced equipment. Construction managers are then challenged with dealing with the classic combinatorial search to find the best selection of modes resulting in the minimization of cost and time to complete the project; For example, a small project modelled by only 15 activities each having 5 discrete modes, the number of possible combinations for a complete enumeration of modes to solve the problem is more than 30 billion combinations. This number increases exponentially with the increase in the number of activities and/or an increase in the number of alternative modes at each activity. Conventionally, the attributes of time and cost for each mode have been implicit in being deterministic with a crisp value, but historical data indicates that those attributes are uncertain and follow a probability distribution function. In the real world, the alternative modes, especially those that are new to the construction industry or new technologies, are not proven and come with higher uncertainty estimate of their time and cost; therefore, some modes for a single activity can be riskier than others and have a probability distribution function with a broader range. When the overlaps among the modes are significant, the problem becomes hard to solve using deterministic methods (Zheng et al. 2005). When incorporating the uncertainty of each mode into the equation, the difficulty of solving the DTCTP becomes exponentially larger. This is due to now becoming a stochastic problem where each mode attribute of time and cost is represented by a probability rather than a deterministic value. This uncertain environment increases the complexity dimension to the already NP-Hard combinatorial problem.

It is known that projects are often late and overrun their budgets. Many of the reasons for this phenomenon are attributed to risk and uncertainty. Every project by nature is unique and has a substantial amount of uncertainty. Nevertheless, those are frequently overlooked, and the project manager develops his estimates for completion times and cost budgets before any mature uncertainty assessment. These estimates then become binding in project contracts, and set the basis for target dates and budget, and are carried forward to set the penalties or incentives. Such overruns in project time and budget contribute to the diminished profits for both contractors and clients. Those contractors and clients are realizing those overruns and are becoming further sophisticated and demanding for a higher level of project risk analysis and optimization for cost and schedule.

The pace of technology is increasingly becoming hectic; this, by nature, is reflected in computer processing becoming more sophisticated and powerful. Clients, on the other hand, try to interact with this trending environment to be more educated than before; however, a mismatch exists between the technological advancements and the client's abilities and demand. Furthermore, the complexity of projects is continuously increasing that project managers cannot comprehend the outlook required to make suitable decisions.

The literature extensively addresses TCTP using continuous, most often assume a linear relationship between time and cost. Sometimes, this assumption makes the model somewhat impractical and cannot adequately solve discrete time-cost trade-off problem (DTCTP) since the relationship between discrete modes may not be represented by a mathematical equation let alone a linear equation; therefore, there is a needed to gain the benefits from integrating the DTCTP with the stochastic analysis of uncertainties and risks in the decision-making process; this becomes more important on complex projects.

A class of projects often include repetitive sections or units for repeat work. Many techniques have been proposed to solve the crew optimization problem on repetitive projects; however, most of the recent techniques have addressed crew work continuity but not effectively capture uncertainties in cost and durations and furthermore, modelling of non-typical activities. The limitations of existing techniques are discussed in Chapter 2.

1.3. Research scope and objectives

This thesis is concerned with analyzing the project cost and time using stochastic methods. Much of this research is devoted to the DTCTP, which is inadequately addressed in the literature. The identification of activity modes, risks, estimation of uncertainty and the qualitative risk analysis is not part of this study.

The main objective of this study is to provide project managers and decision-makers a holistic, integrated cost and time analysis tool that can answer enquiries such as, what are the optimal modes of construction that result in the multi-objective solution for schedule minimization, cost minimization or the optimal joint cost and schedule minimization that meets a targeted joint confidence level of both time and cost. To achieve this objective; this study introduces a new method that:

- Utilizes Monte Carlo simulation of the project to account for uncertainties and estimate the joint confidence level of both time and cost.
- Find the main effect of interchanging modes and their relationship to the optimization objective function.
- Integrate the discrete risk events and their probability of occurrence into the TCTP analysis to gain the benefits of concurrent assessments of uncertainties and risks on the decision-making process.
- Incorporate penalty and/or bonus schemes to account for costs from exceeding or meeting defined milestone completion dates and/or exceeding or meeting defined budget values.

- Application to both traditional and repetitive class projects, accounting for crew work continuity as a deciding factor to minimize the crew idle times on repetitive construction projects, and
- Accounts for non-typical activities in repetitive construction projects.

Answers to those objectives will increase the project manager's probability of making correct decisions that achieve successful project delivery on budget and on time. The results from the numerical examples and parametric study will offer valuable information and guidelines for potential users of the developed method.

The scope of this study is bounded by pre-defined construction modes for each activity, where each mode is a discrete option. The project manager needs to ensure that the details associated with the resources required in each mode are well estimated to achieve the time and cost for that mode, for example, the crew size, quantities of materials and number of equipment.

The developed method can be used at the planning phase of the project and at any intermediate stage when the project is requested to accelerate the completion date and reduce expenditures.

1.4. Thesis Organization

The thesis consists of five chapters and an appendix. Chapter 2 provides an overview of the different methods in the literature on cost and schedule risk analysis, TCTP analysis, repetitive project scheduling optimization methods and a brief overview of their advantages and disadvantages. Different sections are reviewed to building the theoretical background for the developed method is detailed in Chapter three. Chapter four outlines the developed method for the evolutionary stochastic discrete time-cost trade-off and presents the roadmap for the simulation model used to achieve the research objectives. The developed method is demonstrated using numerical examples drawn from the literature, and a comparison is performed with previous studies on TCTP using a hoist of different optimization techniques.

Chapter five outlines the extension of the developed method to accommodate repetitive construction projects demonstrated using numerical examples drawn from the literature for performance comparison against previous studies in this field.

Chapter six summarizes the contributions, limitations and conclusions of the research and proposed future research work. Appendix one includes the computer program for the developed method.

CHAPTER 2 : LITERATURE REVIEW

2.1. Introduction

In this chapter, the different methods in project scheduling, project cost and time risk analysis and TCTP optimization are reviewed. The literature review focuses on the advantages and drawbacks of each method. This chapter will also review the basic definitions and terms used in the construction project management and the statistical analysis methods.

2.2. What is a project?

A project is a set of activities with a start date, having specific objectives, specifications and conditions, with defined responsibilities between the multiple parties involved, it has a budget and a time frame for completion (Turner and Zolin 2012). Turner and Müller (2005), defined a project as an effort in which people, equipment, materials are systematized to assume a scope of certain conditions, within the cost and time constraints to accomplish quantitative and qualitative goals.

Jugdev and Müller (2005), found following their extensive research, that the definition of project success has evolved from focusing on completing a project within time, cost, and scope to expanding the focus by including the requirements of stakeholders. Every project has stakeholders , and the primary stakeholder is the customer who will benefit from the value-added by the delivered project. The second stakeholder is the project manager building the project for the customer.

On some projects, external stockholders may come at play including the public end-users.

The success of the project is generally measured in three dimensions what is known as the triple constraint, performance, schedule and cost as shown in Figure 2.1 (Shrnhur et al. 1997). However, the perspective of these three dimensions can sometimes be conceived differently by each stakeholder. The project manager may have a different take on what constitutes a successful project from that in the eyes of the customer; this is commonly due to the distinct roles, responsibilities and motivations that drive the people's behaviour.

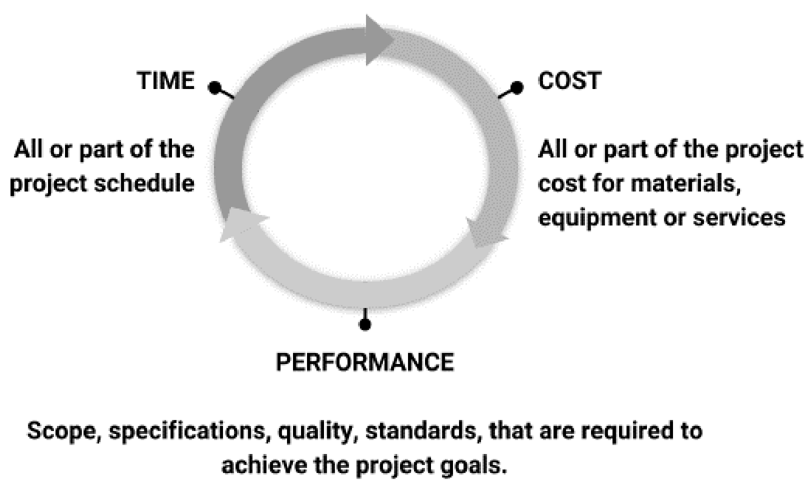


Figure 2.1 Project's Triple Constraints.

Baker et al. (2008) have considered a project successful when it satisfies the technical specification and/or performance criteria, and if key stakeholders have satisfaction concerning the project outcome. Among those stakeholders are the parent organization, the project team and end-users. Many studies have defined project success using two criteria, specifically project management criteria and product success criteria. Where project management is defined successful when meeting time, cost and quality targets, while a product is named successful when it meets the owner's strategic organizational targets, satisfaction of end-users and satisfaction of stakeholders where they relate to the product or future profits or improved business process performance (Belassi and Tukel 1996, and de Wit 1986). Based on the above definitions, the success criteria can be summarized in a fishbone diagram, where the success criteria are the effect driven by the cause of success factors, as shown in Figure 2.2.

In principle, a project is claimed successful when all perspectives and criteria have been successfully satisfied; this is often difficult to achieve due to various internal or external risks that may impact the project.

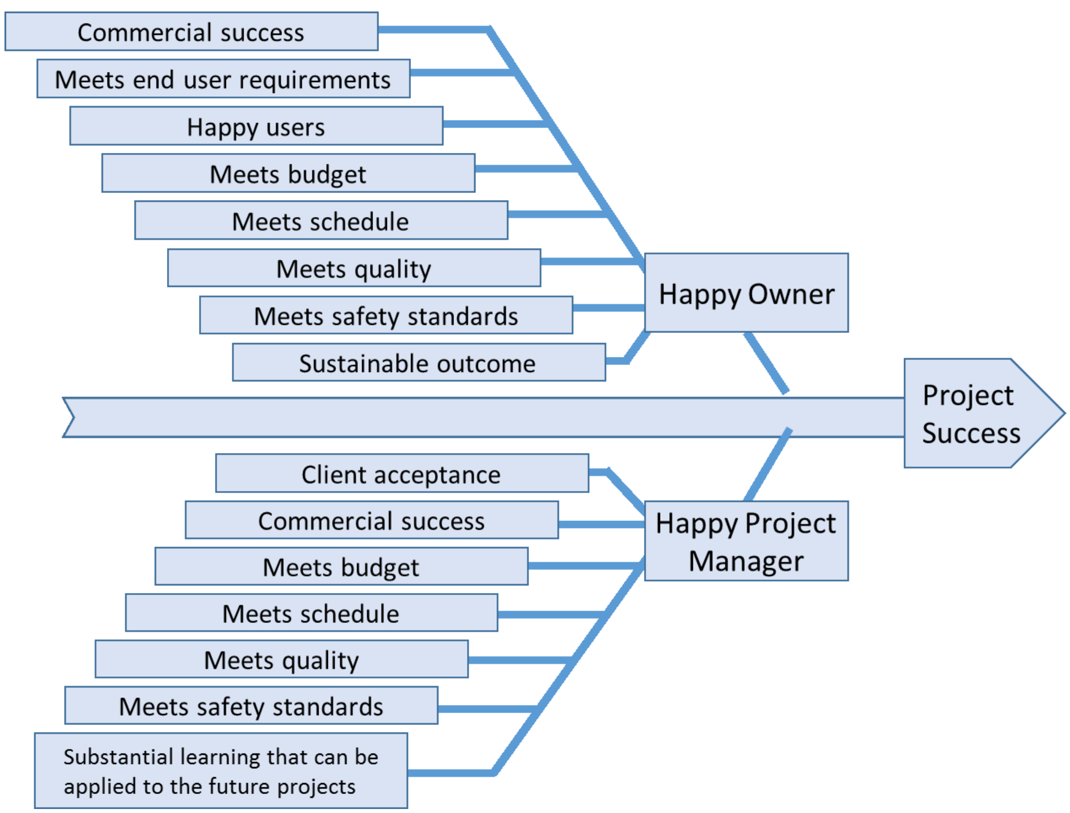


Figure 2.2 Project's Success Criteria.

2.3. What is risk?

Risk is the possible diversion of actual project performance from the original plan due to favourable or unfavourable events or conditions that becomes an intrinsic part of a project. Several definitions of risk are found in the literature evolving around, what it is, and what it involves, who owns the risk and its impact. Many studies (e.g. Williams 1995, Miles and Wilson 1998, and Mullins et al. 1999) define risk as a probability of occurrence for events that influence the success of an objective. Many researchers commonly perceived that risk is a negative attitude towards an objective. Miles and Wilson (1998) describe risks as success barriers. Webster (1997) defines risk as "the chance of injury, damage, or loss; dangerous chance; hazard." Papageorge (1988) describe risk as the possibility of loss or damage to people, assets, or interest. The Construction Industry Institute (CII 1989) describes risk as to the likelihood of a adverse outcome taking place. Many researchers, on the other hand, have suggested another meaning for positive effect risk. Jaafari (2001) describe risk as “exposure to loss/gain, or the probability of the occurrence of loss/gain multiplied by its respective magnitude.” PMBOK (2004) describe risk as “an uncertain event or condition that, if it occurs, has a positive or negative effect on a project's objectives.”; therefore, risk can have a negative or a positive impact on an objective.

Hertz and Thomas (1984), define risk as an absence of certainty of outcomes or consequences for a decision. Their argument introduced the way for other researchers to question the differences between uncertainty and risk. Mullins et al. (1999) describe risk as “the degree of uncertainty and potential loss that may follow from a given behaviour or set of behaviours." Yeo (1990) reasons that uncertainty and risk are the same and are frequently used interchangeably.

2.4. Probability theory

A probability of occurrence is used when future events may have more than one outcome. In a given context, only one of these events will happen yet we cannot state which ahead of time. Such circumstances are called stochastic, rather than deterministic circumstances where the future result is predicted beforehand. The likelihood of a future event is a prediction of its possibility of happening measured as a value in the interim somewhere more than 0% and less than 100%, where an event that is practically sure to happen has a likelihood near 100%, while a remarkably improbable event has a likelihood close 0%.

The outcome of a future event is represented by the value of a function, where the function is generated by random variables. The probability measure for all associated values is a probability distribution. The probability distributions have a significant effect on the results of a risk assessment model; therefore, consideration needs to be provided for the choice of probability distributions to be utilized, particularly to their parameters.

Several probability functions work particularly valuable in construction projects. Some of the popular distribution functions adopted for an activity duration in construction projects include Uniform, Normal, Triangular, and Beta distributions. Similarly, for cost modelling, Triangular, Lognormal and Beta distributions are the favoured.

The Normal distribution function has assumed a focal part in probability and statistics. The numerical estimations of numerous different events can be displayed with this function. Experimental confirmation demonstrates that the Normal distribution function gives decent representation to numerous cases such as estimations on weight, length, instrument errors and rate of return in economics.

Many studies examined utilizing various distribution functions. For instance, Fente et al. (2000) demonstrated that construction projects fell in a beta function and introduced a technique for estimating the beta parameter values. Wilson et al. (1982) examined the triangular versus beta functions; his findings concluded no noteworthy contrasts in results. Another case introduced by Touran (1997) investigated the utilization of Normal and lognormal distributions; his statistical analysis concluded no noteworthy differences in the average project durations.

Back et al. (2000) assert that beta and triangular distributions are more fit for modelling historical cost values. The authors favoured the triangular distribution to avoid the complexity of determining the beta parameters.

2.5. Time-cost trade-off in construction projects

The diversity of available means, resources, materials, technologies and methods for executing an activity provides alternative implementation modes that can be assigned to individual activities in construction projects. In other words, an activity may be executed using alternative materials or with different forms of resources. For example, methods and materials used in earth backfilling activities and dewatering methods for deep excavations. Selection amongst those available modes requires the project manager's decision. The choice of mode may have a different duration for that activity and different direct and indirect costs. Time-cost trade-off problem (TCTP) analysis on construction projects is often used to explore available alternative modes in an attempt to find the optimal non-dominated solution that yields the least overall project's duration and/or total cost. Solving this problem can be quite cumbersome, especially on large and complex projects. Solving the TCTP has been intensively researched since the 1960s. Early methods had a weakness in solving a discrete time-cost trade-off problem (DTCTP) since the relationship between discrete modes may not be represented by a mathematical equation, let alone a linear equation. Instead, the solution space for a DTCTP is a factorial design for the combinatorial nature between the different activities and each of their assigned modes. Hence, the complete enumeration in the solution space of the problem exponentially increases for medium and large size problems. These trade-off problems are known as non-deterministic polynomial-time hard (NP-Hard) (De et al. 1995).

There exist many techniques for the TCTP. Those can be categorized into three areas, namely, mathematical programming, heuristics and simulation. Examples of mathematical techniques that have used linear programming were introduced by Kelley(1961). Patterson and Huber (1974), and Sakellaropoulos and Chassiakos (2004) used the integer programming technique to tackle discrete relationships; however, integer programming requires an extensive computational effort and fail to be applied on large project networks. Elmaghraby (1993) and De et al. (1995) introduced dynamic programming techniques for special class projects that can be decomposed into parallel or a series of sub-projects.

Heuristic methods have dominated much of the research efforts because of their ability to find a good solution within a reasonable computational effort; however, such methods lack mathematical rigour and finding the optimal non-dominated solution is not guaranteed. Examples of heuristic techniques include the structural stiffness method introduced by Moselhi (1993), a branch-and-bound approach introduced by Demeulemeester et al. (1996), and an optimization method using the maximal flow theory introduced by Liu and Rahbar (2004). Some of the more recent researches that have implemented heuristic methods in TCTP and decision-making problems are Chiu and Chiu (2005), Vanhoucke and Debels (2005), and Pendharkar (2015).

Most of the studies in the literature on DTCTP techniques have assumed a certain environment and have represented a single cost and duration value for each activity mode. In real world, those alternative modes, especially those that are new to the construction industry or new technologies, are not proven and come with higher uncertainty estimate of their time and cost; therefore, some modes for a single activity can be riskier than others and have a probability distribution function with a larger range. When the overlaps among the modes are significant, the problem becomes hard to solve using deterministic methods. Solving for the stochastic network has been proven to be a hard problem, as the problem would include an extensive amount of computations for numerical integration from a complete enumeration of all viable solutions; therefore, simulation techniques have become more prevalent in recent researches; however, most of those studies have been focused on either the project duration or costs while fewer studies have been conducted to optimize the project in a multi-objective approach for simultaneous cost and schedule optimization. Furthermore, even fewer of those researches tried to tackle the consideration of uncertainties in TCTP. Zheng et al. (2005), Moselhi and Roofigari (2013), Moselhi and Alshibani (2013), and Kalhor et al. (2011) proposed a fuzzy method for a stochastic TCT problem. Fuzzy set theory dominated those researches due to its capacity to produce the results in fewer runs; on the other hand, Monte Carlo simulation techniques can provide more statistical information exploring the uncertainties of activity durations and costs.

2.6. Discrete-event optimization approaches

Computer simulation is a powerful mean for the evaluation of complex problems, especially when the possible solution space expands. This evaluation is performed as responses to “What If” scenarios. Recently, this has been further extended to provide additional answers to “How to” questions. The "What if" scenarios requires collecting the responses of a problem for a set of variables. While the "How to" questions, attempts to find the optimum values of the variables to maximize or minimize the response. Many studies have been carried out towards establishing procedures for simulation optimization of complex problems that are influenced by several variables. The past twenty-five years have been a rapid development in this field mainly due to the increased computational speed, and the decrease in modelling costs; however, in the construction management industry the literature indicates a lack of integration between the simulation models, optimization and accommodation of program change. Ultimately the objective of modelling construction projects is to better the decision-making process to maximize or minimize selected variables to achieve the desired response.

Solving optimization problems have been carried out using several diverse algorithms, from conventional to recent metaheuristic algorithms (Riley 2013). These algorithms are classified as deterministic or stochastic. Mainly, the conventional algorithms are deterministic, and further can be classified into gradient/ derivative such as the Newton-Raphson algorithm or gradient-free/derivative-free algorithms such as the Hooke-Jeeves pattern search and Nelder-Mead downhill simplex (Yang 2011). The stochastic algorithms, on the other hand, can be heuristic or metaheuristic algorithms. The heuristic algorithms are based on trial and error. The metaheuristic algorithms are based on a partial search algorithm that may provide an approximate solution; those are used primarily when complex, incomplete or imperfect information of the problem exists or when the computation capacity is limited (Manda et al. 2012). Metaheuristics algorithms cannot guarantee to find the optimal solution for complex problems; however, good near-optimal solutions are found faster than other optimization algorithms (Blum and Roli 2003). More details on Metaheuristics algorithms can be found in (Glover and Kochenberger 2006, Talbi 2009).

Meta-heuristic optimization algorithms begin by defining a function for a set of independent variables to obtain a global minimum or maximum for the objective function output. The function is solved by assigning initial values for the variables, which then evolve by iteration of those variables to determine various solutions from the search space. The iterative process is run until a stopping criterion is reached. This stopping criterion can be the number of iterations, the execution time elapsed, reaching the data storage capacity limits, etc. Depending on the function complexity and the total size of the search space, the meta-heuristic approach may only visit some of the possible solutions; therefore, there is no guarantee that the optimal solution found by this approach is the best solution (de Freitas et al. 2010).

The general process steps of a typical meta-heuristic optimization method can be summarized, as shown in Figure 2.3.

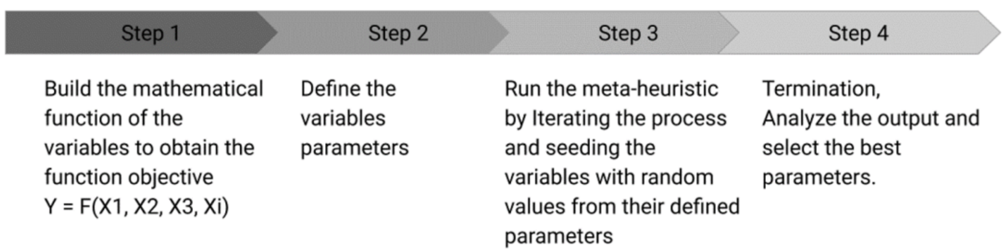


Figure 2.3 General process steps for Meta-heuristic optimization.

There are many types of meta-heuristic algorithms; all use a way to trade-off local search from global search. Yang (2011) concluded in his study that randomization offers a good global search moving away from local solution searches. Blum and Roli (2003) categorized meta-heuristic algorithms into two concepts, ‘intensification and diversification’; where the diversification concept tends to explore the global search space, while intensification focus on local search space when finding a good solution in that space. A balance between the two concepts is important to achieve an efficient and fast algorithm. Too much intensification yields the process to be trapped in local optima, which may be far from the global optima. Too much diversification, on the other hand, may result in a hard to converge system and thus, slow performance (Yang 2011). The means to achieve this balance is the main difference between the various meta-heuristic algorithms. The meta-heuristic algorithms are used to tackle optimization problems where traditional simple methods have failed to be effective. These algorithms are becoming more recognized and applied in many complex fields in machine learning and artificial intelligence. The main advantages of meta-heuristic algorithms can be summarized in the following points (Gholizadeh and Barati 2012):

- Ease to understand.
- Wide-ranging applicability in any field where the problem can be formulated as a function of variables.
- They can be hybridized with other traditional optimization methods.
- They can solve large problems faster than traditional methods.

The method, on the other hand, has disadvantages that can be summarized as:

- Does not promise to find the most optimal solution.
- It lacks a strong mathematical foundation.
- The results accuracy is highly negatively correlated with the number of variables and the fine tune of the variable's parameters; the more refined parameters, the less the optimization performance.
- Different optimization solutions may result when repeated for the same problem with the same initial condition settings.

However, the algorithm can be justified despite its disadvantages by the following points (Talbi 2009):

- Failure of exact techniques to solve optimization due to the complexity of the problem
- Slow performance of exact techniques due to the large number of variables and the fine tune of the variable's parameters.

A brief description of the research pertaining to some of the most popular discrete event simulation optimization algorithms is summarized below with a focus on their advantages and disadvantages reported in the literature.

The Tabu search, created by Fred W. Glover in 1986, is a meta-heuristic algorithm. In this algorithm, the feasible solution space is explored by taking a potential possible solution and moving to its best candidate neighbours. Movement occurs despite degradation in the objective function. Tested solutions are used in Intensification and diversifications strategies to advance the exploration path. The algorithm is used for solution spaces that are characterized by local optima and is only used for discrete event optimization models. Few studies in the literature have compared the accuracy and precision of its results (Glover 1977, Lopez-Garcia et al. 1999).

The Pattern Search method originally published by Hooke and Jeeves in 1961, also known as direct search, is based on stepwise moves towards the direction where the objective function is increasingly improving until failure to find a better solution surrounding the current search point; hence it must be the optimum point. The search is made using two types of movements, the first is exploratory, where the direction and size of the steps are made in small amounts and observing whether the objective function value betters or worsens. When a better solution is found the method then increments the steps in a pattern movement at the improvement direction, this is repeated until failure to find a better solution, at this stage, the search moves backwards to the last memorized solution from which exploratory steps are further performed until stopped by a user-defined convergence test or a tolerance criterion. This heuristic search method is easy to understand and implement; however, its main disadvantage appears in complex functions were an improving direction is not able to be found even when one exists.

The Genetic Algorithm (GA) is a heuristic search optimization method inspired by Charles Darwin’s theory of evolution of genetic selections. The method was first proposed by Holland in 1975 to find good solutions to complex problems fast and relatively at low computational cost. The algorithm adopts medical terms such as “Chromosomes”, where a chromosome is a possible solution that has a set of parameters for the problem variables (or genes). The combination of the best genes is then called the optimal chromosome (or the optimal non-dominated solution). The algorithm starts from randomly generated chromosomes selected from the total population of possible chromosomes. The process is then iterated to produce a new generation by mutating the elite genes defined for their fitness towards generating a better solution for the problem objective. The process continues to evolve until terminated when exceeding a user-defined maximum number of generations, or when a satisfactory fitness level is reached. Figure 2.4 shows the process steps for the traditional GA. Many studies in the literature have concluded that the method is good at identifying the global optimum solution in problems having multiple local optima; however, those studies have also criticized the method for being hard to analyze and difficult to design for complex problems; furthermore, many studies have recommended the need for more theoretical work to test the accuracy of its results. (Michalewiez 1994, Goldberg 1994). The main recognition of the method is crossing the boundaries of medical and mathematical research communities to achieve a fruitful combination. The success of GAs has led to the advancement of wider optimization approaches such as Neural Networks, Ant Colony Optimization, Particle Swarm Optimization and Artificial Immunology (Engelbrecht 2007).

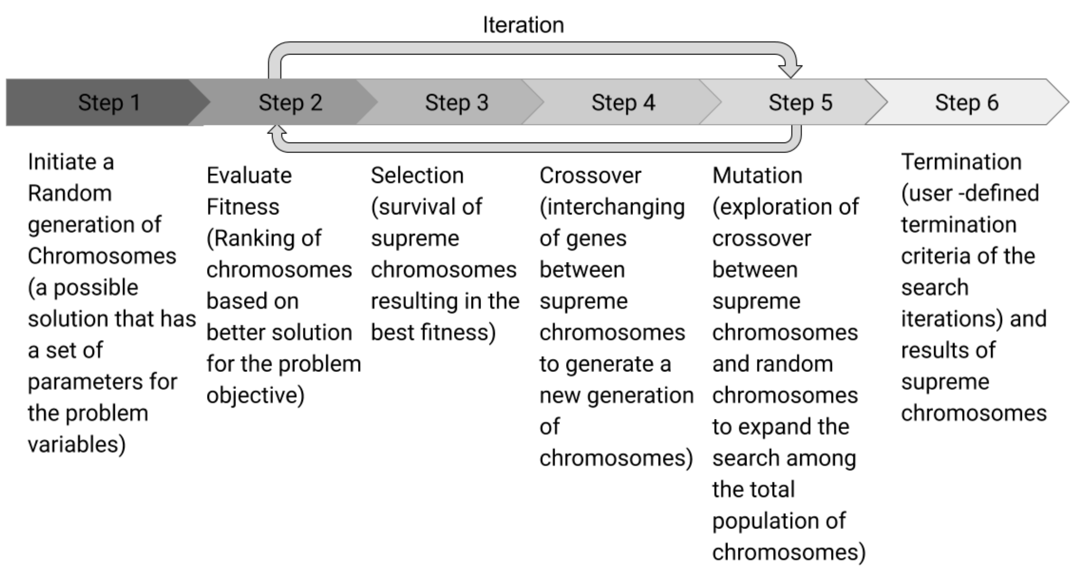


Figure 2.4 Process steps for Genetic Algorithm optimization.

The Simulated Anneal (SA) method, originally proposed by Kirkpatrick et al. in 1983, is inspired by annealing in metallurgy where the metal is gradually cooled until reaching a state of low energy where they are in solid and strong state. The method is a random search in the possible solution space of the objective function.

The search process goes by gradually changing a parameter in resemblance to the temperature. As the temperature decreases in each iteration, the random search becomes likely to move in the direction of better solutions that result in better objective value of the problem. The optimum solution is then found by terminating the process when a pre-defined state is achieved or when the user-defined maximum number of iterations is reached. Many studies have concluded that the SA method is efficient in avoiding getting stuck in local optima. The method is also characterized to have a relatively low computational time at each iteration; however, the large number of iterations may be needed depending on the rate of temperature change and could significantly increase the total time to find the optimal solution (Liu, 1999, Zolfaghari and Liang 1998, Bailey et al.1997, Aarts et al. 1997)

Ant Colony Optimization algorithm (ACO), originally proposed by Dorigo in his Ph.D. thesis in 1992, is a Meta-heuristic optimization technique inspired by the behaviour of ants seeking a path between their colony and source of food. The method starts with applying generations of artificial ants to search for a good solution. A good solution is defined by the shortest path that is discovered via pheromone trails. Each ant moves on a random path and dispose of pheromone, the more pheromone found on a path increases the probability of being followed.

Hybrid Techniques have also been studied by many researchers; those techniques are built on a combination of established techniques such as those described in the previous sections or others. This is generally done to cartel each of their desired features and better the computational cost and speed and the search for the near-optimal solution.

Examples of recent research including stochastic analysis are those by Eshtehardian et al. (2009) and Zahraie and Tavakolan 2009 using fuzzy set theory and genetic algorithms, Gutjahr et al. (2000) using a stochastic model based on a branch-and-bound approach, Aghaie and Mokhtari (2009) and Kalhor et al. (2011) applying a modified ant colony optimization and Monte Carlo simulation, Yang et al. (2013) using the particle swarm optimization approach, and Ke (2014) using a stochastic simulation hybridized model with genetic algorithms.

2.7. Repetitive Construction Projects Optimization

Repetitive construction projects are a class of projects which include repetitive sections or units of work. Common repetitive construction projects are high-rise buildings, housing projects, highways, pipeline networks, and bridges. Scheduling of such projects needs to allow for crew movement and consider crew work continuity constraints. Repeated activities in the different units of a project can be typical or non-typical. Typical activities have the same work amounts; in contrast, non-typical have variation in the work amounts and, therefore, different duration and cost values for different repetitive units. A considerable body of the literature exists for the scheduling optimization of repetitive class projects. These methods differ in their unique characteristics and capabilities; whether they account for the two types of activities referred to above, consider uncertainty, optimized crew formations, allow for interruptions and whether they consider time and/or cost in the optimization process. Table 2.1 summarizes the capabilities and limitations of commonly referred to methods in that domain.

Table 2.1 Capabilities and limitations of existing optimization methods for repetitive class projects.

Reference	**Optimization method**	**Characteristics and capabilities**				
		Activities types	**Crew work continuity**	**Optimization**	**Time-cost trade-off**	**Uncertainty**
Selinger (1980)	Dynamic programming	Typical activities only	No idle time is allowed for any crew	Least schedule Cost is not considered	Not considered	Deterministic
Russell and Caselton (1988)	Dynamic programming	Typical and non-typical actives	Predetermined (User specified) possibilities of interruptions to work continuity	Least schedule Cost is not considered	Not considered	Deterministic
Reda (1990)	Linear programming	Typical activities only	No idle time is allowed for any crew	Least cost schedule	A continuous linear equation is used between time cost trade-off at the activity level.	Deterministic
Moselhi and El-Rayes (1993)	Dynamic programming	Typical and non-typical actives	No idle time is allowed for any crew	Least schedule accounting for cost as a decision variable in the optimization process	Trade-off for discrete modes of time-cost pairs is considered	Deterministic
El-Rayes and Moselhi (2001)	Dynamic programming	Typical and non-typical actives	System calculated set of interruption vectors are considered during scheduling	Least schedule	Not considered	Deterministic
Hegazy and Wassef (2001)	Genetic algorithms	Typical activities only	System calculated set of interruption vectors are considered during scheduling	Least cost	A continuous linear equation is used between time cost trade-off at the activity level.	Deterministic
Moselhi and Hassanein (2003)	Dynamic programming coupled with heuristic rules	Typical and non-typical actives	No idle time is allowed for any crew	Time, cost or both	Not considered	Deterministic

Reference	Optimization method	Characteristics and capabilities				
		Activities types	Crew work continuity	Optimization	Time-cost trade-off	Uncertainty
Nassar (2005)	Genetic algorithms	Typical and non-typical actives	System calculated interruption vectors are considered during scheduling	Least schedule Cost is not considered	Not considered	Deterministic
Liu et al. (2005)	Simulation and genetic algorithm	Typical and non-typical actives	System calculated set of interruption vectors are considered during scheduling	Time, cost or both	Trade-off for discrete modes of time-cost pairs is considered	Deterministic
Ipsilandis (2007)	Linear programming	Typical activities only	System calculated set of interruption vectors are considered during scheduling	Least schedule accounting for cost as a decision variable in the optimization process	Not considered	Deterministic
Senouci and AlDerham (2008)	Genetic algorithms	Typical activities only	No idle time is allowed for any crew	Time, cost or both	Trade-off for discrete modes of time-cost pairs is considered	Deterministic
Srisuwanrat et al. (2008)	Simulation	Typical and non-typical actives	Allow for crew work interruptions using simulation	Least schedule Cost is not considered	Not considered	Probabilistic schedule only.
Hyari et al. (2009)	Genetic algorithms	Typical and non-typical actives	System calculated set of interruption vectors are considered during scheduling	Time, cost or both	Trade-off for discrete modes of time-cost pairs is considered	Deterministic
Long and Ohsato (2009)	Genetic algorithms	Typical and non-typical actives	System calculated set of interruption vectors are considered during scheduling	Time, cost or both	Trade-off based on linear, non-linear, continuous, or discrete relationship models of direct cost relationship to the duration	Deterministic
Maravas and Pantouvakis (2010)	Dynamic programming with fuzzy set theory	Typical activities only	No idle time is allowed for any crew	Least schedule Cost is not considered	Not considered	Schedule uncertainties only.

Reference	Optimization method	Characteristics and capabilities				
		Activities types	**Crew work continuity**	**Optimization**	**Time-cost trade-off**	**Uncertainty**
Bakry et al. (2016)	Dynamic programming with fuzzy set theory	Typical and non-typical actives	No idle time is allowed for any crew	Time, cost or both	Trade-off for discrete modes of time-cost pairs is considered	Fuzzy cost and schedule
Zou et al. (2018)	Dynamic programming	Typical activities only	Allow for crew work interruptions using an automated procedure to reduce the number of interruptions	Time, cost or both	Not considered	Deterministic
Salama and Moselhi (2019)	Dynamic programming with fuzzy set theory	Typical and non-typical actives	System calculated set of interruption vectors are considered during scheduling	Time, cost or both	Trade-off for discrete modes of time-cost pairs is considered	Fuzzy cost and schedule
Developed method	Simulation and optimization	Typical and non-typical actives	Allow for crew work interruptions	Time, cost or both	Trade-off for discrete modes of time-cost pairs is considered	Probabilistic schedule and cost.

The literature in this domain reveals that there are many methods proposed to solve the crew optimization problem on repetitive projects; however, most of the recent techniques have addressed crew work continuity but did not effectively account for uncertainties in cost and durations. Most of the early methods considered a single optimization objective of either time or cost, while, more resent methods accounted for a multi-objective optimization. The literature review in this domain also showed that only a few studies accounted for uncertainties into the optimization function. Those limitations have been addressed in the developed method.

2.8. Existing forecasting methods

Forecasting is crucial for decision making on construction projects. There are two general types of forecasting, deterministic and probabilistic approaches. Deterministic forecasting is a network of tasks connected with dependencies that describe the sequence of work to be performed and the total cost and duration of the project. While probabilistic forecasting are networks with all the elements of a deterministic plan, but the cost and durations of the tasks are modelled with uncertainties.

2.8.1. Deterministic forecasting methods

Critical Path Method (CPM) is a network method originally developed by DuPont in the 1950s to assist in planning, forecasting and control of projects. The method requires careful planning, scheduling and management of interconnected activities. The method initially recognizes critical and non-critical activities and aims to improve work efficiency. By emphasizing the efforts on critical activities, the total project duration is shortened. Many introductions of the CPM can be found (see Oberlender 2000, Winter 2003, Meredith and Mantel 2009, Woolf 2012, Del Pico 2013).

The critical path method performs a forward and backward pass for the project network for calculating the theoretical early dates, and late dates, disregarding limitations of resources, given the durations, relationship logic, lags, and constraints. The resulting early and late start and finish dates are the times where the tasks can be scheduled. The CPM is built on a work breakdown structure

(WBS). The WBS divides the project into discrete tasks. The essential part of CPM is the knowledge that some activities cannot start until others finish. Thus, a sequence of activities can be identified in which each stage must be completed until the next stage can begin. These activities are called sequential activities. Other activities may not depend on the completion of other activities and can be conducted at any time; these activities are called parallel tasks.

The CPM approach has several underlying assumptions. It assumes that a project can be divided into distinguishable activities; the activities are then arranged on a timescale. Each activity is allocated a duration. The activities may also be loaded with resources, such as personnel, cost, equipment, facilities and support services. The best way to present those activities is to draw them as bars over a horizontal time axis what is known as a Gantt chart (Clark et al. 1922).

The CPM is the most popular method used on projects today for the following reasons:

- CPM enables project managers to display the activities graphically and identify the sequence of activities that are required to be completed.
- Identifying the critical path for the project provides project managers focus areas to pave the way for successful completion of critical activities.
- The CPM schedules are updated periodically and hence; the critical paths may change throughout the life of a project due to internal and external issues upsetting the project. the updated CPM schedule offers to identify problem areas where further attention is required due to those changes.

The shortcomings inherent in the CPM have been criticized by many researchers (Cottrell 1999, Lu and AbouRizk 2000). Several disadvantages and assumption surrounding the CPM can be summarized:

- One of the fundamental assumptions of the CPM techniques is the project team's capability to foresee the scope and estimate the duration and costs of each activity; unfortunately, practical experience showed that it is often beyond control (Knoke and Garza 2003).
- The CPM is deterministic in nature and estimating the duration of tasks is most often based on historical information that is conserved within an organization or found in reliable external sources.
- The overall duration of a project is calculated with the assumption that tasks will progress according to plan irrespective of past performance.

The traditional CPM is not the best means for repetitive projects, and the shortcomings of the CPM has led to the reappearance of interest in linear scheduling techniques (Handa and Barcia 1986). Previous work on repetitive project scheduling and optimization is further detailed in section 2.7.

The Earned Value Method (EVM) was developed in the 1960s and has proven its use in projects forecasting differing in size and complexity (Abba 2000). The Project Management Institute (2013) recommends the EVM and is mandated by NASA, DoD, and world-leading contractors. The EVM integrates the cost, schedule and technical parameters for a project. In the planning phase, the baseline can be established by distributing the cost over the duration of the associated work activity. As the activity is performed, the cost value of the activity is earned. The calculated earned value is then used to analyze the cost and schedule variations to enable forecasting the remaining project cost and schedule. The forecasting methods using the traditional EVM have been criticized for unrealistic forecasting of the project status at completion. Short (1993) discussed the undesirable performance of the EVM when non-critical tasks cause the schedule variances and when tasks are executed in a different sequence as opposed to the baseline. To enhance the EVM, several forecasting equations were developed by previous researchers. Lipke (1999) introduced a method for managing the cost and schedule reserves using a cost ratio and a schedule ratio. Later Lipke (2003) presented the Earned Schedule (ES) as a new technique to calculate the schedule variance and schedule performance index. The author concluded that the method provides reliable forecasting results. Henderson (2003, 2004) and later Vandevoorde and Vanhoucke (2006) studied the practicality and consistency of the ES method. In their studies, they concluded that the method is the most suitable way to estimate the project completion time. Moselhi (2011) recommended using the critical activities solely to establish the project baseline curve for the EVM and subsequently for calculating the schedule variances and indices. In his study, the author demonstrated that the method provides realistic forecasting results, particularly in forecasting project durations.

2.8.2. Probabilistic forecasting methods

An extensive amount of studies has been focused on introducing probabilistic capabilities to cost and scheduling forecasting techniques. Program Evaluation and Review Technique (PERT) was one of the first attempts to probabilistic activity cost and durations. Then came the Monte Carlo simulation (MCS) to handle the merge-event bias and account for stochastic task duration (Van Slyke 1963). The Graphical Evaluation and Review Technique (GERT) (Pritsker 1966) was developed to model the uncertainty of precedence activities by allowing probabilistic routing and feedback loops. The Venture Evaluation and Review Technique (VERT) (Moeller and Digman 1981) assesses the risks involved in time, cost, and performance of projects. The Model for Uncertainty Determination (MUD) (Carr 1979) and the Dynamic-Strategy (DYNASTRAT) (Morua Padilla 1986) handles interdependencies of activity durations and the evaluation of the project progress.

PERT was introduced on the Polaris missile program by the U.S. Navy in 1958. The Navy pulled in the Lockheed Aircraft Corporation and the Booz, Allen, and Hamilton management-consulting firms to develop this technique to deal with the variations in the project cost and time. PERT incorporates the element of uncertainty by adding the requirement of three estimates, unlike the CPM, where each activity has a single estimated duration. The three estimates are optimistic, most likely, and pessimistic. PERT uses the critical path to evaluate the total project duration under uncertain activity durations; the expected total project duration is then calculated as the summation of expected durations for all the critical path activities, the computations are made with the assumption that each activity duration follows a beta probability distribution (Sasieni 1986). PERT can calculate the probability of completing a project within a given time frame and the variability in the project completion time. For the purpose of computing the expected total cost, all the activities in the project network are considered to calculate the probability of completing a project within a given cost frame and the variability in the total project cost. Littlefield and Randolph (1991) questioned the validity of the mathematical assumptions used in estimating PERT activity time; they concluded that the estimation process and communication between the project manager and those involved in estimating activity times is more important than the mathematics used for the estimate. Cottrell (1999) proposed a modified technique for estimating the PERT activity durations by using only two durations (most likely and pessimistic) to generate a normal distribution. According to Cottrell, his simplified procedure reduces the level of effort required by conventional PERT because only two estimates, rather than three, are required for each activity. He also concluded that the three-point estimation adds little to the accuracy of deterministic equivalents of stochastic activity times in distributions that are not highly skewed. The PERT method typically underestimates the actual project duration because of the procedure limitation considering one single critical path, where several paths can turn out to be critical due to fluctuations (Diaz and Hadipriono 1993, Hendrickson and Au 2000, and Trietsch et al. 2012). The method also assumes that the task's duration are uncorrelated random variables (Elmaghraby 1977). To overcome this limitation; several research efforts were made. Ang and Tang (1975) developed a Probabilistic Network Evaluation Technique (PNET) for predicting the project duration by representing several paths that are highly correlated by the longest path in the group. The PNET method accuracy was criticized for being varying from "liberal or conservative" that depends on the threshold value of the correlation coefficient, which was left subjective to the scheduler (Diaz and Hadipriono 1993).

Kim (2010) introduced a probabilistic approach to the earned value method by combining the ES method introduced by Lipke (2003) with the Kalman filter algorithm to provide a probabilistic prediction of project duration at completion.

The Monte Carlo simulation (MCS) is the most common technique used to simulate uncertainty. The method simulates a model containing several variables many times by randomly selecting a value from a probability distribution function (PDF) for each variable. The PDF of the resulting overall values is then determined by the iterations until a statistically significant result is obtained. The term Monte Carlo was presented in World War II used as a code name for the atomic bomb program simulation (Eckhardt 1987). The method is used to improve the quantification of risks impacting cost and schedule on construction projects. Project managers can then quantify a schedule contingency, budget contingency, or both to manage the problems that could unfavourably disturb the project. MCS is only as good as the model used for the simulation and the data provided. If the model is weak, the simulation may result in miss-leading information. Newton (1991) reviewed cost modelling in construction; in his research, he explained that the MCS has evolved as a popular tool for the management of construction projects. The MCS of project schedules has been widely studied by many researchers mainly due to the relative complexity in the scheduling techniques for the calculation of the total project duration with probabilistic durations (Lee 2005, Lu and AbouRizk 2000). Williams (2004) explained a drawback on the Monte Carlo simulations stating that the method "simply carry through each iteration unintelligently, assuming no management action". In real life, management will take measures to recover delays; such measures may be a change in the execution sequence, methods and tools. Graves (2001) discussed the types of probability distributions to simulate the activity duration. He also proposed using open-ended distributions explaining that the closed-ended distribution denies the possibility of an activity finishing with a shorter duration than the minimum duration or taking longer than the maximum duration. In practice, issues could arise that were never expected and impact activity durations. The open-ended distributions allow the activity to possibly exceed the activity duration boundaries, resulting in a more realistic simulation. Bennet and Ormerod (1984) summarized the advantages asserted for the method as it offers a boundless capacity for modelling costs and schedules of construction projects.

The MCS can offer the project manager an understanding of factors that are most important and how they interact. Further, The MCS iterative process demonstrates that uncertainties significantly add to the project costs and schedule, and basic additive of the bill of quantities can result in underestimation of costs. Nevertheless, the traditional MCS method has also been described as incomplete for its application on construction projects due to the difficulties to accommodate interdependency (or correlations) between variables (Perry and Hayes 1985). Moselhi and Dimitrov (1993) used real-life data of building projects and proposed a simplified MCS method to quantify the effect of correlation between the discrete cost elements on the estimated cost of construction projects. Many useful references can be found that outline the MCS theory and implementation such as (Clemen et al. 2004, Bedford and Cooke 2001, Raftery 1994, Hartford and Baecher 2004, Vose 2008, Barraza 2010, Kroese et al. 2013).

Another line of research was made to model the uncertainty in the activity durations using a fuzzy set-based theory. Fuzzy set theory was developed by Zadeh (1965) in an effort to provide a basis to handle the uncertainty that is non-statistical in nature. On construction projects, some activities may have been rarely or never done before, and hence no statistical data exists; therefore, the activity durations are described using fuzzy variables through expert opinions. This method uses linguistic terms such as “good” or “bad” that are subjectively quantified (Wu and Hadipriono 1994). Prade (1979) was the first researcher to propose the application of fuzzy set theory in scheduling problems. The Fuzzy set theory has also been used to model the relationship between the activity durations and the factors affecting those durations such as site conditions, weather, labour performance, etc. (AbouRizk and Sawhney 1993, Wu and Hadipriono 1994). Chanas and Kamburowski (1981) developed a fuzzy version of PERT which they named FPERT. The method models the project completion time in the form of a fuzzy set in the time-space. McCahon and Lee (1988) presented a new methodology to calculate the fuzzy completion project time. AbouRizk and Sawhney (1993) developed a computer system called “SIDES” that takes into consideration user subjective factors affecting the duration of activities and their likelihood of occurrence and consequences on the activity. The system then computes the mean, variance and the shape parameters of the beta distribution. Wu and

Hadipriono, (1994) used a technique called “the fuzzy modus ponens deduction” (FMPD) to assess the impacts of factors on activity durations by using the most likely activity durations to calculate the optimistic and pessimistic durations using the angular fuzzy concept; the calculated durations are then used in probabilistic scheduling methods such as PERT (Hadipriono and Sun 1990). Lorterapong and Moselhi (1996) presented a scheduling method built on the fuzzy set theory; the method integrates several techniques for the representation of inexact duration of activities, the calculation of scheduling parameters, and the interpretation of the fuzzy results generated. Chen (2007) proposed an approach for networks with fuzzy activity durations based on the linear programming solutions to the critical path analysis. Yakhchali and Ghodsypour (2010) presented a method for calculating the likely values of the early and late start of an activity using approximate fuzzy durations. Naeni et al. (2011) introduced a fuzzy-based earned value model to incorporate uncertainty in the analysis of the earned value indices and estimating the duration and the cost at completion. The main advantage of the fuzzy set theory as compared to the MCS that it requires less computations; however, it requires an extensive knowledge for the model formulation. Furthermore, the advancement of computing power lessened the fuzzy set theory over the MCS.

2.9. Risk analysis

Projects in the construction industry often overrun their budget and schedule baseline, to some degree in light of the fact that uncertainties and risks are unaccounted for in the cost and schedule estimations. This problem has been addressed in many researches and practices, often by using the MCS.

During project execution, the project is rarely executed as it was planned due to uncertainty resulting from human subjective estimate errors and variability arising from unexpected events. Choi and Mahadevan (2008) arrest that risk evaluations are hard because of the data required for variables. They argue that if historical data is available, it can facilitate a practical application.

Every project is unique and may have different risks impacting cost and schedule. Common examples of such risks in the construction industry are summarized in a fishbone diagram in Figure 2.5. For an in-depth definition of those risks see (Brenner et al. 1996, Eyers 2001, Nasir et al. 2003, Hulett 2015).

The Schedule Risk Analysis (SRA) translates the risk of the various activity durations to offer sensitivity analysis for those activities and evaluate any possible impact caused by the uncertainty on the overall project's duration. Schedule risk analysis provides information to project managers that aids in determining the probability of accomplishing the project objectives on time, determination and monitoring of the schedule reserve, the likelihood of potential problems and the identification of critical and sensitive activities and priorities. Several analysis approaches are present in the literature for performing the schedule risk analysis; those can be grouped into two methods, Qualitative and Quantitative risk analysis. A qualitative risk analysis focuses on quality by prioritizing the risks with an established scale. Risks are ranked according to their impact significance and to the probability of occurrence. The quantitative risk analysis, on the other hand, is a quantitative rating and statistics analysis of risks based on probabilistic inputs. The most common quantitative risk analysis is modelling, and simulation typically performed using the MCS technique. (Neil and Diekmann 1989).

Conventional methods of cost estimating are deterministic and neglect to account for the uncertainties experienced in the real world. The Cost Risk Analysis (CRA) accounts for the uncertainty in cost estimations with the use of probability distribution profiles. Each cost element with a possible variability is represented as a random value; the summation of the random values is then the total cost of the project. The MCS is used to perform iterations representing different scenarios; the probability distribution of the overall outcome value is calculated through the iterations until a statistically significant result is obtained.

Using the three-point estimates to represent the risk profile is common in the construction industry in the absence of historical data. Many studies have concluded that the most recommended distributions for modelling cost are the Normal, lognormal, triangular, beta, and uniform distributions. (Touran and Suphot 1997, Wall 1997, Back et al. 2000, Yang 2005, Hulett 2002).

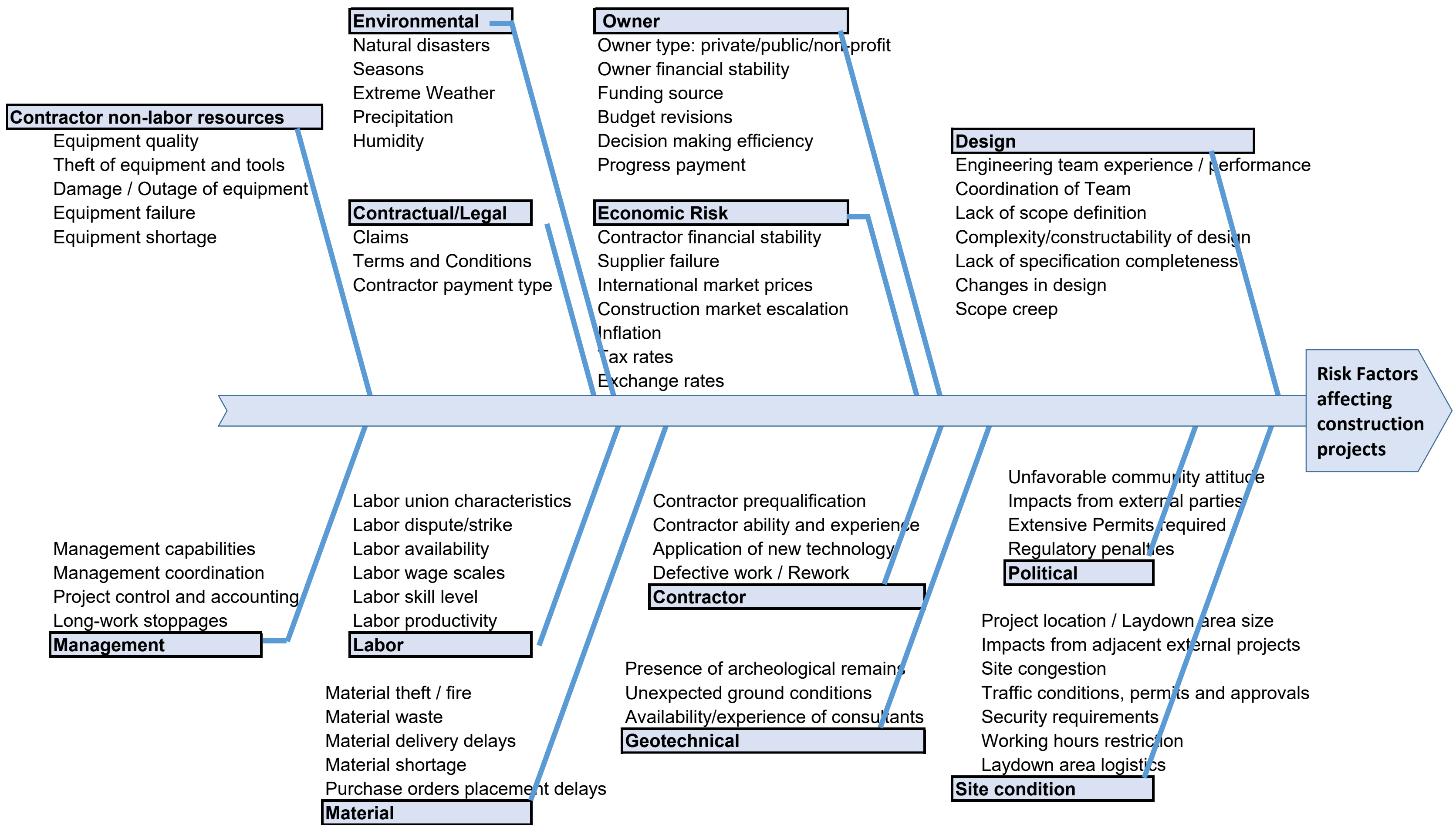


Figure 2.5 Risks affecting construction projects.

The impact of correlation among cost components on the total cost variance of a construction project may have a significant impact on the simulation results. Conventional construction cost estimates are developed with the assumption of independence to avoid the trouble of modelling correlation. (Curran 1990); however, when positive dependence exists, the simulation output is an underestimation of the variable's variance. An approximation method to avoid complex correlation modelling is to roll up the cost elements by grouping correlated items into a single item; however, this approach might produce complications regarding the estimation for large and complex projects (Chau 1995). Among the commonly used measures of dependence are the Pearson correlation coefficient and Spearman's correlation coefficient, where the coefficient represents the degree of association between two cost items. More details can be found (see Chau 1995, Touran and Suphot 1997, Yang 2005).

Cost-Schedule integration is the concurrent consideration of uncertainty arising from the schedule and cost and their combined effect to recognize the risk impacting a project. The analysis offers a betterment for the development of cost and schedule baselines as oppose to independent analysis and can also be adopted for measuring the project performance. Hulett (2002) recommends taking the cost values and apportioning them to the schedule activities to enable an integrated analysis of cost and schedule.

Cost and duration are inter-related; however, in most practices, this relationship is usually not accounted for due to the breakdown structures that differ when developing the cost and schedule of a project. (Poh and Tah 2006, Feng et al.

2010). Isidore and Back (2001 and 2002) introduced the multi-simulation analysis technique (MSAT) for the simultaneous simulation of cost and schedule. The authors joint the results of independent cost and schedule simulation results; however, they do not take into consideration the combined variation in cost and schedule.

The Joint Cost and Schedule Confidence Level (JCL) is an integrated uncertainty analysis of cost and schedule developed by NASA (Coonce 2009). The method integrates the analysis of a project cost, schedule, risk, and uncertainty. The objective result is to estimate the project's cost and schedule for a targeted probability confidence level. For example, a 50% confidence represents the probability for mutually estimating 50% confidence in cost and 50% confidence in schedule. This is unlike estimating independently the 50% cost confidence or estimating the 50% schedule confidence. For large projects, the difference between the 50% cost confidence independent of schedule and joint 50% cost and schedule confidence can be significant. The National Aeronautics and Space Administration (NASA) Cost Estimating Handbook (Version 4.0) recommends the implementation of the JCL, in fact, NASA JCL policy (NPR 7120.5E), mandates projects to perform a JCL with a recommended 70% probability for estimating both cost and schedule (Hulett et al. 2011, Hoffpauir 2015). The methodology for developing a JCL starts with building a probabilistic cost-loaded schedule and systematically integrating cost, schedule, and risk. The method facilitates the establishment of expectations and probabilities of meeting those expectations. It also offers a holistic view to achieve cost and schedule goals and the determination of contingency for cost and schedule for the selected confidence level.

2.10. Project costs

A fundamental relationship exists between a project schedule and its cost. This relationship is vital for the project manager to estimate the project cost; however, it is often hard to quantify this relationship and model. The correlation between the project cost growth and schedule growth is mostly obvious, and many studies have concluded that schedule growth generally leads to growth in cost. It is for this reason that the integrated project schedule is required to correspond to cost estimates to satisfy enough resources assigned to activities to achieve completion within the expected duration. As stated previously, the relationship between the project cost and schedule are often hard to quantify. Typically, a large project takes more time to complete than a small project and most often, accelerating a project schedule can lead to large cost overruns. Figure 2.6 shows the relationship between project cost and schedule based on 50 completed NASA programs. The data analysis shows that when the project is cost-driven, cost and schedule will be more highly correlated, and when the project is schedule driven, the correlation is lower and possibly even negative.

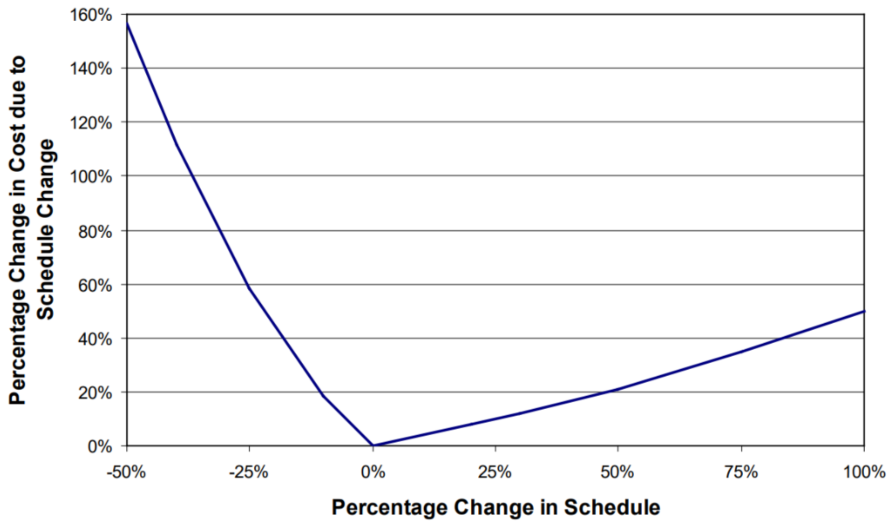


Figure 2.6 Cost and Schedule Interrelationships. Source: Smart, C.B., 2007 NASA Cost Symposium.

In construction projects, costs can be classified as time-independent costs (TIC) and time-dependent costs (TDC). TICs may include material and plant permanent equipment; such costs are referred to in this study as fixed costs (FC). TDC, on the other hand, maybe split into two types, those associated with a project activity or a group of activities according to the project WBS are referred to in this study as variable costs (VC) while costs that are not associated with an activity but rather to the overall project are referred as indirect costs (IC). TDCs are applied as a rate per unit of time and are determined by the length of the activity or the project duration; examples of such costs are those associated with tools and equipment rentals, labour wages, land rentals, insurance, finance expenses and office expenses. The TDC increases with the increased project duration; however, depending on the project complexity, this relationship is not always linear such as in the case where the size of the staff is not constant over the life of the project or the case were the loan interest has complex payment terms.

Other type of costs may be present depending on the contract provisions; such costs that are considered in this study are penalties for schedule delays or bonus payments for early completion. Typical language used in the construction project contract for a liquidated damage clause reads (Halpin et al. 2017):

> ***"Liquidated Damages*** *In case of failure on the part of the Contractor to complete the work within the time fixed in the contract or any extensions thereof, the Contractor shall pay the owner as liquidated damages the sum of $3000 for each calendar day of delay until the work is completed or accepted."*

However, in real-life cases, the amount of the liquidated damage cannot be specified arbitrarily and must be justified for actual damage incurred. Therefore, uncertainty in the application of those amounts may also exist.

Some owners have exercised including an arbitrary high penalty amount, instead of liquidated damages, to scare the contractor into completion, however, a legal precedent was established in the form that when an owner desires to specify a penalty for overrun, a bonus must be offered in the same amount for early completion (Tyler 1994). Figure 2.7 shows the breakdown of the project costs that are considered in this study.

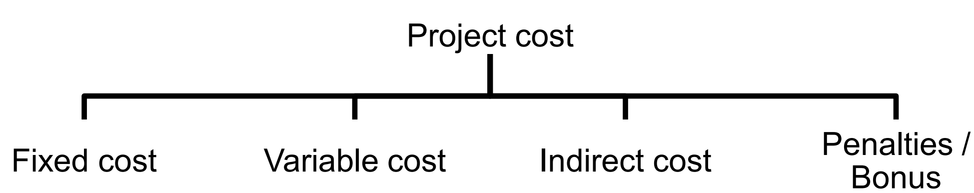


Figure 2.7 Types of project costs

2.11. Design of Experiments

Design of experiments (DOE) is a method used to find the relationship between factors (modes) affecting a system and the response of that system to identify the factors most influential in optimizing the response as shown in Figure 2.8. DOE is a statistical experimental method originally developed by Fisher in 1926. Later Box and Wilson (1951) applied the method on industrial experiments producing the response surface designs; however, it was only in the 1980s with the work of Taguchi (1986) that made statistical experimental design popular and stressed its significance for quality improvement.

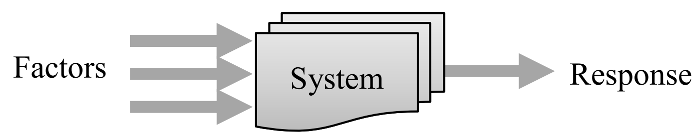


Figure 2.8 Factors and Responses.

The first step to perform a DOE is to define the objective function of the problem and select the variables otherwise known as factors or parameters. The factors can be either discrete qualitative variables or discrete quantitative variables. The number of discrete values for each factor is called levels. Early studies on the DOE assumed the same number of levels for each factor, mainly two levels denoting the high and low range of each factor as [-1, 1]; this was mostly to minimize the problem complexity and reduce the computational requirements; however, the number of levels can be different for each factor.

Different techniques have been developed all evolving around reducing the costs associated with performing the required number of experiments to determine the optimal response of the objective function. The following sections briefly highlights the most popular DOE techniques advantages and disadvantages.

The Randomized Complete Block Design (RCBD) is a DOE technique originally developed by Bernstein (1927), the technique is based on blocking of factors by focusing on a controllable factor (primary factor) and makes it more relevant; while the other factors are considered nuisance factors that may affect the measured result but are not of primary interest. A blocking technique is used to keeps the nuisance factors constant in value. A batch of experiments is made where the primary factor adopts all its possible values. The randomized block design performs a batch of experiments for every possible combination of the nuisance factors. For example, if an objective function has k controllable factors $\{x_1, x_1, \dots\dots, x_K\}$ where one of them is of primary importance. Let each factor have a number of levels $\{L_1, L_1, \dots\dots, L_k\}$. Let N be the number of replications for each experiment, then the sample size for the number of experiments needed to complete an RCBD is $N = (L_1 \times L_2 \times \dots \times L_k) \cdot$ The RCBD technique has an advantage of complete flexibility in the number of variables and blocks. It is a relatively easy statistical method and allows a calculation of unbiased error for specific variables; however, the technique has been criticized for being impractical for large and complex problems and an increased error in the interactions between factors over other DOE methods.

The Latin Square Design (LSD) is built on the RCBD technique with the aim of reducing the total number of experimental runs required. The technique involves selecting a single experiment in each block without confounding the significance of the primary factor. The LSD necessitates certain conditions for applicability, namely the controllable factors k = 3, where x_1 and x_2 are the nuisance factors, and x_3 is the primary factor. The second condition is all factors must have the same number of levels. The LSD is also known as the Graeco-Latin square for $k = 4$, and Hyper-Graeco-Latin square for $k = 5$ and had remained nameless in the literature for any $k > 5$. The advantage of the LSD is its relatively inexpensive in terms of sample size; however, there are disadvantages to the technique due to its pre-requisite conditions and is less flexible compared to the RCBD.

The full factorial experimental design; also known as the Brute force approach; is the most popular technique originally developed in the 19th century by John Bennet Lawes and Joseph Henry Gilbert. In a simple case for a two-level full factorial where there are k factors and two discrete possible level values per factor, the number of experiments is then all possible combinations of the factors. This method does not distinguish between nuisance and primary factors. Typically, the two levels are denoted as ("h" or +1) for high values and ("l" or -1) for low values. For example, if an objective function has 4 factors $\{x_1, x_2, x_3 \text{ and } x_4\}$ each factor has two discrete levels (L_1, L_2) of +1 and -1, then the sample size for the number of experiments needed to complete a full factorial design is $N = L^k$ $(2^4 = 16)$, Table 2.2 shows the 2^4 full factorial experimental design matrix with all possible combinations of the factors.

The full factorial experimental designs can also be extended to the general case to account for a different number of levels for each factor, also known as a mixed-level design. The sample size for the number of experiments needed to complete a mixed-level full factorial design is $N = \prod_{i=1}^{k} L_i$. The full factorial experimental designs have the advantage of being efficient to estimate the interactions that may be present between the factors at several levels on other factors, however, as the number of factors and levels increase the sample size grows exponentially which in turn increases the experimental cost and makes it difficult to interpret; especially when interactions between the factors exist. Those disadvantages have led the way for the development of fractional factorial designs where only a subset of the sample size is examined.

Table 2.2 Example of 2^4 full factorial experimental design.

Experiment runs	Factors			
	X1	X2	X3	X4
1	-1	-1	-1	-1
2	1	-1	-1	-1
3	-1	1	-1	-1
4	1	1	-1	-1
5	-1	-1	1	-1
6	1	-1	1	-1
7	-1	1	1	-1
8	1	1	1	-1
9	-1	-1	-1	1
10	1	-1	-1	1
11	-1	1	-1	1
12	1	1	-1	1
13	-1	-1	1	1
14	1	-1	1	1
15	-1	1	1	1
16	1	1	1	1

Fractional factorial designs were introduced by Finney (1943). In this approach, the sample size is a subset from the full factorial design and is selected at one half or one quarter, and so forth. The sample size is appropriately selected to have the same number of samples for each of its levels. Various strategies are used to ensure an appropriate choice of the sample size. The main advantage is the reduced sample size while doing so, the problem losses on the possibility to distinguish between the main effects and interaction effects between the factors. Several fractional factorial design methods exists in the literature amongst which, some of the most popular are the Homogenous fractional design method used when a large number of factors is required to be explored, The mixed-level fractional design is used when many factors are needed to be assessed for their main effects and when higher-level interactions are considered negligible, the Box-hunter fractional design is used with more than two-level factors or mixed level factors, Plackett-Burman is an efficient screening fractional design used when all interactions are considered negligible, and Taguchi fractional design estimates the main effects concurrently with minimizing variances. The Latin square is a fractional factorial design in the case where one factor is of interest while others are blocking factors.

The Random DOE techniques, also known as “space-filling” techniques depend on uniformly filling the solution space. Such techniques are generally used for creating a response surface especially for large and complex problems; however, due to the fact that space-filling techniques are not level-based, the main effects are not as complete as is the case with full factorial designs. This technique, however, has a drawback in the sense that some samples may be clustered near to each other and fail to fill the solution space uniformly.

Many DOE methods are available to choose from, but there is no best choice. The selection is dependent on the complexity of the problem at hand. The key items to consider are:

- the number of experiments which can be afforded in terms of time and costs required to perform the experiments,
- the number of factors involved in an experiment,
- the number of levels for each factor, and
- the ultimate objective of the experimental design.

Cheap techniques may result in inaccurate results; however, they could jointly be used with other techniques as a preliminary study for estimating the main effects. A good introduction and examples of the various techniques described above can be found in Cavazzuti (2013).

2.12. Findings of the Literature Review

Literature was reviewed in search of existing methods and techniques that perform the processes of scheduling, cost and schedule risk analysis, and optimization for time-cost trade-off in repetitive and non-repetitive class construction projects. The existing scheduling techniques were categorized as deterministic and stochastic techniques. The principles, capabilities and limitations associated with those techniques were highlighted. From the literature review, the following facts can be established:

1. It is a fact that deterministic scheduling is dominantly used due to simplicity; however, it is not as realistic as probabilistic scheduling and could result in misleading information when it comes to setting the basis for target contractual dates and thus the leading cause why projects are often late.

2. Conventional methods of cost estimating are deterministic and neglect to account for the uncertainties experienced in the real world
3. Most established probabilistic techniques; while accounting for estimate uncertainties, fail to account for probable discrete risk events. This ignorance is, in general, a main factor to the cause why projects often overrun their budget.
4. Most established probabilistic forecasting techniques perform either cost or schedule analysis independently in a siloed approach. Only a few of the established techniques account for the joint cost and schedule analysis thus by default neglecting the cost impact of time uncertainty.
5. The fuzzy set theory is developed based on sound theory and provides a tempting approach for stochastic forecasting. When compared to the MCS, it requires less computational efforts; however, it requires an extensive knowledge for the model formulation and interpretations of results which may lack in construction management professionals in the real world. Furthermore, the advancement of computing power lessened the application of fuzzy set theory over the MCS approach.
6. Deterministic time-cost trade-off analysis has dominated the studies in the literature, while only a few and most recent of the studies tackle stochastic analysis; however, it is common sense that different trade-off strategies come with different levels of uncertainties that need to be analyzed simultaneously.

CHAPTER 3 : THEORETICAL BACKGROUND

3.1. Introduction

The purpose of this chapter is to review different sections appropriate for building the theoretical background for the proposed research method.

3.2. Theoretical critical path method

The Critical Path Method (CPM) is a deterministic technique to calculate the earliest time for project completion, by use of dependencies between tasks having a deterministic duration. The method is described using the following steps:

1. Identification of the individual project activity in accordance with the scope of work, size and complexity of the project. This is generally done by the use of a work breakdown structure.
2. Identification of the dependency between activities. This requires listing all predecessors, and the estimation of the lead time.
3. Estimate activities durations. A deterministic duration is estimated usually with the help of historical data of production rates or subject matter expert judgement.
4. Calculate the critical path, which is defined as the longest path that drives the project completion.

Some of the basic definitions used in the method are:

- duration (d): The amount of time the activity takes from start to finish.

- Early Start Time (EST): The earliest possible point in time on which a task can start.
- Early Finish Time (EFT): The earliest possible point in time on which a task can finish.
- Late Start Time (LST): The latest possible point in time on which a task can start.
- Late Finish Time (LFT): The latest possible point in time on which a task can finish.
- Total Float (TF): The total amount of time the activity can slip from its early start without delaying the project finish date.
- Free Float (FF): The amount of time the activity can slip from its early start without delaying the early start of its immediately following activity.
- Critical Path (CP): The sequence of schedule activities that determines the duration of the project. It is the longest path throughout the project.
- Lag: The amount of time delaying the succeeding task.
- Lead (or negative Lag): The amount of time accelerating the succeeding task. (Note that the use of Leeds is uncommon in the construction industry).
- Immediate predecessor: is the activity that must be completed immediately before a given activity can begin.
- Forward pass: is a calculation method to determine the EST and EFT of each activity based on the network logic, the duration of the activities and any imposed constraints and lags. The calculation starts at the start activity or

milestone and works each activity forward through the network, considering all its predecessor activities.

- Backward pass: is a calculation method to calculate the LST and LFT of each activity without delaying the overall project's completion date and any of the imposed constraints.
- Logic links: represent the logical work relationship for an activity and its immediate predecessor. An activity may have several logical links to other project activities. There are four types of relationships used in combination with the lags or leads:
    1. Finish-to-Start (FS), where an activity cannot start until after the finish of the preceding activity.
    2. Start-to-Start (SS), where an activity start time is delayed until after the start of the preceding activity.
    3. Finish-to-Finish (FF), where the activity completion is delayed until the completion of all predecessor activity.
    4. Start-to-Finish (SF), where the activity completion is constrained by the start of the preceding activity.

Figure 3.1 shows an illustration of the four types of relationships.

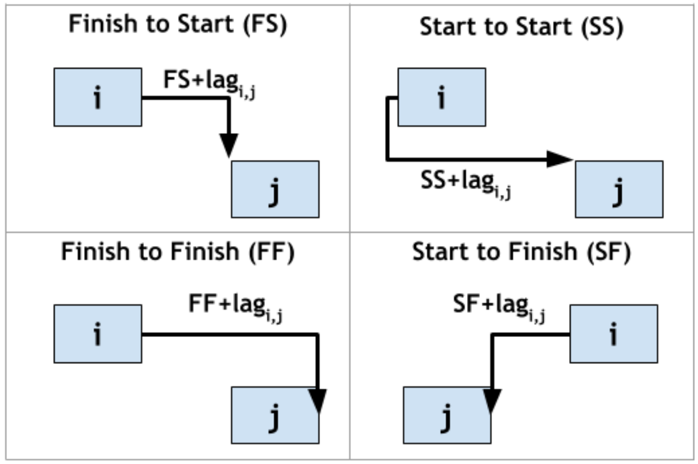


Figure 3.1 Type of logic relationships between the project activities.

The forward pass calculations can be formulated for an activity i having an immediate precedence relationship to activity j using equations (3-1) and (3-2) where j $\in$ the set of predecessor activities $\mathrm{P_i}$. Equations (3-3) computes the overall project's duration D.

$$EST_i = \max_{\forall\, j \in P_i} \begin{cases} EFT_j + lag_{i,j} + 1 & , \; logic_{i,j} = Finish\ to\ Start \\ EST_j + lag_{i,j} & , \; logic_{i,j} = Start\ to\ Start \\ EFT_j + \ lag_{(i,j)} - d_i + 1 & , \; logic_{i,j} = Finish\ to\ Finish \\ EST_j + \ Lag_{(i,j)} - d_i & , \; logic_{i,j} = Start\ to\ Finish \end{cases} \qquad 3\text{-}1$$

$$EFT_i = EST_i + d_i - 1 \qquad 3\text{-}2$$

$$D = \max_{\forall\, i}\{EST_i + d_i - 1\} \qquad 3\text{-}3$$

After completing the forward pass for the entire network, the backward pass calculations are performed to determine the latest start and finish times for each activity without delaying the completion of the overall project. The backward pass calculation starts at the finish activity and logically works back to the beginning. The backward pass calculations can be formulated for an activity i having an immediate successor relationship to activity j using equations (3-4) and (3-5) where j $\in$ the set of successor activities S_i.

$$LFT_i = \min_{\forall j \in S_i} \begin{cases} LST_j - lag_{i,j} - 1 & , \ logic_{i,j} = Finish\ to\ Start \\ LST_j - lag_{(i,j)} + d_i - 1 & , \ logic_{i,j} = Start\ to\ Start \\ LFT_j - lag_{(i,j)} & , \ logic_{i,j} = Finish\ to\ Finish \\ LFT_j - lag_{(i,j)} + 1 & , \ logic_{i,j} = Start\ to\ Finish \end{cases} \quad 3\text{-}4$$

$$LST_i = LFT_i - d_i + 1 \quad 3\text{-}5$$

Once the forward and backward pass calculations are complete, the TF for each activity can be calculated using equation (3-6).

$$TF_i = (LFT_i - EST_i) - d_i + 1 \quad 3\text{-}6$$

A critical activity is one with zero TF and the critical path is determined by the path that runs through the critical activities.

The value of ‘1’ that is added or deducted in equations (3-1 to 3-6) is needed to adjust for the ‘beginning and end of day’ to account for the continuum time calculations as opposed to the calculation using numbers.

3.3. Modelling type of costs

As described in section 2.10, four types of project costs are considered in the present study; those are fixed, variable, indirect and penalty/bonus costs. In preparation for modelling such costs, the estimates should be naked of any contingency as contingency estimates will be the results obtained from the simulation for the specified confidence level. The uncertainty in fixed cost is modelled using equation (3-7):

$$E[FC_i] = [FC_i]_{ML} * \emptyset \quad 3\text{-}7$$

Where $E[FC_i]$ is the expected value of fixed cost for the activity i, $[FC_i]_{\mathrm{ML}}$ is the activity most likely estimate of the fixed cost, and $\emptyset = f(p)$ is a multiplier obtained from the assigned uncertainty PDF using a random probability number p.

To account for the uncertainty in the variable cost, equation (3-8) is used:

$$E[VC_i] = [VCr_i]_{ML} * \partial * [d_i]_{ML} * \gamma \qquad \text{3-8}$$

Where $E[VC_i]$ is the expected value of variable cost for the activity i, $[VCr_i]_{\mathrm{ML}}$ is the most likely estimated variable cost rate per unit of time for an activity or a group of activities under a common work breakdown structure, $[d_i]_{\mathrm{ML}}$ is the most likely estimated value for the activity duration, and $\partial = f(p)$ is the multiplier obtained from the assigned uncertainty PDF using a random probability number p.

$\gamma = g(p)$ is the activity duration multiplier obtained from the assigned duration uncertainty PDF using a random probability number p.

Similarly, to account for the uncertainty in the Indirect cost the following equation is used:

$$E[IC] = [ICr]_{ML} * \omega * E[D] \qquad \text{3-9}$$

Where $E[IC]$ is the expected value of indirect cost for the entire project duration (D), $[\mathrm{ICR}]_{\mathrm{ML}}$ is the project most likely estimated indirect cost rate per unit of time, and $\omega = f(p)$ is the multiplier obtained from the assigned ICr uncertainty PDF using a random probability number p. Figure 3.2 shows an illustration for a linear cumulative distribution of IC cost for a work activity.

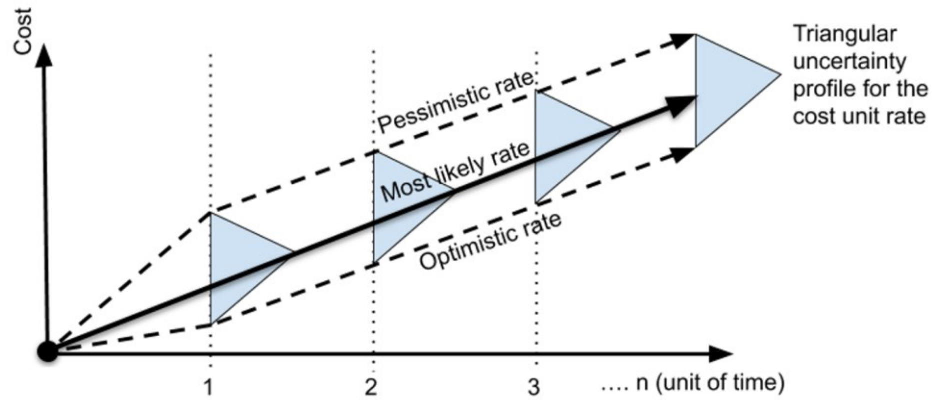


Figure 3.2 Uncertainty modelling of variable and indirect costs.

Depending on the contract provisions, penalties are often used by clients in order to help achieve a certain deadline on schedule-driven projects or meet a targeted budget on budget-driven projects. Such cost only exists if pre-defined conditions are satisfied. Penalty cost (PC) may be applied by the client to cover costs stemming from liquidated damages. Such costs may also be self-applied by the contractor to cover for double handling and storage cost of materials due to self-generated delays in site readiness. Similarly, there is an opportunity for earning bonus costs (BC) for meeting contract milestones or target budgets.

The PC and BC is a form of risk and opportunities (loss / profit); they are treated using the same modelling technique but differ in their sign (+/-) and in their triggering condition. In the case of a penalty, cost accumulates and increases the project total cost estimate, where a negative cost value is accumulated in the case of BC where the earned costs reduce the overall project's cost estimate.

PC and BC can be continuous, applied as a cost rate on a time unit bases, or discrete and applied as a one-time cost. Uncertainty in those costs may also exist, such as uncertainty in the cost estimates for double handling and storage cost. To account for such costs and uncertainty in the developed model, both PC and BC are modelled using the set of equations (3-10 to 3-13):

For schedule-driven projects:

$$E(PC) = \begin{cases} \begin{Bmatrix} PCr * \emptyset * (D - Deadline) & , D > Deadline \\ 0 & , D \leq Deadline \end{Bmatrix} & , Continuous\ function \\ \begin{Bmatrix} PC * \emptyset & , D > Deadline \\ 0 & , D \leq Deadline \end{Bmatrix} & , Discrete\ function \end{cases} \quad 3\text{-}10$$

$$E(BC) = \begin{cases} \begin{Bmatrix} BCr * \emptyset * (D - Deadline) & , D > Deadline \\ 0 & , D \leq Deadline \end{Bmatrix} & , Continuous\ function \\ \begin{Bmatrix} BC * \emptyset & , D > Deadline \\ 0 & , D \leq Deadline \end{Bmatrix} & , Discrete\ function \end{cases} \quad 3\text{-}11$$

The following equations are used for budget-driven projects:

$$E(PC) = \begin{cases} \begin{Bmatrix} PCr * \emptyset * (Budget - C) & , D > Budget \\ 0 & , C \leq Budget \end{Bmatrix} & , Continuous\ function \\ \begin{Bmatrix} PC * \emptyset & , C > Budget \\ 0 & , C \leq Budget \end{Bmatrix} & , Discrete\ function \end{cases} \quad 3\text{-}12$$

$$E(BC) = \begin{cases} \begin{Bmatrix} BCr * \emptyset * (Budget - C) & , C > Budget \\ 0 & , C \leq Budget \end{Bmatrix} & , Continuous\ function \\ \begin{Bmatrix} BC * \emptyset & , C > Budget \\ 0 & , C \leq Budget \end{Bmatrix} & , Discrete\ function \end{cases} \quad 3\text{-}13$$

Where $E[PC]$ and $E[BC]$ is the expected value of penalty and bonus cost, PCr and BCr is the penalty and bonus cost rate per unit of time, $\emptyset = f(p)$ is a multiplier obtained from the uncertainty PDF for penalty and bonus costs using a random probability number p. Figure 3.3 shows an illustration for a linear cumulative distribution of PC and BC cost for a schedule-driven project.

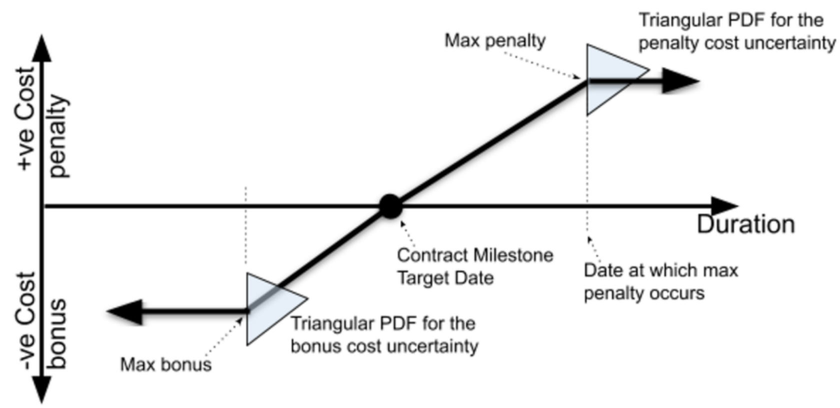


Figure 3.3 Modelling penalty and bonus cost for the schedule-driven project.

For simplicity, a linear function is considered for the VC, IC, PC and BC relationship to time in order to reduce the input required for the analysis model. Non-linear relationships may be considered subject to future enhancements.

3.4. Modelling uncertainty

The Probability Density Function (PDF) is the most common way to model uncertainty. Several PDFs have been concluded adequate for modelling durations and costs of construction projects, the most popular are Normal, Triangular, Beta and Uniform distributions, as shown in Figure 3.4.

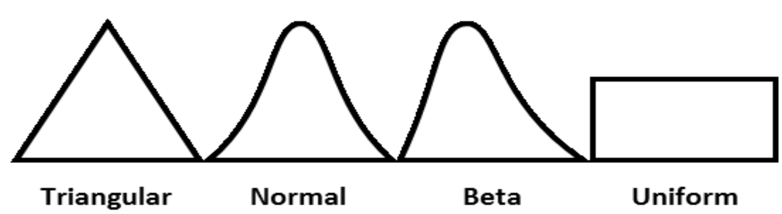


Figure 3.4 Popular probability distribution functions PDFs.

The Normal distribution is the best-known PDF that has significantly contributed to the theories of probability and statistics. The Normal distribution has a probability density function given by:

$$f(x, \mu, \sigma) = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sigma\sqrt{2\pi}} \qquad \textit{3-14}$$

where μ and σ are the mean and standard deviation parameters of the distribution, respectively. The standard Normal distribution has the parameters μ = 0 and σ =1. The probability density function PDF and the cumulative density function CDF, accordingly, are represented by equations (3-15 and 3-16) :

$$PDF \qquad f(x) = \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}} \qquad \textit{3-15}$$

$$CDF \qquad F(x) = \int_{-\infty}^{x} \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}} \qquad \textit{3-16}$$

The mean and standard deviation of the distribution can be represented by:

$$Mean \qquad \mu = \frac{\sum x}{N} \qquad \textit{where N = sample size} \qquad \textit{3-17}$$

$$Standard\ deviation \qquad \sigma = \sqrt{\frac{\sum (x-\mu)^2}{N}} \qquad \textit{3-18}$$

The Normal distribution PDF and CDF is illustrated in Figure 3.5.

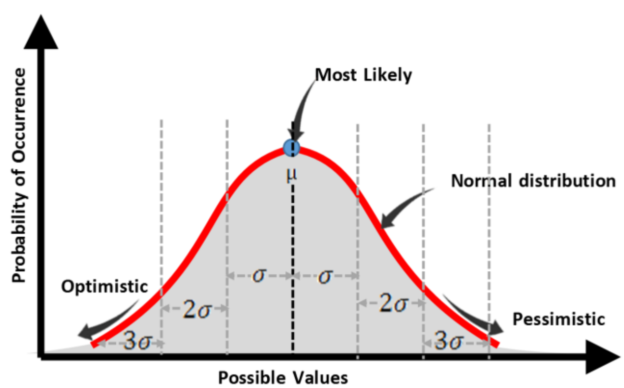


Figure 3.5 Normal probability distribution functions.

The Triangular distribution is a subjective description of a population where limited sample data is available. The Triangular distribution requires 3 parameters for its definition: Optimistic (a): The minimum time or cost at which the task can be completed. Most likely (m): The time or cost required to complete a task based on known conditions and available resources. It is also the mode of distribution, and Pessimistic (b): The maximum time or cost the task to be completed. The triangular distribution PDF is illustrated in Figure 3.6.

The Triangular distribution probability density function PDF and the cumulative density function CDF can be represented by:

$$PDF \quad f(x) = \begin{cases} \frac{2\,(x-a)}{(b-a)(m-a)} & for\ a \le x < m \\ \frac{2(b-x)}{(b-a)(b-m)} & for\ m \le x \le b \end{cases} \tag{3-19}$$

$$CDF \quad F(x) = \begin{cases} \frac{(x-a)^2}{(b-a)(m-a)} & for\ a \le x < m \\ 1 - \frac{(b-x)^2}{(b-a)(b-m)} & for\ m \le x \le b \end{cases} \tag{3-20}$$

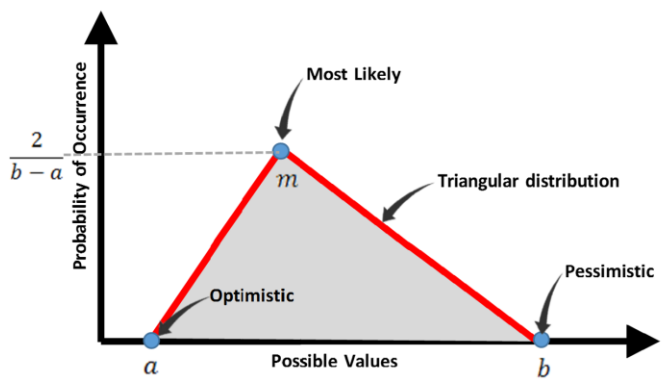


Figure 3.6 Triangular probability distribution functions.

The mean and standard deviation of the distribution can be represented by:

$$Mean \quad \mu = \frac{a+m+b}{3} \tag{3-21}$$

$$Standard\ deviation \quad \sigma = \sqrt{\frac{a^2+m^2+b^2-ab-am-mb}{18}} \tag{3-22}$$

To allow calculation of sampling from a triangular distribution, the inverse transformation of the CDF is formulated as follows:

$$Inverse\ CDF = F^{-1}(x) = \begin{cases} a + \sqrt{p\,(m-a)(b-a)} & for\ 0 \le p < \frac{m-a}{b-a} \\ b - \sqrt{(1-p)(b-m)(b-a)} & for\ \frac{m-a}{b-a} \le p \le 1 \end{cases} \qquad 3\text{-}23$$

Where p is uniformly distributed on (0,1).

Studies have proposed a modification to the triangular function (Vose 2008), which include the Trigen probability distribution function and the Double Triangular probability distribution function. Such modifications were developed to account for the biased estimations of optimistic and pessimistic values to accommodate for optimism bias when needed. The Trigen function short for “triangle generation” requires specifying an optimistic percentage p_a and a pessimistic percentage p_b such as 10% and 90%. Figure 3.7 illustrates the Trigen function. This probability distribution allows to widen the function boundaries to allow for a preselected probability percentage to occur outside those boundaries.

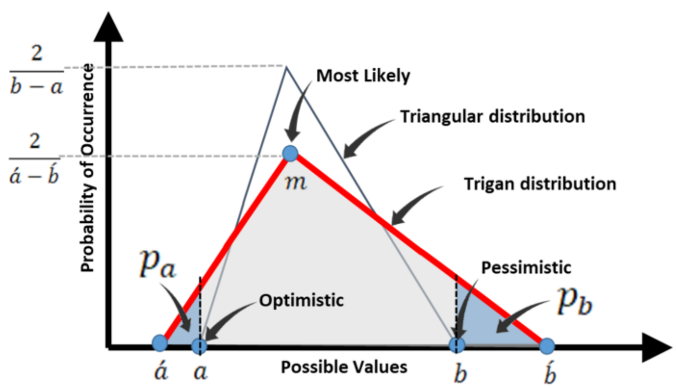


Figure 3.7 Trigen probability distribution functions.

The Double Triangular function is another alternative to the triangular function. In addition to the optimistic, most likely, and pessimistic values, a probability value p between 0 and 1 is also selected for the probability of occurrence from the left side of the triangle, and (1-p) is the probability of occurrence from the right side of the triangle. Figure 3.8 illustrates the Double Triangular function.

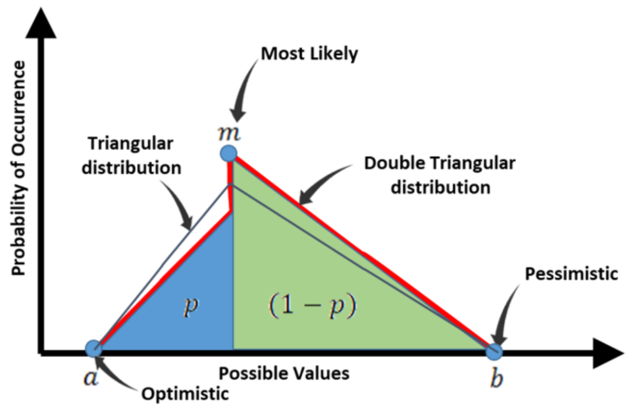


Figure 3.8 Double Triangular probability distribution functions.

The Beta-PERT distribution is a special form of the Beta distribution useful for modelling expert data. The function can be used in a MCS to model uncertainties in cost elements of a project. The fit of the Beta-PERT distribution, like any other distribution, is bounded by the quality of the expert estimates of the parameters. Similar to the Triangular distribution, the Beta-PERT distribution requires the 3 parameters for its definition: Optimistic (a), Most likely (m), and Pessimistic (b). An illustration of the Beta-PERT distribution PDF is shown in Figure 3.9.

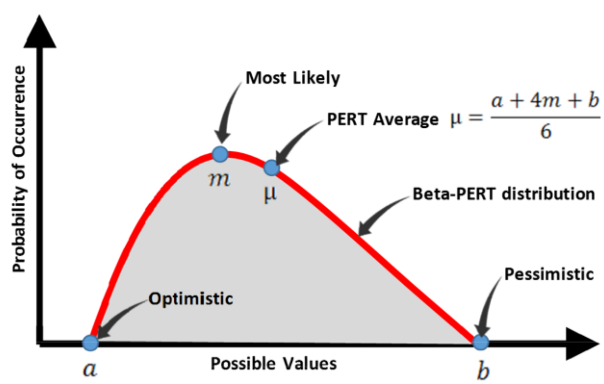


Figure 3.9 Beta-PERT probability distribution functions.

The Beta-PERT function is preferred over the normal function due to its ability to be symmetrical, skewed right towards the pessimistic, or skewed left towards the optimistic (Schexnayder 2005). AbouRizk and Halpin (1992) and Meredith and Mantel (2009) found in their studies that the Beta-PERT distribution is the most suitable in project management models to denote activity duration times given the fact that it allows the portrayal of biases to the right or to the left (Vose 2008).

The Beta-PERT distribution is a special case of the Beta distribution specified by the parameters α, β. Where:

$$\alpha = \frac{4m+b-5a}{b-a} \tag{3-24}$$

$$\beta = \frac{5b-a-4m}{b-a} \tag{3-25}$$

The Beta-PERT distribution probability density function PDF and the cumulative density function CDF can be represented by:

$$PDF \qquad f(x) = \frac{(x-a)^{\alpha-1} * (b-x)^{\beta-1}}{Beta\ (\alpha,\beta) * (b-a)^{\alpha+\beta-1}} \tag{3-26}$$

$$CDF \quad F(x) = I_z(\alpha, \beta) \tag{3-27}$$

Where $Beta\ (\alpha, \beta)$ *is the Beta function;* α *and* β *are the shape factors of the function and* I_z *is the incomplete beta function.*

The mean and standard deviation of the distribution can be represented by:

$$Mean \qquad \mu = \frac{a+4m+b}{6} \tag{3-28}$$

$$Standard\ deviation \qquad \sigma = \frac{b-a}{6} \tag{3-29}$$

Experts are generally more confident to guess the most likely duration than the pessimistic and optimistic values; hence, the Beta-PERT distribution has four times the weighting on the most likely time (m).

The Uniform distribution is applied where quantities vary uniformly between two values. This distribution is only recommended to be used to model less sensitive variables to be conservative for accounting of extreme values of the variable (Pouliquen 1970). The Uniform distribution probability density function PDF and the cumulative density function CDF can be represented by:

PDF $$f(x) = \begin{cases} \frac{1}{(b-a)} & for\ a \leq x \leq b \\ 0 & elsewhere \end{cases}$$ *3-30*

CDF $$F(x) = \begin{cases} 1,\ x \geq b \\ \frac{x-a}{b-a},\ a \leq x\ \leq b \\ 0,\ elsewhere \end{cases}$$ *3-31*

The mean and standard deviation of the distribution can be represented by:

$$Mean \quad \mu = \frac{a+b}{2}$$ *3-32*

$$Standard\ deviation \quad \sigma = \sqrt{\frac{(b-a)^2}{12}}$$ *3-33*

The inverse transformation of the CDF is calculated as follows:

From the CDF, set $F(x) = (X - a)/\ ((b - a) = p$

Solving for X in terms of p yields:

$$X = a + (b - a)p$$ *3-34*

Where p *is uniformly distributed on (0,1).*

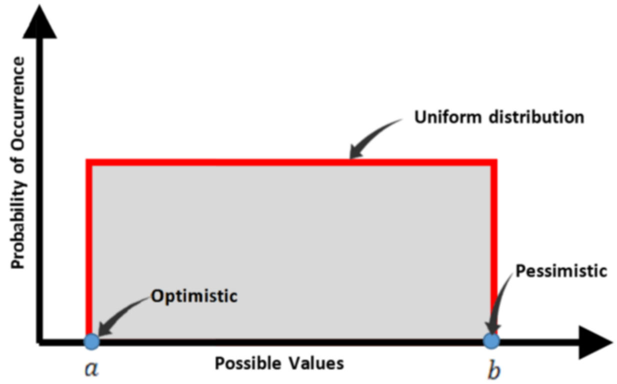


Figure 3.10 Uniform probability distribution functions.

3.5. Joint cost and schedule confidence level

This section outlines a summary of the joint cost and schedule confidence level (JCL) process steps and results as presented by NASA (Hoffpauir 2015). The JLC process requires 5 essential steps that need to be executed in order, as shown in Figure 3.11.

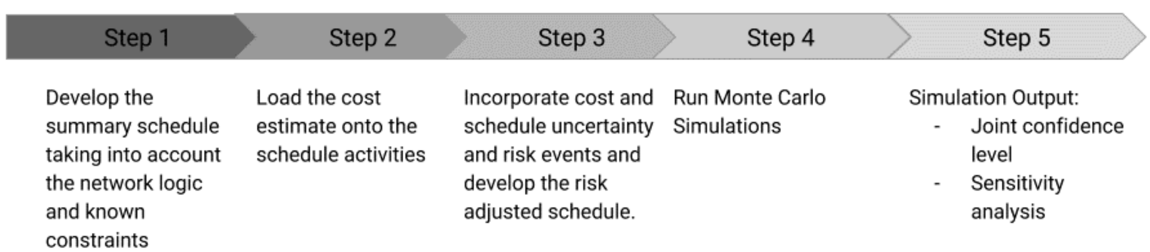


Figure 3.11 Joint Cost and Schedule Confidence Level (JCL) process steps.

The first step is the development of the project schedule with a solid logic network that models the project execution strategy. The schedule is required to be built with the consideration of the goals and objectives of the JCL analysis. Figure 3.12 shows a simple schedule for illustration purposes.

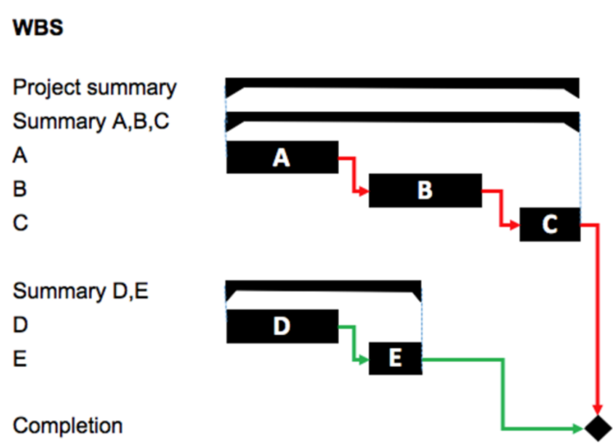


Figure 3.12 A simple schedule network illustration.

The second step is to load the cost estimates. Cost loading is accomplished by mapping the cost to the schedule. Each activity may be loaded with FC and VC. The VC cost are attributed to the duration of activities, where the duration will determine the value of the cost of such activity (unit cost per unit of time). The IC may also be loaded and associated with the overall project duration. The total project cost (C) is then the sum of all FC, VC and IC. Figure 3.13 shows the assignment of FC and VC to the network activities. At this step, the project deterministic schedule and cost can be calculated.

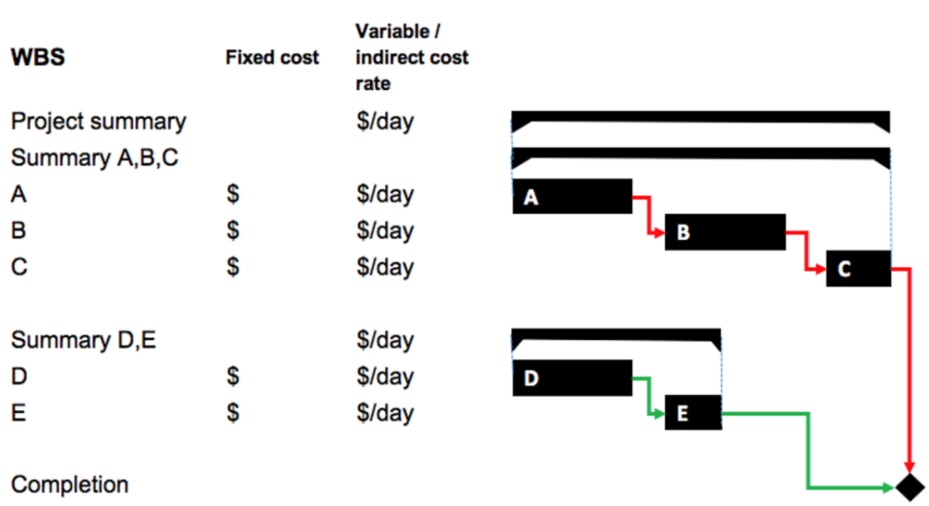


Figure 3.13 Assignment of FC, VC and IC costs on project activities.

The third step is to incorporate the risk events. The identified risks from the project risk register is realized and integrated into the model. The risk events are modelled by adjusting the project schedule network to add a risk activity with its most likely impact duration and cost. This activity represents the additional work required resulting from reacting to the occurrence of the risk. In a deterministic scheduling technique, those activities do not exist and adds nothing to the project cost and schedule. Each activity is then assigned a Bernoulli distribution to model the occurrence of the risk event represented by either True or False (e.g. a risk with a 50% probability of occurrence means it will show an impact in 50% of the simulation runs). Figure 3.14 shows the risk-adjusted model, where Risk events 1 and 2 are added to the network as activities. Risk 1, for example, has a 50% probability of occurrence and when occurs is modelled to extend activity B and will delay the start of activity C by the duration assigned to the risk event. The risk also is loaded with a TIC. The TIC may be for purchasing replacements to damaged plant equipment. To this step in the JCL process, we can perform a quantities risk assessment and its impact on the overall project's schedule and cost.

It is worth noting here that the discrete risk events need to be distinguished from the uncertainties in the cost and duration values of scope activities; therefore, careful consideration of those risk events is required by the project manager to avoid double-dipping in the modelling the impact of risks and uncertainty.

The discrete risk events considered in this study are associated to activities; such risks are not introduced randomly, instead, they incorporate the experience and judgment of the experts to position those risks in the project network. Other types of risks that are not specific to an activity may also exist, such as severe weather conditions; those risks are not considered in the modeling of the developed method and are subject for inclusion in future work.

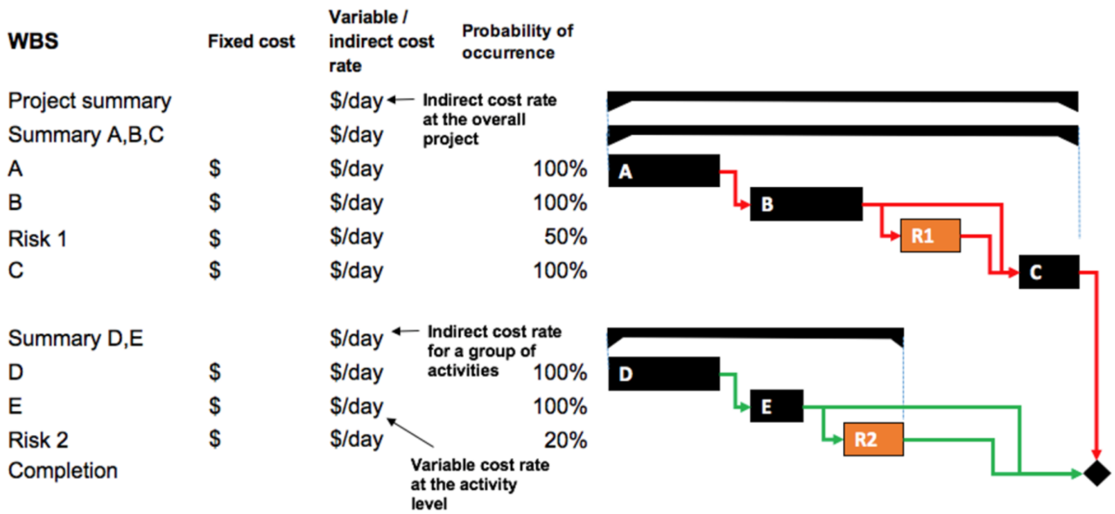


Figure 3.14 Risk-adjusted schedule.

The fourth step is to incorporate the uncertainty in the project cost and schedule estimations. This uncertainty is considered to cover two key facets that can potentially drive cost and schedule:

- Un-identified risks that may have been missed.
- Uncertainty in the estimate of activity duration and costs.

The JLC process calls to distinguish and segregate between risks and uncertainty to avoid double-counting uncertainty caused by risks that are already modelled. Risk is an event that is not part of the base plan; it has a probability of occurrence, and if occurs will have an undesirable impact on the project cost and schedule whereas uncertainty represents the imperfect ability to predict a future event.

The uncertainty is modelled using a probability distribution function. Typically, and for simplicity, the triangular distribution is used. Figure 3.15 shows the uncertainty assigned to the activity durations and costs.

It is worth nothing that not all cost and duration parameters are uncertain. Therefore, the developed method allows a mix between deterministic and uncertain estimations for the activity's duration and cost values.

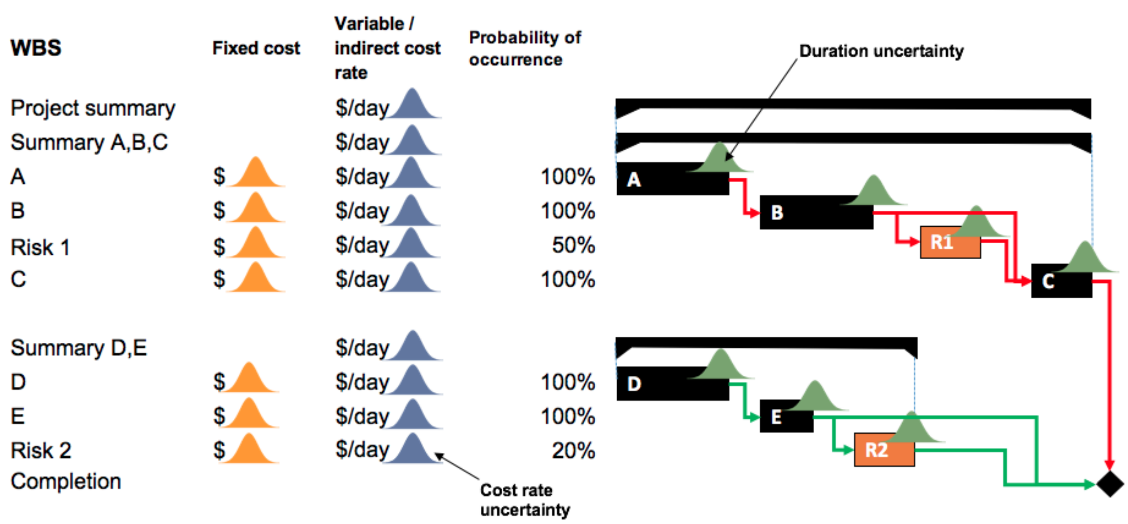


Figure 3.15 Uncertainty assignment to the activity durations and costs.

Step Five: Calculate and view results

In the fifth step in the JLC process, a Monte Carlo simulation is applied, and an extensive risk analysis is made. The discrete risks, cost and schedule, are simulated and iterated many times. The estimates of durations and costs are selected at random for each simulation trial from probability distributions. At each iteration, different finish dates and costs may occur, and the data for cost, duration, start and finish dates and float are collected. From this data, a variety of reports can be generated to analyze the outputs. The most popular report used is the JCL scatterplot chart. An example of this plot is shown in Figure 3.16. The frontier curve joining all the yellow points represents all results that satisfy a 70% joint cost and schedule confidence level.

As expected, the more the selected overall project cost values, the more the confidence level; similarly, for selecting a higher overall project duration. Therefore, the joint confidence distribution is expected to have an positive correlation between time and cost; which, in turn, will result in an increasing concave-up frontier curve. The mathematical proof of this relationship is well detailed in Xu et al. (2014). An illustration of this relationship is provided in Figure 3.17.

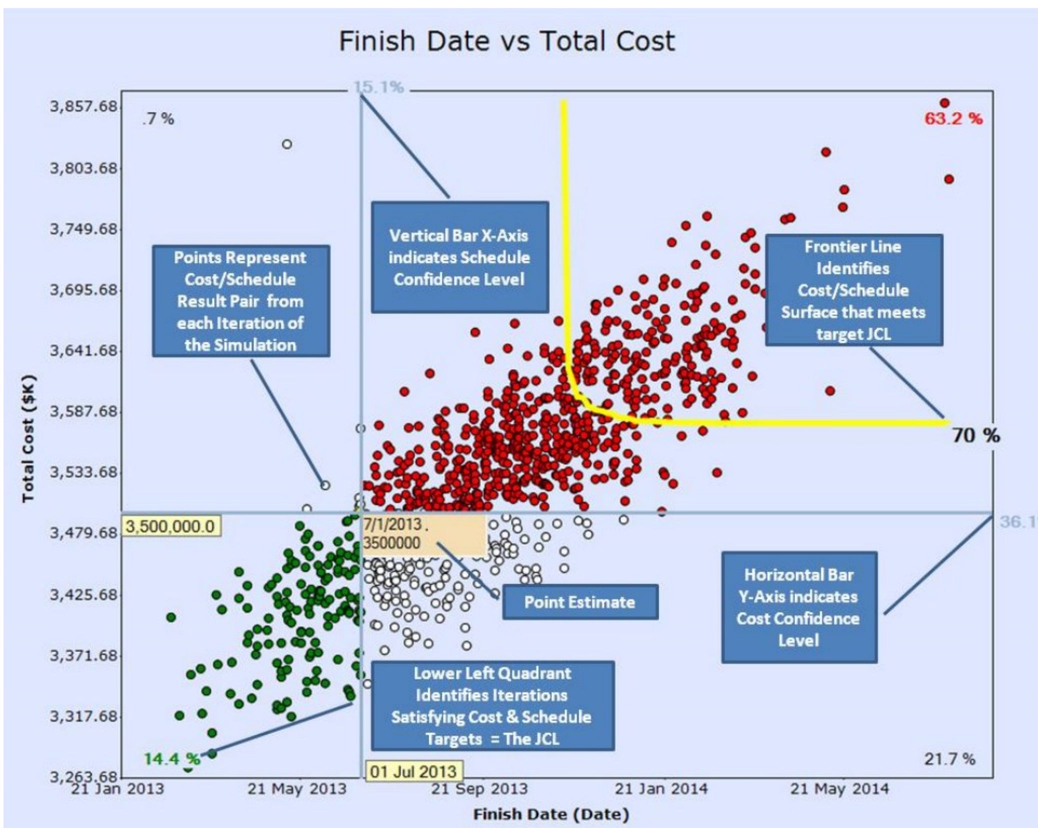


Figure 3.16. JCL Scatterplot.[1]

[1] Source (Hoffpauir 2015).

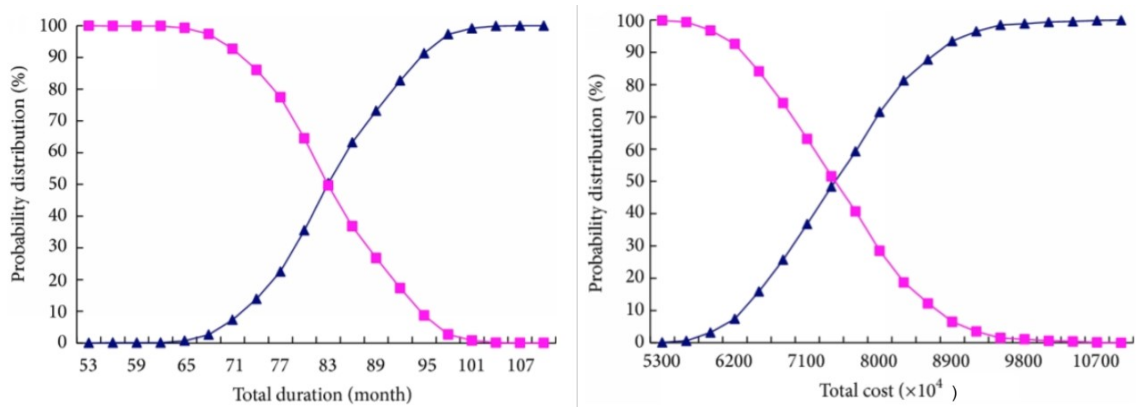


Figure 3.17. The marginal probability distribution and marginal risk probability distribution of total duration and total cost.[2]

3.6. Sensitivity analysis

Another set of outputs of the cost and schedule risk analysis are indices for the degree of an activity criticality and sensitivity that offers an indication for sensitive activities and its effect on the overall project's duration and cost. The most popular indices related to schedule risk analysis are:

- Criticality Index (CI): This index is a measure of the frequency for an activity when observed on the critical path to the total number of iterations as a ratio (between 0 and 1).
- Significance Index (SI): is a measure of the importance of an activity with regards to its influence on the overall project's completion. The follows can be used to calculate the SI:

$$SI = E\left\{\frac{(AD * SD)}{((AD + SL) * E(SD))}\right\} \qquad 3\text{-}35$$

[2] Source (Xe et al. 2014).

Where SD is the project duration resulting from the simulation, AD is the activity duration, SL is the activity slack, and E(x) is the expected value of x (i.e. E(SD) is the mean of SD from all simulation runs).

- Schedule Sensitivity Index (SSI): is a measure of the importance for an activity with regards to the overall project's completion, taking into account the activity criticality index, and is determined as follows:

$$SSI = \frac{(\sigma(AD) * CI)}{\sigma(SD)} \qquad 3\text{-}36$$

Where σ(AD) is the standard deviation of the activity duration, σ(SD) is the standard deviation of the simulated activity duration, and CI is the criticality index described above.

- Cruciality Index (CRI): is a measure for correlation between the activity duration and the overall project's duration.

$$CRI = \left|correlation\big(Avg(AD), Avg(SD)\big)\right| \qquad 3\text{-}37$$

Where Avg(AD) is the average activity duration, and Avg(SD) is the average simulated overall project's duration.

There are three ways to calculate the CRI:

- CRI(r): Pearson's product-moment correlation coefficient.
- CRI(ρ): Spearman's rank correlation coefficient.
- CRI(τ): Kendall's tau rank correlation coefficient.

Sensitivity analysis in the scheduling context has been studied by several authors. For an overview, see (Hall and Posner 2004).

Similarly, a popular index related to cost risk analysis is the Cruciality Index for cost (CRI_c) that measure the correlation between the cost elements and the overall project's cost and is computed using the following equation:

$$CRI_c = |correlation(Avg(AC), Avg(SC))| \quad \text{3-38}$$

Where Avg(AC): Average activity cost and Avg(SC): Average simulated total project cost.

3.7. Time-Cost Trade-off Analysis

This section outlines a summary of the traditional linear time-cost trade-off process steps and results. The trade-off process requires 5 essential steps that need to be executed in its order, as shown in Figure 3.18.

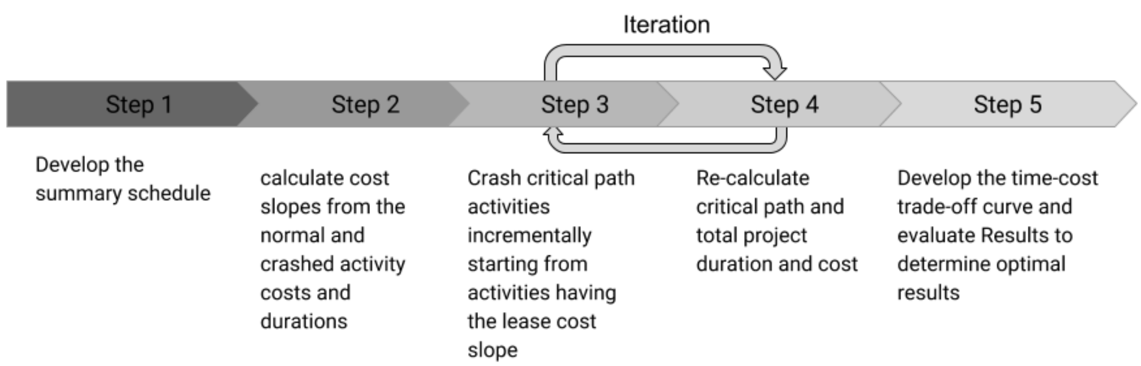


Figure 3.18 Time – Cost trade-off process steps.

The first step is to construct the summary schedule for the project and calculate the start and finish times for each task, taking into account the network logic and known constraints. At this step, each activity duration is estimated at the base case plan, otherwise known as the normal duration (D_n). For the same bases, assumptions, execution methodology and resources, the activities are then loaded with the normal cost estimate (C_n). In the second step, the activities are studied to determine a possible acceleration of the individual activities; this may be done by using alternative execution methods, technologies, additional resources or by extended hours in the workweek. Any change may result in additional costs; this cost is estimated and assigned to the activities as crashed cost (C_c) at its crashed duration (D_c). There are several time-cost relationships; those relationships can be categorized as discrete or continuous. The continuous relationships can, in turn, be linear or nonlinear. Figure 3.19 shows the various relationships that may exist between normal and crashed costs and durations.

The cost slope for each activity is then calculated using the following equation:

$$Cost\ Slope\ = (C_c - C_n)/(D_n - D_c) \qquad 3\text{-}39$$

The third step is to incrementally crash the activities in an iterative process based on a priority basis. The priority is set for activities that lie on the critical path and have the lease cost slope. At this step, a recalculation for the project network is made to re-determine the critical path for every incremental crash. Several critical paths may evolve, and the simultaneous crashing of critical activities at a given iteration is required to achieve a shorter overall project's duration. Step three and four are repeated until the project can no longer be shortened.

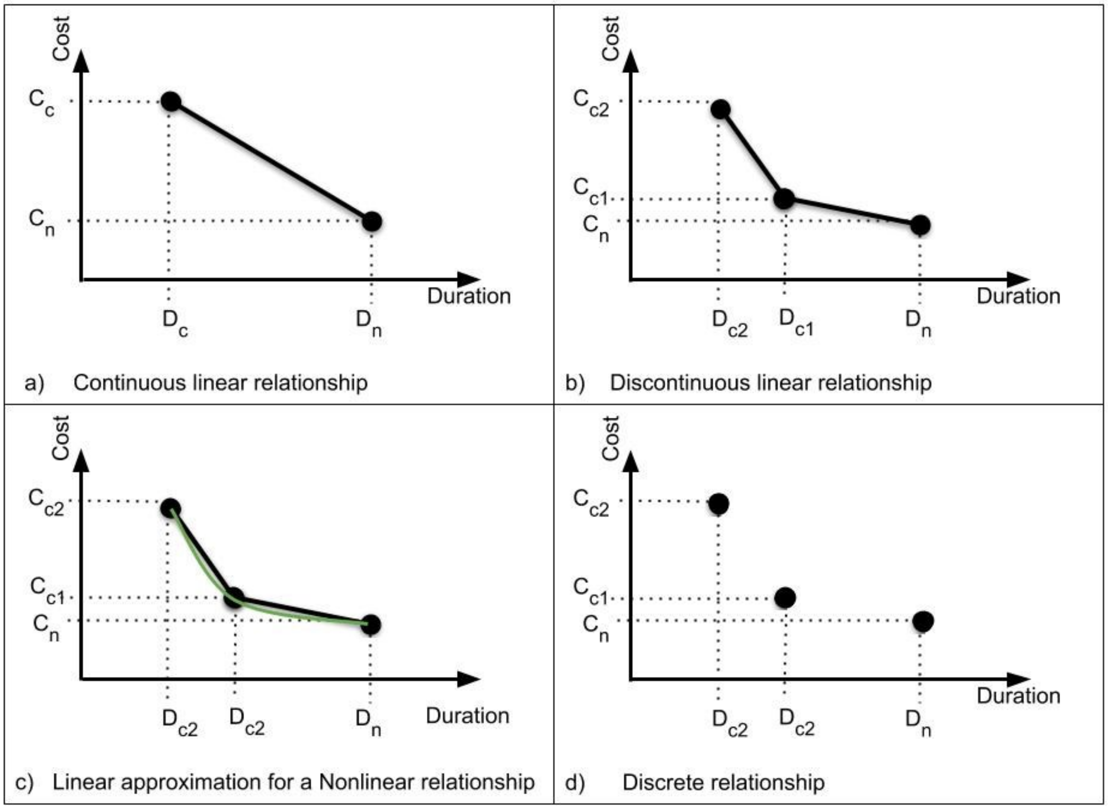


Figure 3.19 Relationships between the normal and crashed time and cost.

The total project cost (C) is the sum of the FC and the IC. The FC is calculated for each crashing step as the sum of the costs for all activities. The IC, on the other hand, is usually calculated as the IC rate per unit of time multiplied by the resulting overall project's duration. Steps one to Step five is to develop the cost summary with different project durations as shown graphically for illustration purposes in Figure 3.20.

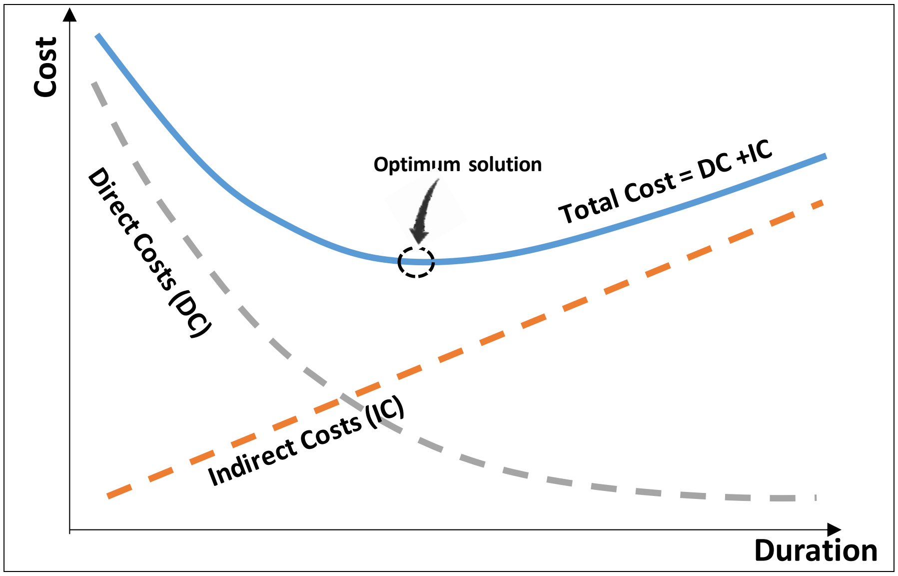


Figure 3.20. Time – cost trade-off relationship.

The deterministic discrete TCTP can be formulated for the multi-objective function to minimize the total project duration and minimize the total project cost using the following set of equations:

$$Objective =$$

$$\begin{cases} Minimize\ C\ \wedge\ Minimize\ D\ , for\ \ Joint\ cost\ and\ schedule\ minimization \\ Minimize\ C\ , \qquad\qquad\qquad\qquad\qquad\qquad for\ \ cost\ minimization \\ Minimize\ D, \qquad\qquad\qquad\qquad\qquad for\ schedule\ minimization \end{cases} \quad 3\text{-}40$$

Subject to:

$$EST_i = max\{EFT_j + 1\} \quad \forall\ j\ \in P_i \qquad 3\text{-}41$$

$$D = \max_{\forall\ i}\{EST_i + d_i - 1\} \qquad 3\text{-}42$$

$$C = \textstyle\sum_{\forall\ i}(FC_i) + (ICr\ \times D) \qquad 3\text{-}43$$

$$d_i = \textstyle\sum_{k=1}^{n_i} M_{i,k} d_{i,k} \quad \forall\ i \qquad 3\text{-}44$$

$$FC_i = \textstyle\sum_{k=1}^{n_i} M_{i,k} FC_{i,k} \quad \forall\ i \qquad 3\text{-}45$$

$$\textstyle\sum_{k=1}^{n_i} M_{i,k} = 1 \quad \forall\ i \qquad 3\text{-}46$$

$$M_{i,k} \in \{1,2,3, \dots n_i\} \quad \forall\, i, k \tag{3-47}$$

Equation (3-49) defines the objective of the problem. The objective can be set for one of three cases, the first is the minimization of both total project duration D and the total project cost C to determine the joint cost and schedule minimization. The second is to minimize C where the total project cost reduction is of interest, and the third is to minimize D when the total project schedule minimization is of interest. Equation (3-50) provides the early start time EST for all activities based on the CPM forward pass calculations taking into account the relationships between activity i and all of the activities in its precedence set P_i, in this case, for simplicity, the FS relationships are considered with no precedence lags, d_i is the duration for activity i . Equations (3-51) and (3-52) computes the overall project's duration D and the total project cost C taking into account the fixed cost (FC) and the indirect cost rate ICr. Equations (3-53) and (3-54) selects the time-cost pair for an activity from the set of defined execution modes $M_{i,k}$, where the number of modes for activity i is n_i. Equations (3-55) and (3-56) ensures that only one mode is selected at a given solution.

CHAPTER 4 : IMPLEMENTATION OF THE DEVELOPED METHOD ON TRADITIONAL NON-REPETITIVE CLASS PROJECTS

4.1. Introduction

This chapter presents the developed object-oriented programming for the Evolutionary Stochastic Discrete Time-Cost Trade-Off Method ESDTCT. The method is implemented in Google Sheets which is a spreadsheet-based application and one of the core components of Google Cloud applications. The Google Apps Script (GAS) is a JavaScript programming language used for the developed procedure. Google BigQuery SQL is used to facilitate the calculation of the method. Google BigQuery is an enterprise data warehouse that “enables super-fast SQL queries, using the processing power of Google's infrastructure” (Fernandes 2015). GAS and BigQuery are executed, not in the browser but remotely on the Google cloud. The application can be executed using any modern browser (preferably Chrome and Firefox), an internet connection running on Windows, Mac OS X or Linux. As mentioned before; Instead of running on the client computer, the program script is executed in the Google Cloud; therefore, the speed of executing the code is not limited by the client computer specifications, instead by the capacity of google servers, which in turn has undergone significant development since a major upgrade in March 2014.

The interrelationship between schedule and cost is critical and is required to be prudently considered before making promises and announcements. Previously, some companies and organizations use separate departments for the cost estimates and schedules of the same project. Sometimes those departments act autonomously of each other and may disagree on key results vital to the success of the project. Figure 4.1 illustrates the siloed risk and contingency assessment processes. Accordingly, a process for the integration of cost, schedule, and risk is gradually recognized as a crucial business practice. Effective integration of cost, schedule and risk necessitates the project manager, scheduler and estimator to collaborate in identifying the execution strategy, methods, activity durations, sequencing, workforce requirements, resources, and constraints. The result in a range of possibilities must be understood to reach the desired objective. Decisions relative to those possibilities are made in an iterative manner, evaluating alternatives on the basis of cost, schedule, in order to reach an agreed-upon schedule and budget solution. Figure 4.2 below outlines of the proposed integrated process.

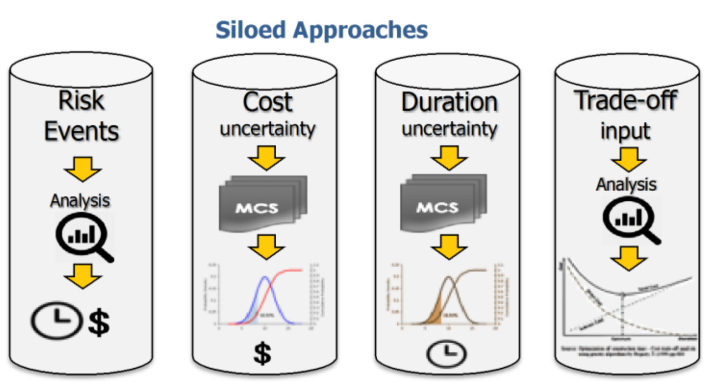


Figure 4.1. Illustration of siloed cost and schedule risk and trade-off assessments.

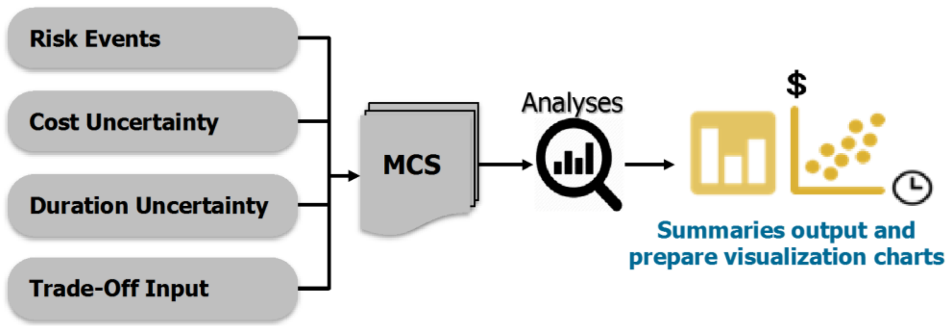


Figure 4.2. Illustration of proposed integrated process.

To overcome the complexity of solving the TCT problem with uncertainty, the proposed method attempts to solve the multi-objective optimization by partitioning the gigantic design of experiment matrix of a full factorial design through blocking of activities. The problem is treated as an experimental design problem where all combinations of possible solutions are determined in a Cartesian product matrix. The major constraint in solving this problem is the large amount of possible combinations. Computation power for this massive data-processing formulation is not only time consuming but also was not available for public use until 2011 where Google BigQuery was rolled out for public utilization, enabling further advanced services and integration with Google Sheets in 2016 (Jordan 2016).

The developed method is composed of two modules, as shown in Figure 4.3. The first identifies the supreme solution (non-dominated solution) representing the combination of modes that yields the bi-objective optimization for cost and/or schedule minimization for the specified joint confidence level of both time and cost. This combination is compared to a chromosome where the genes are the modes of the different activities; while the second module is a random stochastic search that depicts the main effect of changing a mode within a chromosome on the total cost and overall duration of the project for the specified joint confidence level of both time and cost. The following sections provide the background for the set of procedures used simultaneously in setting the computations for the developed evolutionary stochastic discrete time-cost trade-off method (ESDTCT).

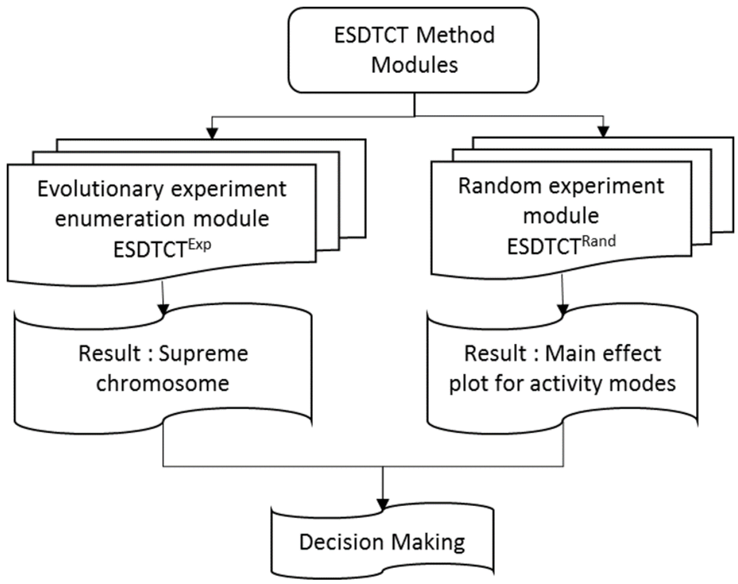


Figure 4.3. Flow chart for the Integration of the developed ESDTCT modules.

4.2. Cloud application used in the ESDTCT method

Due to the fact that most user desktop computers are limited in specifications for hard drive storage capacity, processing power and random-access memory, it was necessary to code the application on a cloud platform. Google Cloud was selected due to its ability to scale seamlessly as computation power demand increases and decreases. Another reason for selecting this platform is its cost-efficiency. The free quota available for the public was found sufficient to solve large instances of project networks. The run time for several test problems will be discussed in section 4.8.

BigQuery has been developed and used by Google for Search engines, Google maps, YouTube, Gmail and Google Docs. To take advantage of Google's powerful computing engine, the project network CPM is coded using standard SQL statements and in-line JavaScript User-Defined-Functions (UDF) to calculate the start and finish dates of the activities. This novel code computes a large network within milliseconds; this, in turn, will enable computing the gigantic number of simulations generated by the developed method in a reasonable short time. Similarly, the uncertainty probability distribution functions are coded in the UDF, taking advantage of the built-in random and other mathematical functions.

There exist many third-party tools and Add-ons for performing CPM calculations, the probability distribution functions and Monte Carlo simulation. However, in this study all computations are coded and executed in one cloud application. By doing so, we are able to eliminate the time necessary to transfer data between different applications, and thus, increasing the computation performance. Furthermore, we were able to avoid the subscription costs to the third-party add-ons.

Google sheets (GS) is a web-based spreadsheet software offered by Google within its Google Drive service. GS is used as the front-end user interface where templates are created for the user data input to describe the project network activities, logic, risks, selection of modes and its respective uncertain cost and time attributes. A GAP code is developed to prepare the BigQuery SQL statements and make Application Program Interface (API) calls to BigQuery which, in turn, is set to perform procedures and protocols.

The design of experiments matrix is decomposed into segments, and parallel Interactive query APIs are concurrently performed. The output of each query is then appended to a master BigQuery table. Using the cloud storage enabled removing the restrictions when scaling the computations to a large network in contrast to using a personal hard drive on a local computer. The concurrent interactive queries significantly contributed to reducing the overall run time.

The purpose of this section is not to explain the coding script but to summarize the use and integration of the variance Google Cloud applications adopted in the developed method. More details are available on GAP, BigQuery and Google quotas and limits in (Ferreira 2014 and Maharana et al. 2015). Figure 4.4 shows the developed method network configuration diagram and the integration of Google Cloud apps.

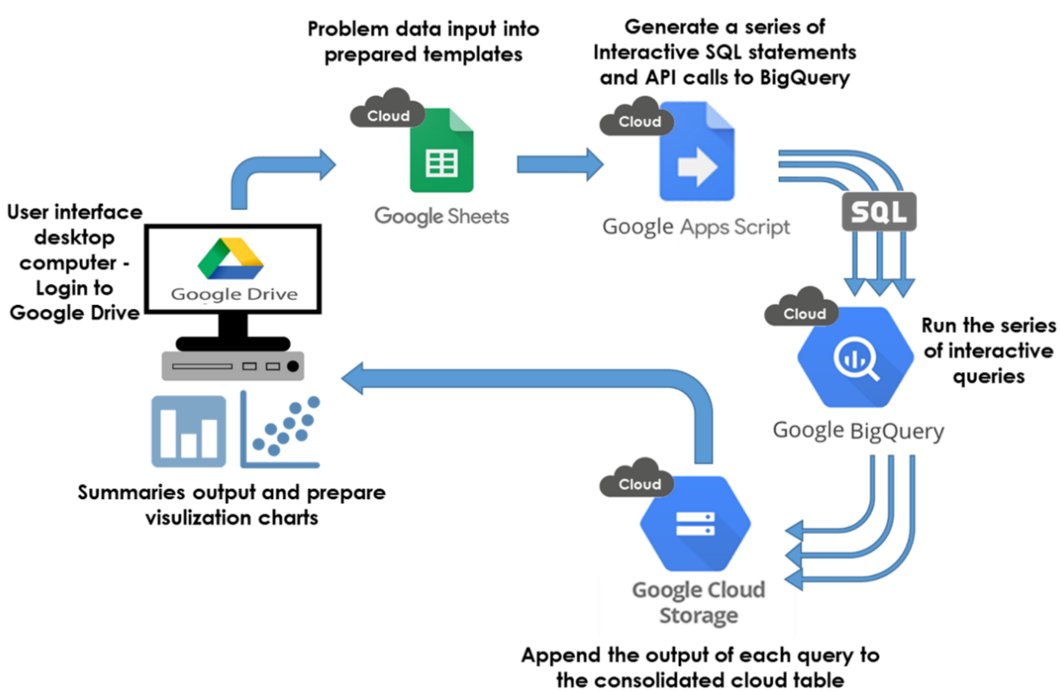


Figure 4.4. ESDTCT cloud network diagram.

4.3. Definition of terms used in the ESDTCT method

The developed method adopts biomedical terms amongst others such as “Evolution”, “Generation” and, “Chromosomes”. Those terms have been interchangeably used in Genetic Algorithm optimization methods and are further defined here to put those terms in the context of the developed method. Figure 4.6 shows a graphical representation of those terms.

Activity (Gene): is an element position of a chromosome. Where this element represents an individual activity within the project network.

Mode: is a variant form of a gene. It represents the resources that are available for use to execute an individual activity.

Elite mode: is the best mode assigned to an activity that results in the optimal solution of the problem. Where the term “best” is defined by the optimality definition for the objective function.

Chromosome: represents the combination of a string of modes for each activity in the network. Accordingly, it depicts one possible solution in the solution space.

Supreme chromosome: is the chromosome that results in the optimal (non-dominated) solution of the problem.

Population: is all the possible chromosomes available in the problem solution space.

Generation: is a sample of the population that represents a set of chromosomes sharing common modes.

Evolution: is the gradual crossover of elite modes to the next generation. The evolutionary technique is used to evolve the elite modes surfacing from previous generations.

Blocking: is a design of experiments (DOE) technique used to eliminate the influence of extraneous factors when running an experiment and focus the analysis on a set of factors having more relevant influence on the objective in a particular generation.

Primary mode activity: is an element position in the project network being investigated for the inclusion of its elite mode in a particular generation. A set of primary activities are identified at each generation using the immediate reverse dominator tree (IRDT) described later.

Observed mode activity: is an element position in the project network where its elite mode has been identified from the previous generation.

Base mode activity: is the base case admitted mode for an activity.

Immediate reverse dominator tree (IRDT): is the set of activities that immediately precedes primary activities from a previous generation. The finish activity is the root of the reversed tree. An illustrative example of the IRDT is shown in Figure 4.5 where a total of two IRDT generations exist to walk backwards through the entire network activities.

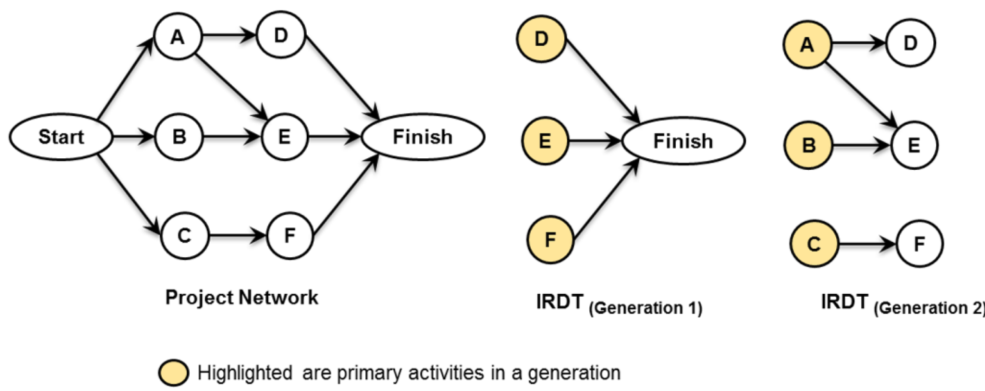


Figure 4.5. Immediate reverse dominator tree.

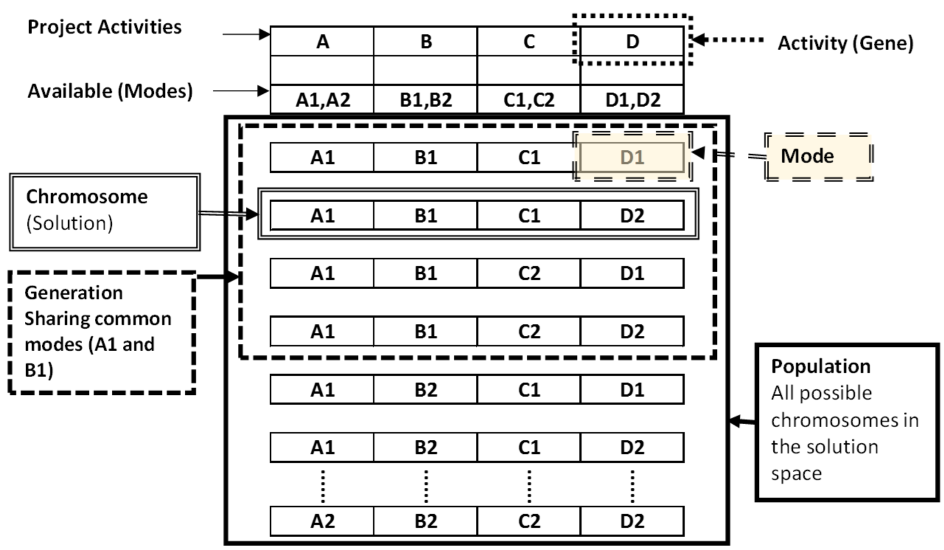


Figure 4.6. Definition and structure of terms.

4.4. Implementation of design of experiments

Design of experiments (DOE) is a process used to find the relationship between factors (modes) affecting a process and the output of that process to identify the factors most influential in optimizing the output. A full factorial design of experiments is the matrix of all combinations of possible modes, also known as the Cartesian product matrix, where each combination can be in resemblance to a chromosome. As illustrated by El-Rayes (1997), a project comprising 20 activities, each having 5 possible crew formations, would result in over ninety-five trillion possible project schedules each represented by a chromosome of discrete selection of modes. The number of chromosomes increases exponentially as the project network activities increase and/or the number of alternative delivery modes at each activity increases; therefore, blocking and randomization techniques are used in the developed method. The blocking technique is used to create homogeneous groups or blocks in which a selection of primary factors can vary while holding the other factors constant, thus allows greater precision in the estimation of primary factors. Randomized designs, on the other hand, allow us to study the effects of factor and compares the values of a response variable based on the discrete levels of the factors. One of the useful plots generated from the output of the randomized designs are the Main Effect Plots. The plot enables to study changes between means among different variables. The plot graphs the mean of all responses for each variable connected by a line. When the line is flat, this demonstrates that there is no main effect and the response is anticipated to be the same over all the levels. When the line not flat, there is a main effect. The more extreme the slope of the line, the more prominent the magnitude of the effect. An illustrative example for four activities each having two modes is shown in Figure 4.5. From visual interrogation of the plot, the change in modes in activity D appears to have more influence on the response. while the change in modes on activity B seems to have a smaller influence.

Figure 4.7 Main Effect Plot.

4.5. Fundamentals of the partitioning process

A novel partitioning method is developed. The purpose of this method is to reduce the number of experiments as oppose to a full factorial design of experiments. The blocking technique is based on partitioning the project network activities into three partitions, namely $(\mathbb{B}, \mathbb{P}, \mathbb{O})$ as shown in Figure 4.8. In this technique, the project network is resembled as a structure, where activities in partition $\mathbb{B}$ apply logical forces to drive the activities in partition $\mathbb{P}$ and so on carried forward to drive the activities in partition $\mathbb{O}$. The set of activities in the intermediate partition $\mathbb{P}$ are labeled as “Primary mode” activities, while the set of activities in its precedence partition $\mathbb{B}$ are labeled as “Base case mode” activities and its successive partition $\mathbb{O}$ are labeled as “Observed mode” activities.

The partitioning procedure is repeated to identify the label of each activity at each generation $\mathrm{G_g}$, where g is the generation number. To establish the partitions for the first generation $\mathrm{G}_{\mathrm{g}=1}$, let $\mathbb{A}$ be the set of all activities in a project network having a total number of activities I , where $i = (1 \ \rightarrow I)$; activity $i = 1$ is the start activity and $i = I$ is the finish activity. Then at any generation (g), $\mathbb{A} = (\mathbb{B}_{\mathrm{G_g}} \cup \mathbb{P}_{\mathrm{G_g}} \cup \mathbb{O}_{\mathrm{G_g}})$. Let $\mathrm{P_i}$ be the sets of all immediate predecessors of activity i where $P_i \in \mathbb{A}$ then $(\mathbb{P}_{\mathrm{G_1}} = [n, P_i] : i = I)$ and $\mathbb{B}_{\mathrm{G_1}} = \mathbb{A} \notin \mathbb{P}_{\mathrm{G_1}}$. For successive generations ($\mathrm{G}_{\mathrm{g}+1}$), the partitions are defined as $\left(\mathbb{P}_{\mathrm{G_g}} = [\mathrm{P_i}] \ \forall \, i \in \mathbb{P}_{\mathrm{G_{g-1}}}\right)$, $\left(\mathbb{O}_{\mathrm{G_g}} = \mathbb{P}_{\mathrm{G_{g-1}}} \cup \mathbb{O}_{\mathrm{G_{\forall\, g\rightarrow 1}}}\right)$ and $\left(\mathbb{B}_{\mathrm{G_g}} = \mathbb{A} \cap (\mathbb{O}_{\mathrm{G_g}} \cup \mathbb{P}_{\mathrm{G_g}})\right)$, Refer to the illustrative network example in Figure 4.9(a to d) , In the first generation of partitions, $\mathbb{P}_{G_1}$ is the set of activities {K, J, I, H} which is a selection of the finish activity K and all its immediate predecessor activities. While $\mathbb{B}_{\mathrm{G_1}}$ are all remaining activities in the network {A, B, C, D, E, F, G}. In the second generation, the $\mathbb{P}_{G_2}$ partition is the set of activities {D, E, F, G}. The $\mathbb{O}_{\mathrm{G_2}}$is the set of activities {K, J, I, H} and the $\mathbb{B}_{\mathrm{G_2}}$ are all remaining activities in the network {A, B, C}. In third generation, $\mathbb{P}_{G_3}$ is the set of activities {A, B, C, D}. The $\mathbb{O}_{\mathrm{G_3}}$is the set of activities {E, F, G, H, I, J, K} and, hence, the $\mathbb{B}_{\mathrm{G_3}} = 0$ as there are no remaining activities.. It should be noted that one or more activities can overlap in several successive parts of the petitioned network. Notice that activity D was categorized as a primary activity in generation 2 and 3; such activities are called here as a multi-generation primary activity as they satisfy the precedence relationship to a primary activity in the previous generation. We continue this process to identify the partitions at each generation until we reach a stage where all activities in the network have been labeled as a primary activity throughout all generations.

In the context of TCTP analysis, the project network is segregated into the partitions described above so as to analyze and solve for the elite modes of "Primary mode" activities at a given generation. The blocked design of experiments matrix is then generated as the Cartesian product where "Primary mode" activities can assume all assigned modes while blocking "Base case" activities to their admitted base case mode (or mode 1). A complete enumeration of the resultant matrix is then performed, and the elite modes for the "Primary mode" activities are then determined as the modes returning the best fit to the objective function. Those elite modes are carried forward in successive generations where Öbserved mode' activities are blocked to the elite modes in building the design of experiments matrix. This process allows narrowing the experiment enumeration by focusing the analysis on evaluating the "Primary mode" activities only at each generation. For the illustrative network example in Figure 4.9, if each activity i have five modes, then the complete full factorial design of experiments matrix for the population in the solution space is (5^{11} = 48,828,125 experiments). The developed partitioning method will allow us to reduce this number. The total number of experiments needed for the first generation is the Cartesian product for the four "Primary mode" activities only (i.e. 5^4 = 625) and so on 625 for each of generation 2 and 3 adding to a total of 1875 experiments (0.0038% of the total population of experiments).

Figure 4.10 shows an illustration of all possible combinations of activity modes at each generation of the illustrative example in hand, where each activity is admitted with k number of modes.

When blocking the modes at a given generation, we sacrifice the ability to estimate the interaction effects between the modes. This is referred to as confounding in building the factorial design of experiments. A detailed explanation on the concept of confounding can be found in Douglas (2019).

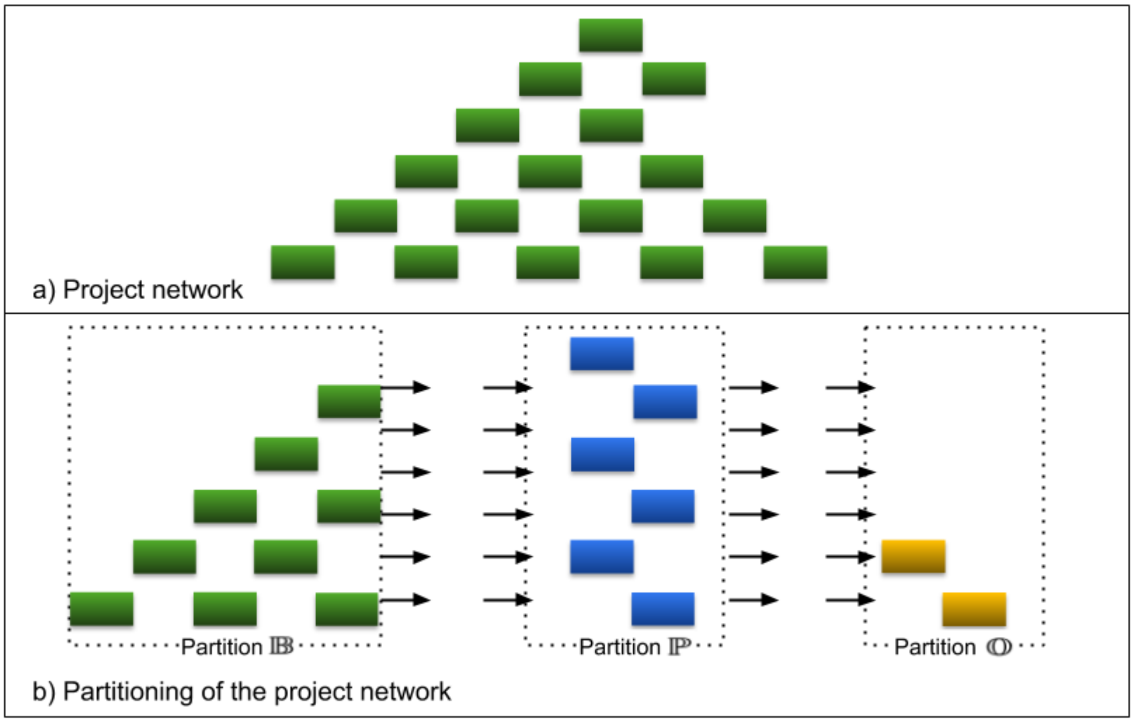


Figure 4.8 Illustration for the project network partitioning.

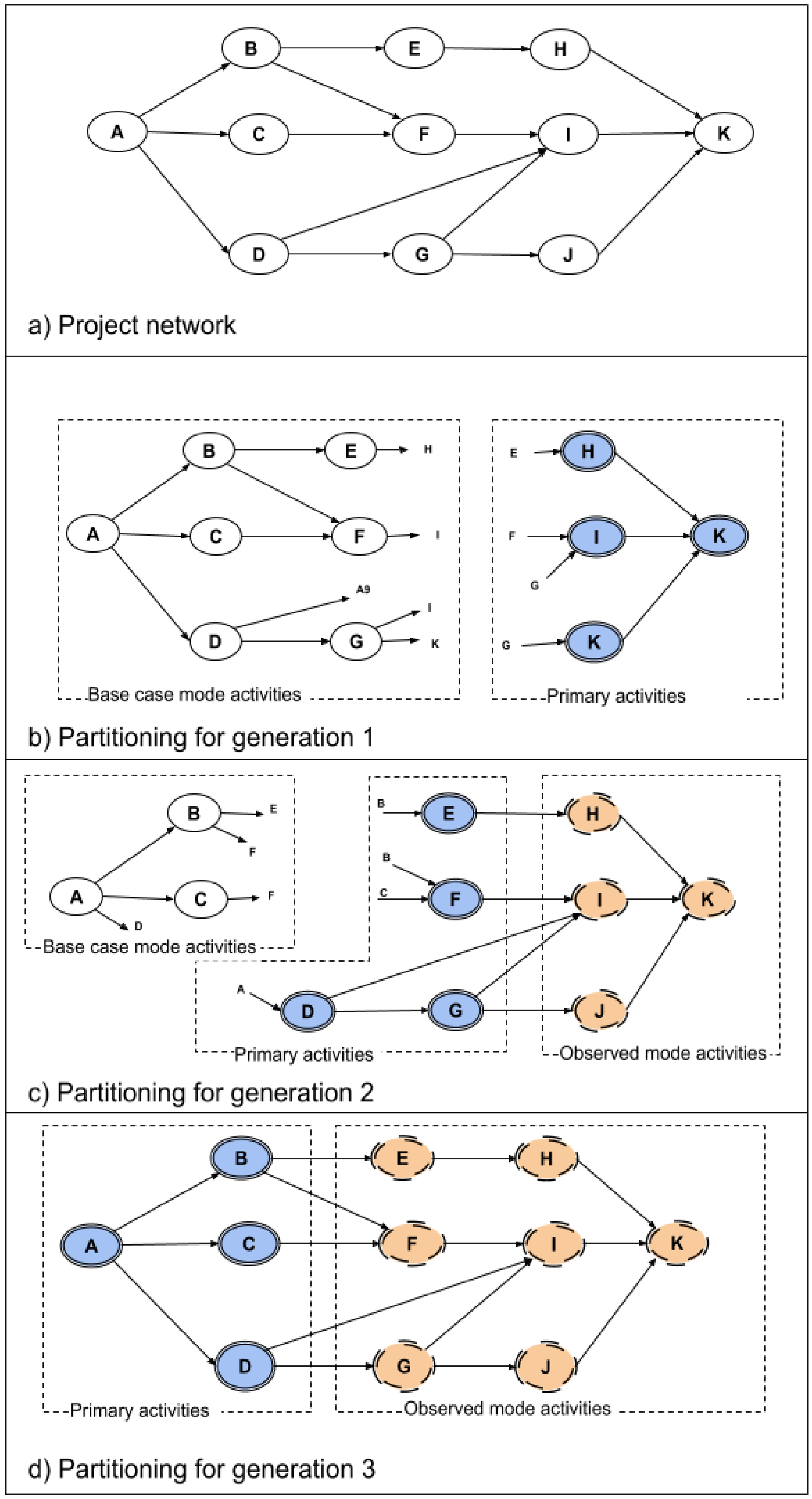


Figure 4.9 Illustrative example for the project network partitioning at each generation.

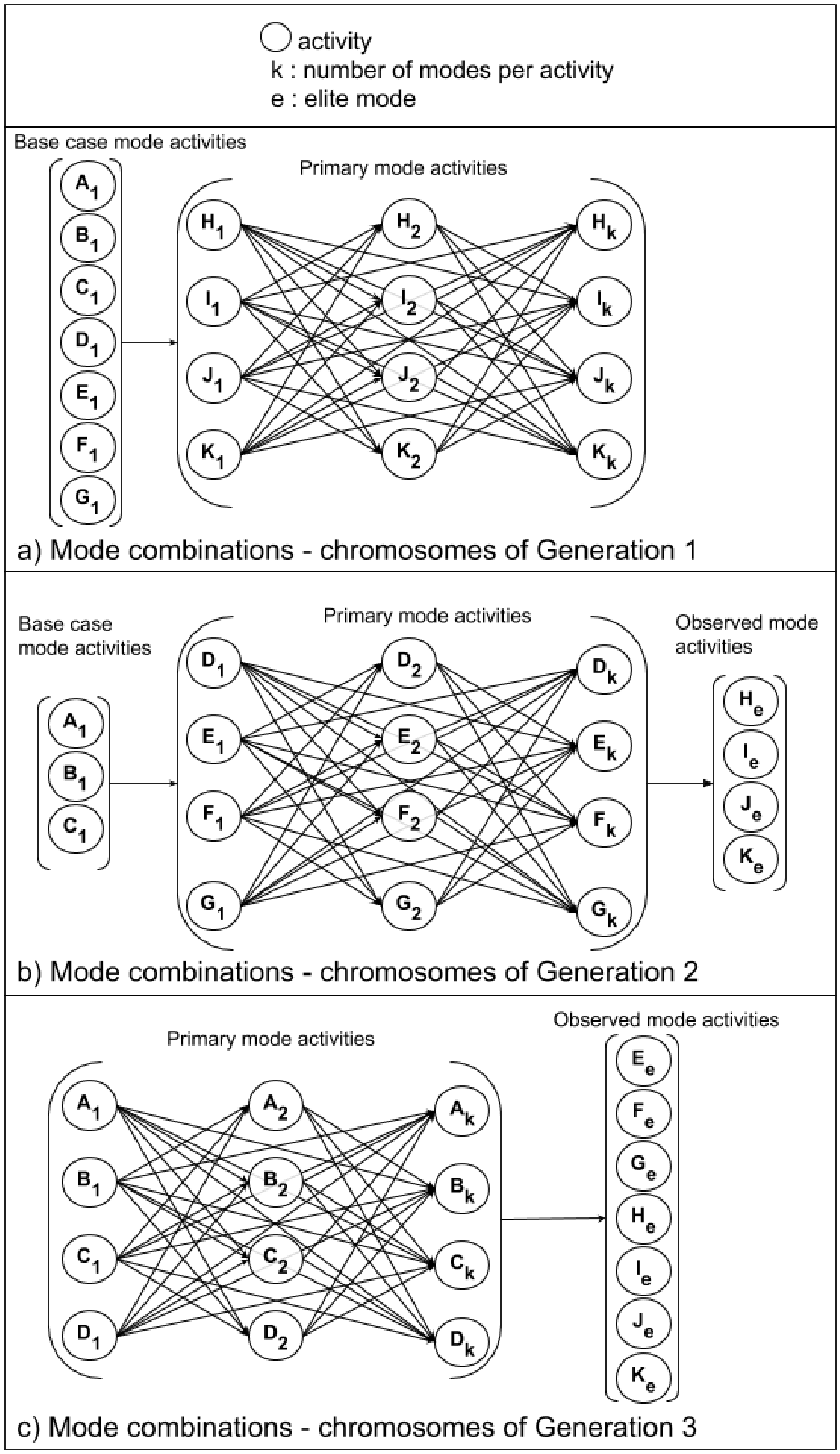


Figure 4.10 Illustrative example for mode combinations – chromosomes of each generation.

4.6. Formulation of the bi-objective fitness function

The developed ESDTCT is tailored to identify the supreme solution (non-dominated solution) representing the combination of modes that yields the single-objective optimization for cost or schedule minimization, or a bi-objective optimization method for cost and schedule minimization; all in an uncertain environment to identify the optimal solution satisfying the specified joint confidence level of both time and cost.

The methodology for formulating the problem objective fitness function starts with building a JCL as detailed in section 3.5. The JCL starts with building a probabilistic cost-loaded schedule and systematically integrating cost, schedule, and uncertainty. The method facilitates the establishment of expectations and probabilities of meeting those expectations. It also offers a holistic view to achieve cost and schedule goals as well as contingency estimation for cost and schedule under a specified confidence level. A frontier curve is developed by joining all the possible combinations of cost and schedule solutions that satisfy a selected confidence level. Figure 4.11 shows a typical scatter plot for all possible solutions resulting from the Monte Carlo simulation of the cost loaded project schedule iterated for the possible values represented by the uncertainty profiles of the individual activity costs and durations; the chart also shows a 50% JCL frontier curve.

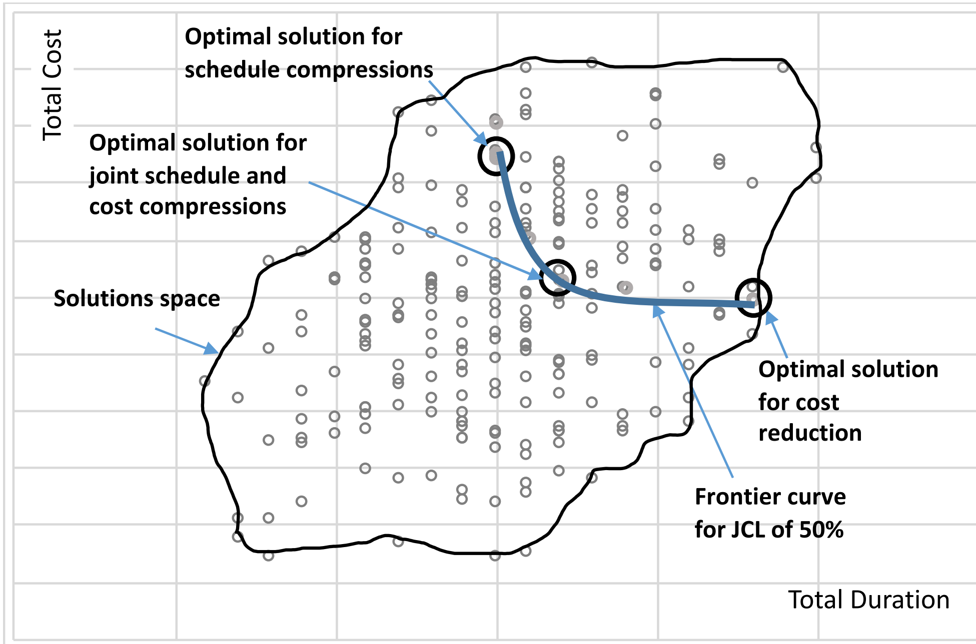


Figure 4.11. Typical JCL scatter plot and the frontier curve showing the different optimization objectives.

The developed ESDTCT method objective fitness function requires reducing the JCL analysis to a single solution. This single solution is determined based on the ultimate objective set forth by the analyst as follows:

1. Cost minimization: this is selected when the objective of the analysis is set to solve for the least cost solution falling on the frontier curve.

2. Schedule minimization: this is selected when the objective of the analysis is set to solve for the least project duration solution falling on the frontier curve.

3. Joint costs and schedule minimization: this is selected when the objective of the analysis is set to solve for the optimal non-dominated solution on the frontier curve while concurrently minimizing the project cost and schedule. This value is defined as the vertex solution on the frontier curve. The calculation method for finding the vertex is further detailed in the following sections. Figure 4.11 shows an illustration of the three objectives.

A calculation procedure is developed to determine the reduced vertex solution in the case of joint costs and schedule minimization as follows:

The cost and duration values are of different units of measurement and magnitudes; therefore, it is necessary to normalize the values so that each is scaled to values in the same range. The solutions found on the frontier are normalized to the range [0,1]. To do this, the nadir and ideal objective vectors can be used to normalize the objective values. The nadir objective vector was recognized and used by the multiple criteria decision-making researchers since the early 1970s. The nadir objective vector is a point constructed from the intersection of the maximum cost and schedule values from all the solutions falling on the frontier curve; similarly, the ideal objective vector is a point constructed from the intersection of the minimum cost and schedule values. The frontier curve cost and schedule values are then normalized to the range [0,1] using the equation (*4-1*):

$$D_j^{norm} = (D_j - D^{ideal}) / \left(D^{nadir} - D^{ideal}\right) \qquad \textit{4-1}$$

Where $\mathrm{D_j^{norm}}$ is the normalized time value for solution $j \in frontier\ curve$, $\mathrm{D^{ideal}}$ and $\mathrm{D^{nadir}}$ are the minimum and maximum time value respectively among the frontier curve solutions.

Similarly, the cost values are normalized using equation (*4-2*):

$$C_j^{norm} = (C_j - C^{ideal}) / \left(C^{nadir} - C^{ideal}\right) \qquad \textit{4-2}$$

The reduced solution of the JCL is the vertex point of the normalized frontier curve. The vertex point is defined as the solution providing the optimal balance between time and cost minimizations and is determined as the point having the shortest vector length connecting to the ideal vector point using the Pythagorean Theorem in equation (*4-3*).

$$l_j = \sqrt{\left(D_j^{norm}\right)^2 + \left(C_j^{norm}\right)^2} \qquad 4\text{-}3$$

Where l_j is the solution response vector length for solution $i \in pareto\ frontier$. Then the optimal solution $j_{optimal}$ is the vertex point solution that has the least response vector length. Figure 4.12 shows an illustration for the nadir and ideal vector points for a selected frontier curve and, Figure 4.13 shows an illustration of the normalized frontier curve.

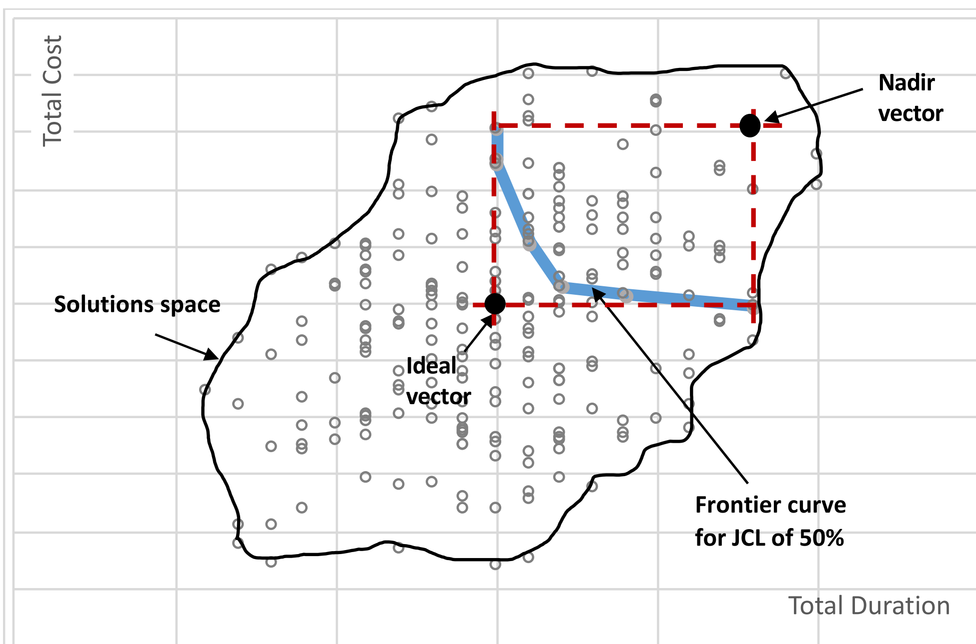


Figure 4.12. Nadir and Ideal vector points for the frontier curve.

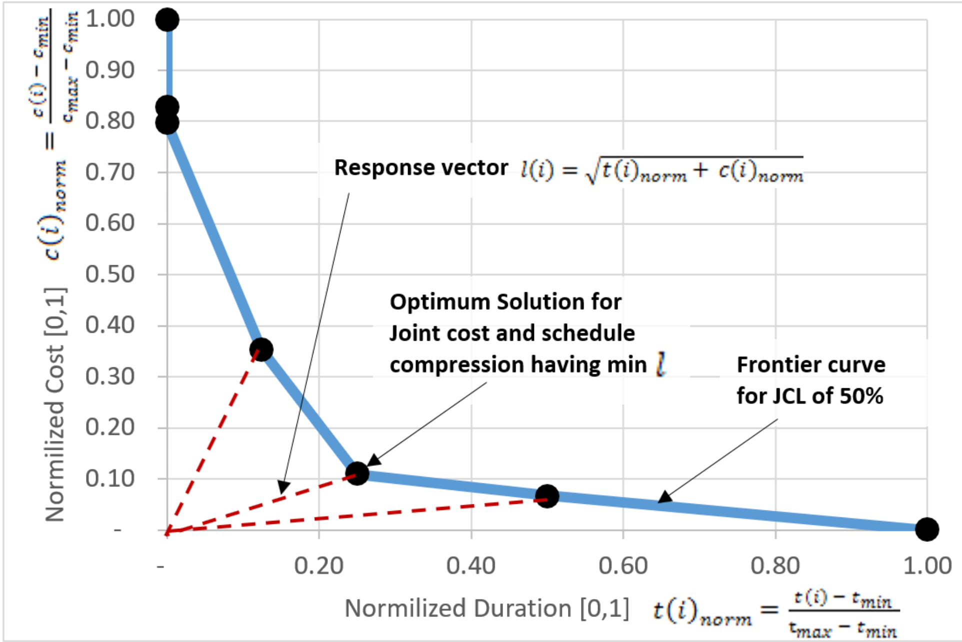


Figure 4.13. Normalized frontier curve.

Throughout the study, a total of 1000 Monte Carlo simulation runs are chosen to represent the solution space at each individual experiment run. This number is selected for the purpose of minimizing the amount of computations and is found to provide good accuracy of results as demonstrated in the test examples. The number of simulation runs can be adjusted according to user preference. The number of simulation runs have a direct relationship with the total execution run time.

4.7. Computational procedure and formulation

The ESDTCT method has two distinct modules; the first is performing a full factorial design of experiments with a blocking technique designated here as ESDTCT^{Exp}. The second module performs a random search of the solution space designated here as ESDTCT^{Rand}. Each of those modules provides answers and insights to the solution space for the stochastic time-cost trade-off problem. The computational procedure and outputs from those modules are explained in the following sections. The flow charts in Figure 4.14 and Figure 4.15 summarizes the $ESDTCT^{EXP}$ and $ESDTCT^{Rand}$ modules procedures developed in this study and coded in Google App Script and Google BigQuery SQL.

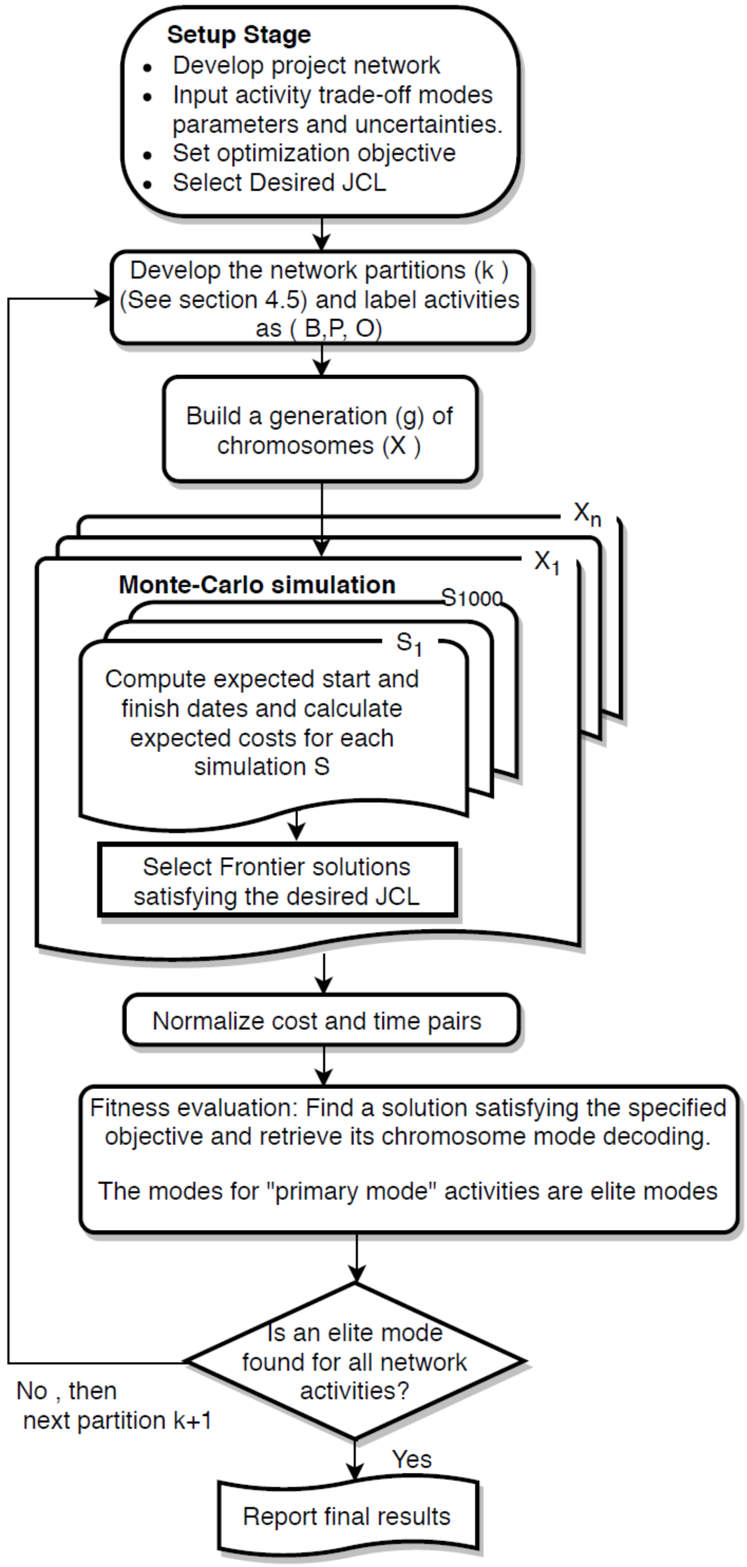


Figure 4.14. Flow chart of ESDTCT^{EXP} method.

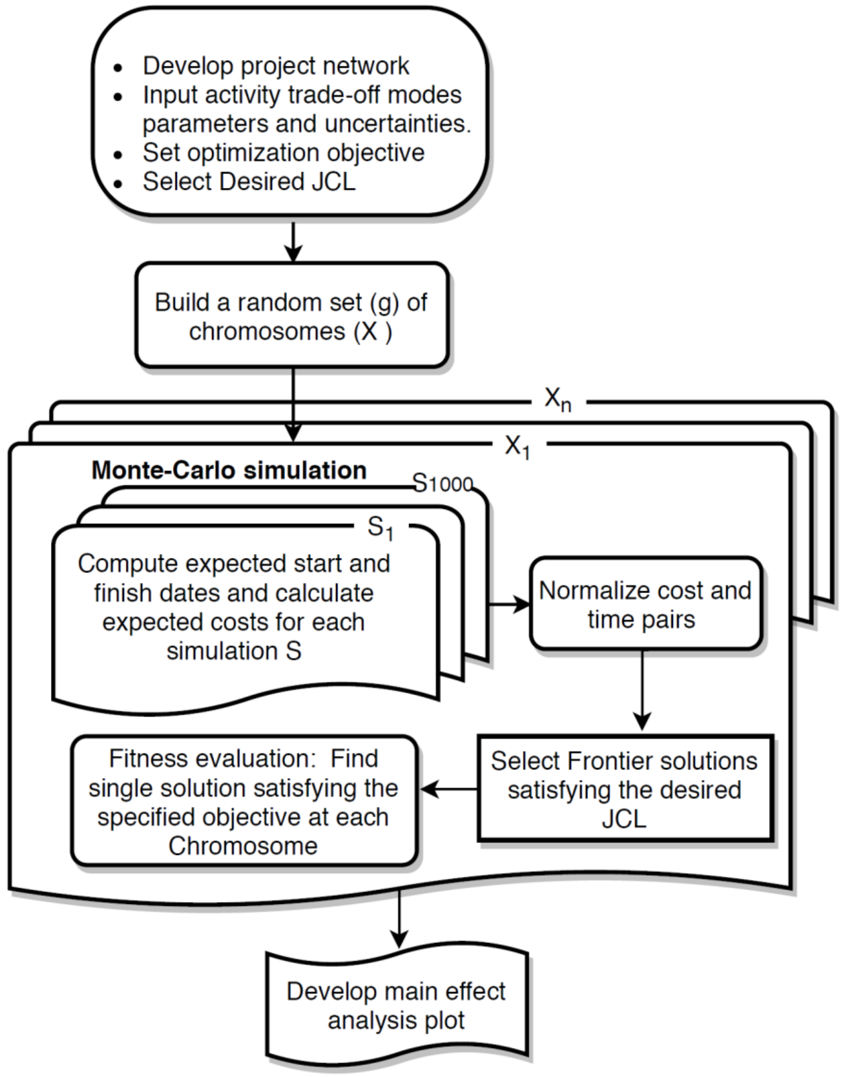


Figure 4.15. Flow chart of ESDTCT^{Rand} method.

4.7.1. Evolutionary experiment enumeration module: ESDTCT^{Exp}

A systematic approach is developed to find the supreme chromosome representing the combination of elite modes that result in the best overall project's cost and/or time minimization for a specified joint confidence level. The method intelligently reduces the number of experiments that are required to search for the supreme chromosome as opposed to the total number of experiments in a complete enumeration of the total solution space. This is achieved by dividing the computation steps into several generations, where each generation is a full factorial DOE matrix with a blocking technique that allows a full enumeration of all possible combinations of modes for primary activities while all other activities in the project network are blocked to a static mode. The primary activities in a generation are selected using the immediate reverse dominator tree (IRDT). The initial generation starts with the completion activity and all its IRDT activities as primary activity and runs backwards in the project network logic to select the next generation of primary activities; by doing so, the number of experiments is reduced to the Cartesian product of the primary activities. The methodology behind this process is to experiment on the mode interchangeability for activities immediately driving the completion activity to determine their elite modes that result in an optimum objective value. Those elite modes are memorized and carried forward as static modes in successive generations. However, the method ignores the higher-level interactions between the modes. In example, the selection of mode A in activity 1 is not correlated to the selection of mode B in activity 2. In an ideal situation, the entire population of the solution space should be studied to estimate those higher-level interactions, but this is almost impossible on large and complex projects. Such interactions will be further studied in future works.

The ESDTCT^{Exp} method can be formulated using the following set of equations:

$$Objective =$$

$$\begin{cases} Minimize\ \widetilde{C}\ \wedge\ Minimize\ \widetilde{D}\,, & for\ Joint\ cost\ and\ schedule\ minimization \\ Minimize\ \widetilde{C}\,, & for\ cost\ minimization \\ Minimize\ \widetilde{D}\,, & for\ schedule\ minimization \end{cases} \qquad 4\text{-}4$$

Constraints:

$$\mathbb{E}(EST_i) = \max_{\forall j \in P_i} \begin{cases} \mathbb{E}(EFT_j) + lag_{i,j} + 1 & , \ logic_{i,j} = Finish\ to\ Start \\ \mathbb{E}(EST_j) + lag_{i,j} & , \ logic_{i,j} = Start\ to\ Start \\ \mathbb{E}(EFT_j) + \ lag_{(i,j)} - \widetilde{d}_{\ i} + 1 & , \ logic_{i,j} = Finish\ to\ Finish \\ \mathbb{E}(EST_j) + \ lag_{(i,j)} - \widetilde{d}_{\ i} & , \ logic_{i,j} = Start\ to\ Finish \end{cases} \qquad 4\text{-}5$$

$$\mathbb{E}(EFT_i) \ = \ \mathbb{E}(EST_i) + \ \widetilde{d}_i - 1 \qquad 4\text{-}6$$

$$\widetilde{d}_\iota = \begin{cases} \sum_{k=1}^{K_i} g_{i,k} \mathbb{E}(d_{i,k} \mid p, a_{d(i,k)}, m_{d(i,k)}, b_{d(i,k)}) & \forall\ i \in \mathbb{P} \\ g_{i,ek} \mathbb{E}(d_{i,ek} | \ p, a_{d(i,ek)}, m_{d(i,ek)}, b_{d(i,ek)}) & \forall\ i \in \mathbb{O} \\ g_{i,k_1} \mathbb{E}(d_{i,k1} | \ p, \forall a_d, b_d = \ m_{d(i,k_1)}) & \forall\ i \in \mathbb{B} \end{cases} \qquad 4\text{-}7$$

$$\widetilde{FC}_i = \begin{cases} \sum_{k=1}^{K_i} g_{i,k} \mathbb{E}(FC_{i,k} \mid p, a_{FC(i,k)}, m_{FC(i,k)}, b_{FC(i,k)}) & \forall\ i \in \mathbb{P} \\ \mathbb{E}(FC_{i,ek} | \ p, a_{FC(i,ek)}, m_{FC(i,ek)}, b_{FC(i,ek)}) & \forall\ i \in \mathbb{O} \\ \mathbb{E}(FC_{i,k1} | \ p, \forall a_{FC}, b_{FC} = \ m_{FC(i,k_1)}) & \forall\ i \in \mathbb{B} \end{cases} \qquad 4\text{-}8$$

$$\widetilde{VC}_i = \begin{cases} \sum_{k=1}^{K_i} g_{i,k} \mathbb{E}(VCr_{i,k} \mid p, a_{VCr(i,k)}, m_{VCr(i,k)}, b_{VCr(i,k)}). \widetilde{d}_\iota & \forall\ i \in \mathbb{P} \\ \mathbb{E}(VCr_{i,ek} | \ p, a_{VCr(i,ek)}, m_{VCr(i,ek)}, b_{VCr(i,ek)}). \widetilde{d}_\iota & \forall\ i \in \mathbb{O} \\ \mathbb{E}\left(VCr_{i,k1} \middle| \ p, \forall a_{VCr}, b_{VCr} = \ m_{VCr(i,k_1)}\right). \widetilde{d}_\iota & \forall\ i \in \mathbb{B} \end{cases} \qquad 4\text{-}9$$

$$\sum_{k=1}^{K_i} g_{i,k} = 1 \quad \forall\ i \qquad 4\text{-}10$$

$$g_{i,k} \ \in \{1,2,3, \ldots K_i\} \quad \forall\ i, k \qquad 4\text{-}11$$

$$\ddot{D} \ = \ \max_{\forall\ i} \{\mathbb{E}(EFT_i)\} \qquad 4\text{-}12$$

$$\ddot{C} = \ \sum_{\forall\ i}(\widetilde{FC}_\iota) + \sum_{\forall\ i}(\widetilde{VC}_i) + \left(\mathbb{E}(ICr \ |p, a, m, b) \ . \ddot{D}\right) \qquad 4\text{-}13$$

$$\ddot{C} = \ \ddot{C} + PC + BC \qquad 4\text{-}14$$

$$\mathbb{Q} = \prod_{g=1}^{G} \prod_{s=1}^{S} \dddot{D}_{(g,s)}, \dddot{C}_{(g,s)} \ \ | \ (P \left\{ \ddot{D} \right\} \le \alpha \ \wedge \ P \left\{ \ddot{C} \right\} \le \alpha) \qquad 4\text{-}15$$

$$\widehat{\mathbb{Q}} = \prod \ddot{D}\,, \ddot{C} \ \ \forall\ s \ \in$$

$$\mathbb{Q} \ | \begin{cases} Minimize\ \ddot{C} \ \wedge \ Minimize\ \ \ddot{D}\,, for\ \ Joint\ cost\ and\ schedule\ minimization \\ Minimize\ \ddot{C} \ , \hfill for\ cost\ minimization \\ Minimize\ \ddot{D} \ , \hfill for\ schedule\ minimization \end{cases} \qquad 4\text{-}16$$

$$\widetilde{D} = \ min \left\{\ddot{D}\right\} \ \forall\ s \ \in \widehat{\mathbb{Q}} \qquad 4\text{-}17$$

$$\widetilde{C} \ = \ min \{\ddot{C}\} \ \forall\ s \ \in \widehat{\mathbb{Q}} \qquad 4\text{-}18$$

The mathematical model is described by Equations (4-5 to 4-18) as constraints and Equations (4-4) for the objective functions. The objective can be set for one of three cases, the first is the minimization of the expected value of both the total project duration $\widetilde{D}$ and the total project cost $\widetilde{C}$ given a confidence level α to determine the joint cost and schedule. The second is to minimize $\widetilde{C}$ where the expected value of the total project cost minimization is of interest and the third is to minimize $\widetilde{D}$ when the expected value for the total project schedule minimization is of interest. In equation (4-5 to 4-7), $\mathbb{E}(\mathrm{EST_i})$ is the expected value of the early start time for activity i; $\mathbb{E}\left(\mathrm{EFT_j}\right)$ is the expected value of the early finish time for its predecessor activity j that belongs in the predecessor set P_i. While $\widetilde{d}$ is the expected value for the activity duration resulting from random sampling of the probability distribution function (PDF). To allow calculation of the expected value, the inverse transformation of the cumulative distribution function is calculated. In the case of triangular PDF, the parameters are (p, a, m, b). Where p is a pseudo-random number generator of a uniform random variable $\in [0,1]$, and a, m, b are the optimistic, most likely and pessimistic values. Other PDFs can be used as detailed in section 3.4. The calculation for the EST and the EFT in equation (4-5) allow for the four types of logical relationships between the activities. Equation (4-7) computes the value of $\tilde{d}$ based on labeling the activity at a given generation. The classification of the network activities is $\mathbb{P}$ for “primary mode” activities, $\mathbb{O}$ for “Observed mode“ activities and $\mathbb{B}$ for “Base case mode” activities. Those labels are described in Section 4.5. The expected value of $\tilde{d}$ is sampled depending on the activity mode assignment, where each activity i can assume a different mode

k amongst the total number of modes K available for the activity. Similarly, equation (4-8) and (4-9) computes the expected value for the fixed cost ($\widetilde{\mathrm{FC}}$) and the variable cost ($\widetilde{\mathrm{VC}}$) for each activity. The resultant matrix from equations (4-7 to 4-9) is a full factorial design of experiments with a blocking technique. Where the activity label belongs to $\mathbb{P}$, the set of parameter values for $\tilde{d}\,, \widetilde{\mathrm{FC}}\,, \widetilde{\mathrm{VC}}$ assume the probabilistic values for all the admitted modes $k = 1 \; \rightarrow K_i$. When the activity label belongs to $\mathbb{O}$, the parameters are blocked to assume only the probabilistic values of the elite mode em and, when the activity label belongs to $\mathbb{B}$, the parameters are blocked to assume only the deterministic values for the base case mode (or mode 1) k_1. The reasoning behind this blocking technique is further described in Section 4.5. Binary variables $g_{i,k}$ in equations (4-10) and (4-11) expresses that only one mode must be admitted for each activity. Equations (4-12) and (4-13) computes the total project duration $\ddot{D}$ and total project cost $\ddot{C}$ at any given simulation run taking into account the expected value of the indirect cost rate per day (ICr). Equation (4-14) adds the penalty cost (PC) and the bonus costs (BC), formulation for such costs were detailed in section 3.3. Equation (4-15) is a matrix $\mathbb{Q}$ of all solutions falling on the frontier curve defined by a specified joint cost – schedule confidence level α. Equation (4-16) provides the matrix $\widehat{\mathbb{Q}}$ which is a subset of $\mathbb{Q}$ for all solutions s satisfying the defined minimization objective function. Equations (4-17) and (4-18) reduces the matrix $\widehat{\mathbb{Q}}$ to the single optimum solution satisfying the objective function.

4.7.2. Complete Random experiments module: ESDTCT^{Rand}

The systematic approach in this module is developed to explore the solution space through a completely randomized DOE sample from the total solution space. The objective here is to predict, approximately, the impact of mode changes on the overall project's cost and time for a specified joint confidence level of time and cost. This provides the decision-maker with an approximate indication of how the project cost and duration will react to alternate modes that may not be the optimal modes but selected based on the project manager's judgement. A typical main effect plot is illustrated in Figure 4.7. This procedure is concurrently computed with the experiment generation module procedure.

The ESDTCT^{Rand} method can be formulated using the following set of equations:

$$Objective =$$

$$\begin{cases} Minimize\ \widetilde{C} \ \wedge \ Minimize\ \widetilde{D}\,, for\ \ Joint\ cost\ and\ schedule\ minimization \\ Minimize\ \widetilde{C}\ , \qquad\qquad\qquad\qquad for\ cost\ minimization \\ Minimize\ \widetilde{D}\ , \qquad\qquad\qquad\qquad for\ schedule\ minimization \end{cases} \tag{4-19}$$

Constraints:

$$\mathbb{E}(EST_i) = \max_{\forall\, j\, \in P_i} \begin{cases} \mathbb{E}(EFT_j) + lag_{i,j} + 1 & , \ logic_{i,j} = Finish\ to\ Start \\ \mathbb{E}(EST_j) + lag_{i,j} & , \ logic_{i,j} = Start\ to\ Start \\ \mathbb{E}(EFT_j) + \ lag_{(i,j)} - \tilde{d}_i + 1 & , \ logic_{i,j} = Finish\ to\ Finish \\ \mathbb{E}(EST_j) + \ lag_{(i,j)} - \tilde{d}_i & , \ logic_{i,j} = Start\ to\ Finish \end{cases} \tag{4-20}$$

$$\mathbb{E}(EFT_i) \ = \ \mathbb{E}(EST_i) + \ \widetilde{d}_i - 1 \tag{4-21}$$

$$\widetilde{d_\imath} = \ g_{i,k_\mu}\mathbb{E}(d_{i,k_\mu} \mid p, a_{d(i,k_\mu)}, m_{d(i,k_\mu)}, b_{d(i,k_\mu)}) \tag{4-22}$$

$$\widetilde{FC}_i = \ g_{i,k_\mu}\mathbb{E}(FC_{i,k_\mu} \mid p, a_{FC(i,k_\mu)}, m_{FC(i,k_\mu)}, b_{FC(i,k_\mu)}) \tag{4-23}$$

$$\widetilde{VC}_i = \ g_{i,k_\mu}\mathbb{E}(VCr_{i,k_\mu} \mid p, a_{VCr(i,k_\mu)}, m_{VCr(i,k_\mu)}, b_{VCr(i,k_\mu)})\ .\ \widetilde{d_\imath} \tag{4-24}$$

$$\textstyle\sum_{k=1}^{K_i} g_{i,k} = 1 \quad \forall\ i \tag{4-25}$$

$$g_{i,k} \in \{1,2,3, \dots K_i\} \quad \forall\, i, k \qquad 4\text{-}26$$

$$\ddot{D} = \max_{\forall i}\{\mathbb{E}(EFT_i)\} \qquad 4\text{-}27$$

$$\ddot{C} = \sum\nolimits_{\forall i}(\widetilde{FC}_i) + \sum\nolimits_{\forall i}(\widetilde{VC}_i) + \left(\mathbb{E}(ICr \mid p, a, m, b) \,.\, \ddot{D}\right) \qquad 4\text{-}28$$

$$\dddot{C} = \ddot{C} + PC + BC \qquad 4\text{-}29$$

$$\mathbb{Q} = \prod\nolimits_{s=1}^{S} \dddot{D}_{(s)}, \dddot{C}_{(s)} \mid \left(P\{\ddot{D}\} \le \alpha \wedge P\{\ddot{C}\} \le \alpha\right) \qquad 4\text{-}30$$

$$\widehat{\mathbb{Q}} = \prod \ddot{D}, \ddot{C} \quad \forall\, s \in$$

$$\mathbb{Q} \mid \begin{cases} \text{Minimize } \ddot{C} \wedge \text{Minimize } \ddot{D}, & \text{for Joint cost and schedule minimization} \\ \text{Minimize } \ddot{C}, & \text{for cost minimization} \\ \text{Minimize } \ddot{D}, & \text{for schedule minimization} \end{cases} \qquad 4\text{-}31$$

The mathematical model is described by Equations (4-20 to 4-31) as constraints and Equations (4-19) for the objective functions. The formulation is similar to that of ESDTCT^{Exp} method detailed in section (4.7.1) with the difference in equations (4-22 to 4-24) where the expected values for $\tilde{d}, \widetilde{DC}$ and $\widetilde{VC}$ are calculated based on a complete random design of experiments where μ is a random general location parameter for a discrete mode k_i . The experimental runs are generated until an execution user-defined time or a computational cost threshold is reached. The resulting matrix $\widehat{\mathbb{Q}}$ provides a data set to generate the main effect plot from the following pseudocode:

```
FOR EACH Activity i
  FOR Mode j = 1 TO M // M = total number of modes assigned to
                                the activity.
    CASE: optimization is set for cost minimization: THEN
      Response (i,j) = Average (total cost (i,j))

    CASE: optimization is set for schedule minimization: THEN
      Response (i,j) = Average (total duration (i,j))

    CASE: optimization is set for
```

```
             Joint cost and schedule minimization: THEN
      Response (i,j) = Average (response vector (i,j))

    NEXT Mode
  Next Activity
  Plot the Main Effect Charts
  END
```

To enable the objective of this module, the developed procedure produces two charts. The first is the main effect chart; while, the second is a tornado chart for the activity relative importance factor (ARI) with regards to its effect on the objective function. The chart shows a measure for the correlation between the activity mode interchangeability to the overall project's response vector. The factor describes the likelihood that a change in the selection of an activity mode will cause a proportional change in the objective response. An activity with a high ARI suggests that the decision-maker should make careful selection amongst the activity admitted modes. The correlation factor (ARI) is calculated using the absolute value of Parsons correlation factor formulated using equations (4-32 to 4-34) with x representing the numerical number value of the admitted modes for the activity i of and y representing the response values of the objective function. An example of the tornado chart is shown in Figure 4.23.

$$ARI_i \; = \left| correlation(x_i, y) \right| \qquad \textit{4-32}$$

$$ARI_i \; = \left| \frac{\sum(x_i - \bar{x}_i)(y - \bar{y})}{\sqrt{\sum(x_i - \bar{x}_i)^2 (y - \bar{y})^2}} \right| \qquad \textit{4-33}$$

$$where \quad y = \begin{cases} I, & for\ Joint\ cost\ and\ schedule\ compression\ (see\ equation\ 4\text{-}3) \\ \ddot{C}, & for\ cost\ reduction \\ \ddot{D}, & for\ schedule\ compression \end{cases} \qquad \textit{4-34}$$

4.8. Numerical Examples

The following series of numerical examples is set to illustrate the basic concept and test the performance and accuracy of the developed ESDTCT modules. The results are compared against those published by other researches using meta-heuristic optimization algorithms for the discrete time-cost trade-off problem. The examples 1 to 3 are performed on a relatively small project having 18 activities network. This example is drawn from the literature and was solved by several researchers under certain and uncertain TCTP optimization algorithms. The rich data obtained from the literature will allow us to validate and compare performance and accuracy for the developed method.

Examples 4 and 5 are performed on a large size 63 activity project also drawn from the literature. Previous researchers solved this example under certain environment, and to the best knowledge on the literature research made in this field, no attempt has previously been made to solve this size problem under uncertain environment due to the excessive solution space. The sixth and seventh examples are solved for the 18-activities network of example 1 to incorporate and test the effect of discrete risk events.

4.8.1. Example 1: Testing ESDTCT^{Exp} module on a small size project under uncertain environment.

To demonstrate the application of the developed method, a simple 18 activities network example is used. The example was originally developed by Feng et al. (1997) and Hegazy (1999) who solved the TCTP in GA-based application under crisp cost and time values, i.e., deterministic solution. The same example was later extended to account for uncertainty by Eshtehardian et al. (2009) and solved using a multi-objective genetic algorithm (MOGA) using fuzzy time and cost values. Kalhor et al. (2011) also solved the same example using a fuzzy non-dominated archiving ant colony approach (NA-ACO) and confirmed the same results obtained by the aforementioned MOGA approach. The network configuration of the example is shown in Figure 4.16. For each activity, there are two to five alternative modes making a total number of 5,9 billion possible mode alternatives. Each mode has a time and associated cost of which their uncertainties are expressed by a triangular function defined by the optimistic, most likely and, pessimistic estimates as presented in Table 4.1. The indirect cost is identical to that of Eshtehardian et al. (2009), i.e., triangular probability function with $185 optimistic, $200 most likely, and $235 pessimistic values. The variable cost for the activities was not considered by Eshtehardian; therefore, has been set to zero to allow for proper comparison of the results. The objective of the problem is to identify the supreme chromosome solution that results in schedule minimization that satisfies a 50% joint cost and schedule confidence level. The performance criteria are set such that the optimal chromosome result should be no worse than the aforementioned optimal values using the MOGA and the NA-ACO approaches.

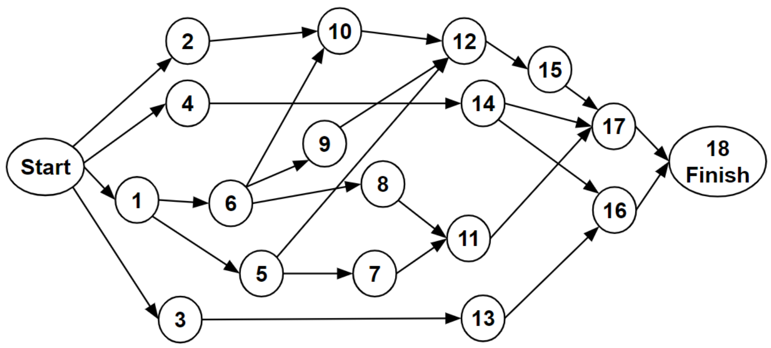


Figure 4.16 Example 1: Network configuration

The ESDTCT^{Exp} procedure took five generations to calculate the supreme mode of all the activities and accordingly arrive at the supreme chromosome. The overall number of experimental chromosome runs is 844 as compared to the total size of the solution space of 5,9 billion possible chromosomes. The example was solved in under 3 min using the programing approach described earlier in section 4.2. The selection of primary activities at each generation is shown in Figure 4.17. The summary results of each generation are included in Table 4.2. Due to the extensive size of data, the details about results from each generation are not presented for which a cumulatively 850 thousand simulation iterations were performed.

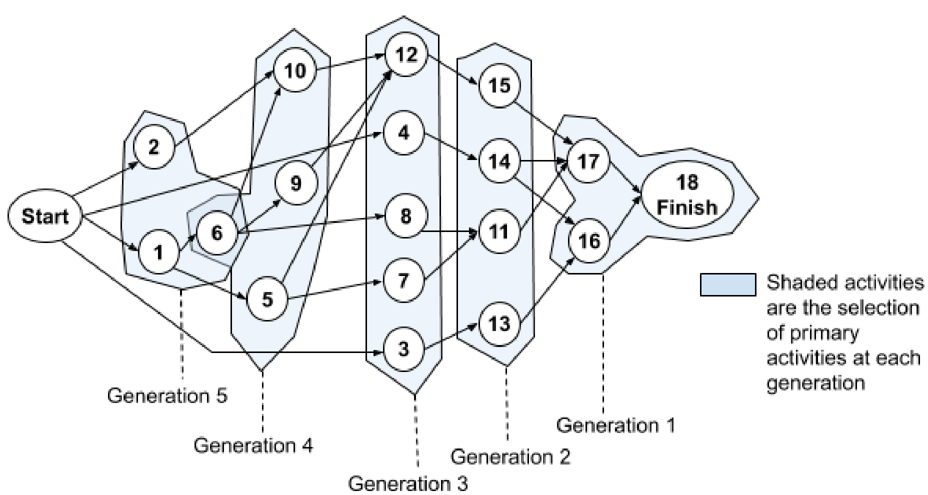


Figure 4.17 Example 1: Network configuration showing primary activities at each generation in the ESDTCT method.

The developed evolutionary experiment enumeration module starts with partitioning the project network to identify the first generation of blocked activities. The primary activities in the first generation are identified by the project completion activity and all its immediate predecessors. In this case, activities 18,17 and 16 are considered primary and assume all their respective admitted modes, while all other activities are blocked at their deterministic duration and cost of their base case mode (mode 1). By doing so, we can evaluate the effects of mode variability of primary activities on the overall project's time and cost. The resulting design of experiment matrix is then the multiplications of the number of modes for the primary activities. i.e., 5 x 3 x 3 = 45 runs. The MCS is performed using 1000 iterations for each of the 45-design experiment runs. At each run, the results of the MCS are collected, and the solutions that fall on the 50% joint confidence level frontier are then analyzed to obtain the supreme chromosome using equation 4.1 and 4.2. The modes of the supreme chromosome for the primary activities are then defined as the elite modes. The second generation is then developed by blocking the immediate predecessor activities for the activities that were considered primary in the previous generation, in this case, activities 11,13,14 and 15. Those activities assume all their respective admitted modes while the elite modes found from the previous generation are held constant for their respective activity and the balance of activities are held constant and taken at their most likely duration and cost of their respective mode 1. This process is repeated to produce successive generations until all the project activities are solved for their elite modes. The supreme chromosome is then determined as the solution that holds all elite modes.

The developed ESDTCT is completely coded in google app script and integrated to google sheets and the computational power of Google BigQuery. By using the free – quota limits of Google BigQuery, the example in hand was solved in under 3 minutes. This duration may decrease significantly by linking the BigQuery to a billable account where quota limits can increase significantly.

Table 4.1 Example 1: Data input (adapted from Eshtehardian et al. 2009)

	Mode **Duration in days (Optimistic, Most Likely, Pessimistic)** **Fixed cost $ (Optimistic, Most Likely, Pessimistic)**			Triangular Probability Distribution Function selected for both time and cost	
Activity	**Mode 1**	**Mode 2**	**Mode 3**	**Mode 4**	**Mode 5**
1	(10,14,19) (2100,2400,2890)	(11,15,20) (1900,2150,2560)	(13,16,20) (1720,1900,2280)	(16,21,24) (1250,1500,2000)	(19,24,31) (985,1200,1750)
2	(12,15,21) (2870,3000,3420)	(16,18,23) (2185,2400,2850)	(18,20,25) (1650,1800,2255)	(20,23,28) (1300,1500,1950)	(19,25,30) (900,1000,1190)
3	(11,15,23) (4250,4500,4990)	(17,22,29) (3850,4000,4460)	(29,33,40) (2985,3200,3560)		
4	(10,12,16) (42050,45000,48800)	(13,16,21) (38500,35000,39000)	(17,20,28) (28500,30000,33550)		
5	(19,22,25) (18500,20000,22550)	(22,24,27) (16000,17500,19950)	(24,28,33) (14150,15000,17050)	(29,30,34) (8500,10000,12600)	
6	(12,14,17) (38500,40000,42860)	(17,18,21) (29800,32000,34550)	(21,24,29) (16550,18000,21000)		
7	(8,9,10) (28500,30000,33670)	(11,15,19) (21670,24000,28560)	(16,18,23) (20000,22000,23560)		
8	(11,14,16) (185,220,282)	(13,15,19) (182,215,255)	(13,16,21) (182,200,245)	(17,21,25) (175,208,234)	(21,24,29) (110,120,132)
9	(11,15,20) (290,300,313)	(16,18,23) (212,240,288)	(18,20,22) (165,180,225)	(20,23,28) (125,150,196)	(21,25,28) (85,100,124)
10	(13,15,19) (420,450,492)	(21,22,27) (385,400,485)	(30,33,39) (290,320,356)		
11	(10,12,16) (410,450,510)	(14,16,20) (313,350,395)	(16,20,26) (284,300,352)		
12	(18,22,29) (1850,2000,2450)	(19,24,30) (1565,1750,2050)	(20,28,36) (1325,1500,1880)	(21,30,45) (915,1000,1350)	
13	(12,14,18) (3650,4000,4540)	(16,18,20) (2970,3200,3385)	(23,24,26) (1595,1800,2160)		
14	(7,9,11) (2580,3000,3685)	(13,15,19) (2200,2400,2880)	(16,18,23) (2080,2200,2850)		
15	(10,12,15) (4385,4500,4850)	(13,16,18) (3200,3500,3750)			
16	(18,20,23) (2650,3000,3850)	(19,22,26) (1850,2000,2480)	(20,24,30) (1340,1750,2240)	(22,28,34) (1250,1500,1950)	(23,30,39) (860,1000,1320)
17	(12,14,18) (3750,4000,4670)	(15,18,22) (3000,3200,3530)	(22,24,29) (1650,1800,2140)		
18	(8,9,12) (2850,3000,3575)	(11,15,20) (2030,2400,2950)	(14,18,22) (1950,2200,2660)		
Indirect cost $ (185, 200, 235)					

Table 4.2 Example 1: ESDTCT^{Exp} supreme chromosome results at each generation (for a JCL = 50%)

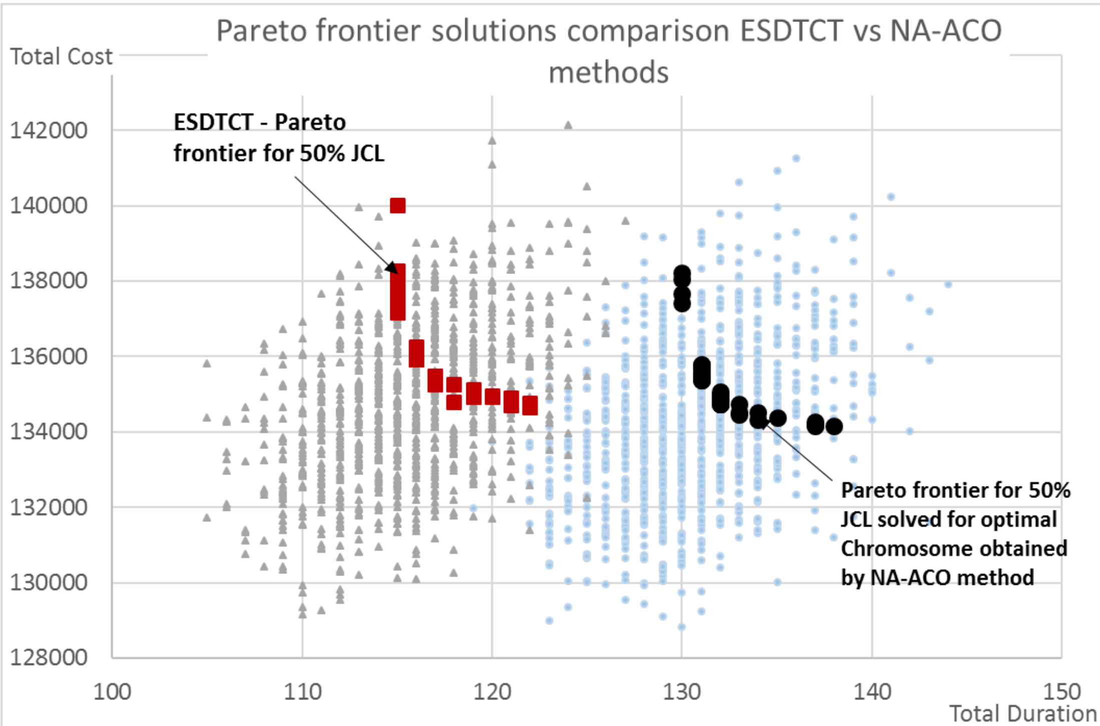
			Activity ID >>	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	1
			Number of admitted modes>>	5	5	3	3	4	3	3	5	5	3	3	4	3	3	2	5	3	3
Generation	No. of Experimental runs	Total duration	Total cost	<< Chromosome >>																	
1	45	102	$191,053	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	5	1	1
2	54	102	$188,518	1	1	1	1	1	1	1	1	1	1	3	1	3	2	1	5	1	1
3	540	103	$165,896	1	1	3	3	1	1	3	4	1	1	3	1	3	2	1	5	1	1
4	180	115	$138,555	1	1	3	3	4	3	3	4	1	1	3	1	3	2	1	5	1	1
5	75	118	$134,795	1	5	3	3	4	3	3	4	1	1	3	1	3	2	1	5	1	1
Supreme Chromosome →		118	$134,795	1	5	3	3	4	3	3	4	1	1	3	1	3	2	1	5	1	1
	∑ = 894 runs																				

$Activity\ i \in \mathbb{O}$, elite mode
$Activity\ i \in \mathbb{P}$, variable modes
$Activity\ i \in \mathbb{B}$, Most likely value of mode 1

Interpretation of results:

It is noted that the supreme chromosome generated by the developed ESDTCT^{Exp} method is different from that reported by Kalhor et al. (2011) using the NA-ACO method referenced in Figure 4.18, where activities 1, 8, 14, 15 and 17 have resulted in a different elite mode. To compare the two results, a hypothesis is made here that the supreme chromosome resulting from ESDTCT^{Exp} is better than that produced by the MOGA and NA-ACO methods. To test the hypothesis, the Monte Carlo simulation is performed to identify the JCL frontier curve for the optimal chromosome reported from the MOGA and NA-ACO methods. Figure 4.18 shows the difference in those results where the ESDTCT^{Exp} supreme chromosome resulted in a solution with a slightly different cost but a significantly lower overall project's duration; therefore, it can be concluded here that the performance of the developed ESDTCT^{Exp} is superior to that of the MOGA and NA-ACO methods.

The effect of the decision-maker's appetite for risk acceptance is represented by his/her selection of a joint confidence level (JCL). The higher the JCL, the more confidence the project is in meeting the resultant cost and schedule values. The effect of diverse selection of JCL is tested. Results of using a JCL value of 20%, 50%, and 80% are listed in Table 4.3. The analysis shows that solving the problem for the objective of joint cost and schedule minimization resulted in different supreme chromosome combination for different JCL values as noticed in differing modes for activities 2,3,8 and 11.



Activity ID >>>>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ESDTCT Supreme Chromosome	1	5	3	3	4	3	3	4	1	1	3	1	3	2	1	5	1	1
NA-ACO Optimal Chromosome *	3	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	3	1
Differences	⇧							⇧						⇧	⇧		⇧	

* Obtained by Kalhor et al. (2011)

Figure 4.18 Example 1: ESDTCT supreme chromosome frontier curve results verses NA-ACO method optimal chromosome reported by (Kalhor et al. 2011).

Another parametric study is also made to test the effect of the decision-maker's objective settings for the optimization problem. The problem is solved for the three objectives of cost minimization, schedule minimization, and joint cost-schedule minimization. The analysis results are listed in Table 4.3. It can be noticed that different objectives produced different supreme chromosome combinations. For example, the supreme chromosome at the specified 50% JCL is almost entirely different for each optimization objective. In other words, different supreme chromosome solutions are found more fitting as a result of the objective settings and the specified JCL.

Presumably, the “base case” is the one with the initial mode assignment for each activity. To compare against the base case scenario the mode identifier 1 for each activity is assigned here as the base case mode. The deterministic and probabilistic analysis is performed on the base case chromosome. The results of the probabilistic JCL analysis for a JCL percentage of (20%, 50%, and 80%) and the results of the deterministic analysis based on the most likely cost and schedule values of the activity respective base case mode (1) are listed in Table 4.3. Figure 4.20 shows the cost and schedule contingency calculations for the base case scenario for the three objective settings at the 50% JCL. Figure 4.21 shows that the probabilistic time-cost trade-off resulted in better chromosomes to that of the base case with lower project costs. For example, the supreme chromosome from the joint cost-schedule minimization objective at a 50% JCL resulted in a $62,209 (31%) less cost than the base case with an additional 12 days.

Table 4.3 Example 1: ESDTCT optimal chromosome for different JCL and different optimization objectives.

				Activity ID >>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
				Num of available modes >>	5	5	3	3	4	3	3	5	5	3	3	4	3	3	2	5	3	3
Study Type	Optimization objective	Selected joint cost and schedule confidence level %	Total duration	Total cost	<< Chromosome >>																	
Probabilistic Trade-off	Schedule minimization	20%	100	$198,356	1	3	1	1	1	1	1	5	1	1	3	1	2	3	1	3	1	1
		50%	104	$181,280	1	4	1	1	3	1	3	1	1	1	1	1	2	1	1	5	1	1
		80%	107	$174,272	1	3	1	3	2	1	2	1	1	1	3	1	2	3	1	3	1	1
Probabilistic Trade-off	Cost minimization	20%	125	$132,310	2	5	3	3	4	3	3	5	2	1	3	2	3	3	2	5	1	1
		50%	128	$133,786	3	5	3	3	4	3	3	2	1	1	3	1	3	3	2	5	2	1
		80%	129	$135,690	2	5	3	3	4	3	3	3	1	1	3	1	3	3	2	5	2	1
Probabilistic Trade-off	Joint cost - schedule Minimization	20%	111	$134,177	1	5	2	3	4	3	3	4	1	1	1	1	3	2	1	5	1	1
		50%	118	$134,795	1	5	3	3	4	3	3	4	1	1	3	1	3	2	1	5	1	1
		80%	118	$138,458	1	4	3	3	4	3	3	1	1	1	1	1	3	2	1	5	1	1
Base case - Probabilistic	Schedule minimization	20%	103	$195,279	<< (Mode 1 of each activity) >>																	
		50%	104	$200,537																		
		80%	108	$200,245																		
	Cost minimization	20%	107	$194,321																		
		50%	108	$196,660																		
		80%	113	$199,095																		
	Joint cost - schedule Minimization	20%	103	$195,004																		
		50%	106	$197,004																		
		80%	108	$199,408																		
Base case - Deterministic		100%	100	$189,820	<< (Mode 1 of each activity at Most Likely cost and duration values) >>																	

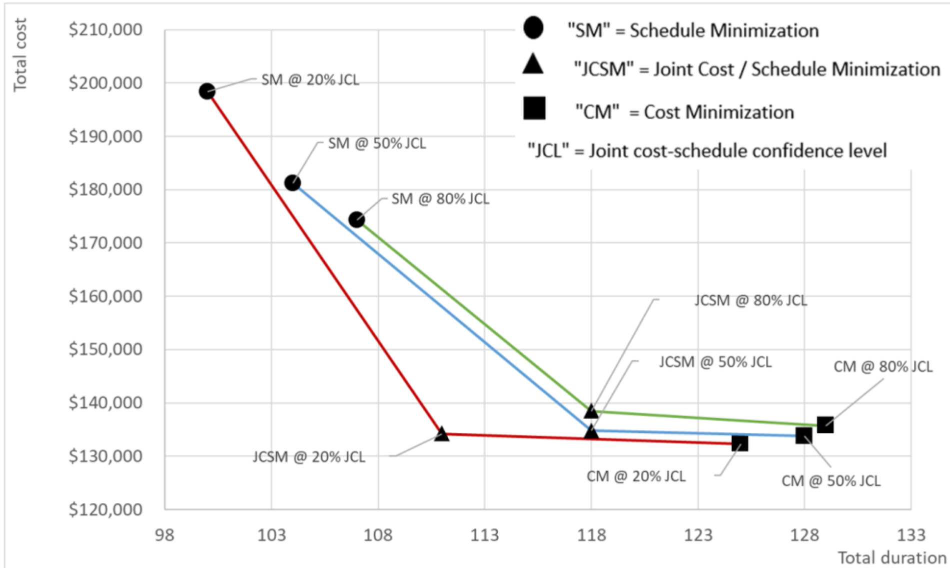


Figure 4.19 Example 1: ESDTCT^{Exp} optimal chromosome for different JCL and different optimization objectives

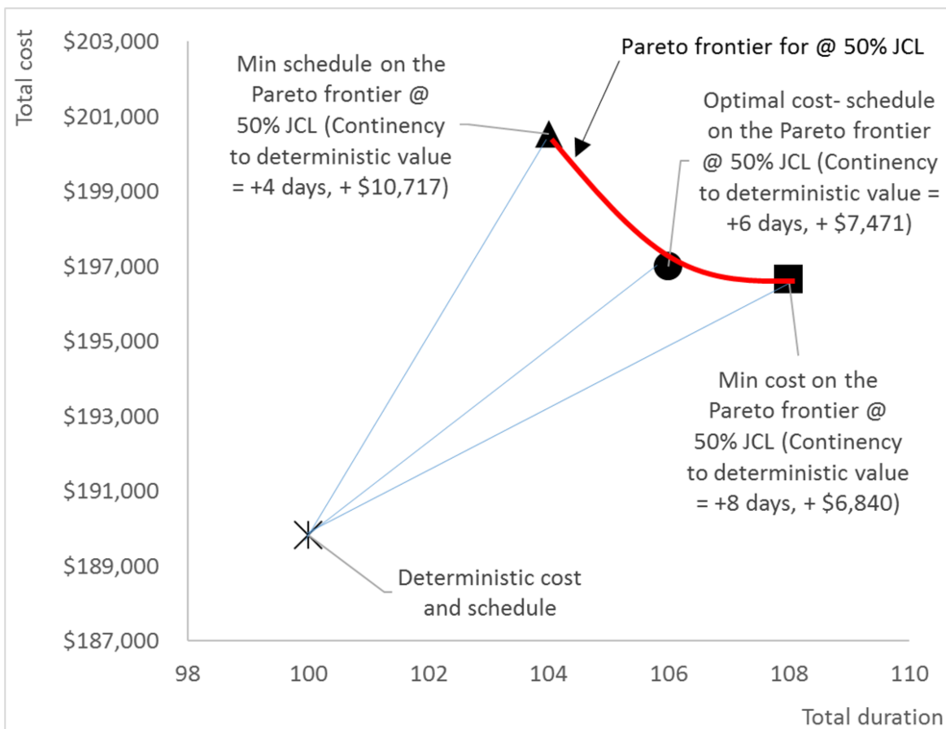


Figure 4.20 Example 1: base case 50% JCL frontier curve showing cost and schedule contingency calculations.

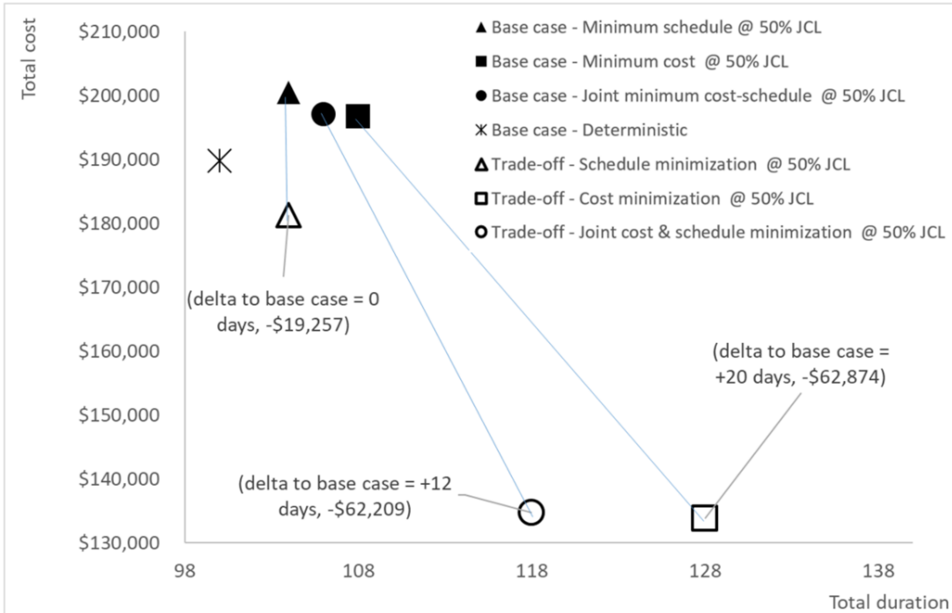


Figure 4.21 Example 1: ESDTCT^{Exp} optimal chromosome for different JCL and different optimization objectives.

4.8.2. Example 2: Testing ESDTCT^{Rand} module on a small size project under uncertain environment.

To test the developed ESDTCT^{Rand} module, the example is solved to obtain the main effect plot that represents the most influential activities and variables (modes) for the total project cost and time at the specified 50% joint confidence level for the objective of joint overall time and cost minimization. A random search of the total solution space is performed, and a total of 75 million iterations were made for a 750 thousand random selection of chromosomes in under 4 minutes. This random selection represents 0.01% of the total solution space. The results of this simulation were collected, and the main effect plot is generated by taking the average of the response vector length against each variable mode. The main effect plot is a tool that can indicate to the project manager what would be the effect of choosing an alternative mode for an activity for convenience. The project manager can then decide the desired chromosome of modes that results in an acceptable objective which may not be the most optimal chromosome. The main effect plot for the example in hand is shown in Figure 4.22.

Interpretation of the ESDTCT main effect plot

By visual analysis of the plot, it can be noticed, for example, that there are insignificant differences between choosing any of the 5 modes associated with activities 8. The reason for these insignificance differences could be related to any or a combination of the following three items:

- The activity in hand has a relatively small cost value assigned to the modes of this activity as compared to the other project activities or as compared to the overall project's cost,
- Another reason can be due to the non-criticality of this activity, which does not significantly affect the overall project's duration,
- A third reason may be due to the significant overlap of the probability distributions among the different mode's cost and duration estimates; in other words, there are insignificant differences in time and cost distributions between the activity modes.

Conversely, activity 4 is observed to have a wide range influence on the objective response vector where the small response vector value indicates a more optimal solution resulting in an expected lower cost and duration of the project. Similarly, the interpretations of this influence can be explained by the following items:

- The activity has a relatively significant cost value over the entire project,
- A small overlap or no overlap in some cost and duration estimates among the different modes.
- The activity is on the critical path and variations in durations are highly correlated to the increased project duration and hence, the indirect cost.

While there is a more significant reduction in cost and duration when choosing mode 3 for the said activity, it is worth noting here that the main effect plot only explored a small sample of the total solution space and can only provide a rough indication of the effects. For the example in hand, the optimal modes found using the ESDTCT^{Exp} module are generally matching to those modes showing the least response vector visualized in the main effect plot in Figure 4.22. However, this rough indication from the main effect plots may be more accurate for small projects and less accurate for large projects; therefore, once the final selection of the desired modes is made, it is recommended to perform a final simulation run to validate the impact on the objectives of total cost and time. Figure 4.23 shows a tornado chart for the activity relative importance with regards to its mode interchangeability.

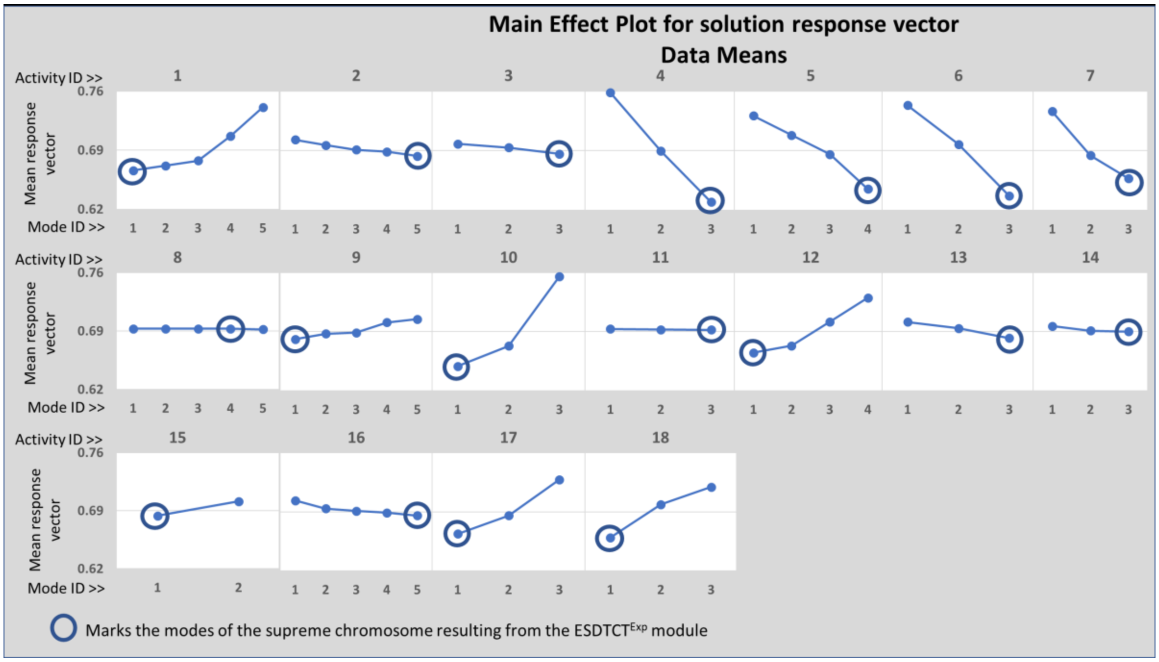


Figure 4.22 Example 2: Main effect plot for response vectors

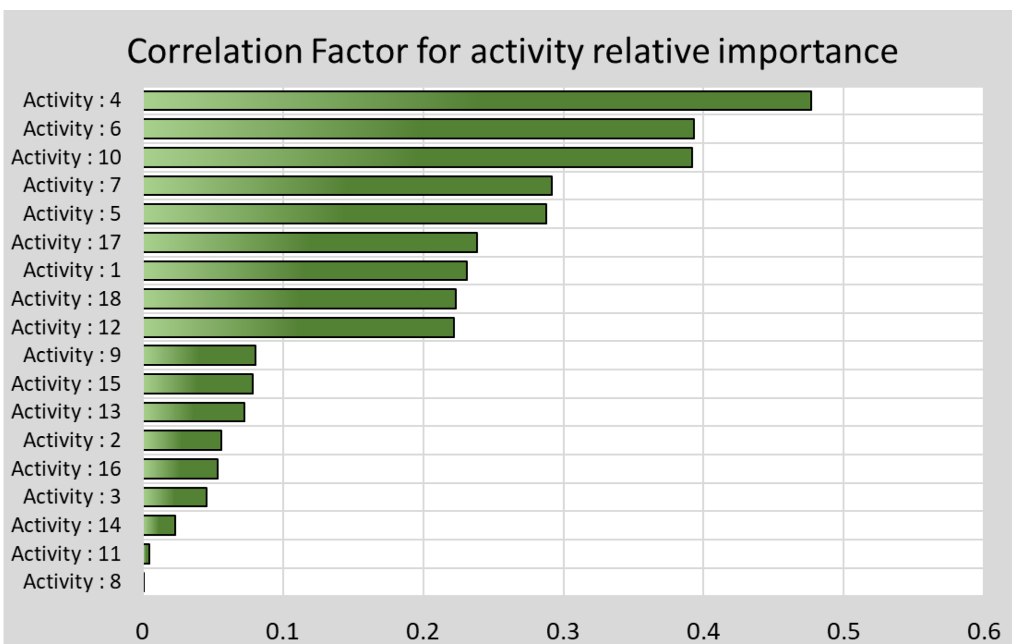


Figure 4.23 Example 2: Tornado chart for activity relative importance.

4.8.3. Example 3: Testing ESDTCT under certain environment.

This example is set to test the accuracy of the developed $ESDTCT^{Exp}$ module in comparison with previous solutions obtained from the literature. Feng et al. 1997 and Hegazy (1999) solved this example for the TCTP in GA-based application under crisp cost and time values, i.e. deterministic solution. Later, Kalhor et al. (2011) solved the same example using a fuzzy non-dominated archiving ant colony approach (NA-ACO) under a certain deterministic environment and confirmed the same results obtained by Feng and Hegazy. Table 4.4 summarizes those optimal results showing the optimal time and cost pairs and their corresponding chromosome structure.

The project network configuration is the same as that of example 1 shown in Figure 4.16. The time and cost data of activities are the same as those used in example 1, taking into consideration only the most likely duration and fixed cost of each mode. Presumably, when the minimum and maximum values for time and cost are the same as that of most likely, the ESDTCT method is converted to a deterministic approach. The indirect cost is the same as the aforementioned literature studies at $200 per day.

The example is solved using the $ESDTCT^{Exp}$ method to determine the optimal solution falling on the frontier curve at the deterministic 100% confidence level. The example is repeatedly solved with three objectives. (1) cost minimization, (2) schedule minimization, and (3) joint cost and schedule minimization. The results of each generation are tabulated in Table 4.5 to Table 4.7. The results show the supreme chromosome from the $ESDTCT^{Exp}$ method firmly confirms to those reported using the GA model and the NA-ACO model in a certain context. Using Google Cloud BigQuery applications, the results were produced in under 2 minutes.

The main effect plot is generated from the ESDTCT^{Rand} module as shown in Figure 4.24 to Figure 4.26 respectively for the three objectives. By visual analysis of the plot in Figure 4.24, it can be noticed that interchanging of modes amongst activities 4,5,6 and 7 have the most sensitivity effect on the objective of cost minimization; therefore, careful decisions should be made towards those activities; while a different set of sensitive activities are found when setting the objective for schedule minimization as seen visually in Figure 4.25 and similarly for the objective of joint cost and schedule minimization as seen visually in Figure 4.26.

Table 4.4 Optimal results for certain TCTP adapted from Kalhor et al. (2011).

		Activity ID >>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
		Number of admitted modes>>	5	5	3	3	4	3	3	5	5	3	3	4	3	3	2	5	3	3
No	Duration	Cost	<< Chromosome >>																	
1[a]	100	$153,320	1	5	3	3	3	1	3	5	1	1	2	1	3	3	1	5	1	1
2	101	$148,520	1	5	3	3	4	1	3	5	1	1	2	1	3	3	1	5	1	1
3	102	$148,470	2	5	3	3	4	1	3	5	1	1	2	1	3	3	1	5	1	1
4	103	$148,420	3	5	3	3	4	1	3	5	1	1	2	1	3	3	1	5	1	1
5	104	$141,120	1	5	3	3	4	2	3	5	1	1	2	1	3	3	1	5	1	1
6	105	$141,070	2	5	3	3	4	2	3	5	1	1	2	1	3	3	1	5	1	1
7	106	$141,020	3	5	3	3	4	2	3	5	1	1	2	1	3	3	1	5	1	1
8	108	$140,870	1	5	3	3	4	2	3	5	1	1	3	1	3	3	2	5	1	1
9	109	$140,820	2	5	3	3	4	2	3	5	1	1	3	1	3	3	2	5	1	1
10[b]	110	$128,270	1	5	3	3	4	3	3	5	1	1	3	1	3	3	1	5	1	1
11	111	$128,220	2	5	3	3	4	3	3	5	1	1	3	1	3	3	1	5	1	1
12	112	$128,170	3	5	3	3	4	3	3	5	1	1	3	1	3	3	1	5	1	1
13	114	$128,070	1	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	1	1
14	115	$128,020	2	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	1	1
15	116	$127,970	3	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	1	1
16	124	$127,870	1	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	3	1
17	125	$127,820	2	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	3	1
18[c]	126	$127,770	3	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	3	1

[a] Chromosome solution at minimum duration
[b] Chromosome solution at minimum cost and duration
[c] Chromosome solution at minimum cost

Table 4.5 Example 3: ESDTCT^{Exp} supreme chromosome results at each generation for cost minimization.

			Activity ID >>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
			Number of admitted modes>>	5	5	3	3	4	3	3	5	5	3	3	4	3	3	2	5	3	3
Generation	No. of Experimental runs	Total duration	Total cost	<< Chromosome >>																	
1	45	110	$187,620	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	5	3	1
2	54	114	$184,070	1	1	1	1	1	1	1	1	1	1	3	1	3	3	2	5	3	1
3	540	114	$159,870	1	1	3	3	1	1	3	5	1	1	3	1	3	3	2	5	3	1
4	180	124	$129,870	1	1	3	3	4	3	3	5	1	1	3	1	3	3	2	5	3	1
5	75	126	$127,770	3	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	3	1
Supreme Chromosome →		126	$127,770	3	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	3	1
	∑ = 894 runs																				

Activity $i \in \mathbb{O}$, elite mode

Activity $i \in \mathbb{P}$, variable modes

Activity $i \in \mathbb{B}$, Most likely value of mode 1

Table 4.6 Example 3: ESDTCT^{Exp} supreme chromosome results at each generation for schedule minimization.

			Activity ID >>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
			Number of admitted modes>>	5	5	3	3	4	3	3	5	5	3	3	4	3	3	2	5	3	3
Generation	No. of Experimental runs	Total duration	Total cost	<< Chromosome >>																	
1	45	100	$187,820	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	5	1	1
2	54	100	$184,720	1	1	1	1	1	1	1	1	1	1	2	1	3	3	1	5	1	1
3	540	100	$160,120	1	1	3	3	1	1	3	5	1	1	2	1	3	3	1	5	3	1
4	180	100	$154,920	1	1	3	3	3	1	3	5	1	1	2	1	3	3	1	5	3	1
5	75	100	$153,320	1	5	3	3	3	1	3	5	1	1	2	1	3	3	1	5	1	1
Supreme Chromosome →		100	$153,320	1	5	3	3	3	1	3	5	1	1	2	1	3	3	1	5	1	1
	∑ = 894 runs																				

Activity $i \in \mathbb{O}$, elite mode

Activity $i \in \mathbb{P}$, variable modes

Activity $i \in \mathbb{B}$, Most likely value of mode 1

Table 4.7 Example 3: ESDTCT^{Exp} supreme chromosome results at each generation for joint cost and schedule minimization.

			Activity ID >>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
			Number of admitted modes>>	5	5	3	3	4	3	3	5	5	3	3	4	3	3	2	5	3	3
Generation	No. of Experimental runs	Total duration	Total cost	<< Chromosome >>																	
1	45	100	$187,820	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	5	1	1
2	54	100	$184,670	1	1	1	1	1	1	1	1	1	1	3	1	3	3	1	5	1	1
3	540	100	$160,270	1	1	3	3	1	1	3	5	1	1	3	1	3	3	1	5	1	1
4	180	110	$130,270	1	1	3	3	4	3	3	5	1	1	3	1	3	3	1	5	1	1
5	75	110	$128,270	1	5	3	3	4	3	3	5	1	1	3	1	3	3	1	5	1	1
Supreme Chromosome →		110	$128,270	1	5	3	3	4	3	3	5	1	1	3	1	3	3	1	5	1	1
	∑ = 894 runs																				

$Activity\ i \in \mathbb{O}$, elite mode
$Activity\ i \in \mathbb{P}$, variable modes
$Activity\ i \in \mathbb{B}$ / Most likely value of mode 1

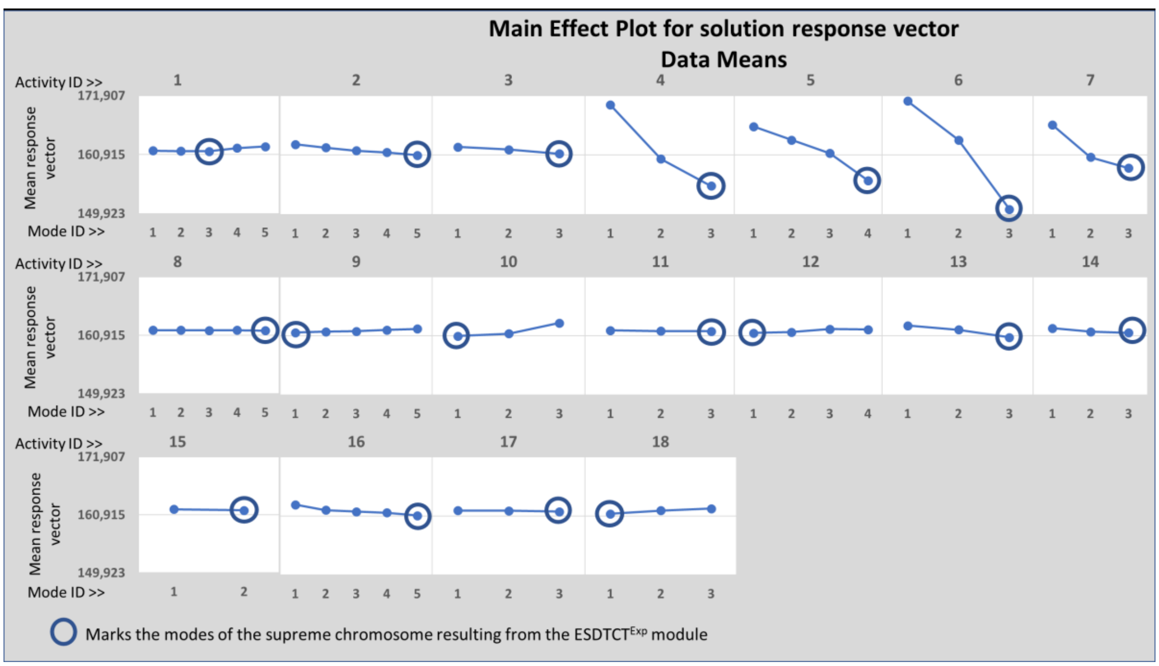


Figure 4.24 Example 3: Main effect plot for response vectors for the objective of cost minimization.

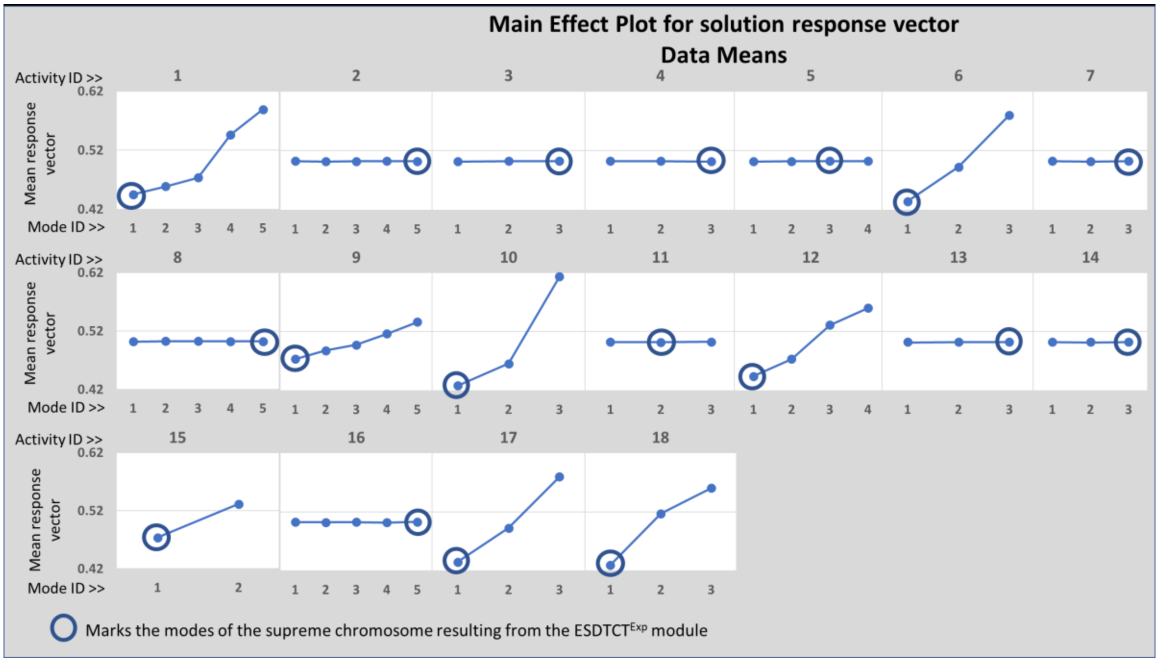


Figure 4.25 Example 3: Main effect plot for response vectors for the objective of schedule minimization.

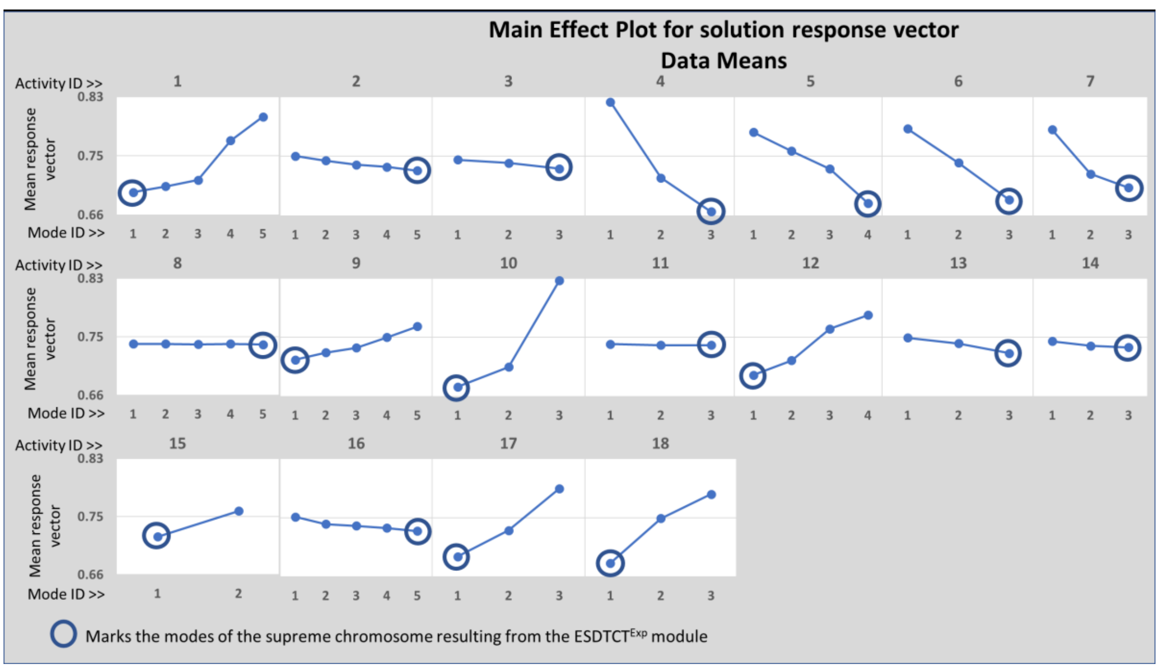


Figure 4.26 Example 3: Main effect plot for response vectors for the objective of joint cost and schedule minimization.

4.8.4. Example 4: Testing ESDTCT on a large size project under certain environment.

This example is set to test the performance and accuracy of the developed method on a large size project composed of 63-activities each having between 3 to 5 different modes; as a result, the problem has a total solution space of 1.373 tredecillion possible chromosome combinations (1.373 x 10^{42}) making the problem impossible to perform exhaustive enumeration. The network configuration of the example is shown in Figure 4.27, and the cost and duration values for each mode is listed in Table 4.8. The constant daily indirect cost is taken at $2,300. This data and network configuration for the project is adopted from Bettemir 2009; in his Ph.D. thesis the problem was solved in eight different meta-heuristic algorithms all under certain cost and time values. The data from Bettemir's research solving for the most optimum cost and time values for the objective of cost minimization are depicted in Table 4.11. To allow for comparison of the results, the developed ESDTCT method is converted to a certain environment by parsing the minimum and maximum values for time and cost as the same as that of most likely.

The ESDTCT took 11 generations to solve for the supreme chromosome. The primary activities at each generation is indicated in Figure 4.27. The primary activities at each generation are simply selected as the immediate predecessor activities to the previous generation. Starting with the project finish milestone and walking backwards through the project network. It can be noticed here that some activities are varied in multiple generations; this phenomenon increases as the logic complexity increases as some of the activities have more than one predecessor. For example, activity 52 is varied in generation 2 to solve for its elite mode; however, because this activity is also a predecessor to activity 55 that belongs in the same generation, the activity is put back into the experimental analysis for the search of its elite mode. Activities (1,10,24 and 36) are similarly analyzed.

The ESDTCT^{Exp} module results are listed at each generation and for the final supreme chromosome in Table **4.10**. The ESDTCT method was able to reduce the total solution space of 1.373 x 10^{42} experiments to only 677649 experiments (i.e. 4.94 x 10^{-35} %), which is extremely low and efficient. The supreme chromosome to achieve the cost minimization resulted in a total project cost of $5,421,320 for a total project duration of 633 days. The supreme chromosome cost and duration values firmly confirm to the aforementioned optimal lowest cost using the particle swarm optimization and the genetic algorithm simulated annealing optimization methods which in turn returned the lowest cost among other referenced optimization methods. The ESDTCT experimental module took less than 9 minutes to solve for the 11 generations and arrive at the supreme chromosome solution.

The main effect plot is generated from the developed ESDTCT^{Rand} module to allow predicting the effect of the decision-maker's favouring one mode over another on the objective cost minimization solution. Results for a total of 3 million random experiments were computed in parallel time to the ESDTCT^{Exp} module and shown in Figure 4.28. The greater slope on the main effect plot for an activity, the more inclination of results to influence the total project cost value. Conversely, the low slope has less influence. A decision-maker can then choose to assign preferred modes knowing how the problem will likely react. As data depicts, despite the small number of experiments representing at random only 2.18 x 10^{-34} % of the total solution space, nearly all the activities having the least response vector are consistent with that elite mode from the ESDTCT^{Exp} module. Figure 4.29 shows a tornado chart for the activity relative importance with regards to its mode interchangeability.

The base case scenario is defined by mode 1 of each activity. The base case is solved to have a total cost of $5,487,020 and a total duration of 708 days. The supreme chromosome resulted in a cost minimization of 1.21% of the base case and a 11.85% of the total duration.

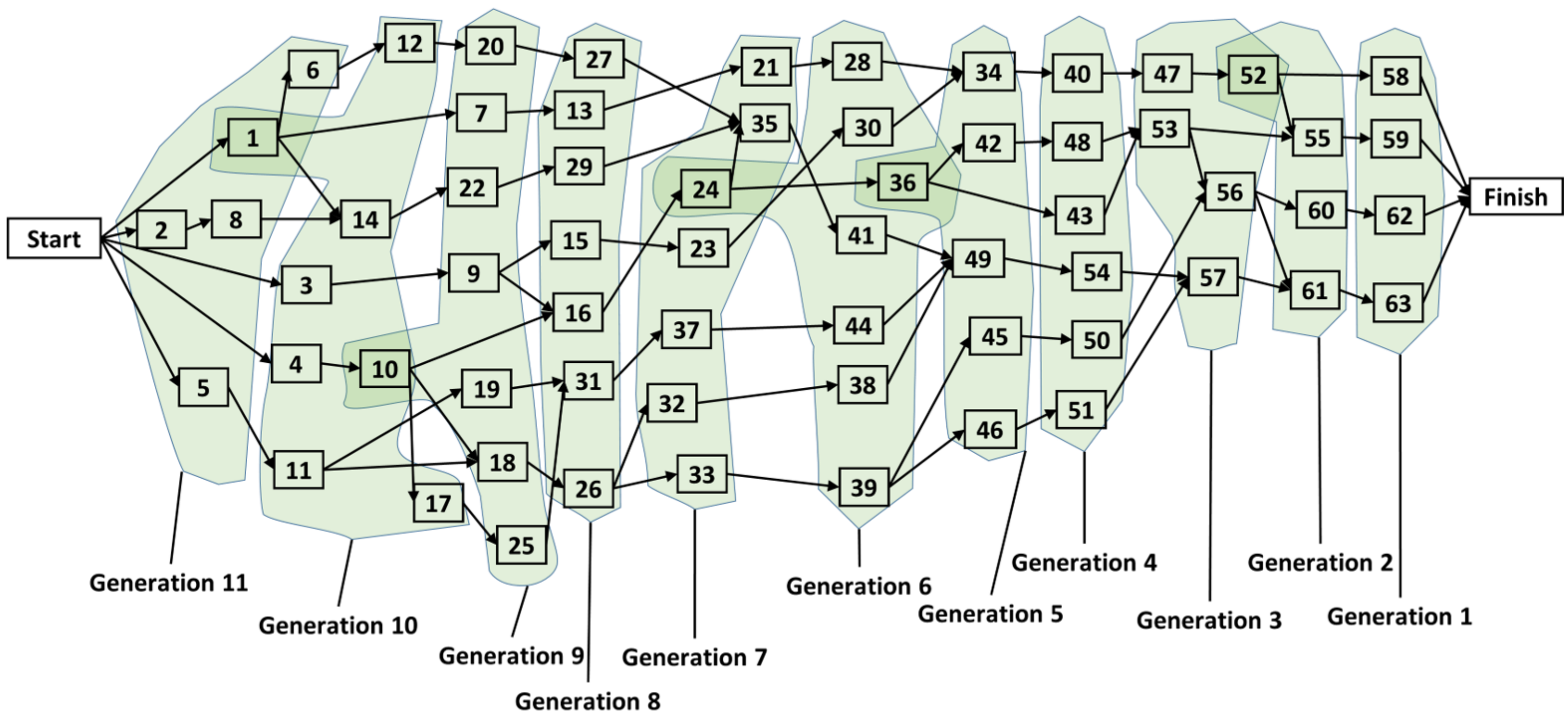


Figure 4.27 Example 4: Network configuration showing primary activities at each generation in the ESDTCT method.

Table 4.8 Example 4: Data input activities 1 to 46) (adapted from Bettemir 2009)

Activity	Mode (Duration (day), cost ($))		Triangular Probability Distribution Function selected for both time and cost		
	Mode 1	**Mode 2**	**Mode 3**	**Mode 4**	**Mode 5**
1	(14, $3750)	(12, $4250)	(10, $5400)	(9, $6250)	
2	(21, $11250)	(18, $14800)	(17, $16200)	(15, $19650)	
3	(24, $22450)	(22, $24900)	(19, $27950)	(17, $31650)	
4	(19, $17800)	(17, $19400)	(15, $21600)		
5	(28, $31180)	(26, $34200)	(23, $38250)	(21, $41400)	
6	(44, $54260)	(42, $58450)	(38, $63225)	(35, $68150)	
7	(39, $47600)	(36, $50750)	(33, $54800)	(30, $59750)	
8	(52, $62140)	(47, $69700)	(44, $72600)	(39, $81750)	
9	(63, $72750)	(59, $79450)	(55, $86250)	(51, $91500)	(49, $99500)
10	(57, $66500)	(53, $70250)	(50, $75800)	(46, $80750)	(41, $86450)
11	(63, $83100)	(59, $89450)	(55, $97800)	(50, $104250)	(45, $112400)
12	(68, $75500)	(62, $82000)	(58, $87500)	(53, $91800)	(49, $96550)
13	(40, $34250)	(37, $38500)	(33, $43950)	(31, $48750)	
14	(33, $52750)	(30, $58450)	(27, $63400)	(25, $66250)	
15	(47, $38140)	(40, $41500)	(35, $47650)	(32, $54100)	
16	(75, $94600)	(70, $101250)	(66, $112750)	(61, $124500)	(57, $132850)
17	(60, $78450)	(55, $84500)	(49, $91250)	(47, $94640)	
18	(81, $127150)	(73, $143250)	(66, $154600)	(61, $161900)	
19	(36, $82500)	(34, $94800)	(30, $101700)		
20	(41, $48350)	(37, $53250)	(34, $59450)	(32, $66800)	
21	(64, $85250)	(60, $92600)	(57, $99800)	(53, $107500)	(49, $113750)
22	(58, $74250)	(53, $79100)	(50, $86700)	(47, $91500)	(42, $97400)
23	(43, $66450)	(41, $69800)	(37, $75800)	(33, $81400)	(30, $88450)
24	(66, $72500)	(62, $78500)	(58, $83700)	(53, $89350)	(49, $96400)
25	(54, $66650)	(50, $70100)	(47, $74800)	(43, $79500)	(40, $86800)
26	(84, $93500)	(79, $102500)	(73, $111250)	(68, $119750)	(62, $128500)
27	(67, $78500)	(60, $86450)	(57, $89100)	(56, $91500)	(53, $94750)
28	(66, $85000)	(63, $89750)	(60, $92500)	(58, $96800)	(54, $100500)
29	(76, $92700)	(71, $98500)	(67, $104600)	(64, $109900)	(60, $115600)
30	(34, $27500)	(32, $29800)	(29, $31750)	(27, $33800)	(26, $36200)
31	(96, $145000)	(89, $154800)	(83, $168650)	(77, $179500)	(72, $189100)
32	(43, $43150)	(40, $48300)	(37, $51450)	(35, $54600)	(33, $61450)
33	(52, $61250)	(49, $64350)	(44, $68750)	(41, $74500)	(38, $79500)
34	(74, $89250)	(71, $93800)	(66, $99750)	(62, $105100)	(57, $114250)
35	(138, $183000)	(126, $201500)	(115, $238000)	(103, $283750)	(98, $297500)
36	(54, $47500)	(49, $50750)	(42, $56800)	(38, $62750)	(33, $68250)
37	(34, $22500)	(32, $24100)	(29, $26750)	(27, $29800)	(24, $31600)
38	(51, $61250)	(47, $65800)	(44, $71250)	(41, $76500)	(38, $80400)
39	(67, $81150)	(61, $87600)	(57, $92100)	(52, $97450)	(49, $102800)
40	(41, $45250)	(39, $48400)	(36, $51200)	(33, $54700)	(31, $58200)
41	(37, $17500)	(31, $21200)	(27, $26850)	(23, $32300)	
42	(44, $36400)	(41, $39750)	(38, $42800)	(32, $48300)	(30, $50250)
43	(75, $66800)	(69, $71200)	(63, $76400)	(59, $81300)	(54, $86200)
44	(82, $102750)	(76, $109500)	(70, $127000)	(66, $136800)	(63, $146000)
45	(59, $84750)	(55, $91400)	(51, $101300)	(47, $126500)	(43, $142750)
46	(66, $94250)	(63, $99500)	(59, $108250)	(55, $118500)	(50, $136000)

Table 4.9 Example 4: Data input activities 53 to 63) (adapted from Bettemir 2009)

Activity	Mode (Duration (day), cost ($))		Triangular Probability Distribution Function selected for both time and cost		
	Mode 1	**Mode 2**	**Mode 3**	**Mode 4**	**Mode 5**
47	(54, $73500)	(51, $78500)	(47, $83600)	(44, $88700)	(41, $93400)
48	(41, $36750)	(39, $39800)	(37, $43800)	(34, $48500)	(31, $53950)
49	(173, $267500)	(159, $289700)	(147, $312000)	(138, $352500)	(121, $397750)
50	(101, $47800)	(74, $61300)	(63, $76800)	(49, $91500)	
51	(83, $84600)	(77, $93650)	(72, $98500)	(65, $104600)	(61, $113200)
52	(31, $23150)	(28, $27600)	(26, $29800)	(24, $32750)	(21, $35200)
53	(39, $31500)	(36, $34250)	(33, $37800)	(29, $41250)	(26, $44600)
59	(27, $34600)	(24, $37500)	(22, $41250)	(19, $46750)	(17, $50750)
60	(31, $28500)	(29, $30500)	(27, $33250)	(25, $38000)	(21, $43800)
61	(29, $22500)	(27, $24750)	(25, $27250)	(22, $29800)	(20, $33500)
62	(25, $38750)	(23, $41200)	(21, $44750)	(19, $49800)	(17, $51100)
63	(27, $9500)	(26, $9700)	(25, $10100)	(24, $10800)	(22, $12700)
Indirect cost $ ($2300)					

Table 4.10 Example 4: ESDTCT^{Exp} supreme chromosome results at each generation under certain environment.

Activity ID >>				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Num of admitted modes >>				4	4	4	3	4	4	4	4	5	5	5	5	4	4	4	5	4	4	3	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	5	5	5	5	5	5	5	5	4	5	5	5	5	5	5	5	5	5	5	5	5	5
Generation	Num of Experimental runs	Total duration	Total cost	<< Chromosome >>																																																														
1	625	703	$5,478,720	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	5
2	625	694	$5,469,020	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	5
3	3125	683	$5,459,320	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	5	1	5
4	12500	678	$5,455,620	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	5	1	1	5	1	1	1	5	1	5
5	15625	652	$5,440,320	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3	1	1	1	1	5	1	1	5	1	1	1	5	1	5
6	312500	639	$5,425,220	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4	1	1	1	1	1	1	1	3	1	1	1	1	5	1	1	5	1	1	1	5	1	5
7	78125	638	$5,424,520	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	4	1	1	1	1	1	1	1	3	1	1	1	1	5	1	1	5	1	1	1	5	1	5
8	50000	638	$5,424,520	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	4	1	1	1	1	1	1	1	3	1	1	1	1	5	1	1	5	1	1	1	5	1	5
9	120000	633	$5,421,320	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	4	1	1	1	1	1	1	1	3	1	1	1	1	5	1	1	5	1	1	1	5	1	5
10	96000	633	$5,421,320	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	2	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	4	1	1	1	1	1	1	1	3	1	1	1	1	5	1	1	5	1	1	1	5	1	5
11	1024	633	$5,421,320	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	2	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	4	1	1	1	1	1	1	1	3	1	1	1	1	5	1	1	5	1	1	1	5	1	5
Supreme Chromosome		633	$5,421,320	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	2	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	4	1	1	1	1	1	1	1	3	1	1	1	1	5	1	1	5	1	1	1	5	1	5
	∑ = 677649 runs																																																																	

Activity $i \in \mathbb{O}$, elite mode

Activity $i \in \mathbb{P}$, variable modes

Activity $i \in \mathbb{B}$, Most likely value of mode 1

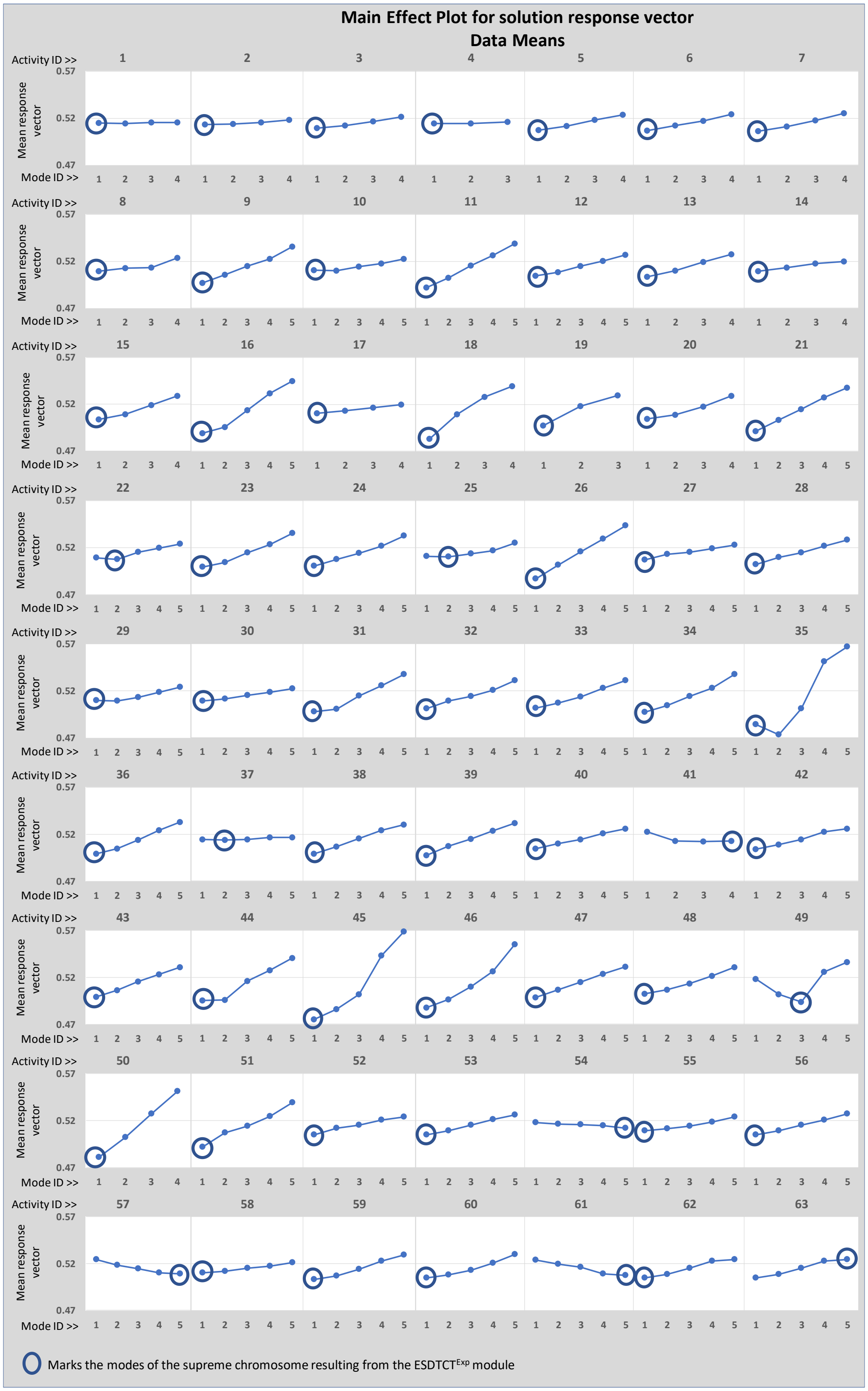


Figure 4.28 Example 4: Main effect plot for response vectors

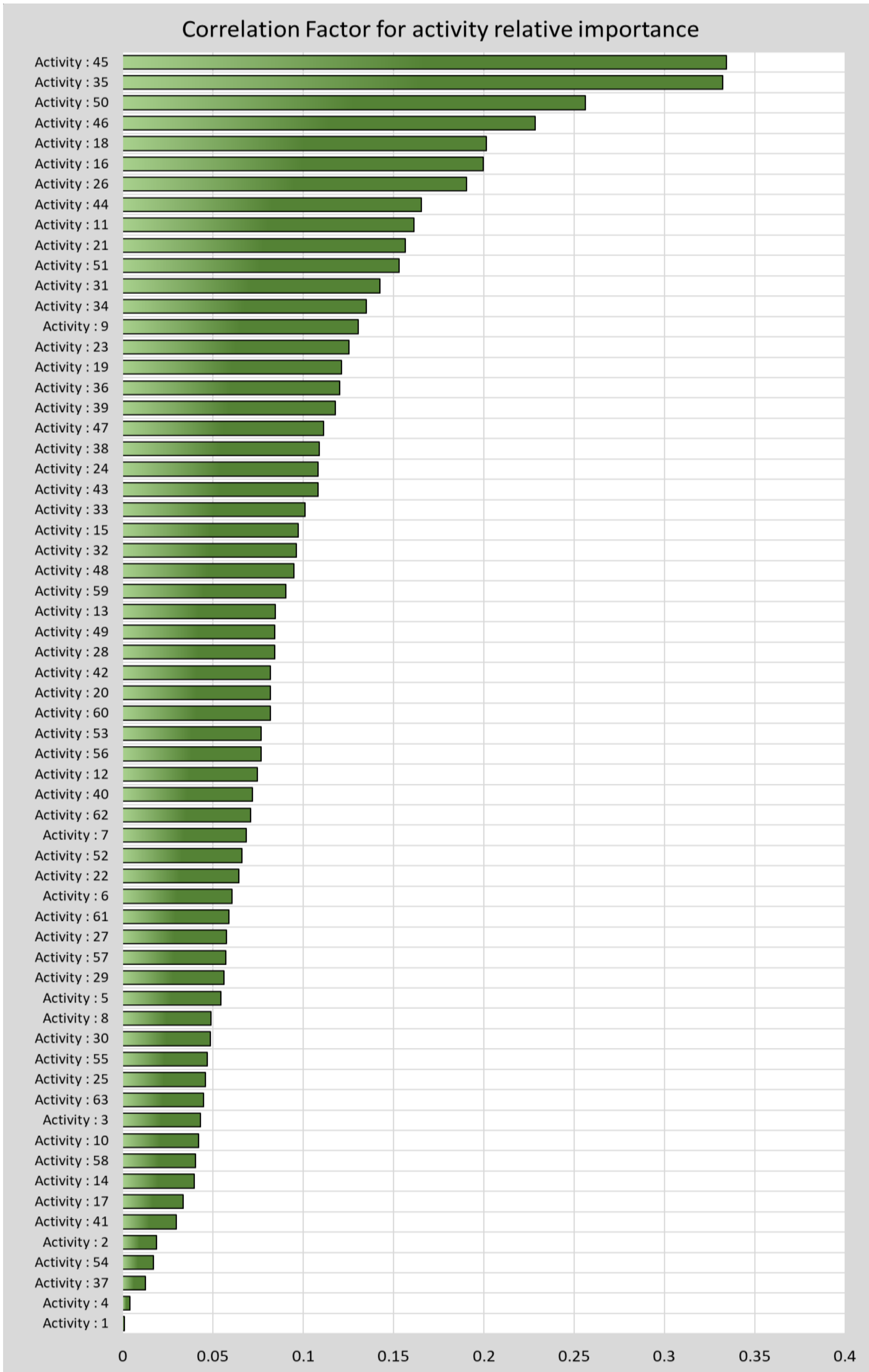


Figure 4.29 Example 4: Tornado chart for activity relative importance towards trade-off.

Table 4.11 Time and cost optimal solution using various meta-heuristic optimization methods (reported by Bettemir 2009).

Method		Cost	Duration
GMASA	Genetic Memetic Algorithm Simulated Annealing.	$5,421,120	630
HGAQSA	Hybrid Genetic Algorithm with Quantum Simulated Annealing.	$5,421,120	630
GASAVNS	Genetic algorithm with Simulated Annealing and Variable Neighborhood Search.	$5,421,120	630
ACO	Any Colony Optimization	$5,492,210	635
EMS	Electromagnetic Scatter Search.	$5,532,920	622
PSO	Particle Swarm Optimization.	$5,421,320	633
GASA	Genetic Algorithm Simulated Annealing	$5,421,620	633
GA	Genetic Algorithm	$5,690,790	623

4.8.5. Example 5: Testing ESDTCT on a large size project under uncertain environment.

Example 4 is extended here to solve the TCT problem under uncertain environment. To the best of our knowledge, this problem exceeds the largest and hardest test cases found in the literature that was solved for the TCT with uncertainty in both cost and schedule. The problem is solved to optimize for cost minimization with a 50% JCL. The uncertainty in the cost and duration values of each activity at each mode is listed in Table 4.12. The uncertainty in the indirect daily cost is ($2070, $2300, $2760) for the optimistic, most likely and pessimistic values respectively. The triangular PDF is used to represent the uncertainty profile for all cost and duration values. The ESDTCT experimental module run time took 11 minutes to solve for the 11 generations and arrive at the supreme chromosome solution. The ESDTCT experimental module results are listed at each generation and for the final supreme chromosome in Table 4.13. As data depicts, the uncertainty parameters resulted in 20 of the activities having different elite mode within the supreme chromosome structure as compared to the solution from example 4 performed under a certain environment. Those differences indicated that different modes of an activity are more fit for supremacy influenced by the uncertainty parameters. The optimum total cost is $5,954,231 for a total duration of 665 days at 50% JCL. For comparison purposes, the base case is solved for cost minimization and resulted in a probabilistic 50% JCL of $5,997,035 and 712 days. In contrast, optimal cost minimization is only $42,804 (0.72%) less than the probabilistic base case and a more significant 7.07% schedule savings.

The main effect plot is generated from the developed ESDTCT^{Rand} method to allow predicting the effect of mode selection on the objective of cost minimization at 50% JCL. The main effect plot from a total of 3 million random experiments was computed in parallel time to the ESDTCT^{Exp} module and shown in Figure 4.30. It can be noted from the plot that 52 of the 63 of the elite modes from the ESDTCT^{Exp} experimental module are confirming to the mode having the lowest cost from the ESDTCT^{Rand}. When selecting those modes from the main effect plot, the chromosome structure will be [1, 1, 1, 2, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 4, 1, 1, 1, 2, 2, 1, 1, 2, 1, 1, 2, 2, 1, 2, 1, 1, 2, 1, 1, 4, 1, 1, 2, 3, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 5, 1, 1, 5, 1, 1, 1, 1, 1, 1, 5] for activities 1 to 63 respectively. The balance 11 differing modes are those for activities [6, 10, 11, 13, 20, 27, 31, 42, 45, 48, 63]. To test if those differing modes are truly more superior to those obtained from the ESDTCT^{Exp} module, the probabilistic analysis is performed on this chromosome structure for the lowest cost on the 50% JCL frontier. The results where a total cost of $5,955,177 and a total duration of 668. This cost and schedule are higher than that of the $ESDTCT^{Exp}$ supreme chromosome by only $946 and 3 days. This proximity of the two solutions indicates the accuracy of the $ESDTCT^{Rand}$ and the main effect of modes. The $ESDTCT^{Rand}$ searched an extremely low of only 2.18×10^{-34} % of the total solution space. Presumably, the $ESDTCT^{Rand}$ can provide a decision guide for the selection of appropriate modes; however, the final solution resulting from the selected mode combination needs to be verified using the JCL analysis. Figure 4.31 shows a tornado chart for the activity relative importance with regards to its mode interchangeability.

Table 4.12 Example 5: Data input

Activity	**Mode** **Duration in days (Optimistic, Most Likely, Pessimistic)** **Fixed cost $ (Optimistic, Most Likely, Pessimistic)**			Triangular Probability Distribution Function selected for both time and cost	
	Mode 1	**Mode 2**	**Mode 3**	**Mode 4**	**Mode 5**
1	(13,14,17) (2813,3750,4200)	(9,12,15) (3825,4250,5525)	(9,10,13) (4860,5400,6372)	(6,9,11) (5625,6250,8625)	
2	(17,21,26) (10125,11250,12825)	(13,18,24) (13320,14800,17612)	(13,17,20) (14580,16200,18792)	(13,15,20) (17685,19650,26135)	
3	(20,24,30) (20205,22450,28736)	(20,22,30) (22410,24900,34860)	(17,19,24) (25155,27950,35217)	(15,17,23) (28485,31650,34815)	
4	(16,19,22) (16020,17800,22962)	(15,17,20) (17460,19400,21340)	(11,15,18) (19440,21600,27864)		
5	(24,28,33) (28062,31180,37728)	(22,26,36) (30780,34200,40356)	(20,23,31) (34425,38250,44370)	(17,21,26) (37260,41400,55062)	
6	(33,44,55) (48834,54260,61856)	(30,42,58) (52605,58450,66633)	(33,38,43) (56903,63225,85354)	(28,35,44) (61335,68150,86551)	
7	(33,39,43) (42840,47600,56644)	(30,36,46) (45675,50750,56840)	(29,33,36) (49320,54800,71240)	(27,30,40) (53775,59750,81260)	
8	(36,52,65) (55926,62140,83268)	(37,47,53) (62730,69700,92004)	(37,44,51) (65340,72600,101640)	(28,39,48) (73575,81750,94830)	
9	(45,63,83) (65475,72750,97485)	(45,59,69) (71505,79450,90573)	(48,55,63) (77625,86250,108675)	(41,51,62) (82350,91500,115290)	(40,49,68) (89550,99500,125370)
10	(51,57,80) (59850,66500,91105)	(43,53,59) (63225,70250,82193)	(38,50,59) (68220,75800,103088)	(35,46,56) (72675,80750,112243)	(36,41,54) (77805,86450,111521)
11	(55,63,85) (74790,83100,108030)	(44,59,66) (80505,89450,108235)	(39,55,63) (88020,97800,134964)	(39,50,58) (93825,104250,115718)	(34,45,50) (101160,112400,157360)
12	(53,68,80) (67950,75500,83050)	(45,62,73) (73800,82000,100040)	(45,58,68) (78750,87500,120750)	(39,53,66) (82620,91800,119340)	(42,49,64) (86895,96550,106205)
13	(32,40,47) (30825,34250,40758)	(30,37,46) (34650,38500,44275)	(26,33,44) (39555,43950,61530)	(27,31,36) (43875,48750,56550)	
14	(29,33,37) (47475,52750,59080)	(22,30,40) (52605,58450,76570)	(20,27,31) (57060,63400,84956)	(18,25,34) (59625,66250,88775)	
15	(35,47,59) (34326,38140,44242)	(28,40,48) (37350,41500,56440)	(31,35,44) (42885,47650,53368)	(26,32,44) (48690,54100,61674)	
16	(56,75,96) (85140,94600,116358)	(58,70,88) (91125,101250,123525)	(53,66,75) (101475,112750,129663)	(44,61,82) (112050,124500,160605)	(43,57,68) (119565,132850,152778)

Activity	Mode Duration in days (Optimistic, Most Likely, Pessimistic) Fixed cost $ (Optimistic, Most Likely, Pessimistic)			Triangular Probability Distribution Function selected for both time and cost	
	Mode 1	**Mode 2**	**Mode 3**	**Mode 4**	**Mode 5**
17	(44,60,70) (70605,78450,108261)	(43,55,74) (76050,84500,100555)	(39,49,58) (82125,91250,122275)	(39,47,54) (85176,94640,104104)	
18	(73,81,109) (114435,127150,170381)	(58,73,86) (128925,143250,180495)	(47,66,81) (139140,154600,177790)	(53,61,82) (145710,161900,192661)	
19	(28,36,46) (74250,82500,95700)	(26,34,46) (85320,94800,104280)	(24,30,39) (91530,101700,126108)		
20	(29,41,50) (43515,48350,57537)	(30,37,50) (47925,53250,68693)	(31,34,45) (53505,59450,70746)	(24,32,37) (60120,66800,79492)	
21	(54,64,70) (76725,85250,108268)	(46,60,81) (83340,92600,108342)	(43,57,75) (89820,99800,130738)	(45,53,70) (96750,107500,123625)	(44,49,69) (102375,113750,126263)
22	(50,58,79) (66825,74250,95040)	(39,53,67) (71190,79100,99666)	(42,50,67) (78030,86700,120513)	(40,47,64) (82350,91500,111630)	(33,42,58) (87660,97400,130516)
23	(34,43,52) (59805,66450,79740)	(34,41,45) (62820,69800,88646)	(30,37,42) (68220,75800,105362)	(23,33,45) (73260,81400,100122)	(26,30,35) (79605,88450,118523)
24	(55,66,75) (65250,72500,80475)	(50,62,77) (70650,78500,91060)	(45,58,78) (75330,83700,97092)	(44,53,66) (80415,89350,122410)	(43,49,56) (86760,96400,124356)
25	(41,54,59) (59985,66650,85979)	(37,50,67) (63090,70100,81316)	(39,47,54) (67320,74800,86768)	(36,43,49) (71550,79500,90630)	(32,40,53) (78120,86800,104160)
26	(71,84,117) (84150,93500,105655)	(64,79,109) (92250,102500,112750)	(63,73,86) (100125,111250,145738)	(52,68,95) (107775,119750,167650)	(54,62,75) (115650,128500,165765)
27	(59,67,93) (70650,78500,91845)	(47,60,67) (77805,86450,118437)	(41,57,74) (80190,89100,116721)	(50,56,74) (82350,91500,113460)	(43,53,59) (85275,94750,131703)
28	(51,66,85) (76500,85000,119000)	(47,63,83) (80775,89750,123855)	(53,60,73) (83250,92500,125800)	(44,58,67) (87120,96800,127776)	(38,54,65) (90450,100500,111555)
29	(53,76,90) (83430,92700,127926)	(55,71,79) (88650,98500,137900)	(48,67,92) (94140,104600,116106)	(56,64,70) (98910,109900,121989)	(52,60,79) (104040,115600,147968)
30	(24,34,44) (24750,27500,30800)	(25,32,41) (26820,29800,39038)	(26,29,39) (28575,31750,43498)	(23,27,33) (30420,33800,41574)	(19,26,31) (32580,36200,42716)
31	(74,96,132) (130500,145000,189950)	(74,89,117) (139320,154800,185760)	(70,83,110) (151785,168650,219245)	(55,77,89) (161550,179500,233350)	(55,72,80) (170190,189100,219356)
32	(37,43,52) (38835,43150,56527)	(33,40,51) (43470,48300,61824)	(26,37,44) (46305,51450,57110)	(29,35,44) (49140,54600,69342)	(27,33,46) (55305,61450,78042)
33	(37,52,59) (55125,61250,71050)	(44,49,60) (57915,64350,74646)	(33,44,59) (61875,68750,83188)	(33,41,54) (67050,74500,96850)	(29,38,46) (71550,79500,88245)

Activity	Mode **Duration in days (Optimistic, Most Likely, Pessimistic)** **Fixed cost $ (Optimistic, Most Likely, Pessimistic)**				Triangular Probability Distribution Function selected for both time and cost
	Mode 1	**Mode 2**	**Mode 3**	**Mode 4**	**Mode 5**
34	(54,74,99) (80325,89250,124058)	(51,71,94) (84420,93800,105994)	(57,66,74) (89775,99750,123690)	(51,62,76) (94590,105100,147140)	(48,57,71) (102825,114250,155380)
35	(104,138,157) (164700,183000,203130)	(102,126,160) (181350,201500,227695)	(83,115,150) (214200,238000,271320)	(87,103,123) (255375,283750,380225)	(73,98,113) (267750,297500,330225)
36	(41,54,63) (42750,47500,58425)	(44,49,68) (45675,50750,70035)	(38,42,53) (51120,56800,76112)	(29,38,42) (56475,62750,84713)	(27,33,37) (61425,68250,94868)
37	(29,34,46) (20250,22500,28575)	(23,32,36) (21690,24100,27956)	(21,29,39) (24075,26750,33438)	(21,27,33) (26820,29800,33972)	(22,24,30) (28440,31600,43924)
38	(36,51,68) (55125,61250,85138)	(41,47,63) (59220,65800,86856)	(33,44,49) (64125,71250,88350)	(30,41,53) (68850,76500,104040)	(32,38,48) (72360,80400,105324)
39	(54,67,75) (73035,81150,108741)	(43,61,79) (78840,87600,106872)	(40,57,79) (82890,92100,113283)	(37,52,60) (87705,97450,131558)	(43,49,61) (92520,102800,117192)
40	(34,41,55) (40725,45250,55205)	(32,39,53) (43560,48400,58080)	(27,36,46) (46080,51200,57856)	(27,33,46) (49230,54700,65093)	(23,31,35) (52380,58200,79734)
41	(26,37,43) (15750,17500,23450)	(26,31,42) (19080,21200,26288)	(22,27,30) (24165,26850,34368)	(18,23,30) (29070,32300,40375)	
42	(37,44,61) (32760,36400,49504)	(36,41,54) (35775,39750,50880)	(32,38,46) (38520,42800,55212)	(29,32,39) (43470,48300,53130)	(24,30,41) (45225,50250,60300)
43	(54,75,103) (60120,66800,90180)	(62,69,84) (64080,71200,90424)	(52,63,75) (68760,76400,105432)	(48,59,72) (73170,81300,111381)	(41,54,64) (77580,86200,120680)
44	(65,82,93) (92475,102750,136658)	(68,76,104) (98550,109500,148920)	(56,70,79) (114300,127000,173990)	(57,66,89) (123120,136800,184680)	(55,63,81) (131400,146000,167900)
45	(53,59,70) (76275,84750,105938)	(45,55,73) (82260,91400,117906)	(38,51,67) (91170,101300,123586)	(42,47,65) (113850,126500,174570)	(34,43,60) (128475,142750,161308)
46	(59,66,92) (84825,94250,112158)	(55,63,77) (89550,99500,117410)	(51,59,71) (97425,108250,125570)	(46,55,77) (106650,118500,159975)	(43,50,65) (122400,136000,156400)
47	(44,54,70) (66150,73500,97020)	(39,51,64) (70650,78500,91060)	(37,47,52) (75240,83600,94468)	(33,44,50) (79830,88700,116197)	(32,41,50) (84060,93400,110212)
48	(30,41,56) (33075,36750,46305)	(33,39,48) (35820,39800,43780)	(31,37,51) (39420,43800,60882)	(27,34,41) (43650,48500,66930)	(25,31,38) (48555,53950,73372)
49	(133,173,202) (240750,267500,358450)	(129,159,202) (260730,289700,393992)	(125,147,168) (280800,312000,374400)	(120,138,163) (317250,352500,412425)	(90,121,146) (357975,397750,505143)
50	(90,101,118) (43020,47800,53058)	(63,74,86) (55170,61300,74786)	(44,63,77) (69120,76800,89856)	(37,49,57) (82350,91500,128100)	

Activity	**Mode** **Duration in days (Optimistic, Most Likely, Pessimistic)** **Fixed cost $ (Optimistic, Most Likely, Pessimistic)**				Triangular Probability Distribution Function selected for both time and cost
	Mode 1	**Mode 2**	**Mode 3**	**Mode 4**	**Mode 5**
51	(71,83,109) (76140,84600,94752)	(55,77,93) (84285,93650,113317)	(60,72,81) (88650,98500,136915)	(52,65,88) (94140,104600,128658)	(43,61,84) (101880,113200,155084)
52	(28,31,42) (20835,23150,26623)	(20,28,33) (24840,27600,37812)	(23,26,35) (26820,29800,35760)	(18,24,31) (29475,32750,38973)	(17,21,26) (31680,35200,48224)
53	(34,39,46) (28350,31500,34965)	(26,36,50) (30825,34250,44525)	(29,33,45) (34020,37800,51786)	(21,29,35) (37125,41250,45788)	(19,26,32) (40140,44600,57980)
54	(17,23,30) (14850,16500,22110)	(17,22,25) (16020,17800,23496)	(15,21,29) (17775,19750,24885)	(16,20,25) (19080,21200,28832)	(16,18,21) (21870,24300,27945)
55	(23,29,33) (21060,23400,26676)	(20,27,36) (22725,25250,32825)	(21,26,30) (24210,26900,33356)	(21,24,31) (26460,29400,37632)	(15,22,26) (29250,32500,42900)
56	(29,38,42) (37125,41250,53625)	(31,35,48) (40185,44650,60724)	(29,33,37) (43020,47800,64052)	(23,31,36) (46260,51400,65278)	(22,29,34) (49905,55450,69867)
57	(29,41,47) (34020,37800,51030)	(33,38,45) (37125,41250,55688)	(25,35,46) (41040,45600,50616)	(28,32,44) (44775,49750,69153)	(23,30,40) (48060,53400,58740)
58	(19,24,30) (11250,12500,16375)	(16,22,28) (12240,13600,17816)	(18,20,27) (13725,15250,16928)	(13,18,25) (15120,16800,18648)	(13,16,19) (17505,19450,24702)
59	(23,27,37) (31140,34600,43596)	(17,24,27) (33750,37500,49500)	(19,22,29) (37125,41250,53213)	(16,19,23) (42075,46750,52360)	(14,17,20) (45675,50750,69020)
60	(27,31,39) (25650,28500,31920)	(20,29,33) (27450,30500,36905)	(23,27,31) (29925,33250,41563)	(18,25,35) (34200,38000,42560)	(16,21,28) (39420,43800,56064)
61	(23,29,34) (20250,22500,31050)	(20,27,38) (22275,24750,29453)	(20,25,29) (24525,27250,34335)	(16,22,26) (26820,29800,33674)	(16,20,23) (30150,33500,40535)
62	(21,25,30) (34875,38750,44563)	(20,23,28) (37080,41200,45320)	(18,21,25) (40275,44750,52805)	(14,19,26) (44820,49800,67230)	(12,17,23) (45990,51100,70518)
63	(24,27,33) (8550,9500,12160)	(18,26,35) (8730,9700,12610)	(19,25,31) (9090,10100,13938)	(17,24,29) (9720,10800,14364)	(19,22,29) (11430,12700,17145)
Indirect cost $ ($2070, $2300, $2760)					

Table 4.13 Example 5: ESDTCT^{Exp} supreme chromosome results at each generation under uncertain environment.

			Activity ID >>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
			Num of admitted modes >>	4	4	4	3	4	4	4	4	5	5	5	5	4	4	4	5	4	4	3	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	5	5	5	5	5	5	5	5	4	5	5	5	5	5	5	5	5	5	5	5	5	5
Generation	Num of Experimental runs	Total duration	Total cost	<< Chromosome >>																																																														
1	625	709	$5,761,011	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	5
2	625	696	$5,756,752	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	5
3	3125	700	$5,763,543	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	5	1	1	1	1	1	5
4	12500	695	$5,771,843	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	2	1	1	1	1	1	5	1	1	5	1	1	1	1	1	5
5	15625	680	$5,770,326	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	2	1	2	1	1	2	1	1	2	3	1	1	1	1	5	1	1	5	1	1	1	1	1	5
6	312500	666	$5,790,749	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	2	1	1	1	1	1	2	1	2	1	1	2	1	1	2	3	1	1	1	1	5	1	1	5	1	1	1	1	1	5
7	78125	671	$5,812,595	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	2	1	1	1	1	1	2	1	1	4	1	1	2	3	2	1	1	2	1	1	2	3	1	1	1	1	5	1	1	5	1	1	1	1	1	5
8	50000	666	$5,859,888	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	2	1	1	1	1	1	4	2	1	1	1	1	1	2	1	1	4	1	1	2	3	2	1	1	2	1	1	2	3	1	1	1	1	5	1	1	5	1	1	1	1	1	5
9	120000	661	$5,905,150	1	1	1	1	1	1	1	1	2	3	1	1	2	1	1	1	1	1	1	3	2	1	1	1	2	1	4	2	2	1	1	1	1	2	1	1	4	1	1	2	3	2	1	1	2	1	1	2	3	1	1	1	1	5	1	1	5	1	1	1	1	1	5
10	96000	652	$5,935,790	1	1	1	2	1	1	1	1	2	3	2	1	2	1	1	1	4	1	1	3	2	2	1	1	2	1	4	2	2	1	1	1	1	2	1	1	4	1	1	2	3	2	1	1	2	1	1	2	3	1	1	1	1	5	1	1	5	1	1	1	1	1	5
11	1024	665	$5,954,231	1	1	1	2	1	1	1	1	2	3	2	1	2	1	1	1	4	1	1	3	2	2	1	1	2	1	4	2	2	1	1	1	1	2	1	1	4	1	1	2	3	2	1	1	2	1	1	2	3	1	1	1	1	5	1	1	5	1	1	1	1	1	5
Supreme Chromosome		665	$5,954,231	1	1	1	2	1	2	1	1	2	3	2	1	2	1	1	1	4	1	1	3	2	2	1	1	2	1	4	2	2	1	1	1	1	2	1	1	4	1	1	2	3	2	1	1	2	1	1	2	3	1	1	1	1	5	1	1	5	1	1	1	1	1	5
	∑ = 677649 runs																																																															

$Activity\ i \in \mathbb{O}$, elite mode

$Activity\ i \in \mathbb{P}$, variable modes

$Activity\ i \in \mathbb{B}$, Most likely value of mode 1

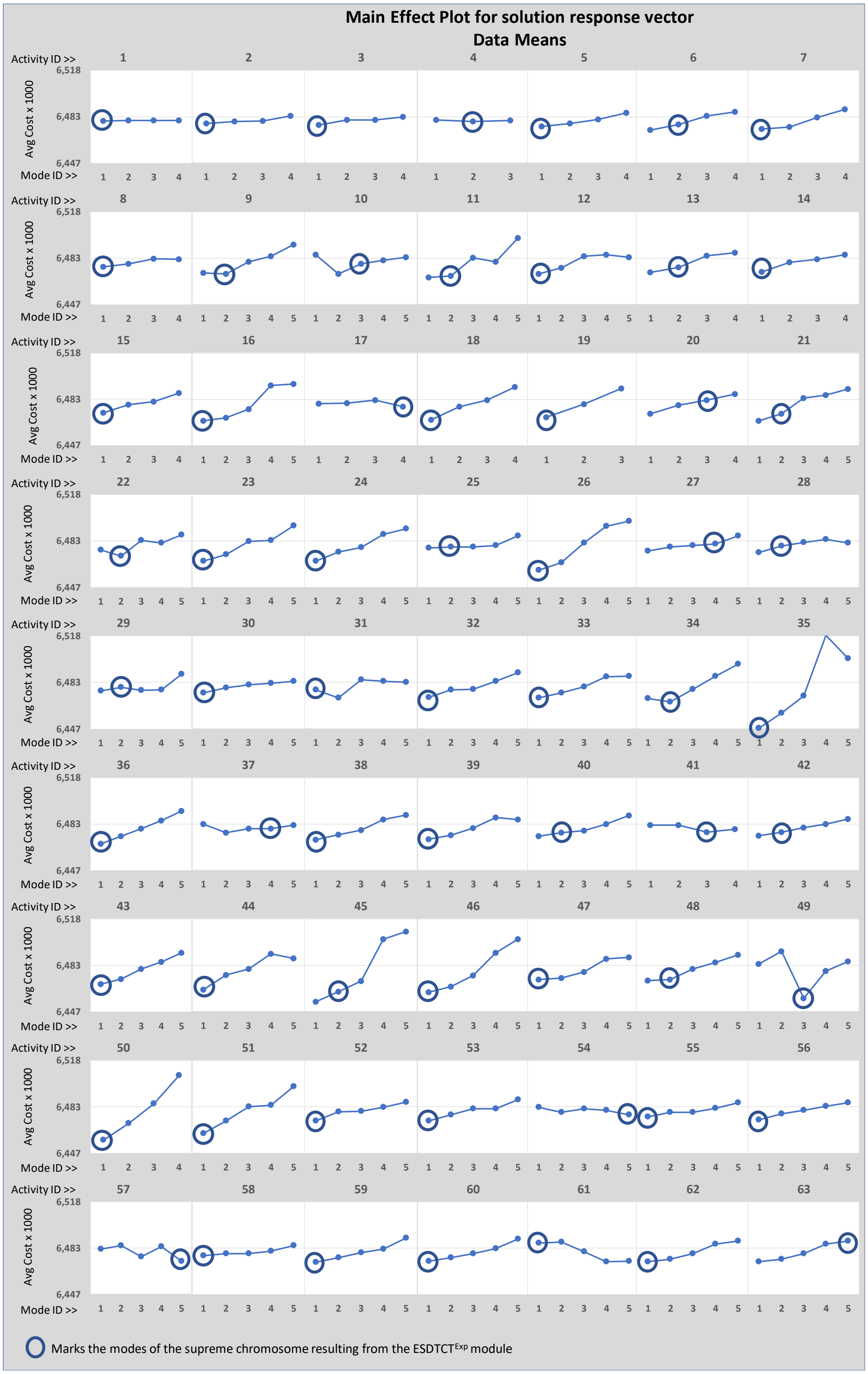


Figure 4.30 Example 5: Main effect plot for response vectors

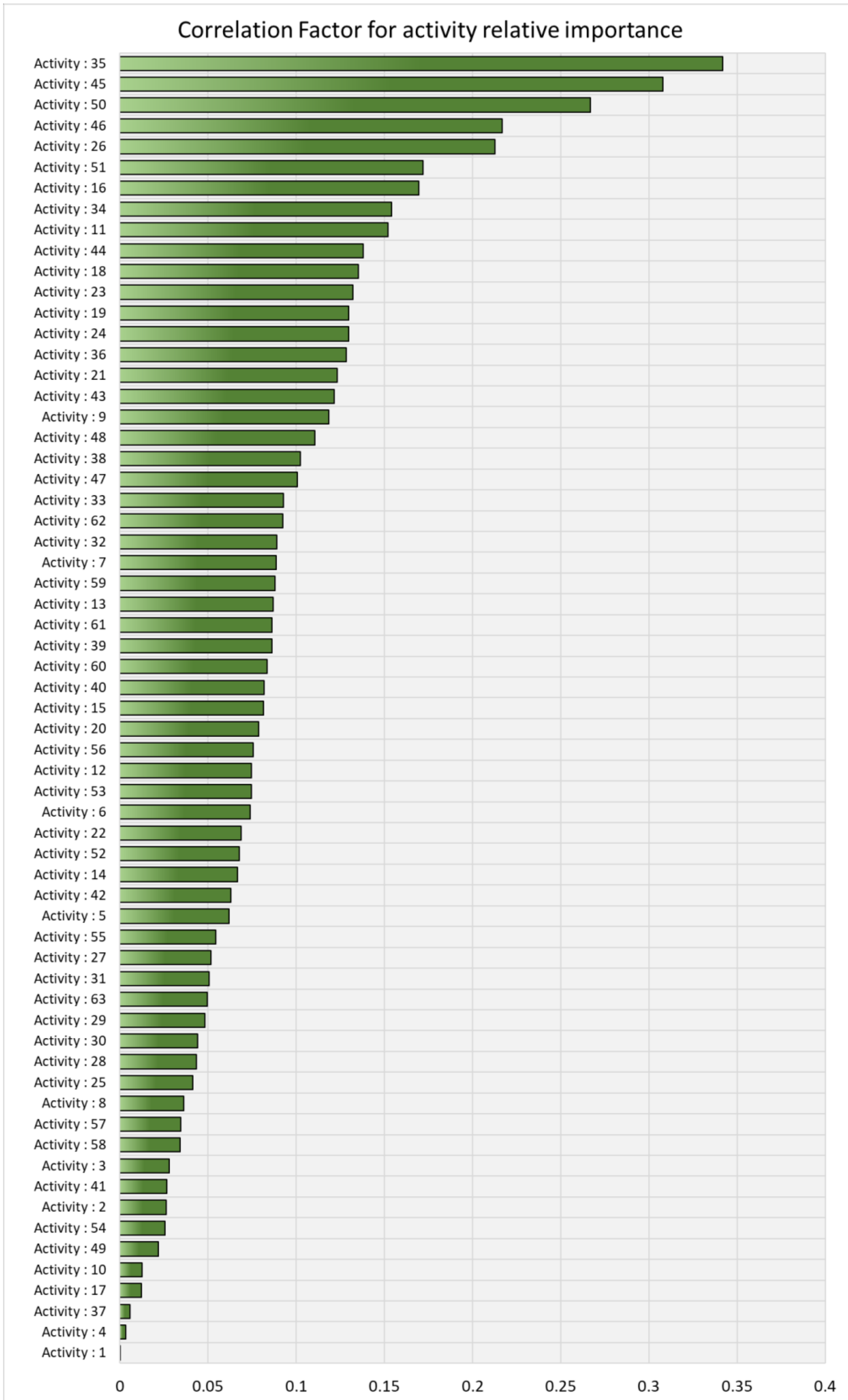


Figure 4.31 Example 5: Tornado chart for activity relative importance towards trade-off.

4.8.6. Example 6: Testing with discrete risk events.

This example is set to test the incorporation of discrete risk events into the TCT optimization model. The project network configuration and data of example 2 are extended to incorporate 3 discrete risk events denoted as R1, R2 and R3, each of which has a probability of occurrence. The time and cost data of risk events and their assigned probability of occurrence are listed in Table 4.14. The indirect cost is also a triangular probability function with \$185 optimistic, \$200 most likely, and \$235 pessimistic values. The table also lists the different modes of execution for treating the risk event when it occurs. Presumably, the activities in the network are assigned a probability of occurrence of 100% as those represent defined scope for the project in hand, while a discrete risk event may or may not occur, however, when the risk occurs it will have an impact on the project schedule and cost as per the network configuration.

The first step is to adjust the project network to incorporate the risk event, as shown in Figure 4.32. Risk event 1 when occurs is modelled to delay the start of activity 13. Similarly, risk event 2 delays the start of activity 15, while risk event 3 is modelled to possibly delay the start of activity 18 depending on the criticality of the path leading to activity 16. It is worth noting that this design is arbitrary for illustration purposes and more complex designs can be introduced when more than one predecessor and successor can be logically linked to the risk event to model the risk situation and possible impacts.

The ESDTCT^{Exp} module is used to solve for the supreme chromosome resulting in the most optimal joint cost and schedule minimization for a JCL of 50%.

The total solution space of the problem is the product function of the number of modes for each activity and risk; this equates to 1.41718 x 10^{11} possible chromosome solutions. The $ESDTCT^{Exp}$ took 5 generations to solve for the supreme chromosome. The primary activities at each generation is indicated in Figure 4.33. The total number of experiments is 9,075, which is an extremely efficient low number of only 6.403 x 10^{-8} % of the total solution space. The total run time for this example was under 3.5 minutes. The results of each generation are tabulated in Table 4.15.

The main effect plot generated from the $ESDTCT^{Rand}$ module is showing in Figure 4.34. Amongst the discrete risk events, R1 is observed to have a main effect on the optimization objective function. Figure 4.35 shows a tornado chart for the activity relative importance with regards to its mode interchangeability.

The supreme chromosome from the $ESDTCT^{Exp}$ is slightly different than that solved in example 1 where no risk events are considered. Only 4 of the 18 activities have different elite modes. Those differences are found in activities 1,2,3 and 14. The optimal total cost is found 14% higher than that of example 1 due to the effect of discrete risk events; the optimal treatment of the risk events are modes 3,1 and 2 for risks R1, R2 and R3 respectively.

Table 4.14 Numerical example 6: Data input for discrete risk events.

Activity	Probability of occurrence	Mode Duration in days (Optimistic, Most Likely, Pessimistic) Fixed cost $ (Optimistic, Most Likely, Pessimistic)			Triangular Probability Distribution Function selected for both time and cost
		Mode 1	Mode 2	Mode 3	Mode 4
R1	50%	(8,9,12) (38500,40000,42860)	(21,22,27) (29800,32000,34550)	(30,33,39) (16550,18000,21000)	
R2	20%	(10,12,15) (42050,45000,48800)	(13,16,21) (38500,35000,39000)	(17,20,28) (28500,30000,33550)	(19,22,25) (18500,20000,22550)
R3	30%	(14,18,22) (14150,15000,17050)	(29,30,34) (8500,10000,12600)		

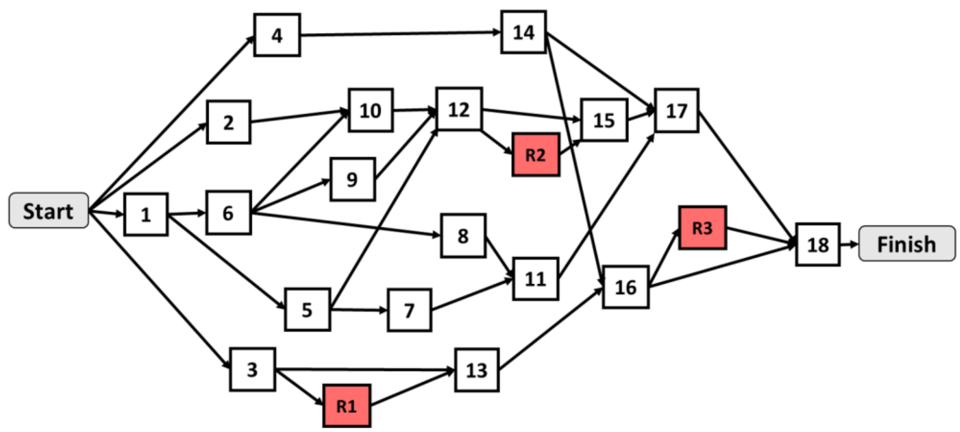


Figure 4.32. Numerical example 6: Risk adjusted network configuration.

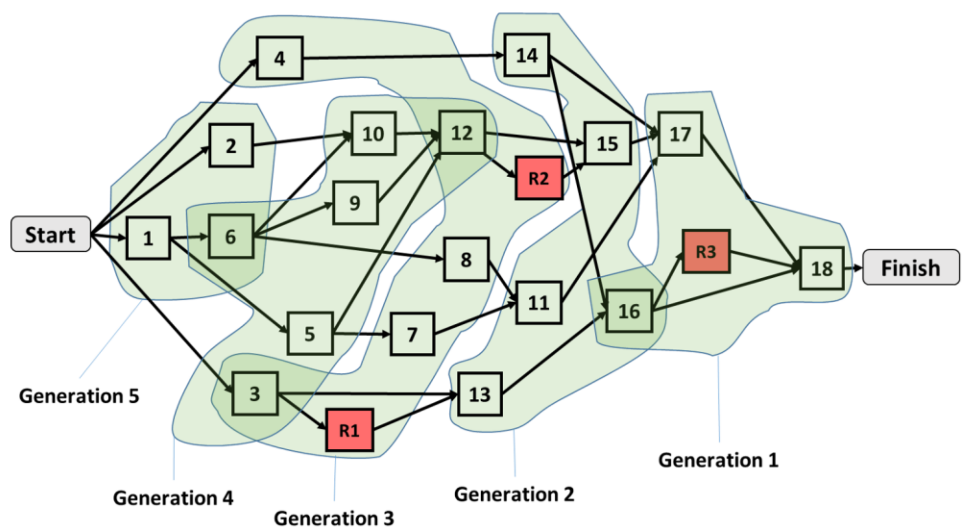


Figure 4.33 Example 6: Network configuration showing primary activities at each generation in the ESDTCT method.

Table 4.15 Example 6: ESDTCT^{Exp} supreme chromosome results at each generation (for a JCL = 50%)

			Activity ID >>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	R1	R2	R3
			Number of admitted modes>>	5	5	3	3	4	3	3	5	5	3	3	4	3	3	2	5	3	3	3	4	2
Generation	No. of Experimental runs	Total duration	Total cost	<< Chromosome >>																				
1	45	102	$231,653	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	5	1	1	1	1	2
2	270	103	$228,885	1	1	1	1	1	1	1	1	1	1	3	1	3	3	1	5	1	1	1	1	2
3	2160	112	$202,143	1	1	1	1	1	1	3	4	1	1	3	1	3	3	1	5	1	1	3	1	2
4	6480	118	$158,297	1	1	1	3	4	3	3	4	1	1	3	1	3	3	1	5	1	1	3	1	2
5	75	118	$157,255	1	4	1	3	4	3	3	4	1	1	3	1	3	3	1	5	1	1	3	1	2
Supreme Chromosome →		118	$157,255	1	4	1	3	4	3	3	4	1	1	3	1	3	3	1	5	1	1	3	1	2
	∑ = 9075 runs																							

$Activity\ i \in \mathbb{O}$, elite mode
$Activity\ i \in \mathbb{P}$, variable modes
$Activity\ i \in \mathbb{B}$, Most likely value of mode 1

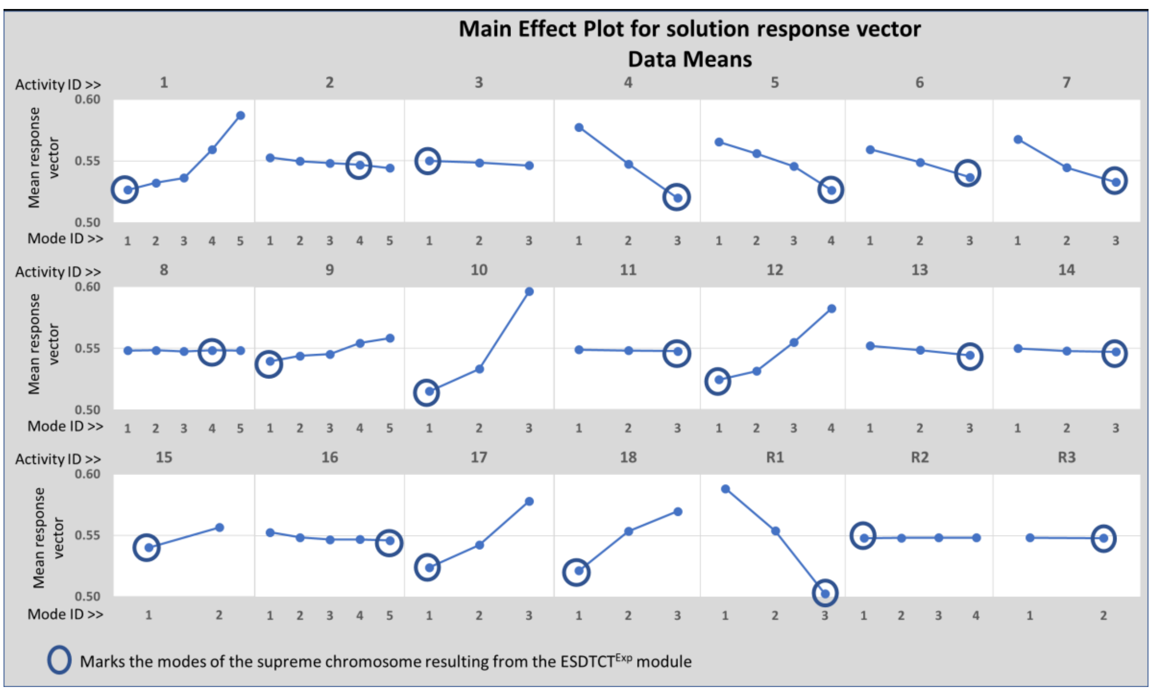


Figure 4.34 Example 6: Main effect plot for response vectors

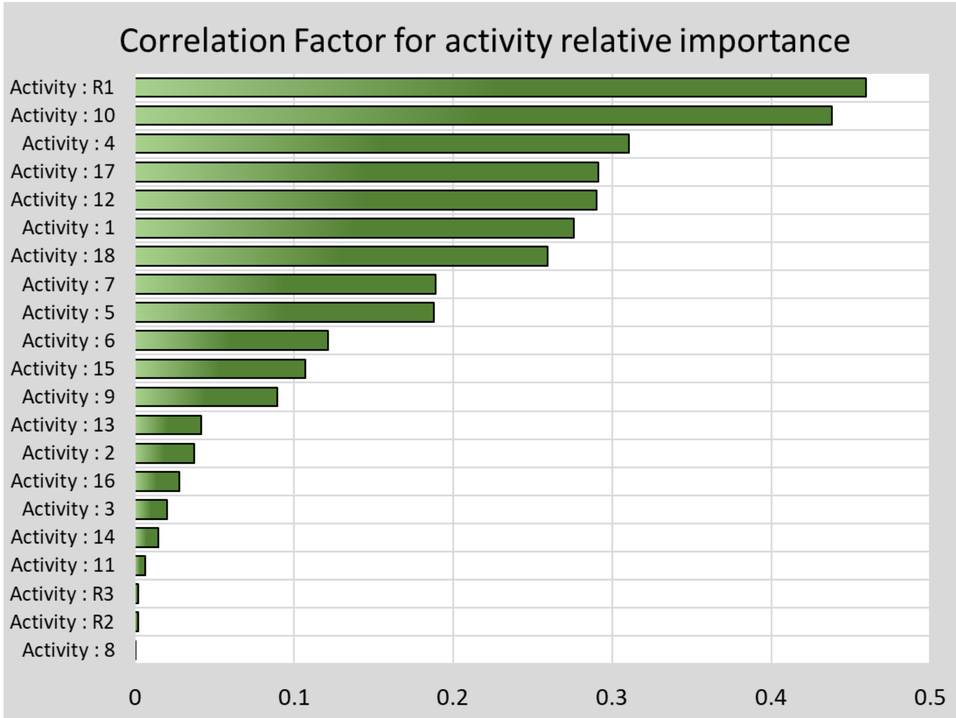


Figure 4.35 Example 6: Tornado chart for activity relative importance towards trade-off.

CHAPTER 5 : IMPLEMENTATION OF THE DEVELOPED METHOD ON REPETITIVE CLASS PROJECTS

5.1. Introduction

A particular type of construction projects is repetitive in nature consisting of a number of similar or identical units, most of which have their own sets of possible crew formations, the challenge on such projects is to determine which is the optimum crew formation for each repetitive activity that results in the best overall project's cost and/or time minimization? The objective of this chapter is to introduce a systematic approach method that provides an answer to this crucial question, taking into considerations for:

- Optimization for crew work formation.
- Optimization for crew work continuity; i.e. minimization of crew idle times.
- Non-typical activities in repetitive units,
- Uncertainty in crew productivity factors and cost estimates.

Non-typical activities in repetitive projects are in general attributed to dissimilarities in the quantities of work or productivity of crews and machinery used to execute the work. For example, the quantity of earthwork can vary from one unit to another due to site topography or the soil conditions; other variables can be the learning curve effect and weather change from one season to another. Consequently, activity durations in non-typical activities can be varied along the repeated units.

Figure 5.1 shows an illustration of a repetitive project consisting of typical and non-typical activities.

Idle time is the duration where no work is performed for an employed crew and their equipment, i.e. the crew is interrupted from achieving a product output. Most often, those interruptions result in a cost burden on the project. Eliminating crew work interruptions by minimizing the idle time of each crew also leads to maximizing the learning curve and momentum of the crew; however, its strict application may result in a longer overall project's duration. Work interruptions are permitted in the developed method. The present method incorporates cost as a decision variable in the optimization of crew interruption times and uses traditional CPM calculations and, therefore, can be easily developed for any project.

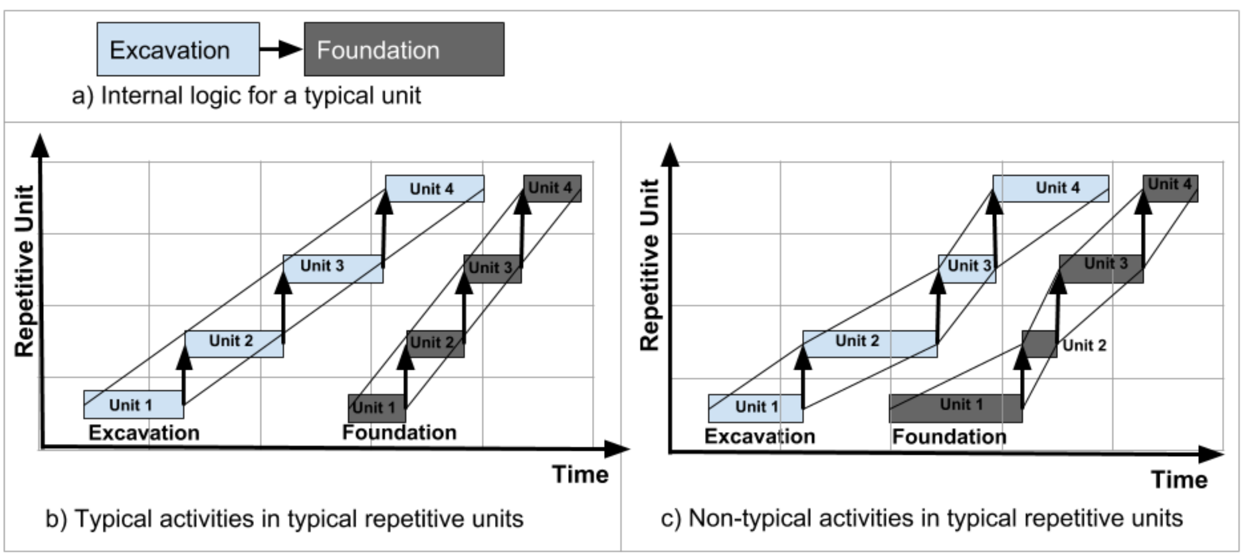


Figure 5.1 Typical and Non-typical activities in repetitive projects.

5.2. Computational procedure

A systematic approach method for discrete time-cost trade-off optimization of repetitive activities in construction projects is developed. This method is an extension to the developed ESDTCT^{EXP} method, explained in the previous chapter (see section 4.6), to account for repetitive projects, and named here as Repetitive Project Evolutionary Stochastic Discrete Time-Cost Trade-off - Experiment enumeration module (RP-ESDTCT^{EXP}). The method extends the ESDTCT^{EXP} method from a bi-objective optimization of time and cost minimization to a tri-objective optimization adding the crew work continuity vector as a decision factor. The method is composed of five stages:

1. Setup stage: in this stage, the project network is developed in a CPM-like schedule. The CPM network naturally has a ladder-like appearance due to the repetitive character of the activities. The logic within a typical unit is used to maintain continuity of workflow, abbreviated here as workflow logic (WFL). The project network is then extended to account for resource flow logic to successive repetitive units, abbreviated here as (RFL). The RFL logic is considered a finish-to-start relationship to ensure a timely movement of the crew from one unit to the next. Figure 5.2 shows a representation of the WFL and RFL logic.
2. Forward stage: In this stage a CPM forward pass calculation is performed throughout the project network starting with the first activity in the first unit and ending with the last activity in the last unit in compliance with the WFL logic and

RFL Logic and thus resulting in computing the early start dates for each activity and the overall project's early completion date.

3. Optimization stage for crew work formation. In this stage, the computational procedure described in section 4.7 is performed for the discrete crew formation modes to search for the optimal crew work formation for each activity. The assumption made here is that a selected crew formation will continue to be deployed on the project to work on the specific activity in a typical unit with movement of the crew from one unit to the next. This assumption was necessary to minimize the matrix size of the full factorial design of experiment runs and to reduce the total search space. Furthermore, this assumption reasonable simulates practical applications in the real world, as this is clearly inefficient since it requires the demobilization and restarting of the alternative crew.
4. Backward stage: In this stage, the computation procedure propagates throughout the project to compute the idle (interruption) time between the finish time of an activity i in a given unit n and the start of the similar activity in the successive repetitive unit $(n+1)$ in compliance with the crew work continuity constraint. The summation of all idle time in activity i throughout all repetitive units is then the total idle time for the crew assigned to this repetitive activity.
5. Optimization stage for crew work continuity. In this stage, the optimal solution obtained from the third stage is further analyzed. An iterative nested enumeration method is developed to search for the optimal crew work continuity. A shift variable is considered for each activity which represented the

amount of time that the activity in the first repetitive unit could be delayed beyond its early start date. The shifts were defined to range from zero, that represented an early start schedule for all activities, to a maximum value calculated as the summation of all idle time between an activity and its matching in successive repetitive units.

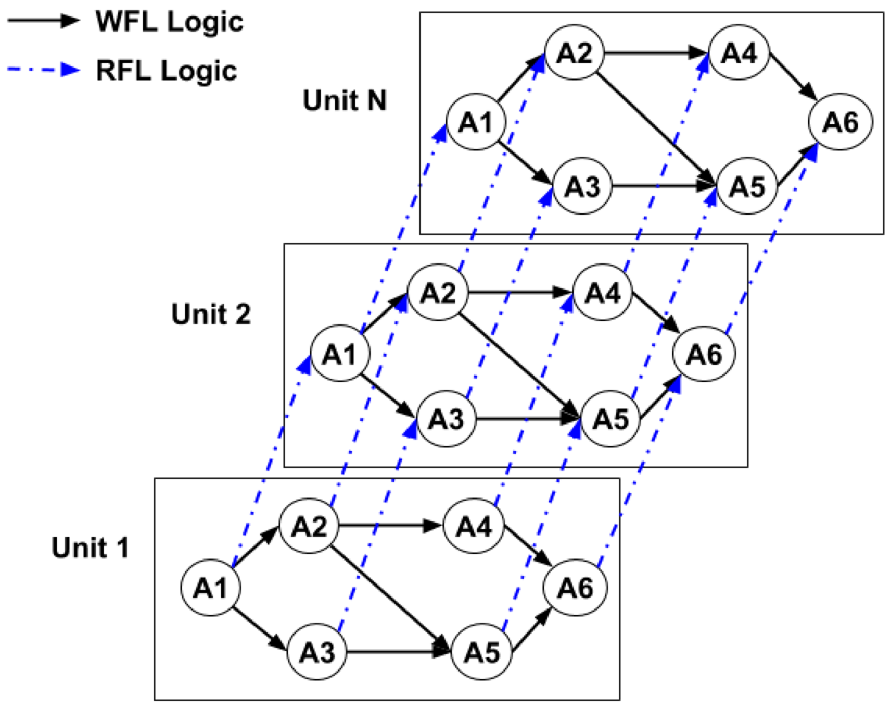


Figure 5.2 Repetitive project network logic.

5.3. Calculations for crew work interruption time.

On construction projects, an interruption in resource continuity may be required to meet some known or anticipated conditions or most often to reduce and account for the costs associated with such interruptions. In such instances, those interruptions need to be incorporated in the baseline schedule and budget. Figure

5.3 (a to e) shows a simplified example to demonstrate the various feasible schedules and the calculation for interruptions allowing the shift in the activity start date. The example is a representation of a simple project having three activities (A, B and C) with a finish to start relationship. Figure 5.3b shows the feasible schedule for the earliest start and finish dates. This schedule clearly shows idle time in activity B as the assigned crew moves from one unit to another. The total idle time for crew B is 3 days. Figure 5.3 (c to e) shows all feasible schedules allowing for shifting the start dates of activity B incrementally to the maximum value of the total idle time of 3 days. Figure 5.3e shows strict compliance with the rule of no work interruption; however, applying this rule clearly shows an extension to the overall project duration.

The shift vector SV_i for each activity i at the first typical unit can assume the various shift values T_i ranging from zero to the total idle time by an increment of one day. In this simplified example, SV for activity A and C can assume one value equals to zero, while SV for activity B can assume four values equal to $\{0,1,2,3\}$. Accordingly, the total number of all feasible combinations of shift vectors (*NSV*) can be identified as follows:

$$NSV = \prod_{i=1}^{I} \max_{T \in SV_i}(T_i) \qquad \textit{5-1}$$

Where I is the total number of activities in a typical unit. (for this oversimplified example, $NSV = 4$); however, in a real-life project, there are many more activities in a typical unit with a more complex WFL. For example, an SV_i of 5 days in a project comprising of 20 activities in a typical unit can result in more than ninety-five trillion possible combinations of shift vectors (20^5). This vast search space renders the optimization problem practically infeasible in polynomial time; therefore, an innovative method is developed to reduce the search space.

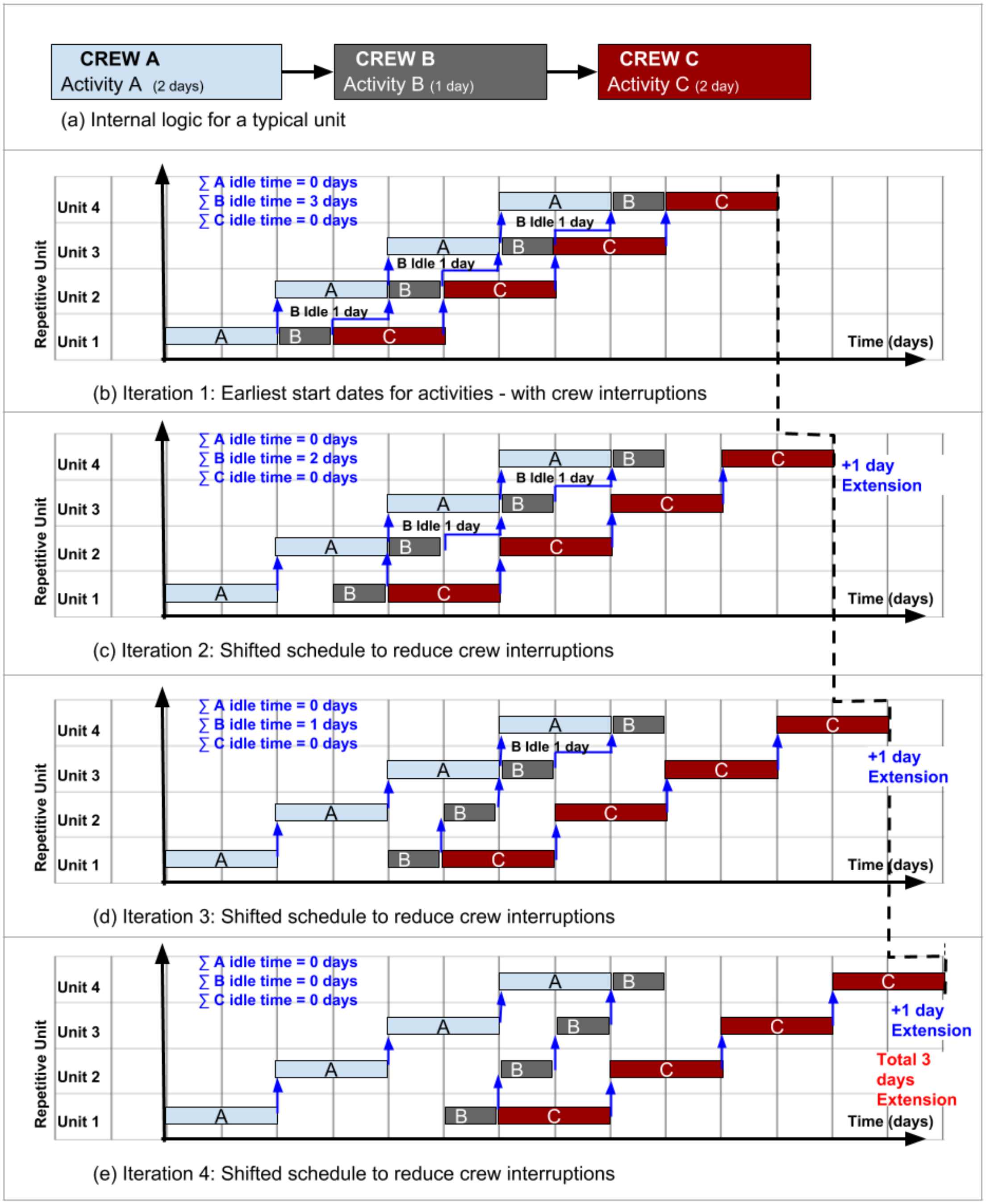


Figure 5.3 Iterative shift of start dates to reduce crew interruption times.

5.4. Formulation of the tri-objective fitness function

This section presents an extension to previously described bi-objective fitness function (see section 4.6) to include simultaneous optimizing of cost, schedule and the third dimension for resource interruption in repetitive construction and thus allowing for crew work continuity.

A vector R is calculated representing the total project interruption days for all crews using Equation (5-10):

As previously described, the cost and schedule vectors are of different magnitudes and thus normalized to the range [0,1] using equations (4-1 and 4-2) Similarly; the interruption vectors can be normalized using equation (5-2):

$$R^{norm} = (R - R^{ideal}) / (R^{nadir} - R^{ideal}) \qquad \textit{5-2}$$

A frontier surface is developed by joining all the possible combinations of cost, schedule and interruption solutions that satisfy a selected JCL confidence level. The vertex point of the normalized frontier surface is the solution having the least vector length $\vec{v}$, providing the optimal balance between time, cost and interruption minimizations. The vector magnitude can be calculated using equation (5-3).

$$\vec{v}_j = \sqrt{}((D_j^{norm})^2 + (C_j^{norm})^2 + (R_j^{norm})^2) \qquad \textit{5-3}$$

Then, the optimal solution $j_{optimal}$ is the vertex point solution that has the least response vector length. Figure 5.4 shows an illustration for the frontier surface and the response vector for a feasible solution.

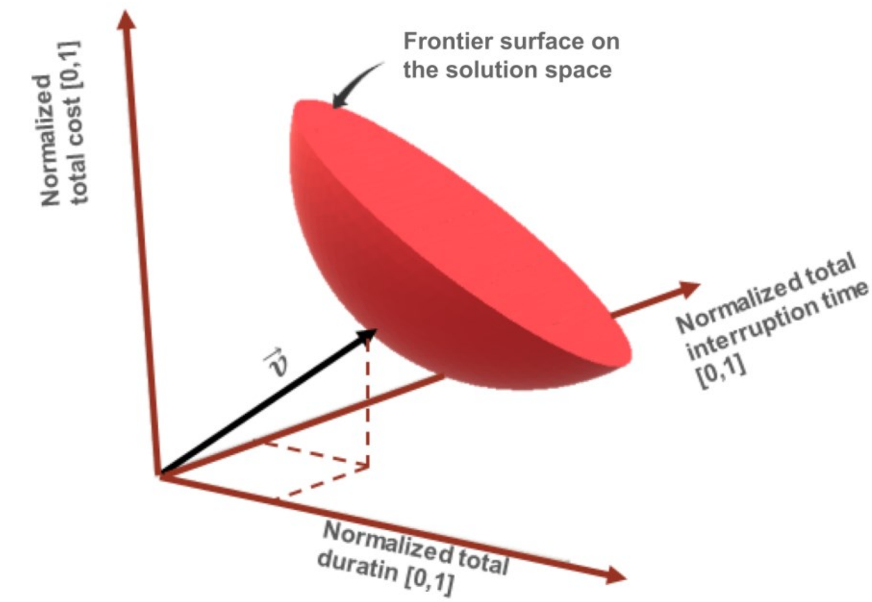


Figure 5.4. Normalized frontier surface.

5.5. Optimization for crew work continuity.

The developed method is based on the blocking technique explained in section 4.5 with a modification for the context of crew work continuity optimization. The project network is segregated into the partitions ($\mathbb{B}, \mathbb{P}, \mathbb{O}$) to analyze and solve for the elite T of primary activities at a given generation. The blocked design of experiments matrix is then generated as the Cartesian product where primary activities can assume all T_i values $\in SV_i$ while blocking all other activities to the maximum SV_i value. A complete enumeration of the resultant matrix is then performed and the elite SV_i shift value for the primary activities are then determined as the shifts satisfying the ultimate tri-objective optimization fitness function, the formation of which will be detailed in the next section. Those elite shift values are carried forward in successive generations where observed activities are blocked to the elite shift values in building the design of experiments matrix. This process allows narrowing the experiment enumeration by focusing the analysis on evaluating the primary activities at each generation.

The data type of the SV parameters is a continuous time factor, and thus by nature, it can be divided into an infinite number of time increments; therefore, for simplicity and for practicality in the application for construction projects, the Interruption days increment are limited to integer values with a minimum of one complete time unit. In this study, the time unit is 1 day. By doing this, the level of factors will not assume part days and thus reducing the search space from an infinite to a finite number of values.

On complex projects, where SV can assume large numbers for a primary activity at a given generation, an iterative method is proposed to further reduce the combinatorial space for the search of the optimal T value. The method segregates the continuous SV parameters into domains and then iteratively narrow the domains until the single elite T value is determined. For example, when the total idle time for a given crew is 10 days, then the SV vector can assume 11 values equal to $\{0,1,2,3, \ldots\ldots, 10\}$. The SV in such a case is then segregated into 2 domains using the equation $domain_i = \{\mho\colon \left\lceil \frac{SV_i}{2} \right\rceil ,\ \Omega\colon SV_i\}$ where the lower factor $\mho$ is the smallest integer greater than or equal to half the SV_i value while the upper factor Ω is equal to the total SV_i. The two domains in this example is then $\{\mho\colon 5\ ,\ \ \Omega\colon 10\}$. Those are then evaluated for fitness in the objective function. If for example, $\mho$ was found more fit, then the domain representing the values from {0 to 5} is dominate to that of the values {6 to 10}. The domain $\mho$ is then classified as elite and considered for the next iteration. In the next iteration the SV_i is adjusted to the elite domain. Accordingly, the upper and lower values are then $\{\mho: 3\ ,\ \ \Omega: 5\}$. If Ω is found elite, then the next iteration will be $\{\mho: 4\ ,\ \ \Omega: 5\}$. If $\mho$ is elite, then that will conclude the optimal T as no further division is foreseen for the value of 4. Figure 5.5 shows a flow chart diagram for the developed crew work interruption time optimization.

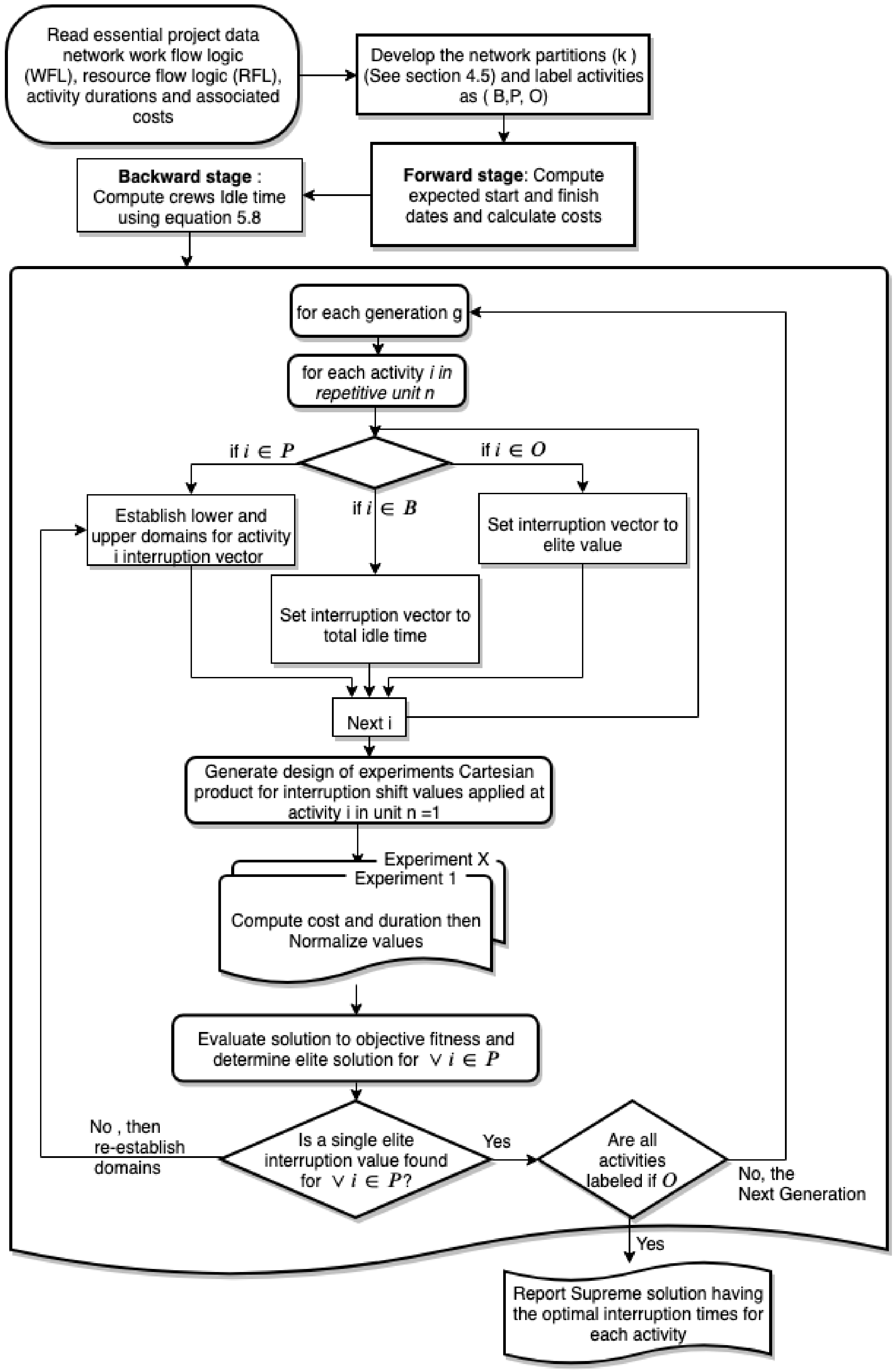


Figure 5.5. Flow chart of proposed crew work interruption time optimization.

5.6. Formulation of the RP-ESDTCT^{Exp}

The RP-ESDTCT method can be formulated using the following set of equations:

$$Objective =$$

$$\begin{cases} Minimize\ \widetilde{C}\ , & for\ \ cost\ minimization \\ Minimize\ \widetilde{D}\ , & for\ schedule\ minimization \\ Minimize\ \vec{v}\ , & for\ Joint\ cost, schedule\ and\ Interruption\ minimization \end{cases} \tag{5-4}$$

Constraints:

$$\mathbb{E}(EST_i^n) =$$

$$\max_{\forall\, j \in P_i} \begin{cases} (\mathbb{E}(EFT_j^n) + lag_{(i,j)} + 1\ ,\ \mathbb{E}(EFT_i^{n-1}) + 1 + T_i) & ,\ logic_{i,j} = Finish\ to\ Start \\ (\mathbb{E}(EST_j^n) + lag_{(i,j)}\ \ ,\ \mathbb{E}(EFT_i^{n-1}) + 1 + T_i) & ,\ logic_{i,j} = Start\ to\ Start \\ (\mathbb{E}(EFT_j^n) +\ lag_{(i,j)} - \widetilde{d}\,_i^n + 1\ ,\ \mathbb{E}(EFT_i^{n-1}) + 1 + T_i) & ,\ logic_{i,j} = Finish\ to\ Finish \\ (\mathbb{E}(EST_j^n) +\ lag_{(i,j)} - \widetilde{d}\,_i^n\ \ ,\ \mathbb{E}(EFT_i^{n-1}) + 1\ + T_i) & ,\ logic_{i,j} = Start\ to\ Finish \end{cases} \tag{5-5}$$

$$\mathbb{E}(EFT_i^n)\ =\ \mathbb{E}(EST_i^n) +\ \widetilde{d}\,_i^n - 1 \tag{5-6}$$

$$\widetilde{d}\,_i^n = \begin{cases} \sum_{k=1}^{K_i} g_{i,k}\mathbb{E}\left(\frac{qty_{i,k}^n}{PF_{i,k}}\middle|p, a_{PF_{i,k}}, m_{PF_{i,k}}, b_{PF_{i,k}}\right) \forall i \in \mathbb{P} \\ \mathbb{E}\left(\frac{qty_{i,ek}^n}{PF_{i,ek}}\middle|p, a_{PF_{i,ek}}, m_{PF_{i,ek}}, b_{PF_{i,ek}}\right) \qquad \forall i \in \mathbb{O} \\ \mathbb{E}\left(\frac{qty_{i,k1}^n}{PF_{i,k1}}\middle|p, \forall a_{PF_i}, b_{PF_i} =\ m_{PF_{i,k1}}\right) \qquad \forall i \in \mathbb{B} \end{cases} \tag{5-7}$$

$$sd_i^n\ =\ \mathbb{E}(EST_i^{n+1}) -\ \mathbb{E}(EFT_i^n\,) \qquad 2 \le n \le N \tag{5-8}$$

$$Sd_i^N\ =\ \textstyle\sum_{n=1}^{N} sd_i^n \tag{5-9}$$

$$R\ =\ \textstyle\sum_{i=1}^{I}\sum_{n=1}^{N} sd_i^n \tag{5-10}$$

$$SV_i\ \in \{0 \to Sd_i^N\} \qquad \forall\, i \tag{5-11}$$

$$T_i = \begin{cases} \begin{Bmatrix} \sum_{r=0}^{r=Sd_i^N} h_{i,k}SV_{i,k} & \forall\, i \in \mathbb{P} \\ SV_{i,er} & \forall\, i \in \mathbb{S} \\ Sd_i^N & \forall\, i \in \mathbb{N} \end{Bmatrix} & ,\ n = 1 \\ 0 & ,\ n \neq 1 \end{cases} \tag{5-12}$$

$$h_i\ \in \{1 \to Sd_i^N + 1\} \qquad \forall\, i \tag{5-13}$$

$$\textstyle\sum_{r=1}^{Sd_i^N} h_{i,r} = 1 \qquad \forall\, i \tag{5-14}$$

$$\widetilde{FC_i^n} = \begin{cases} \sum_{k=1}^{K_i} g_{i,k}\mathbb{E}(FC_{i,k}^n \mid p, a_{FC_{i,k}^n}, m_{FC_{i,k}^n}, b_{FC_{i,k}^n}) & \forall\, i \in \mathbb{P} \\ \mathbb{E}(FC_{i,ek}^n \mid p, a_{FC_{i,ek}^n}, m_{FC_{i,ek}^n}, b_{FC_{i,ek}^n}) & \forall\, i \in \mathbb{O} \\ \mathbb{E}(FC_{i,k1}^n \mid p, \forall\ a_{FC_i^n}, b_{FC_i^n} = m_{FC_{i,k1}^n}) & \forall\, i \in \mathbb{B} \end{cases} \qquad \textit{5-15}$$

$$\widetilde{VC_i^n} = \begin{cases} \sum_{k=1}^{K_i} g_{i,k}\mathbb{E}(VCr_{i,k}^n \mid p, a_{VCr_{i,k}^n}, m_{VCr_{i,k}^n}, b_{VCr_{i,k}^n}) \,.\, \widetilde{d}_i^n & \forall\, i \in \mathbb{P} \\ \mathbb{E}(VCr_{i,ek}^n \mid p, a_{VCr_{i,ek}^n}, m_{VCr_{i,ek}^n}, b_{VCr_{i,ek}^n}).\, \widetilde{d}_i^n & \forall\, i \in \mathbb{O} \\ \mathbb{E}(VCr_{i,k1}^n \mid p, \forall\ a_{VCr_i^n}, b_{VCr_i^n} = m_{VC_{ri,k1}^n}).\, \widetilde{d}_i^n & \forall\, i \in \mathbb{B} \end{cases} \qquad \textit{5-16}$$

$$\sum_{k=1}^{K_i} g_{i,k} = 1 \quad \forall\, i \qquad \textit{5-17}$$

$$g_{i,k} \in \{1,2,3, \ldots K_i\} \quad \forall\, i, k \qquad \textit{5-18}$$

$$sc_i^N = sd_i^n \times \widetilde{VC_i^n} \qquad \textit{5-19}$$

$$Sc_i^N = \sum_{n=1}^{N} sc_i^n \qquad \textit{5-20}$$

$$\ddot{D} = \max_{\forall\, i,n}\left\{\mathbb{E}(EFT_i^n)\right\} \qquad \textit{5-21}$$

$$\ddot{C} = \sum_{\forall\, i,n}\left(\widetilde{FC_i^n}\right) + \sum_{\forall\, i,n}\left(\widetilde{VC_i^n}\right) + \left(ICr\,.\,\ddot{D}\right) + \sum_{\forall\, i}(Sc_i^N)\,) \qquad \textit{5-22}$$

$$\dddot{C} = \ddot{C} + PC + BC \qquad \textit{5-23}$$

$$\mathbb{Q} = \prod_{g=1}^{G}\prod_{s=1}^{S} \dddot{D}_{(g,s)}, \dddot{C}_{(g,s)} \mid (P\left\{\dddot{D}\right\} \le \alpha \ \wedge\ P\left\{\dddot{C}\right\} \le \alpha) \qquad \textit{5-24}$$

$$\widehat{\mathbb{Q}} = \prod \dddot{D}, \dddot{C} \ \ \forall\, s \in$$

$$\mathbb{Q} \mid \begin{cases} Minimize\ \dddot{C} \ \wedge\ Minimize\ \dddot{D}, for\ Joint\ cost\ and\ schedule\ minimization \\ Minimize\ \dddot{C}\ , \qquad for\ cost\ minimization \\ Minimize\ \dddot{D}\ , \qquad for\ schedule\ minimization \end{cases} \qquad \textit{5-25}$$

$$\widetilde{D} = min\left\{\dddot{D}\right\}\ \forall\, q \in \widehat{\mathbb{Q}} \qquad \textit{5-26}$$

$$\tilde{C} = min\left\{\dddot{C}\right\}\ \forall\, q \in \widehat{\mathbb{Q}} \qquad \textit{5-27}$$

$$\tilde{R} = min\left\{R\right\}\ \forall\, q \in \widehat{\mathbb{Q}} \qquad \textit{5-28}$$

$$\vec{v} = \sqrt{(\left(\widetilde{D}^{norm}\right)^2 + \left(\widetilde{C}^{\,norm}\right)^2 + \left(\widetilde{R}^{\,norm}\right)^2)} \qquad \textit{5-29}$$

The mathematical model is described by Equations (5-5 to 5-28) as constraints and Equation (5-4) for the objective functions. The objective can be set for one of three cases. The first is to minimize the $\widetilde{C}$ when the value of the total project cost minimization is of interest, the second is to minimize $\widetilde{D}$ when the value for the total project schedule minimization is of interest and, the third is the tri-objective minimization $\widetilde{D}$, $\widetilde{C}$ and, vector R representing the total interruption time.

In equation (5-5 to 5-6), $\mathbb{E}(EST_i)$, $\mathbb{E}(EFT_j)$ are the expected value of the early start time for activity i and the expected value of the early finish time for its predecessor activity j. While $\tilde{d}$ is the expected value for the activity duration. To account for non-typical repetitive activities, $\tilde{d}$ is taken as a function of the activity quantities qty and the productivity factor PF for the assigned crew, then $d = qty/PF$. The value T_{i} represent a shift to the early start of an activity to allow for work interruptions. The calculation for the EST in equation (5-5) allows for the four types of logical relationships between the activities and allows for the logic between the repetitive units n and $(\mathrm{n}-1)$ as finish-to-start relationship to ensure a timely movement of the crew from one unit to the next. Equation (5-7) computes the value of $\widetilde{d}$ based on the activity classification at a given generation. The labeling of the network activities is $\mathbb{P}$ for primary mode activities, $\mathbb{O}$ for observed activities and $\mathbb{B}$ for base case mode activities. Those labeling of activities at a given generation is described in Section 4.5. Where each activity i can assume a different mode k amongst the total number of modes K available for the activity. The activity mode assignment is applied to all similar activities i across all repetitive units $n \rightarrow N$. The reasoning behind this application is that the selected crew formation option and its characteristics for cost and schedule performance will be the same carried through from one unit to another throughout the life span of the project; further reasoning is also explained in section 5.2. To account for uncertainty in $\widetilde{d}$, the expected value is taken as a random sampling of the probability distribution function applied to the uncertainty in the crew PF. To allow calculation of the expected value, the inverse transformation of the cumulative distribution function is calculated. In the case of triangular PDF, the parameters are (p, a, m, b). Where p is a pseudo-random number generator of a uniform random variable on [0,1], and a, m, b are the optimistic, most likely and pessimistic values. The value for T_{i} is calculated using Equations (5-8 to 5-12) where the idle time sd for activity i in unit n is calculated in Equation (5-8) as simply the time between the finish date and the start date of the matching activity in unit $n+1$. The overall project's idle time Sd for a given crew assigned to activity i is then calculated in Equation (5-9). Equation (5-10) calculates the magnitude of the vector R representing the total interruption time for all crews. Equation (5-12) applies the shift in start dates at the activities in the first unit. h_{i} is an arbitrary decimal value ranging between 1 and $Sd_i^N + 1$. This value is used here to tag the increment value SV_i as a discrete option that is in turn used in the developed method to perform the enumeration experiments to shift the start time of activity i until full compliance to work continuity with zero interruptions is achieved. Equation (5-14) expresses that only one shift value must be admitted for each activity at any given experiment run. The resultant matrix from equation (5-12) is a full factorial design of experiments with a blocking technique. Where the activity classification belongs to $\mathbb{P}$, the values for T_i assume all the admitted options ($h_i = 1 \ \rightarrow (Sd_i^N + 1))$. When the activity classification belongs to $\mathbb{S}$, the shift value T_i are blocked to assume only the value of the elite option $SV_{i,er}$ and when the activity classification belongs to $\mathbb{N}$, the parameters are blocked to assume only the total idle time values for the activities Sd_i^N. The reasoning behind this blocking technique is further described in Section 4.5.

Similarly, equation (5-15) and (5-16) computes the expected value for the fixed cost ($\widetilde{\mathrm{FC}}$) and the expected value for the variable cost ($\widetilde{\mathrm{VC}}$) for each activity.

The resultant matrix from equations (5-7), (5-15) and (5-16) is a full factorial design of experiments with a blocking technique. Where the activity classification belongs to $\mathbb{P}$, the set of parameter values for $d, \widetilde{\mathrm{FC}}, \widetilde{\mathrm{VC}}$ assume the values for all the admitted modes $k \ \rightarrow K$. When the activity classification belongs to $\mathbb{S}$, the parameters are blocked to assume only the values of the elite mode ek and when the activity classification belongs to $\mathbb{N}$, the parameters are blocked to assume only the values for the base case mode $k_1(\text{or mode } 1)$. The reasoning behind this blocking technique is further described in Section 4.5. Binary variables $g_{i,k}$ in equations (5-17) and (5-18) expresses that only one mode must be admitted for each activity. Equations (5-19) and (5-20) computes the cost resultant from a crew idle time. The assumption made here is that the ($\widetilde{\mathrm{VC}}$) will still be incurred on the project during the idle time. The reasoning behind this assumption is the project will need to bear the cost associated with tools, equipment and the labour salaries during the idle times.

Equation (5-22) computes the total project cost $\ddot{C}$ at any given experiment run taking into account the expected value of the indirect cost rate per day (IC) over the overall project's duration $\ddot{D}$ computed in equation (5-21) . Equation (5-23) adds the penalty cost (PC) and the bonus costs (BC), the formulation for such costs were detailed in section 3.3. Equation (5-24) is a matrix $\mathbb{Q}$ of all solutions falling on the frontier curve defined by a specified joint cost – schedule confidence level α. Equation (5-25) provides the matrix $\widehat{\mathbb{Q}}$ which is a subset of $\mathbb{Q}$ for all solutions q satisfying the defined minimization objective function. Equations (5-26, 5-27 and 5-28) reduces the matrix $\widehat{\mathbb{Q}}$ to the single optimum solution satisfying the objective function. Equation (5-29) computes the magnitude of the response vector $\vec{v}$ required for the joint optimization of time, cost and interruptions.

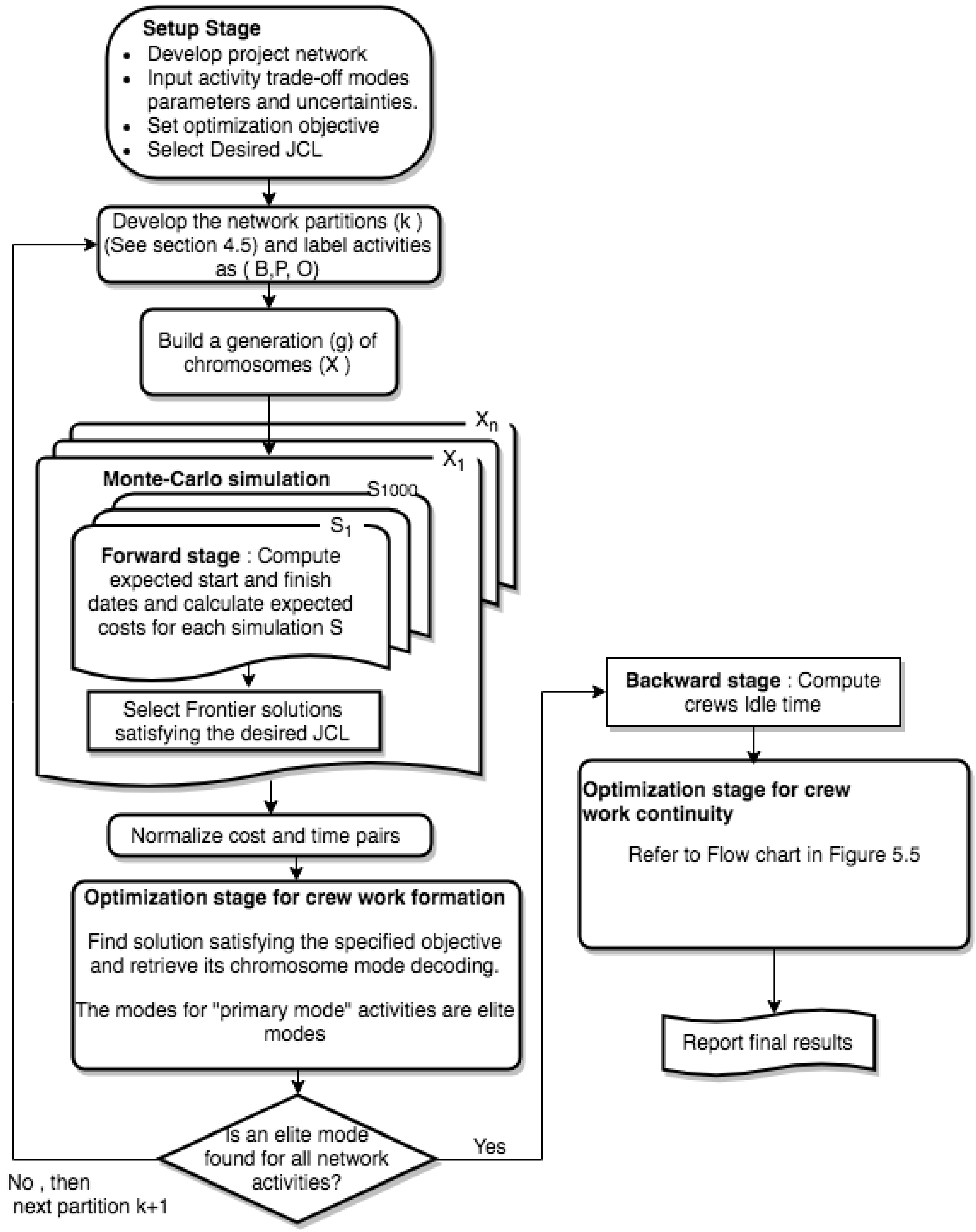


Figure 5.6. Flow chart of RP-ESDTCT^{EXP} method.

5.7. Numerical examples of repetitive projects

5.7.1. Example 7: Testing RP-ESDTCT^{Exp} module under certain environment.

This example is set for the purpose of verification against a deterministic scenario to solve for crew formation chromosome optimization. In this scenario, the input data parameters are modelled by converting the adopted triangular probability distribution to a deterministic single value, where (a, m and b) are equal values equal to the most likely value m. By doing so, the data input for the example is matched to the basic data input and thus allows for comparative studies with other deterministic approaches introduced by previous researchers. The example is drawn from the literature known as the three-span concrete bridge is analyzed to demonstrate the capabilities of the developed method in searching the optimal time – cost trade-offs for non-typical activities in repetitive projects. The project consists of four typical units each having five non-typical activities; those activities are excavation, foundation, columns, beams and slabs. The precedence relationships among these successive activities are finish-to-start with no lag time. The non-typical activities are due to different quantities associated with the different units. Figure 5.7 shows an illustration of the example.

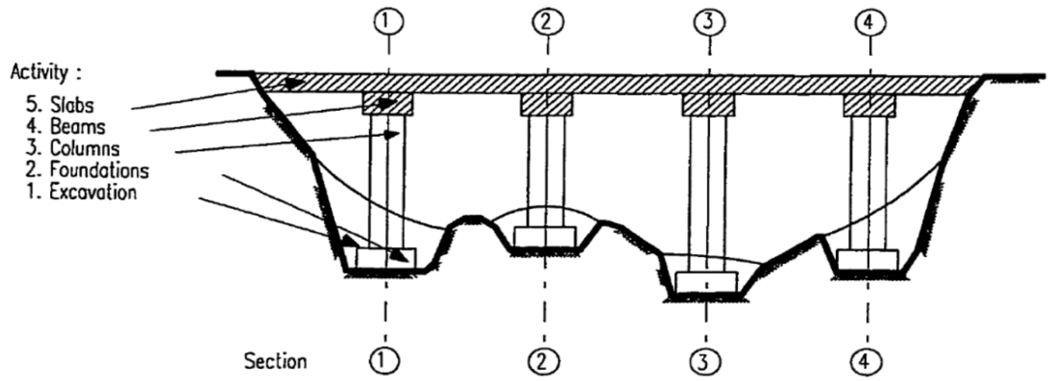


Figure 5.7 Three-span concrete bridge illustration (Selinger 1980).

This example was adopted in many research efforts in the past forty years for validation of a wide range of schedule optimization methods for linear repetitive projects. The example was originally developed by Selinger (1980) using deterministic dynamic programming to solve for reducing the project duration while maintaining resource continuity. Russell and Caselton (1988) used this example to verify their model that extended the work of Selinger (1980) and added the ability to accommodate typical and non-typical actives and to allow for the possibility of having user-specified work interruptions using deterministic dynamic programming. Moselhi and El-Rayes (1993) and El-Rayes (1997) adopted this example in their deterministic dynamic programming method; technique while accounting for cost as an important decision variable in the optimization process for crew work formation. El-Rayes (1997) used this example with modifications to solve for optimization of crew formations using a deterministic dynamic programming technique. Later, El-Rayes and Moselhi (2001) further developed their deterministic dynamic programming technique to automate the generation of interruptions during scheduling to make the interruption more feasible and bounded; the authors used this example to solve for the optimization of resource utilization. Hyari and El- Rayes (2004) and Hyari et. al (2009) also used this example to validate their deterministic multi-objective genetic algorithm and scheduling algorithm introducing the capability of simultaneous minimization of both project duration and work interruptions for construction crews. Nassar (2005) solved this example using a deterministic genetic algorithm to optimally assign resources for repetitive construction projects with the aim to find the optimal crew formations and interruption times that results in least project duration while simultaneously reducing the number of interruption days. Liu and Wang (2007) solved the example using a constraint programming backtracking approach to optimize as the searching algorithm for model formulation for deterministic optimization of either total project cost is or project duration. Long and Ohsato (2009) analyzed this example using a genetic algorithm-based method to solve for the minimization of project duration, project cost, or both of them. Bakry et al. (2016) also used this example to validate their fuzzy dynamic programming model accounting for schedule and cost uncertainties. Lately, Eid et al. (2018) used this example to validate their deterministic genetic algorithms and Pareto front sorting model. Salama and Moselhi (2019) used the example to validate their uncertain multi-objective optimization model using an integration of linear scheduling with the critical chain scheduling method.

The extensive utilization of the selected example in the literature provides a sound basis for the validation of the developed method. The network configuration of the example is shown in Figure 5.8. Each mode represents a crew formation in which the labour cost and productivity are presented in Table 5.1 along with quantities of work.

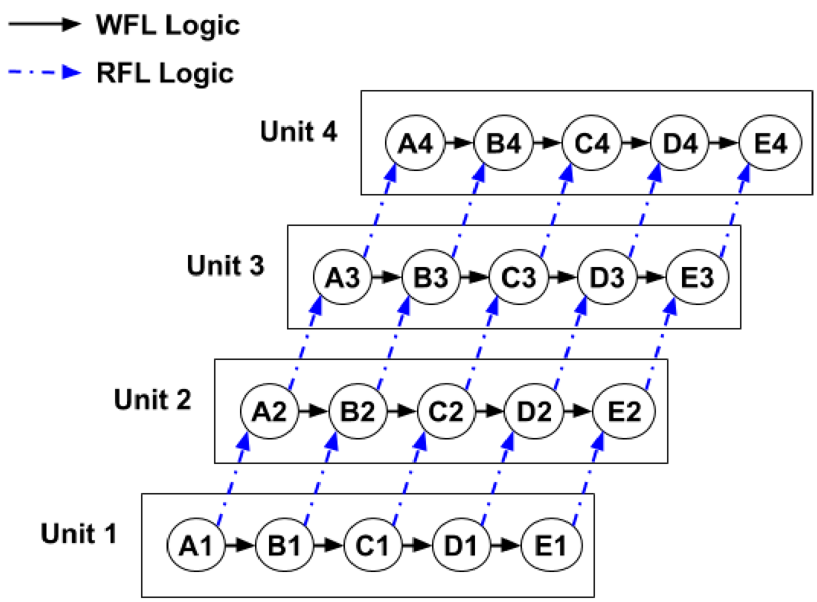


Figure 5.8 Example 7: Network configuration.

Table 5.1 Example 7: Data input (adapted from El-Rayes (1997))

Activity	Quantity of work (*qty*) in m3				Material cost rate in $/m3	Material cost ($) = qty * Material cost rate or (fixed cost)			
Repetitive unit (n)	**1**	**2**	**3**	**4**	**1**	**1**	**2**	**3**	**4**
A) Excavation	1147	1434	994	1529	0	0	0	0	0
B) Foundation	1032	1077	943	896	92	94944	99084	86756	82432
C) Columns	104	86	129	100	479	49816	41194	61791	47900
D) Beams	85	92	101	80	195	16575	17940	19695	15600
E) Slabs	0	138	114	145	186	0	25668	21204	26970

Activity	Crew Formation No. (Mode)	Productivity Factor (*PF*) in m3/day	Duration of work in days = *qty / PF*				Lab. and equip. cost rate ($/day) or (variable cost rate)
Repetitive unit (n)			**1**	**2**	**3**	**4**	
A) Excavation	1	91.75	12.50	15.63	10.83	16.66	906
B) Foundation	1	89.77	11.50	12.00	10.50	9.98	4678
	2	71.81	14.37	15.00	13.13	12.48	3508
	3	53.86	19.16	20.00	17.51	16.64	2338
C) Columns	1	5.73	18.15	15.01	22.51	17.45	2160
	2	6.88	15.12	12.50	18.75	14.53	2809
	3	8.03	12.95	10.71	16.06	12.45	3456
D) Beams	1	9.9	8.59	9.29	10.20	8.08	4246
	2	8.49	10.01	10.84	11.90	9.42	3497
	3	7.07	12.02	13.01	14.29	11.32	2748
	4	5.66	15.02	16.25	17.84	14.13	1998
E) Slabs	1	8.73	0.00	15.81	13.06	16.61	2407
	2	7.76	0.00	17.78	14.69	18.69	2027

The performance criteria for the present method are set such that the supreme chromosome result should be no worse than that reported by Hyari et al. (2009) solved using a genetic algorithm approach.

The present method is run with two scenarios. The first scenario is for the purpose of verification against deterministic approaches to solve for crew formation chromosome optimization. In this scenario, the input data parameters are modelled by converting the adopted triangular probability distribution function to a deterministic single value, where (a, m and b) are equal values equal to the most likely value m. By doing so, the data input for the example is matched to the basic data input presented by El-Rayes (1997) and thus allows for comparative studies with other deterministic approaches introduced by previous researchers.

Table 5.3 lists the comparative results for optimum crew formation chromosomes having the objective function set for schedule minimization. The total duration, crew formations and total crews interruption time are found firmly confirming the best solutions reported by Long and Ohsato (2009) and Hyari et al. (2009). Selinger (1980) formulation was with a strict no crews interruption time; for this reason, the total duration was extended to 117.9 days; resulting in a different crew formation chromosome. Russell and Caselton (1988) formulation allowed for a user prespecified set of interruption times and resulted in an improvement for the total project duration; however, it was unable to obtain the optimum solution.

A daily project indirect cost of $2,500 is used and identical to that used by El-Rayes and Moselhi (2001) and Hyari et al. (2009).

Table 5.2 Example 7: RP-ESDTCT^{EXP} supreme chromosome different optimization objectives under certain environment.

						Activity ID >>	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E	Total Interruption (days)
						Num of available modes >>	1	3	3	4	2											
Optimization objective	Total duration	Total cost	Total Material cost ($) (Fixed cost)	Total Lab. and equip. Cost ($) (Variable cost)	Total indirect cost	Total cost for idle crew time	Chromosome					Shift Time (Delayed start of activity at unit 1) (days)					Total Crew Idle Time (Interruption) (days)					
a) Schedule minimization	106.8	$1,740,753	$707,753	$699,572	$266,931	$66,497	1	1	3	1	1	0	3	0	3	9	0	6	0	8	0	14
b) Cost minimization	123.6	$1,667,018	$707,753	$650,268	$308,997	$0	1	2	2	4	1	0	1	0	0	20	0	0	0	0	0	0
c) Joint cost - schedule - Interruption minimization	134.2	$1,726,357	$707,753	$642,814	$335,416	$40,374	1	3	1	1	1	0	0	3	18	9	0	0	1	9	1	11

Table 5.3 Example 7: Comparative results for the schedule minimization under certain environment.

Activity ID >>		A	B	C	D	E	A	B	C	D	E	Total Interruption (days)
Num of available modes >>		1	3	3	4	2						
Author	Total duration	Chromosome					Total Crew Idle Time (Interruption) (days)					
Selinger (1980) One state variable formulation	117.9	1	2	3	3	1	0	0	0	0	0	0
Russell and Caselton (1988) Two state variable formulation	110.4	1	1	3	1	1	0	4	0	12	0	16
El-Rayes and Moselhi (2001) Two state variable formulation	106.8	1	1	3	1	1	0	6	0	9	0	15
Long and Ohsato (2009) Genetic algorithm-based formulation	106.8	1	1	3	1	1	0	6	0	8	0	14
Hyari et al. (2009) Two state variable formulation	106.8	1	1	3	1	1	Not reported					14
Present method	106.8	1	1	3	1	1	0	6	0	8	0	14

Table 5.4 lists the comparative results for optimum crew formation chromosomes having the objective function set for cost minimization. The total cost, duration, crew formations and total crews interruption time is found confirming to that reported by El-Rayes and Moselhi (2001) and Hyari et al. (2009). There is a 0.11% difference in cost values between the different results generally attributed to rounding of numbers exercised by previous studies.

Table 5.4 Example 7: Comparative results for the cost minimization under certain environment.

						Activity ID >>	A	B	C	D	E	A	B	C	D	E	
						Num of available modes >>	1	3	3	4	2						
Author	Total duration	Total cost	Total Material cost ($) (Fixed cost)	Total Lab. and equip. Cost ($) (Variable cost)	Total indirect cost	Total cost for idle crew time	Chromosome					Total Crew Idle Time (Interruption) (days)					Total Interruption (days)
El-Rayes and Moselhi (2001) Two state dynamic programming formulation	123.6	$1,665,247	$1,356,348		$308,899	NA*	1	2	2	4	1	NA*					NA*
Hyari et al. (2009) Two state variable formulation	124	$1,668,021	$1,358,021		$310,000		1	2	2	4	1	0	0	0	0	0	0
Present model	123.6	$1,667,018	$707,753	$650,268	$308,997	$ -	1	2	2	4	1	0	0	0	0	0	0

* Data was not reported by the author

The example was also solved to find a solution satisfying the joint cost, schedule and Interruption minimization as listed in Table 5.2. It is noted that this solution was not available in previous studies. The solution resulted in a different crew formation that, together with permitted crew interruptions, provide a balance between the cost, schedule and interruption vectors.

The developed method is coded in a spreadsheet-based application using Google sheets cloud applications using Google Apps Script. The application was not run in the browser but rather remotely on the Google cloud. Google BigQuery SQL is used to facilitate the calculation procedure. The total run time to obtain the supreme chromosome solution for the desired objective averaged at 41 seconds using the Google cloud application and the free BigQuery quota limitations (Maharana et al. 2015).

5.7.2. Example 8: Testing RP-ESDTCT^{Exp} module under uncertain environment.

The example is aimed to illustrate the developed method capability to account for uncertainties in the cost and schedule parameters and solved to find a solution satisfying the joint cost, schedule and Interruption minimization. The same project network and details of example 7 is used. The first step was to transform the deterministic data into uncertain data represented by a triangular PDF. The original deterministic value was used as the most likely (m) value. The deterministic value was once multiplied by a factor less than 1.0 to get the optimistic (a) value, and once by a factor greater than 1.0 to get the pessimistic (b) value. The uncertainty multiplier factors were generated randomly between 70% and 130% of the deterministic value and applied to material cost, productivity factors and labour and equipment costs rates. Table 5.5 lists the generated multiplier factors, and the produced uncertain data numbers are listed in Table 5.6. Uncertainty is also presented in the daily project indirect cost represented by the triangular PDF ($2,125, $2,500, $3,000). The present model was run for a selected JCL = 70% to find the supreme crew formation chromosome satisfying the objectives a) schedule minimization, b) cost minimization and c) Joint cost, schedule and interruption minimization. Run outputs are summarized in Table 5.7. The results produced higher schedule and cost values in comparison to those produced in example 7 under a deterministic environment. Also, it is noticed that except for the schedule minimization objective, the supreme chromosome is found different in certain gens, where different crew formation selections are more fit for an objective function than the other. In the case of schedule minimization objective, the supreme chromosome is found to be [1,1,3,1,1] for activities A to E, respectively. This chromosome is the same in both the deterministic and uncertain scenarios and constantly produced the minimum project duration.

In order to study the sensitivity of the different targeted JCL, the same numerical example is analyzed after setting the JCL at different values ranging from 10% to 90% confidence levels; several scenario runs were analyzed under different optimization objectives. For each scenario, the supreme chromosome solution is identified as shown in Table 5.8. This illustrates that different crew formation selections are more fit for an objective function than the other and vary according to the targeted JCL.

The total run time to obtain the supreme chromosome solution for the desired objective averaged at 65 seconds. It can be noticed that the average run execution time for the developed method is not that significantly different from that in a deterministic scenario in spite of the fact that the execution under uncertainty resulted in an exponential increase in computations required to simulate each experiment and furthermore the solutions are examined for optimum crew work continuity. The reason for this is the expandable nature of the selected computational platform (BigQuery) being executed online on Google servers. BigQuery automates the computational power needed depending on the query size and complexity. The query is divided and then concurred for parallel executions.

Table 5.5 Example 8: Uncertainty multiplier factors for cost and productivity factors.

Activity	Uncertainty multipliers for total material cost ($) (Fixed cost) (a, m, b)			
Repetitive unit (n)	**1**	**2**	**3**	**4**
A) Excavation	0	0	0	0
B) Foundation	a: 0.88 m: 1.00 b: 1.13	a: 0.87 m: 1.00 b: 1.29	a: 0.84 m: 1.00 b: 1.22	a: 0.91 m: 1.00 b: 1.22
C) Columns	a: 0.91 m: 1.00 b: 1.29	a: 0.73 m: 1.00 b: 1.2	a: 0.95 m: 1.00 b: 1.13	a: 0.84 m: 1.00 b: 1.1
D) Beams	a: 0.7 m: 1.00 b: 1.15	a: 0.94 m: 1.00 b: 1.13	a: 0.88 m: 1.00 b: 1.13	a: 0.87 m: 1.00 b: 1.25
E) Slabs	a: 0.74 m: 1.00 b: 1.14	a: 0.83 m: 1.00 b: 1.06	a: 0.93 m: 1.00 b: 1.12	a: 0.8 m: 1.00 b: 1.26

Activity	Crew formation No.	Uncertainty multipliers for productivity factor (PF) in m3/day (a, m, b)				Lab. and equip. Variable cost rate ($/day) (a, m, b)			
Repetitive unit (n)		**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
A) Excavation	1	a: 0.8 m: 1.00 b: 1.08	a: 0.78 m: 1.00 b: 1.29	a: 0.78 m: 1.00 b: 1.13	a: 0.93 m: 1.00 b: 1.18	a: 0.76 m: 1.00 b: 1.23	a: 0.89 m: 1.00 b: 1.19	a: 0.74 m: 1.00 b: 1.27	a: 0.81 m: 1.00 b: 1.28
B) Foundation	1	a: 0.7 m: 1.00 b: 1.05	a: 0.82 m: 1.00 b: 1.19	a: 0.82 m: 1.00 b: 1.07	a: 0.93 m: 1.00 b: 1.28	a: 0.84 m: 1.00 b: 1.26	a: 0.73 m: 1.00 b: 1.12	a: 0.71 m: 1.00 b: 1.05	a: 0.73 m: 1.00 b: 1.28
	2	a: 0.91 m: 1.00 b: 1.1	a: 0.87 m: 1.00 b: 1.27	a: 0.87 m: 1.00 b: 1.23	a: 0.7 m: 1.00 b: 1.23	a: 0.92 m: 1.00 b: 1.2	a: 0.82 m: 1.00 b: 1.15	a: 0.78 m: 1.00 b: 1.3	a: 0.82 m: 1.00 b: 1.16
	3	a: 0.84 m: 1.00 b: 1.09	a: 0.72 m: 1.00 b: 1.3	a: 0.95 m: 1.00 b: 1.13	a: 0.78 m: 1.00 b: 1.14	a: 0.84 m: 1.00 b: 1.13	a: 0.71 m: 1.00 b: 1.27	a: 0.73 m: 1.00 b: 1.21	a: 0.82 m: 1.00 b: 1.24
C) Columns	1	a: 0.85 m: 1.00 b: 1.21	a: 0.84 m: 1.00 b: 1.18	a: 0.78 m: 1.00 b: 1.27	a: 0.75 m: 1.00 b: 1.12	a: 0.71 m: 1.00 b: 1.2	a: 0.86 m: 1.00 b: 1.15	a: 0.85 m: 1.00 b: 1.09	a: 0.89 m: 1.00 b: 1.06
	2	a: 0.84 m: 1.00 b: 1.26	a: 0.92 m: 1.00 b: 1.3	a: 0.75 m: 1.00 b: 1.21	a: 0.9 m: 1.00 b: 1.06	a: 0.85 m: 1.00 b: 1.3	a: 0.91 m: 1.00 b: 1.18	a: 0.7 m: 1.00 b: 1.23	a: 0.87 m: 1.00 b: 1.27
	3	a: 0.82 m: 1.00 b: 1.11	a: 0.9 m: 1.00 b: 1.09	a: 0.92 m: 1.00 b: 1.11	a: 0.95 m: 1.00 b: 1.12	a: 0.93 m: 1.00 b: 1.13	a: 0.71 m: 1.00 b: 1.06	a: 0.73 m: 1.00 b: 1.05	a: 0.84 m: 1.00 b: 1.17
D) Beams	1	a: 0.84 m: 1.00 b: 1.06	a: 0.88 m: 1.00 b: 1.1	a: 0.81 m: 1.00 b: 1.05	a: 0.91 m: 1.00 b: 1.13	a: 0.9 m: 1.00 b: 1.25	a: 0.75 m: 1.00 b: 1.13	a: 0.91 m: 1.00 b: 1.24	a: 0.72 m: 1.00 b: 1.07
	2	a: 0.72 m: 1.00 b: 1.25	a: 0.72 m: 1.00 b: 1.12	a: 0.8 m: 1.00 b: 1.15	a: 0.83 m: 1.00 b: 1.19	a: 0.88 m: 1.00 b: 1.24	a: 0.85 m: 1.00 b: 1.13	a: 0.89 m: 1.00 b: 1.3	a: 0.94 m: 1.00 b: 1.14
	3	a: 0.82 m: 1.00 b: 1.09	a: 0.87 m: 1.00 b: 1.05	a: 0.84 m: 1.00 b: 1.06	a: 0.93 m: 1.00 b: 1.08	a: 0.91 m: 1.00 b: 1.27	a: 0.9 m: 1.00 b: 1.24	a: 0.84 m: 1.00 b: 1.12	a: 0.94 m: 1.00 b: 1.11
	4	a: 0.7 m: 1.00 b: 1.21	a: 0.71 m: 1.00 b: 1.05	a: 0.93 m: 1.00 b: 1.15	a: 0.82 m: 1.00 b: 1.11	a: 0.91 m: 1.00 b: 1.12	a: 0.86 m: 1.00 b: 1.18	a: 0.94 m: 1.00 b: 1.09	a: 0.89 m: 1.00 b: 1.05
E) Slabs	1	a: 0.82 m: 1.00 b: 1.21	a: 0.8 m: 1.00 b: 1.13	a: 0.7 m: 1.00 b: 1.1	a: 0.91 m: 1.00 b: 1.27	a: 0.82 m: 1.00 b: 1.27	a: 0.9 m: 1.00 b: 1.15	a: 0.7 m: 1.00 b: 1.25	a: 0.83 m: 1.00 b: 1.19
	2	a: 0.79 m: 1.00 b: 1.14	a: 0.8 m: 1.00 b: 1.21	a: 0.74 m: 1.00 b: 1.11	a: 0.75 m: 1.00 b: 1.23	a: 0.89 m: 1.00 b: 1.17	a: 0.89 m: 1.00 b: 1.19	a: 0.71 m: 1.00 b: 1.15	a: 0.7 m: 1.00 b: 1.15

Table 5.6 Example 8: Uncertain Data input.

Activity ID	Crew Formation	Repetitive Unit	Activity duration			Activity fixed cost			Activity variable cost		
			a	m	b	a	m	b	a	m	b
A	1	1	10.0	12.5	13.5	$0	$0	$0	$689	$906	$1,114
A	1	2	12.2	15.6	20.2	$0	$0	$0	$806	$906	$1,078
A	1	3	8.4	10.8	12.2	$0	$0	$0	$670	$906	$1,151
A	1	4	15.5	16.7	19.7	$0	$0	$0	$734	$906	$1,160
B	1	1	8.1	11.5	12.1	$83,551	$94,944	$107,287	$3,930	$4,678	$5,894
B	1	2	9.8	12.0	14.3	$87,194	$99,084	$111,965	$3,415	$4,678	$5,239
B	1	3	8.6	10.5	11.2	$76,345	$86,756	$98,034	$3,321	$4,678	$4,912
B	1	4	9.3	10.0	12.8	$72,702	$82,616	$93,356	$3,415	$4,678	$5,988
B	2	1	13.1	14.4	15.8	$83,551	$94,944	$107,287	$3,227	$3,508	$4,210
B	2	2	13.1	15.0	19.1	$87,194	$99,084	$111,965	$2,877	$3,508	$4,034
B	2	3	11.4	13.1	16.1	$76,345	$86,756	$98,034	$2,736	$3,508	$4,560
B	2	4	8.8	12.5	15.4	$72,702	$82,616	$93,356	$2,877	$3,508	$4,069
B	3	1	16.1	19.2	20.9	$83,551	$94,944	$107,287	$1,964	$2,338	$2,642
B	3	2	14.4	20.0	26.0	$87,194	$99,084	$111,965	$1,660	$2,338	$2,969
B	3	3	16.6	17.5	19.8	$76,345	$86,756	$98,034	$1,707	$2,338	$2,829
B	3	4	13.0	16.7	19.0	$72,702	$82,616	$93,356	$1,917	$2,338	$2,899
C	1	1	15.5	18.2	22.0	$45,333	$49,816	$64,263	$1,534	$2,160	$2,592
C	1	2	12.6	15.0	17.7	$37,487	$41,194	$53,140	$1,858	$2,160	$2,484
C	1	3	17.6	22.5	28.6	$56,230	$61,791	$79,710	$1,836	$2,160	$2,354
C	1	4	13.1	17.5	19.6	$43,589	$47,900	$61,791	$1,922	$2,160	$2,290
C	2	1	12.7	15.1	19.0	$45,333	$49,816	$64,263	$2,388	$2,809	$3,652
C	2	2	11.5	12.5	16.3	$37,487	$41,194	$53,140	$2,556	$2,809	$3,315
C	2	3	14.1	18.8	22.7	$56,230	$61,791	$79,710	$1,966	$2,809	$3,455
C	2	4	13.1	14.5	15.4	$43,589	$47,900	$61,791	$2,444	$2,809	$3,567
C	3	1	10.7	13.0	14.4	$45,333	$49,816	$64,263	$3,214	$3,456	$3,905
C	3	2	9.6	10.7	11.7	$37,487	$41,194	$53,140	$2,454	$3,456	$3,663
C	3	3	14.8	16.1	17.9	$56,230	$61,791	$79,710	$2,523	$3,456	$3,629
C	3	4	11.9	12.5	14.0	$43,589	$47,900	$61,791	$2,903	$3,456	$4,044
D	1	1	7.2	8.6	9.1	$11,603	$16,575	$19,061	$3,821	$4,246	$5,308
D	1	2	8.2	9.3	10.2	$12,558	$17,940	$20,631	$3,185	$4,246	$4,798
D	1	3	8.3	10.2	10.7	$13,787	$19,695	$22,649	$3,864	$4,246	$5,265
D	1	4	7.4	8.1	9.2	$10,920	$15,600	$17,940	$3,057	$4,246	$4,543
D	2	1	7.2	10.0	12.5	$11,603	$16,575	$19,061	$3,077	$3,497	$4,336
D	2	2	7.8	10.8	12.1	$12,558	$17,940	$20,631	$2,972	$3,497	$3,952
D	2	3	9.5	11.9	13.7	$13,787	$19,695	$22,649	$3,112	$3,497	$4,546
D	2	4	7.8	9.4	11.2	$10,920	$15,600	$17,940	$3,287	$3,497	$3,987
D	3	1	9.8	12.0	13.1	$11,603	$16,575	$19,061	$2,501	$2,748	$3,490
D	3	2	11.3	13.0	13.7	$12,558	$17,940	$20,631	$2,473	$2,748	$3,408
D	3	3	12.0	14.3	15.2	$13,787	$19,695	$22,649	$2,308	$2,748	$3,078
D	3	4	10.5	11.3	12.2	$10,920	$15,600	$17,940	$2,583	$2,748	$3,050
D	4	1	10.5	15.0	18.2	$11,603	$16,575	$19,061	$1,818	$1,998	$2,238
D	4	2	11.6	16.3	17.1	$12,558	$17,940	$20,631	$1,718	$1,998	$2,358
D	4	3	16.6	17.8	20.5	$13,787	$19,695	$22,649	$1,878	$1,998	$2,178
D	4	4	11.6	14.1	15.7	$10,920	$15,600	$17,940	$1,778	$1,998	$2,098
E	1	1	0.0	0.0	0.0	$0	$0	$0	$1,974	$2,407	$3,057
E	1	2	12.6	15.8	17.9	$18,994	$25,668	$29,262	$2,166	$2,407	$2,768
E	1	3	9.2	13.1	14.4	$15,691	$21,204	$24,173	$1,685	$2,407	$3,009
E	1	4	15.1	16.6	21.1	$19,958	$26,970	$30,746	$1,998	$2,407	$2,864
E	2	1	0.0	0.0	0.0	$0	$0	$0	$1,804	$2,027	$2,372
E	2	2	14.2	17.8	21.5	$18,994	$25,668	$29,262	$1,804	$2,027	$2,412
E	2	3	10.9	14.7	16.3	$15,691	$21,204	$24,173	$1,439	$2,027	$2,331
E	2	4	14.0	18.7	23.0	$19,958	$26,970	$30,746	$1,419	$2,027	$2,331

Table 5.7 Example 8: RP-ESDTCT^{EXP} supreme chromosome different optimization objectives under Uncertain environment.

Optimization objective	Total duration	Total cost	Total Material cost ($) (Fixed cost)	Total Lab. and equip. Cost ($) (Variable cost)	Total indirect cost	Total cost for idle crew time	Chromosome					Shift Time (Delayed start of activity at unit 1) (days)					Total Crew Idle Time (Interruption) (days)					Total Interruption (days)
						Activity ID >>	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E	
						Num of available modes >>	1	3	3	4	2											
a) Schedule minimization	109.7	$1,808,853	$709,744	$730,527	$313,857	$54,727	1	1	3	1	1	0	1	0	3	9	0	2	0	10	1	13
b) Cost minimization	129.1	$1,753,842	$735,179	$697,543	$302,653	$18,468	1	1	3	4	1	0	9	1	0	19	0	4	0	0	1	5
c) Joint cost - schedule - Interruption minimization	137.8	$1,831,851	$737,696	$701,643	$354,619	$37,891	1	3	1	1	1	0	0	0	26	9	0	0	3	7	1	11

Table 5.8 Example 8: Impact of different JCL %.

							Activity ID >>	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E	
							Num of available modes >>	1	3	3	4	2											
Optimization objective	JCL %	Total duration	Total cost	Total Material cost ($) (Fixed cost)	Total Lab. and equip. Cost ($) (Variable cost)	Total indirect cost	Total cost for idle crew time	Chromosome					Shift Time (Delayed start of activity at unit 1) (days)					Total Crew Idle Time (Interruption) (days)					Total Interruption (days)
a) Schedule minimization	JCL = 0.1	103.5	$1,707,293	$702,387	$695,159	$254,020	$55,728	1	1	3	1	1	0	3	0	2	9	0	4	0	8	1	13
	JCL = 0.3	105.9	$1,759,222	$724,760	$693,460	$287,345	$53,658	1	1	3	1	1	0	4	0	4	9	0	5	0	7	1	13
	JCL = 0.5	107.8	$1,768,941	$744,536	$694,281	$285,328	$44,796	1	1	3	1	1	0	4	1	3	10	0	3	0	7	0	10
	JCL = 0.7	109.7	$1,808,853	$709,744	$730,527	$313,857	$54,727	1	1	3	1	1	0	1	0	3	9	0	2	0	10	1	13
	JCL = 0.9	113.3	$1,838,778	$727,867	$753,632	$280,344	$76,934	1	1	3	1	1	0	6	2	1	10	0	5	2	10	1	18
b) Cost minimization	JCL = 0.1	143.9	$1,621,230	$676,118	$581,482	$359,627	$4,003	1	3	1	4	2	0	0	7	4	13	0	0	0	2	1	3
	JCL = 0.3	129.9	$1,657,469	$737,314	$626,723	$288,182	$5,250	1	2	3	4	2	0	0	7	0	15	0	0	2	0	1	3
	JCL = 0.5	122.1	$1,738,381	$727,420	$650,238	$331,198	$29,524	1	1	3	4	2	0	3	0	0	14	0	6	0	0	1	7
	JCL = 0.7	129.1	$1,753,842	$735,179	$697,543	$302,653	$18,468	1	1	3	4	1	0	9	1	0	19	0	4	0	0	1	5
	JCL = 0.9	142.6	$1,833,713	$714,636	$712,982	$369,731	$36,365	1	2	1	4	1	0	0	0	4	21	0	0	0	8	8	16
c) Joint cost - schedule - Interruption Minimization	JCL = 0.1	117.3	$1,731,494	$718,862	$638,832	$325,202	$48,598	1	1	1	1	1	0	4	0	12	10	0	10	0	6	1	17
	JCL = 0.3	120.2	$1,776,892	$712,519	$675,624	$325,544	$63,206	1	1	1	1	1	0	0	0	19	9	0	8	0	6	1	15
	JCL = 0.5	136.3	$1,743,744	$721,038	$640,198	$349,014	$33,493	1	3	1	1	1	0	0	0	21	9	0	0	0	9	1	10
	JCL = 0.7	137.8	$1,831,851	$737,696	$701,643	$354,619	$37,891	1	3	1	1	1	0	0	0	26	9	0	0	3	7	1	11
	JCL = 0.9	144.4	$1,828,154	$720,288	$696,568	$385,445	$25,854	1	3	1	2	1	0	0	2	23	11	0	0	2	6	8	16

5.7.3. Example 9: Testing including penalty costs

In order to study the sensitivity of the supreme solution to the project penalty costs, the same network configuration and data of example 8 is analyzed after setting the project penalty costs at different values and different schemes. Additional costs are accumulated at any simulation run when the penalty conditions are met. The problem is solved to find the solution for the objective of cost minimization at a targeted 70% joint confidence level of both time and cost. Scenarios 1 to 3 are set for a schedule-driven project where a penalty is set for exceeding a deadline of 120 days. Scenarios 4 to 6 are set for a budget-driven project where a percentile penalty is applied on the cost overrun in excess of $1.7 million. A combination of schedule and budget penalties schemes are also analyzed. The deadline and budget values are chosen in contrast to the cost and schedule values identified for the supreme chromosome solution in example 8 (where no penalties/bonus is considered). The model for the penalties at each scenario is capped to a maximum value, as is commonly exercised in the contract provisions for construction projects. In this example, the maximum schedule penalty is set to $200,000 (i.e. in the case of scenario 1, this value will be applicable if the expected schedule from a simulation run is late by more than 25 day). For each scenario, the supreme chromosome is identified, and the optimum cost and schedule results are shown in Table 5.9. As expected, different chromosomes are obtained in this analysis for the different scenarios. This phenomenon can be attributed to the fact that penalty values are only triggered when the conditions are satisfied and thus affects the expected cost values for some of the simulation runs. Depending on the cost and schedule uncertainty profiles, certain crew formations are found more fit to the objective function. This illustrates that modelling the penalty cost scheme is an important factor in the determination of the supreme chromosome solution.

Table 5.9 Example 9: Impact of different penalty schemes.

Scenario	Penalty Scheme	Activity ID >>		A	B	C	D	E	Total Interruption (days)
		Total duration	Total cost	Chromosome					
1	Schedule Penalty of $8,000/day for exceeding a deadline of 120days. Maximum schedule penalty is $200,000	122.4	$1,759,181	1	2	3	3	2	1
2	Schedule Penalty of $12,000/day for exceeding a deadline of 120days. Maximum schedule penalty is $200,000	121.7	$1,774,560	1	1	3	3	2	1
3	Schedule Penalty of $16,000/day for exceeding a deadline of 120days. Maximum schedule penalty is $200,000	122.7	$1,799,644	1	1	3	3	2	1
4	Budget Penalty of 20 % x Budget overrun in excess of $1,700,000. Maximum Budget penalty is $100,000	143.4	$1,792,348	1	2	1	4	2	8
5	Budget Penalty of 30 % x Budget overrun in excess of Maximum Budget penalty is $100,000	134.8	$1,781,130	1	2	2	4	2	2
6	Budget Penalty of 40 % x Budget overrun in excess of Maximum Budget penalty is $100,000	132.6	$1,789,035	1	2	3	4	2	4
7	Scenario 1 and 4 Combined	125.9	$1,786,859	1	2	3	3	2	4
8	Scenario 2 and 5 Combined	123.8	$1,772,111	1	1	3	3	2	1
9	Scenario 3 and 6 Combined	121.7	$1,787,718	1	1	3	3	2	1

CHAPTER 6 : CONCLUSIONS, CONTRIBUTIONS AND FUTURE WORK.

6.1. Conclusions

Project teams are facing an increasing challenge to deliver the project at the lowest time and with the lowest cost. These conflicting objectives require exploring several execution modes for the project activities, each of which has uncertainty around its own time and cost attributes, making the decisions about trade-offs for these conflicting objectives an essential issue. The complete enumeration in the solution space of the problem exponentially increases for medium and large size problems; hence, these trade-off problems are known as non-deterministic polynomial-time hard (NP-Hard) (De et al. 1995). In this study, we have developed a method using a combination of simulation and optimization techniques to solve discrete time–cost trade-off problem under uncertainty. The aim of the developed method is two-fold. On the one hand, it is to find the optimal chromosome representing the combination of activity modes while minimizing the time and/or cost of the project and concurrently maintain the joint confidence level of time and cost at the specified level. This is achieved by evolving generations of potential solutions where each generation is composed by partitioning the project network activities to solve for the optimal modes of primary activities, on the other hand, the aim is to identify the main effect of mode selection and provides insight into the relationship between cost and schedule and its influence on the overall time and cost of the project at the specified joint confidence level. The method was further extended to optimize for crew work continuity on repetitive class projects under uncertainty with consideration to scheduling typical and non-typical activities. The method addresses two main limitations of previous works related to repetitive construction projects: these are the shortage of optimization techniques accounting for uncertainty in both cost and schedule attributes and the lack of a comprehensive crew work interruption optimization methods. The method hence presents a comprehensive systematic approach circumventing the limitations of previous works.

Several examples taken from the literature are solved to illustrate the basic concept and test the performance and accuracy of the developed method; hence, several conclusions were drawn. The method was able to match the optimum results by others under a certain environment, which is considered a validation of the developed method. The method is capable of accounting for uncertainties associated with input variables using the Monte Carlo simulation technique, where for large test cases the optimal solution under a stochastic context was unknown so far. The outcomes prove that depending on the shapes, range and overlaps of the probability distribution functions of each mode and depending on the penalty/bonus schemes the method generates different optimal chromosome solutions for different joint confidence levels.

The method provides project managers with a tool to set project schedules and budgets that depict project environments more reasonable, accounting for uncertainty, thus reducing the potential construction cost and time overruns.

Only a small percentage of the solution space was searched to obtain the optimal solutions with reasonable execution time. This was achieved utilizing the power of the computational engine of Google's BigQuery with the free BigQuery quota limitations. Those limitations are expandable, where users can purchase an additional number of slots to use for query processing to improve the performance of the developed queries. The execution time is not linearly proportioned to the size of the project since big queries are run concurrently on the server. Since the run time is reasonably short, it is safe to say that the developed method can be applied for solving larger problems while maintaining the same accuracy.

6.2. Contributions

The aim of this thesis was to study the modelling of a discrete time-cost trade-off method under a stochastic context. We have developed an evolutionary technique for staged enumeration of the solution space.

The main contribution of this study is the development of a computerized method named ESDTCT that efficiently hybridizes the techniques of which (1) The CPM algorithm, (2) Monte Carlo simulation, (3) Joint cost and schedule risk analysis, and (4) Design of experiments for the enumeration of the solution space.

In parallel with the need to find the supreme chromosome solution, there is a need to apply managerial flexibility towards the selection of execution modes. This arises from the bias of the project manager to favour certain modes that may not be the optimal modes. The developed method provides a main effect plot that shows the relationship between the mode selection and the objective function; this provides the decision-maker a guideline for making informed decisions.

The method also brings a number of features representing improvements to existing methodologies.

- Utilizes Monte Carlo simulation to model uncertainties associated with an activity duration, and the activity fixed, variable and indirect costs.
- The joint cost and schedule confidence level (JCL) is used to identify the solution satisfying a targeted confidence level in the produced cost and schedule frontier curve solutions.
- identifying the supreme chromosome solution satisfying the tri-objective function for cost minimization, schedule minimization and crew work interruption minimization.
- Modelling traditional and repetitive class projects; accounting for non-typical activities in repetitive projects.
- Accounting for penalty/bonus schemes in the simulation model.
- Integrating the discrete risk events and their probability of occurrence to gain the benefits of concurrent assessments of uncertainties and risks on the trade-off process.
- Incorporate the risk or opportunity of incurring a penalty or bonus cost when exceeding or meeting defined milestone completion dates and/or exceeding or meeting defined budget values.

6.3. Limitations of the developed method

Several limitations exist in the developed method and summarized in the following sections:

The ESDTCT method is coded in BigQuery graphical web User-Interface (UI) in the Cloud Console using BigQuery resources and standard SQL queries. The developed query uses In-line JavaScript User-Defined Functions (UDF) to make calls for solving the Monte Carlo simulation at each experimental run. The UDF function has a limitation by Google where output data of the function must be 5MB or less in size. To put things in context, the query is designed to solve the batch of 1000 Monte Carlo simulation for each experiment. The output of the function is the expected start and finish dates of each activity and its relative expected costs both calculated accounting for the uncertainty PDF; therefore, by inheriting this limitation from Google BigQuery, the maximum number of activities that can exist in a project network is restricted. The larger the number of activities, the longer the execution run time; therefore, it is recommended to simplify the project network to a management summary, also called a Summary Master Schedule (SMS) which depicts the overall project broken down into its major components by area and is used for higher-level management reporting.

Although the developed method can handle a large network schedule, it has limitations to the problem complexity. This is mainly determined by the number of relationship logic between the network activities. The size of a Cartesian product matrix for the full factorial experiment enumeration for primary activities in a single generation is governed by the number of primary activities and the number of assigned modes to each activity. The maximum size that can be processed is restricted by the Google BigQuery limitation of 2000 maximum bound for concurrent slots per project for on-demand pricing. This limitation, however, is the current state for the free quota and can be scalable upon request from Google by adding instances of virtual machines to the Google compute engine. With the free quota limitations, we were able to process 8 primary tasks each having 5 modes at any given generation. This equates to approximately 400,000 experimental runs at each generation. This was achieved by breaking the Cartesian matrix into the Google maximum limit of 50 concurrent "Interactive" queries. More Google "Batch" queries can be deployed; however, will increase the execution run time. For more details on Google BigQuery limitations and details on Batch and Interactive queries, the reader is referenced to Google quotas and limits (Maharana et al. 2015).

Another limitation of the developed method is ignoring the higher-level interactions between the modes. In example, the selection of mode A in activity 1 is not correlated to the selection of mode B in activity 2. This assumption was necessary to minimize the matrix size of the full factorial design of experiments runs. Such interactions will be further studied in future works.

The discrete risk events considered in this study are associated to a project activity that may or may not occur, when occur, their impacts are uncertain. However, other types of risks that are not specific to an activity may also exist, such as severe weather conditions; those risks are not considered in the modeling of the developed method and are subject for inclusion in future work.

A loss of accuracy is found in the case of the developed complete random experiments module ($ESDTCT^{Rand}$) based on statistics of the sample size considered in the analysis. In an ideal situation, the entire population of the solution space should be studied but this is almost impossible on large and complex projects. This loss of accuracy is augmented by the developed Evolutionary experiment enumeration module ($ESDTCT^{Exp}$). Therefore, it is recommended to perform a final simulation run to validate the impact of selecting the various construction modes the objectives of total cost and time.

6.4. Directions for further research

Looking to the future, more emphasis is required on blurring the boundaries between the simulation model and the optimization techniques.

Future work will be to further enhance the applicability of the developed method by:

- Adding a knowledge-based component to narrow the search space.
- Modelling non-discrete risk events that are not associated with a specific work activity, such as severe weather conditions.
- Modelling probabilistic branches of the network to account for different network logic for different combination of construction modes.
- Modelling correlation between the selection of construction modes amongst the different activities.

REFERENCES

Aarts, E. H. L., Korst, J. H. M., & van Laarhoven, P. J. M. (1997). Simulated annealing. *Local search in combinatorial optimization*, 91.

Abba, W. (2000, April). How earned value got to primetime: A short look back and a glance ahead. In *Project management institute seminars and symposium in Houston, TX*.

AbouRizk, S. M., & Halpin, D. W. (1992). Statistical properties of construction duration data. *Journal of Construction Engineering and Management*, *118*(3), 525-544.

AbouRizk, S. M., & Sawhney, A. (1993). Subjective and interactive duration estimation. *Canadian journal of civil engineering*, *20*(3), 457-470.

Aghaie, A., & Mokhtari, H. (2009). Ant colony optimization algorithm for stochastic project crashing problem in PERT networks using MC simulation. *The International Journal of Advanced Manufacturing Technology*, 45(11-12), 1051.

Ang, A. HS. and Tang, WH (1975). Probability Concepts in Engineering Planning and Design, Volume I: Basic Principles.

Back, W. E., Boles, W. W., & Fry, G. T. (2000). Defining triangular probability distributions from historical cost data. *Journal of Construction Engineering and Management*, *126*(1), 29-37.

Bakry, I., Moselhi, O., & Zayed, T. (2016). Optimized scheduling and buffering of repetitive construction projects under uncertainty. *Engineering, Construction and Architectural Management*, *23*(6), 782-800.

Bailey, R. N., Garner, K. M., & Hobbs, M. F. (1997). Using simulated annealing and genetic algorithms to solve staff-scheduling problems. *Asia-Pacific journal of operational research*, *14*(2), 27.

Baker, B. N., Murphy, D. C., & Fisher, D. (2008). Factors affecting project success. *Project Management Handbook, Second Edition*, 902-919.

Barraza, G. A. (2010). Probabilistic estimation and allocation of project time contingency. *Journal of Construction Engineering and Management*, *137*(4), 259-265.

Bedford, T., and Cooke, R. M. (2001). *Probabilistic risk analysis: foundations and methods*, Cambridge University Press, Cambridge, UK; New York, NY.

Belassi, W., & Tukel, O. I. (1996). A new framework for determining critical success/failure factors in projects. *International journal of project management*, *14*(3), 141-151.

Bennett, J., & Ormerod, R. N. (1984). Simulation applied to construction projects. *Construction Management and Economics*, *2*(3), 225-263.

Bernstein, S. (1927). On the extension of the limit theorem of the computation of the probabilities to the sums of dependent quantities. *Mathematische Annalen* , *97* (1), 1-59.

Bettemir, Ö. H. (2009). "Optimization of time-cost-resource trade-off problems in project scheduling using meta-heuristic algorithms." Ph.D. thesis, Middle East Technical Univ., Ankara, Turkey.

Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR)*, *35*(3), 268-308.

Box, G. E. P. (1951). Wilson KB, The experimental attainment of optimal conditions. *J. Roy. Statist. Soc*, B13.

Brenner, L. A., Koehler, D. J., Liberman, V., & Tversky, A. (1996). Overconfidence in probability and frequency judgments: A critical examination. *Organizational Behavior and Human Decision Processes*, *65*(3), 212-219.

Carr, R. I. (1979). Simulation of construction project duration. *Journal of the Construction Division*, *105*(2), 117-128.

Cavazzuti, M. (2013). Design of experiments. In *Optimization Methods* (pp. 13-42). Springer Berlin Heidelberg.

Chanas, S., & Kamburowski, J. (1981). The use of fuzzy variables in PERT. *Fuzzy sets and systems*, *5*(1), 11-19.

Chau, K. W. (1995). Monte Carlo simulation of construction costs using subjective data. *Construction Management and Economics*, *13*(5), 369-383.

Chen, S. P. (2007). Analysis of critical paths in a project network with fuzzy activity times. *European Journal of Operational Research*, *183*(1), 442-459.

Chiu, Y. S. P., & Chiu, S. W. (2005). Incorporating expedited time and cost of the end product into the product structure diagram. *International Journal of Machine Tools and Manufacture*, *45*(7-8), 987-991.

Choi, H. H., & Mahadevan, S. (2008). Construction project risk assessment using existing database and project-specific information. *Journal of Construction*

Engineering and Management, *134*(11), 894-903.

Clark, W., Polakov, W. N., & Trabold, F. W. (1922). *The Gantt chart: A working tool of management*. Ronald Press Company.

Clemen, R. T., & Reilly, T. (2004). Making hard decisions with decision tools suite update edition. *Cengage Learning, Pacific Grove, CA*, *1*.

Cottrell, W. D. (1999). Simplified program evaluation and review technique (PERT). *Journal of construction Engineering and Management*, *125*(1), 16-22.

Coonce, T. 2009. Joint Confidence Level – Yes We Are. NASA, Headquarters Cost Analysis Division. 2009 NASA Cost Workshop, April 28th, 2009.

Curran, M. W. (1990). Range estimating reduces iatrogenic risk. *AACE International Transactions*, K-3.

de Freitas, F. G., Maia, C. L. B., de Campos, G. A. L., & de Souza, J. T. (2010). Optimization in software testing using metaheuristics. *Revista de Sistemas de Informação da FSMA*, *5*, 3-13.

de Wit, A. (1986, September). Measuring project success: An illusion. In *Proceedings of the 18th Annual Seminar/Symposium (Montreal/Canada)* (pp. 13-21).

De, P., Dunne, E. J., Ghosh, J. B., & Wells, C. E. (1995). The discrete time-cost tradeoff problem revisited. *European Journal of Operational Research*, *81*(2), 225-238.

Del Pico, W. J. (2013). *Project Control: Integrating Cost and Schedule in Construction*. John Wiley & Sons.

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.

Diaz, C. F., & Hadipriono, F. C. (1993). Nondeterministic networking methods. *Journal of construction engineering and management*, *119*(1), 40-57.

Douglas, C. M. (2019). *Design and Analysis of Experiments*. John Wiley & Sons.

Dorigo, M. (1992). Optimization, learning and natural algorithms. *PhD Thesis, Politecnico di Milano.*

Eckhardt, R. (1987). Stan ulam, john von neumann, and the Monte Carlo method. *Los Alamos Science*, *15*(131-136), 30.

Eid, M. S., Elbeltagi, E. E., & El-Adaway, I. H. (2018). Simultaneous multi-criteria optimization for scheduling linear infrastructure projects. *International Journal of Construction Management*, 1-15.

Elmaghraby, S. E. (1993). Resource allocation via dynamic programming in activity networks. *European Journal of Operational Research*, *64*(2), 199-215.

Elmaghraby, S. E. (1977). Probabilistic activity networks (PANs): a critical evaluation and extension of the PERT model. *Activity Networks: Project Planning and Control by Network Models*, 228-320.

El-Rayes, K. A. (1997). *Optimized scheduling for repetitive construction projects* (Doctoral dissertation, Concordia University).

El-Rayes, K., & Moselhi, O. (2001). Optimizing resource utilization for repetitive construction projects. *Journal of Construction Engineering and Management*, *127*(1), 18-27.

Engelbrecht, A. P. Computational Intelligence: An Introduction, (2007). *John Wily& Sons*.

Eshtehardian, E., Afshar, A., & Abbasnia, R. (2009). Fuzzy-based MOGA approach to stochastic time–cost trade-off problem. *Automation in Construction*, *18*(5), 692-701.

Eyers, K. (2001). *Belief network analysis of direct cost risk in building construction* (Doctoral dissertation, National Library of Canada= Bibliothèque nationale du Canada).

Feng, C. W., Chen, Y. J., & Huang, J. R. (2010). Using the MD CAD model to develop the time–cost integrated schedule for construction projects. *Automation in Construction*, *19*(3), 347-356.

Feng, C. W., Liu, L., & Burns, S. A. (1997). Using genetic algorithms to solve construction time-cost trade-off problems. *Journal of computing in civil engineering*, *11*(3), 184-189.

Fente, J., Schexnayder, C., & Knutson, K. (2000). Defining a probability distribution function for construction simulation. *Journal of construction engineering and management*, *126*(3), 234-241.

Finney, D. J. (1943). The fractional replication of factorial arrangements. *Annals of Human Genetics*, *12*(1), 291-301.

Fernandes, S., & Bernardino, J. (2015, July). What is bigquery?. In *Proceedings of the 19th International Database Engineering & Applications Symposium* (pp. 202-203). ACM.

Ferreira, J. (2014). *Google Apps Script: Web Application Development Essentials*. " O'Reilly Media, Inc.".

Fisher, R. (1926). Introduction to “The arrangement of field experiments”. *J Minist Agric G B*, *33*, 503-13.

Gholizadeh, S., & Barati, H. (2012). A COMPRATIVE STUDY OF THREE METAHEURISTICS FOR OPTIMUM DESIGN OF TRUSSES.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, *13*(5), 533-549.

Glover, F. (1977). Heuristics for integer programming using surrogate constraints. *Decision sciences*, *8*(1), 156-166.

Glover, F. W., & Kochenberger, G. A. (Eds.). (2006). *Handbook of metaheuristics* (Vol. 57). Springer Science & Business Media.

Goldberg, D. E. (1994). Genetic and evolutionary algorithms come of age. *Communications of the ACM*, *37*(3), 113-119.

Graves, R. (2001). Open and Closed: The Monte Carlo Model While some late-finishing and over-budget projects may be attributable to poor project management, a more likely reason is that the original budgets and schedules were unrealistic. *PM NETWORK*, *15*(12), 48-52.

Gutjahr, W. J., Strauss, C., & Wagner, E. (2000). A stochastic branch-and-bound approach to activity crashing in project management. *INFORMS Journal on Computing*, 12(2), 125-135.

Hadipriono, F. C., & Sun, K. (1990). Angular fuzzy set models for linguistic values. *Civil Engineering Systems*, *7*(3), 148-156.

Hall, N. G., & Posner, M. E. (2004). Sensitivity analysis for scheduling problems. *Journal of scheduling*, *7*(1), 49-83.

Handa, V. K., & Barcia, R. M. (1986). Linear scheduling using optimal control theory. *Journal of construction engineering and management*, *112*(3), 387-393.

Hartford, D. N., & Baecher, G. B. (2004). *Risk and uncertainty in dam safety*. Thomas Telford.

Hauser, J. R. (2014). Consideration-set heuristics. *Journal of Business Research*, *67*(8), 1688-1699.

Hertz, D. B. & Thomas, H. (1984). Practical Risk Analysis: and Approach Through Case Histories. John Wiley and Sons. Chichester, UK: taken from Edwards, P and Bowen, P (1999)

Hegazy, T. (1999). Optimization of construction time-cost trade-off analysis using genetic algorithms. *Canadian Journal of Civil Engineering*, *26*(6), 685-697.

Henderson, K. (2003). Earned schedule: A breakthrough extension to earned value theory? A retrospective analysis of real project data. *The Measurable News*, *1*(2), 13-23.

Henderson, K. (2004). Further developments in earned schedule. *The measurable news*, *1*(1), 15-22.

Hendrickson, C., & Au, T. (2000). Project Management for Construction: Fundamental Concepts for Owners. *Engineers, Architects and Builders, Prentice Hall, Pitsburgh*.

Hoffpauir, D. (2015). A Look into NASA's Joint Cost and Schedule Confidence.

Holland, J. (1975). "Adaptation in Natural and Artificial Systems". The University of Michigan Press, *Ann Arbor.*

Hooke, R., & Jeeves, T. A. (1961). ``Direct Search''Solution of Numerical and Statistical Problems. *Journal of the ACM (JACM)*, *8*(2), 212-229.

Hulett, D. T. (2002). Project cost risk analysis. *Los Angeles, CA: Hullet & Associates*.

Hulett, D. T. (2015). What Should We Do with Unknowns in Schedule Risk Analysis?. *PM world Journal*.

Hulett, D. T., Caddell, C. P., Clarke, T., Cretu, O., Curran, P. K. M., Curran, M. W., ... & Hollmann, J. K. (2011). Integrated cost and schedule risk analysis using Monte Carlo simulation of a CPM model. *Morgantown, WV: AACE International Recommended Practice*, (57R-09).

Hyari, K., & El-Rayes, K. (2004). A multi-objective model for optimizing construction planning of repetitive infrastructure projects. In *Proceedings of the Tenth International Conference on Computing in Civil and Building Engineering, ICCCBE* (pp. 2-4).

Hyari, K. H., El-Rayes, K., & El-Mashaleh, M. (2009). Automated trade-off between time and cost in planning repetitive construction projects. *Construction Management and Economics*, *27*(8), 749-761.

Ioannou, P. G., & Srisuwanrat, C. (2006). Sequence step algorithm for continuous resource utilization in probabilistic repetitive projects. In *Proceedings of the 2006 Winter Simulation Conference* (pp. 1731-1740). IEEE.

Isidore, L. J., & Back, W. E. (2001). Probabilistic optimal-cost scheduling. *Journal of Construction Engineering and Management*, *127*(6), 431-437.

Isidore, L. J., & Back, W. E. (2002). Multiple simulation analysis for probabilistic cost and schedule integration. *Journal of Construction Engineering and Management*, *128*(3), 211-219.

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

Jordan Novet (May 6, 2016). "Google BigQuery now lets you analyze data from Google Sheets".

Jugdev, K., & Müller, R. (2005). A retrospective look at our evolving understanding of project success. *Project management journal*, *36*(4), 19-31.

Kalhor, E., Khanzadi, M., Eshtehardian, E., & Afshar, A. (2011). Stochastic time–cost optimization using non-dominated archiving ant colony approach. *Automation in Construction*, *20*(8), 1193-1203.

Kavanagh, D. P. (1985). SIREN: A repetitive construction simulation model. *Journal of construction engineering and management*, *111*(3), 308-323.

Ke, H. (2014). A genetic algorithm-based optimizing approach for project time-cost trade-off with uncertain measure. *Journal of Uncertainty Analysis and Applications*, 2(1), 8.

Kelley Jr, J. E. (1961). Critical-path planning and scheduling: Mathematical basis. *Operations research*, *9*(3), 296-320.

Kim, B. C., & Reinschmidt, K. F. (2010). Probabilistic forecasting of project duration using Kalman filter and the earned value method. *Journal of Construction Engineering and Management*, *136*(8), 834-843.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, *220*(4598), 671-680.

Knoke, J. R., & de la Garza, J. (2003). Practical cost/schedule modeling for CIP management. *AACE International Transactions*, PM61.

Kroese, D. P., Taimre, T., & Botev, Z. I. (2013). *Handbook of Monte Carlo methods* (Vol. 706). John Wiley & Sons.

Lee, D. E. (2005). Probability of project completion using stochastic project scheduling simulation. *Journal of construction engineering and management*, *131*(3), 310-318.

Lipke, W. (1999). Applying management reserve to software project management. *Journal of Defense Software Engineering*, *3*, 17-21.

Lipke, W. (2003). Schedule is different. *The Measurable News*, *31*(4), 31-34.

Littlefield, T. K., & Randolph, P. H. (1991). PERT duration times: mathematics or MBO. *Interfaces*, *21*(6), 92-95.

Liu, J. (1999). The impact of neighbourhood size on the process of simulated annealing: computational experiments on the flowshop scheduling problem. *Computers & Industrial Engineering*, *37*(1-2), 285-288.

Liu, J., & Rahbar, F. (2004). Project time-cost trade-off optimization by maximal flow theory. *Journal of construction engineering and management*, *130*(4), 607-609.

Liu, S. S., & Wang, C. J. (2007). Optimization model for resource assignment problems of linear construction projects. *Automation in Construction*, *16*(4), 460-473.

Long, L. D., & Ohsato, A. (2009). A genetic algorithm-based method for scheduling repetitive construction projects. *Automation in Construction*, *18*(4), 499-511.

Lopez-Garcia, L., & Posada-Bolivar, A. (1999). A simulator that uses Tabu search to approach the optimal solution to stochastic inventory models. *Computers & Industrial Engineering*, *37*(1-2), 215-218.

Lorterapong, P., & Moselhi, O. (1996). Project-network analysis using fuzzy sets theory. *Journal of Construction Engineering and Management*, *122*(4), 308-318.

Lu, M., & AbouRizk, S. M. (2000). Simplified CPM/PERT simulation model. *Journal of Construction Engineering and Management*, *126*(3), 219-226.

Maharana, S. B., Iswarya, R., Kumar, V. S., & Vamsi, M. R. (2015). Google App Engine. *International Journal of Engineering and Management Research (IJEMR)*, *5*(6), 536-538.

Manda, K., Satapathy, S. C., & Poornasatyanarayana, B. (2012). Population based meta-heuristic techniques for solving optimization problems: A selective survey. *International Journal of Emerging Technology and Advanced Engineering*, *2*(11), 206-211.

Maravas, A., & Pantouvakis, J. P. (2010). Fuzzy repetitive scheduling method for projects with repeating activities. *Journal of Construction Engineering and Management*, *137*(7), 561-564.

McCahon, C. S., & Lee, E. S. (1988). Project network analysis with fuzzy activity times. *Computers & Mathematics with applications*, *15*(10), 829-838.

Meredith, J. & Mantel, S. (2009), *Project Management: A Managerial Approach*, John Wiley & Sons, Inc. New York.

Michalewicz, Z. (1994). Evolutionary computation techniques for nonlinear programming problems. *International Transactions in Operational Research*, *1*(2), 223-240.

Miles, F. M., & Wilson, T. G. (1998). Managing project risk and the performance envelope. In *Applied Power Electronics Conference and Exposition, 1998. APEC'98. Conference Proceedings 1998., Thirteenth Annual* (Vol. 1, pp. 247-253). IEEE.

Moeller, G. L., & Digman, L. A. (1981). Operations planning with VERT. *Operations Research*, *29*(4), 676-697.

Morua Padilla, E. (1986). *Resource strategies for dynamic construction project management* (Doctoral dissertation, Ph. D. Dissertation, Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI).

Moselhi, O. (1993). Schedule compression using the direct stiffness method. *Canadian Journal of Civil Engineering*, *20*(1), 65-72.

Moselhi, O., & Dimitrov, B. (1993). Discussion of “Monte Carlo Technique with Correlated Random Variables” by Ali Touran and Edward P. Wiser (June, 1992, Vol. 118, No. 2). *Journal of construction engineering and management*, *119*(3), 658-660.

Moselhi, O. (2011). The use of earned value in forecasting project duration. In *28th International Symposium on Automation and Robotics in Construction, ISARCInternational Symposium on Automation and Robotics in Construction (ISARC)*.

Moselhi, O., & El-Rayes, K. (1993). Scheduling of repetitive projects with cost optimization. *Journal of Construction Engineering and Management*, *119*(4), 681-697.

Moselhi, O., & Roofigari-Esfahan, N. (2013). Project schedule compression: a multi-objective methodology. *Construction Innovation*, *13*(4), 374-393.

Moselhi, O., & Alshibani, A. (2013). Schedule compression using fuzzy set theory and contractors judgment. *Journal of Information Technology in Construction (ITcon)*, *18*(4), 64-75.

Mullins, J. W., Forlani, D., & Walker Jr, O. C. (1999). Effects of organizational and decision-maker factors on new product risk taking. *Journal of Product Innovation Management: AN INTERNATIONAL PUBLICATION OF THE PRODUCT DEVELOPMENT & MANAGEMENT ASSOCIATION*, *16*(3), 282-294.

Naeni, L. M., Shadrokh, S., & Salehipour, A. (2011). A fuzzy approach for the earned value management. *International Journal of Project Management*, *29*(6), 764-772.

NASA. 2015. Cost estimating handbook. Version 4.0. NASA. Washington, D.C.

Nassar, K. (2005). Evolutionary optimization of resource allocation in repetitive construction schedules. *Journal of Information Technology in Construction (ITcon)*, *10*(18), 265-273.

Nasir, D., McCabe, B., & Hartono, L. (2003). Evaluating risk in construction–schedule model (ERIC–S): construction schedule risk model. *Journal of construction engineering and management*, *129*(5), 518-527.

Newton, S. (1991). 'An Agenda For Cost Modelling Research', *Construction Management and Economics*, 9, pp. 97-112.

Neil, J. N., & Diekmann, J. (1989). Management of project risks and uncertainties. *Construction Industry Institute, Univ. of Texas at Austin, Austin, TX*.

Oberlender, G. D. (2000). *Project management for engineering and construction*, 2nd Ed., McGraw-Hill, Boston, MA.

Patterson, J. H., & Huber, W. D. (1974). A horizon-varying, zero-one approach to project scheduling. *Management Science*, *20*(6), 990-998.

Pendharkar, P. C. (2015). Cost minimizing target setting heuristics for making inefficient decision-making units efficient. *International Journal of Production Economics*, *162*, 1-12.

Perry, J.G. and Hayes, R.W. (1985), 'Risk and Its Management in Construction Projects', *Proceedings of the Institution of Civil Engineers*, Part 1 (78), June, pp 499-521.

Poh, Y. P., & Tah, J. H. M. (2006). Integrated duration–cost influence network for modelling risk impacts on construction tasks. *Construction Management and Economics*, *24*(8), 861-868.

Pouliquen, L. Y. (1970). *Risk analysis in project appraisal* (No. E10 P874). Banco Mundial, Washington, DC (EUA).

Prade, H. (1979). Using fuzzy set theory in a scheduling problem: a case study. *Fuzzy sets and systems*, *2*(2), 153-165.

Pritsker, A. A. B. (1966). GERT: Graphical Evaluation and Review Technique, Part II. Probabilistic and Industrial Engineering. *J. Ind. Eng.*, (6), 293-301.

Project Management Institute. (2013). A guide to the project management body of knowledge: PMBOK guide, Fifth Edition. ISBN 978-1-935589-67-9, 2013, *Project Management Institute Inc*., Newtown Square, PA.

Raftery, J. (1994). *Risk analysis in project management*, E & FN Spon, London; NewYork.

Riley, L. A. (2013, July). Discrete-event simulation optimization: a review of past approaches and propositions for future direction. In *Proceedings of the 2013 Summer Computer Simulation Conference* (p. 47). Society for Modeling & Simulation International.

Russell, A. D., & Caselton, W. F. (1988). Extensions to linear scheduling optimization. *Journal of Construction Engineering and Management*, *114*(1), 36-52.

Sakellaropoulos, S., & Chassiakos, A. P. (2004). Project time–cost analysis under generalised precedence relations. *Advances in Engineering Software*, *35*(10-11), 715-724.

Salama, T., & Moselhi, O. (2019). Multi-objective optimization for repetitive scheduling under uncertainty. *Engineering, Construction and Architectural Management*.

Sasieni, M. W. (1986). Note—A note on PERT times. *Management Science*, *32*(12), 1652-1653.

Schexnayder, C., Knutson, K., & Fente, J. (2005). Describing a beta probability distribution function for construction simulation. *Journal of construction engineering and management*, *131*(2), 221-229.

Schoderbek, P. P., & Digman, L. A. (1967). 3RD GENERATION, PERT-LOB. *Harvard Business Review*, *45*(5), 100-110.

Senouci, A., & Al-Derham, H. R. (2008). Genetic algorithm-based multi-objective model for scheduling of linear construction projects. *Advances in Engineering Software*, *39*(12), 1023-1028.

Short, J. W. (1993). Using schedule variance as the only measure of schedule performance. *Cost Engineering*, *35*(10), 35.

Shrnhur, A. J., Levy, O., & Dvir, D. (1997). Mapping the dimensions of project success. *Project management journal*, *28*(2), 5-13.

Selinger, S. (1980). Construction planning for linear projects. *Journal of the Construction Division*, *106*(ASCE 15504).

Srisuwanrat, C., & Ioannou, P. G. (2007). Optimal scheduling of probabilistic repetitive projects using completed unit and genetic algorithms. In *2007 Winter Simulation Conference* (pp. 2151-2158). IEEE.

Taguchi, G. (1986). *Introduction to quality engineering: designing quality into products and processes* (No. 658.562 T3).

Talbi, E. G. (2009). *Metaheuristics: from design to implementation* (Vol. 74). John Wiley & Sons.

Tyler, S. M. (1994). No (easy) way out: Liquidating stipulated damages for contractor delay in public construction contracts. Duke LJ, 44, 357.

Construction Industry Institute (CII). (1989). Management of project risks and uncertainties (Publication 6-8). Austin, TX: Construction Industry Institute

Touran, A. (1997). Probabilistic model for tunneling project using Markov chain. *Journal of construction engineering and management*, *123*(4), 444-449.

Touran, A., & Suphot, L. (1997). Rank correlations in simulating construction costs. *Journal of construction engineering and management*, *123*(3), 297-301.

Trietsch, D., Mazmanyan, L., Gevorgyan, L., & Baker, K. R. (2012). Modeling activity times by the Parkinson distribution with a lognormal core: Theory and validation. *European Journal of Operational Research*, *216*(2), 386-396.

Turner, J. R., & Müller, R. (2005, June). The project manager's leadership style as a success factor on projects: A literature review. Project Management Institute.

Turner, R., & Zolin, R. (2012). Forecasting success on large projects: developing reliable scales to predict multiple perspectives by multiple stakeholders over multiple time frames. *Project Management Journal*, *43*(5), 87-99.

Van Slyke, R. M. (1963). Letter to the editor— Monte Carlo methods and the pert problem. *Operations Research*, *11*(5), 839-860.

Vandevoorde, S., & Vanhoucke, M. (2006). A comparison of different project duration forecasting methods using earned value metrics. *International journal of project management*, *24*(4), 289-302.

Vanhoucke, M., & Debels, D. (2005). The discrete time/cost trade-off problem under various assumptions exact and heuristic procedures. *Universiteit Gent, FEeB*.

Vose, D. (2008). *Risk analysis: a quantitative guide*. John Wiley & Sons.

Wall, D. M. (1997). Distributions and correlations in Monte Carlo simulation. *Construction Management & Economics*, *15*(3), 241-258.

Williams, T. (1995). A classified bibliography of recent research relating to project risk management. *European journal of operational research*, *85*(1), 18-38.

Williams, T. (2004). Why Monte-Carlo simulations of project networks can mislead. *Project Management Journal*, *35*(3), 53-61.

Wilson, J. R., Vaughan, D. K., Naylor, E., & Voss, R. G. (1982). Analysis of Space Shuttle ground operations. *Simulation*, *38*(6), 187-203.

Winter, R. M. (2003). Computing the near-longest path. *AACE International Transactions*, PS111.

Woolf, M. B. (2012). CPM Mechanics: The Critical Path Method of Modeling Project Execution Strategy.

Wu, R. W. K., & Hadipriono, F. C. (1994). Fuzzy modus ponens deduction technique for construction scheduling. *Journal of Construction Engineering and Management*, *120*(1), 162-179.

Xu, Z., Yu, J., & Li, H. (2014). Analyzing integrated cost-schedule risk for complex product systems R&D projects. *Journal of Applied Mathematics*, 2014.

Yakhchali, S. H., & Ghodsypour, S. H. (2010). Computing latest starting times of activities in interval-valued networks with minimal time lags. *European Journal of Operational Research*, *200*(3), 874-880.

Yang, I. T. (2005). Simulation-based estimation for correlated cost elements. *International Journal of Project Management*, *23*(4), 275-282.

Yang, X. S. (2011). Bat algorithm for multi-objective optimisation. *International Journal of Bio-Inspired Computation*, *3*(5), 267-274.

Yang, I. T., Lin, Y. C., & Lee, H. Y. (2013). Use of support vector regression to improve computational efficiency of stochastic time-cost trade-off. *Journal of Construction Engineering and Management*, 140(1), 04013036.

Yeo, K. T. (1990). Risks, classification of estimates, and contingency management. *Journal of Management in Engineering*, *6*(4), 458-470.

Zadeh, L. A. (1965). Information and control. *Fuzzy sets*, *8*(3), 338-353.

Zahraie, B., & Tavakolan, M. (2009). Stochastic time-cost-resource utilization optimization using nondominated sorting genetic algorithm and discrete fuzzy sets. *Journal of Construction Engineering and Management*, 135(11), 1162-1171.

Zheng, D. X., Ng, S. T., & Kumaraswamy, M. M. (2005). Applying Pareto ranking and niche formation to genetic algorithm-based multiobjective time–cost optimization. *Journal of Construction Engineering and Management*, *131*(1), 81-91.

Zolfaghari, S., & Liang, M. (1998). Machine cell/part family formation considering processing times and machine capacities: a simulated annealing approach. *Computers & industrial engineering*, *34*(4), 813-823.

Zou, X., Zhang, L., & Zhang, Q. (2018). A Biobjective Optimization Model for Deadline Satisfaction in Line-of-Balance Scheduling with Work Interruptions Consideration. *Mathematical Problems in Engineering*, *2018*.

APPENDIX A: ESDTCT method script

a) ESDTCT^{Exp}

```
#standardSQL


--  ESDTCT_EXP Model

--######################################################################
-- This code is developed using BigQuery SQL statements with Temporary
-- functions having an in-line Java script programing. Readers of this
-- code need advanced programing skills.
-- The input to this script is two BigQuery tables namely (Tasks_bq
-- , Modes_bq and Experiments). The input tables are created in Google
-- sheets and uploaded to BigQuery using the Google BigQuery API in
-- Apps Script.
-- (Note: This is an advanced service that must be enabled before use.
-- Reference:
-- https://developers.google.com/apps-script/guides/services/advanced)
-- Tasks_bq table contains data input describing the network
-- activities and logic relationship for a typical unit in a repetative
-- project.
-- Modes_bq table contains data input describing the admitted modes to
-- each activity and probabilistic estimates parameters for the cost
-- and duration data.
-- Experiments table contains the cartisian product for the enummeration
-- of primary activities
-- A Number of code parameters, where noted in the script, are modified
-- using the Apps Script before executing the API command.
--######################################################################
-- The Simulation_and_JCL_Calculation_Function performs a Monte Carlo
-- simulation and identify the frontier curve solution satisfying the
-- user defined JCL.
--######################################################################
  CREATE TEMPORARY FUNCTION Simulation_and_JCL_Calculation_Function (arr ARRAY<STRING>)
  RETURNS ARRAY<STRUCT<TRADE_OFF_RUN INT64, SIMULATION_RUN  INT64,
  S_Activity_ID INT64,
  P_Activity_ID INT64, U_Activity_ID INT64,  START FLOAT64 ,
  FINISH FLOAT64, TC FLOAT64 , JCL FLOAT64,
  CHROMOSOME String ,
  Mode_Option INT64,
  Activity_Type string, P1 INT64 , P2 INT64, P3 INT64, P4 INT64,
  P5 INT64, P6 INT64, Ladder_Seq INT64,  RDUR FLOAT64, RFC FLOAT64,
  RVC FLOAT64 , ROHC FLOAT64, FC FLOAT64, VC FLOAT64, OHC FLOAT64,
  PC FLOAT64,  BC FLOAT64,
  Lable STRING, Num_Of_Modes INT64
  >>
  LANGUAGE js AS """
  var result = [];
  for (var i = 0; i < arr.length; i++){
  result.push(JSON.parse(arr[i])) }
  for (var i = 0; i < arr.length; i++){
  if (result[i].Probability_of_Occurrence < Math.random()){
  result[i].RDUR ==  0
```

```
result[i].RFC ==   0
result[i].RVC ==  0
}}

result.sort(function (a,b) {
  if (a.SIMULATION_RUN > b.SIMULATION_RUN) return  1;
  if (a.SIMULATION_RUN < b.SIMULATION_RUN) return -1;
  if (a.Ladder_Seq > b.Ladder_Seq) return  1;
  if (a.Ladder_Seq < b.Ladder_Seq) return -1;
  return 0;
});
var arry = []
for (var i = 0; i < result.length; i++) {
arry.push (result[i].S_Activity_ID)
}
for (var i = 0; i < result.length; i++) {

var x1 = arry.indexOf(result[i].P1);
var x2 = arry.indexOf(result[i].P2);
var x3 = arry.indexOf(result[i].P3);
var x4 = arry.indexOf(result[i].P4);
var x5 = arry.indexOf(result[i].P5);
var x6 = arry.indexOf(result[i].P6);

 if (x1 == -1) {var FX1 = 0} else {var FX1 = result[x1].FINISH}
 if (x2 == -1) {var FX2 = 0} else {var FX2 = result[x2].FINISH}
 if (x3 == -1) {var FX3 = 0} else {var FX3 = result[x3].FINISH}
 if (x4 == -1) {var FX4 = 0} else {var FX4 = result[x4].FINISH}
 if (x5 == -1) {var FX5 = 0} else {var FX5 = result[x5].FINISH}
 if (x6 == -1) {var FX6 = 0} else {var FX6 = result[x6].FINISH}

result[i].START = Math.max( FX1, FX2,FX3,FX4,FX5,FX6)
+ 0  /*continuum_time_calculations*/;
result[i].FINISH = result[i].START + result[i].RDUR
- 0  /*continuum_time_calculations*/;
}
result.sort(function (a,b) {
  if (a.S_Activity_ID > b.S_Activity_ID) return  1;
  if (a.S_Activity_ID < b.S_Activity_ID) return -1;
  return 0;
});
var Num_of_Project_Tasks = 5/* No of Activities in a Typical unit */
  var arry2 = []
for (var i = 0; i < result.length; i++) {
arry2.push (result[i].S_Activity_ID)
}
var TTC = 0; var TFC = 0 ; var TVC = 0;
var TOHC = 0 ;
   for (var i = 0; i < result.length; i++) {

   TFC = TFC + (result[i].RFC)
   if ( result[i].P_Activity_ID == Num_of_Project_Tasks){
   result[i].FC = TFC , TFC = 0}
   TVC = TVC + (result[i].RVC * result[i].RDUR)
   if ( result[i].P_Activity_ID == Num_of_Project_Tasks){
   result[i].VC = TVC , TVC = 0}
   if ( result[i].P_Activity_ID == Num_of_Project_Tasks){
   result[i].OHC = result[i].ROHC * result[i].FINISH}

   TTC +=
   (result[i].RFC + (result[i].RVC * result[i].RDUR))
```

```
     if ( result[i].P_Activity_ID == Num_of_Project_Tasks){
     result[i].TC = TTC + result[i].ROHC * result[i].FINISH , TTC = 0

   // Adding Penalty / Bonus Cost

   // Linear continous schemes for Penalty / Bonus Cost

   // Schedule deadline driven projects - Penalty Cost scheme:
    if ((result[i].FINISH - Deadline_) > 0){
     result[i].PC =
     Math.min((Schedule_Penalty_Cost * (result[i].FINISH - Deadline_)),
     Max_Schedule_Penalty__Cost )
     }
   // Schedule deadline driven projects - Bonus Cost scheme:
    if ((result[i].FINISH - Deadline_) < 0){
     result[i].BC =
     Math.max((Schedule_Bonus_Cost * (result[i].FINISH - Deadline_)),
     - Max_Schedule_Bonus__Cost)
     }

   // Budget driven projects - Penalty Cost scheme:

    if ((result[i].TC - Budget__) > 0){
     result[i].PC +=
     Math.min((Budget_Penalty_Cost * (result[i].TC - Budget__)),
     Max_Budget_Penalty__Cost )
     }
   // Budget driven projects - Bonus Cost scheme:
    if ((result[i].TC - Budget__) < 0){
    result[i].BC +=
     Math.max((Budget_Bonus_Cost * (result[i].TC - Budget__)),
     - Max_Budget_Bonus__Cost)
     }
     result[i].TC += (result[i].PC + result[i].BC)
     }
     }
     var Final_result = result.filter(function (dataRow) {
     return dataRow.TC > 0;
     });

     for (var i = 0; i < Final_result.length; i++){
     var count = 0
     for (var j = 0; j < Final_result.length; j++){
     if (Final_result[i].TC >= Final_result[j].TC  &&
     Final_result[i].FINISH >= Final_result[j].FINISH ){count +=1 }
     }
     Final_result[i].JCL = count / (Final_result.length )
     }

       var Final_result1 = Final_result.filter(function (dataRow) {
     return dataRow.JCL > 0.5 && dataRow.JCL < 0.6/* Lower and upper JCL:
     Change Numbers using App Script */;
     });
     return Final_result1
     """;

   --######################################################################
   -- The Normalization_of_Frontier_Solutions_Function performs calculations
   -- to nomalize the cost and schedule pairs of the frontier solutions to
   -- the range [0,1]
   --######################################################################
```

```
    CREATE TEMPORARY FUNCTION Normalization_of_Frontier_Solutions_Function (arr ARRAY<STRING>)
    RETURNS ARRAY<STRUCT<TRADE_OFF_RUN INT64, SIMULATION_RUN INT64,
    S_Activity_ID INT64, P_Activity_ID INT64,
    U_Activity_ID INT64, START FLOAT64 ,  FINISH FLOAT64, TC FLOAT64 ,
    JCL FLOAT64,
    CHROMOSOME String ,
    Mode_Option INT64,
    Activity_Type string, P1 INT64 , P2 INT64, P3 INT64, P4 INT64,
    P5 INT64, P6 INT64, Ladder_Seq INT64, RDUR FLOAT64, RFC FLOAT64,
    RVC FLOAT64 , ROHC FLOAT64, FC FLOAT64, VC FLOAT64,
     OHC FLOAT64, PC FLOAT64,  BC FLOAT64,
     Lable STRING, Num_Of_Modes INT64 ,
     Vector FLOAT64
     >>
    LANGUAGE js AS """
    var Final_result = [];
    for (var i = 0; i < arr.length; i++){
    Final_result.push(JSON.parse(arr[i])) }
    var maxTC = Math.max.apply(Math, Final_result.map(function(v) {
    return v.TC;}));
  //  var minTC = Math.min.apply(Math, Final_result.map(function(v){
  //  return v.TC;}));
     var minTC = 0
     var maxFinish = Math.max.apply(Math, Final_result.map(function(v)
     { return v.FINISH;}));
  // var minFinish = Math.min.apply(Math, Final_result.map(function(v){
  // return v.FINISH;}));
     var minFinish = 0
    for (var i = 0; i < Final_result.length; i++){
    Final_result[i].Vector =
    Math.sqrt(Math.pow(((Final_result[i].FINISH - minFinish )/
    (maxFinish - minFinish)), 2)
    + Math.pow((Final_result[i].TC - minTC)/(maxTC - minTC),2) )
        }
    return Final_result
    """;


     CREATE TEMP FUNCTION TriDist_Sampling(P FLOAT64 ,
     a FLOAT64, m FLOAT64, b FLOAT64)
    RETURNS FLOAT64
    LANGUAGE js AS """
    var d ;   var x
    d = b - a
    if (d != 0){x = (m - a) / d} else {x = b;}
    if (P <= x){return (a + ((Math.sqrt(P * x)) * d))} else {
    return  (b - ((Math.sqrt((1 - P) * (1 - x))) * d));}
    """;

  -- END OF TEMPORARY FUNCTIONS
  --#####################################################################
  -- The below series of BigQuery SQL Sub Queries are set to perform the
  -- data input set up for the described Temporary functions above.
  --#####################################################################
    WITH

  --#####################################################################
  -- The sub query Simulation below prepares a replication of the project
  -- to a user defined number of simulation scenarios.
  --#####################################################################
```

```
    Simulation as (
    Select
      SIMULATION_RUN  , rec.*
     , RAND() as P_Dur
     , RAND() as P_FC
     , RAND() as P_VC
     , RAND() as P_OHC
     from
      ESDTCT_Database.Tasks_bq rec
       Cross JOIN
       UNEST((GENERATE_ARRAY(1,1000) )) AS SIMULATION_RUN
  --      Change SIMULATION_RUN Number using App Script
  --    order by SIMULATION_RUN ,Activity_ID
       )
    ,
  --####################################################################
  -- The sub query Tasks below prepares the initial network details for
  -- a typical unit and creates the global network for the user defined
  -- number of repetitive units.
  --####################################################################
      Tasks as (
      Select
      SIMULATION_RUN, Activity_Type,
      Probability_of_Occurrence, Ladder_Seq,
      Lable, Num_Of_Modes,
      Activity_ID as U_Activity_ID  ,
      Activity_ID    as P_Activity_ID
      , (Activity_ID  + (SIMULATION_RUN-1) *
      ( Num))   as S_Activity_ID
      , P1 + (SIMULATION_RUN - 1) *
      ( Num)   as P1
      , P2  + (SIMULATION_RUN - 1) *
      ( Num)   as P2
      , P3  + (SIMULATION_RUN - 1) *
      ( Num)   as P3
      , P4  + (SIMULATION_RUN - 1) *
      ( Num)   as P4
      , P5  + (SIMULATION_RUN - 1) *
      ( Num)   as P5
      , P6  + (SIMULATION_RUN - 1) *
      ( Num)   as P6
       , P_Dur , P_FC , P_VC , P_OHC
       from Simulation rec

               Cross JOIN
   UNNEST((GENERATE_ARRAY(5/* No of Activities in a Typical unit */,
        5/* No of Activities in a Typical unit */ ) ))
        AS Num /*: Change Number using App Script */

    --   order by S_Activity_ID
         )
  ,
  --####################################################################
  -- The sub query Modes below reads the user parameter inputs for each
  -- activity admitted modes
  --####################################################################
    Modes AS (
    SELECT * from `ESDTCT_Database.Modes_bq`
    )
    ,
  --####################################################################
```

```
  -- The sub query Experiment_run below prepares the activity modes input
  -- to activities.
  --######################################################################
    Experiment_run AS (
    SELECT TRADE_OFF_RUN ,
    STRING_AGG(Cast( CHROMOSOME as string)) CHROMOSOME
    FROM `ESDTCT_Database.Experiments` rec
    group by TRADE_OFF_RUN
    )
    ,
  --######################################################################
  -- The sub query SETUP below joins the data input tables and assign a
  -- random cost and duration values based on the trinagular probability
  -- distribution function.
  --######################################################################
    SETUP as (
    SELECT TRADE_OFF_RUN, SIMULATION_RUN, S_Activity_ID ,
    U_Activity_ID, P_Activity_ID,
    Probability_of_Occurrence , Mode_Option,
    Activity_Type, P1, P2, P3, P4, P5, P6, Ladder_Seq, 0.0 START,
    0.0 FINISH, 0 JCL, 0 IDLE,
    TriDist_Sampling( P_Dur, MinDur, MLDur, MaxDur) as RDUR,
    TriDist_Sampling( P_FC, MinFC , MLFC , MaxFC ) as RFC,
    TriDist_Sampling( P_VC, MinVC , MLVC , MaxVC ) as RVC,
    TriDist_Sampling( P_OHC, MinOHC , MLOHC , MaxOHC ) as ROHC,
    0 as TC , 0 as FC, 0 as VC, 0 as OHC, 0 as PC,  0 as BC,
    CHROMOSOME , Lable,  Num_Of_Modes
    FROM
    Experiment_run , UNNEST(SPLIT( CHROMOSOME )) oid WITH OFFSET tid
    JOIN Tasks t1 ON t1.P_Activity_ID = tid + 1
    JOIN Modes t2 ON t2.Activity_IID = t1.U_Activity_ID
     and  CAST(REPT_UNIT AS STRING)  =
     cast(Ceil ((tid + 1)/5/* No of Activities in a Typical unit */)
     as string)
    AND CAST(Mode_Option AS STRING) = oid
  -- order by TRADE_OFF_RUN, S_Activity_ID
    )
    ,
  --######################################################################
  -- The sub querys JCL_Calc and JCL_Calc2 calls the defined temporary
  -- functions to determine the supreme chromosome solution.
  --######################################################################
    JCL_Calc as (
      SELECT  rec.*
    FROM (  SELECT ARRAY_AGG(TO_JSON_STRING(t)) AS data
    FROM SETUP as t
    GROUP BY TRADE_OFF_RUN
    ) as t
   , UNNEST(Simulation_and_JCL_Calculation_Function(data)) AS rec
   )

    SELECT  1 as GENERATION,
    FINISH,        TC,        FC,        VC,
    OHC, 0.0 as        IDLE_C,        PC,
    BC, ""         as Shift_Combination_String,
    "" as IDLE_Combination_, 0.0 as        Interruption,
    "" as        Lable_Combination_String,
    CHROMOSOME
    FROM (
    SELECT
    ARRAY_AGG(TO_JSON_STRING(t)) AS data
```

```
      FROM JCL_Calc as t
      ), UNNEST(Normalization_of_Frontier_Solutions_Function(data)) AS rec

      -- Optimization Method
    -- for  Joint cost,schedule,Interruption minimization Change the
    --        Order by to: Vector, FINISH, TC
    -- for  cost minimization Change the Order by to: TC , FINISH
    -- for schedule minimization Change the Order by to: FINISH, TC

    Order by Vector , FINISH, TC
    Limit 1
```

b) ESDTCT^{Rand}

```
  #standardSQL

--  ESDTCT_RAND Model
-- Run Number 1

--####################################################################
-- This code is developed using BigQuery SQL statements with Temporary
-- functions having an in-line Java script programing. Readers of this
-- code need advanced programing skills.
-- The input to this script is two BigQuery tables namely (Tasks_bq
-- and Modes_bq). A sub query is designed to generate a random sample
-- of experiments. The input tables are created in Google
-- sheets and uploaded to BigQuery using the Google BigQuery API in
-- Apps Script.
-- (Note: This is an advanced service that must be enabled before use.
-- Reference:
-- https://developers.google.com/apps-script/guides/services/advanced)
-- The Output Table is saved temporarily in BigQuery Tables.
-- 100 Concurrent Runs of this script is made and all data is apended to
-- the same output table.
-- Tasks_bq table contains data input describing the network
-- activities and logic relationship for a typical unit in a repetative
-- project.
-- Modes_bq table contains data input describing the admitted modes to
-- each activity and probabilistic estimates parameters for the cost
-- and duration data.
-- A Number of code parameters, where noted in the script, are modified
-- using the Apps Script before executing the API command.
--####################################################################
-- The Simulation_and_JCL_Calculation_Function performs a Monte Carlo
-- simulation and identify the frontier curve solution satisfying the
-- user defined JCL.
--####################################################################
  CREATE TEMPORARY FUNCTION Simulation_and_JCL_Calculation_Function (arr ARRAY<STRING>)
  RETURNS ARRAY<STRUCT<TRADE_OFF_RUN INT64, SIMULATION_RUN  INT64,
  S_Activity_ID INT64,
  P_Activity_ID INT64, U_Activity_ID INT64,  START FLOAT64 ,
  FINISH FLOAT64, TC FLOAT64 , JCL FLOAT64,
```

```
CHROMOSOME String ,
Mode_Option INT64,
Activity_Type string, P1 INT64 , P2 INT64, P3 INT64, P4 INT64,
P5 INT64, P6 INT64, Ladder_Seq INT64,  RDUR FLOAT64, RFC FLOAT64,
RVC FLOAT64 , ROHC FLOAT64, FC FLOAT64, VC FLOAT64, OHC FLOAT64,
PC FLOAT64,  BC FLOAT64,
Lable STRING, Num_Of_Modes INT64
>>
LANGUAGE js AS """
var result = [];
for (var i = 0; i < arr.length; i++){
result.push(JSON.parse(arr[i])) }
for (var i = 0; i < arr.length; i++){
if (result[i].Probability_of_Occurrence < Math.random()){
result[i].RDUR ==  0
result[i].RFC ==   0
result[i].RVC ==  0
}}

result.sort(function (a,b) {
  if (a.SIMULATION_RUN > b.SIMULATION_RUN) return  1;
  if (a.SIMULATION_RUN < b.SIMULATION_RUN) return -1;
  if (a.Ladder_Seq > b.Ladder_Seq) return  1;
  if (a.Ladder_Seq < b.Ladder_Seq) return -1;
  return 0;
});
var arry = []
for (var i = 0; i < result.length; i++) {
arry.push (result[i].S_Activity_ID)
}
for (var i = 0; i < result.length; i++) {

var x1 = arry.indexOf(result[i].P1);
var x2 = arry.indexOf(result[i].P2);
var x3 = arry.indexOf(result[i].P3);
var x4 = arry.indexOf(result[i].P4);
var x5 = arry.indexOf(result[i].P5);
var x6 = arry.indexOf(result[i].P6);

 if (x1 == -1) {var FX1 = 0} else {var FX1 = result[x1].FINISH}
 if (x2 == -1) {var FX2 = 0} else {var FX2 = result[x2].FINISH}
 if (x3 == -1) {var FX3 = 0} else {var FX3 = result[x3].FINISH}
 if (x4 == -1) {var FX4 = 0} else {var FX4 = result[x4].FINISH}
 if (x5 == -1) {var FX5 = 0} else {var FX5 = result[x5].FINISH}
 if (x6 == -1) {var FX6 = 0} else {var FX6 = result[x6].FINISH}

result[i].START = Math.max( FX1, FX2,FX3,FX4,FX5,FX6)
+ 0  /*continuum_time_calculations*/;
result[i].FINISH = result[i].START + result[i].RDUR
- 0  /*continuum_time_calculations*/;
}
result.sort(function (a,b) {
  if (a.S_Activity_ID > b.S_Activity_ID) return  1;
  if (a.S_Activity_ID < b.S_Activity_ID) return -1;
  return 0;
});
var Num_of_Project_Tasks = 5/* No of Activities in a Typical unit */
  var arry2 = []
for (var i = 0; i < result.length; i++) {
arry2.push (result[i].S_Activity_ID)
}
```

```
    var TTC = 0; var TFC = 0 ; var TVC = 0;
    var TOHC = 0 ;
    for (var i = 0; i < result.length; i++) {

    TFC = TFC + (result[i].RFC)
    if ( result[i].P_Activity_ID == Num_of_Project_Tasks){
    result[i].FC = TFC , TFC = 0}
    TVC = TVC + (result[i].RVC * result[i].RDUR)
    if ( result[i].P_Activity_ID == Num_of_Project_Tasks){
    result[i].VC = TVC , TVC = 0}
    if ( result[i].P_Activity_ID == Num_of_Project_Tasks){
    result[i].OHC = result[i].ROHC * result[i].FINISH}

    TTC +=
    (result[i].RFC + (result[i].RVC * result[i].RDUR))
    if ( result[i].P_Activity_ID == Num_of_Project_Tasks){
    result[i].TC = TTC + result[i].ROHC * result[i].FINISH , TTC = 0

  // Adding Penalty / Bonus Cost

  // Linear continous schemes for Penalty / Bonus Cost

  // Schedule deadline driven projects - Penalty Cost scheme:
   if ((result[i].FINISH - Deadline_) > 0){
    result[i].PC =
    Math.min((Schedule_Penalty_Cost * (result[i].FINISH - Deadline_)),
    Max_Schedule_Penalty__Cost )
    }
  // Schedule deadline driven projects - Bonus Cost scheme:
   if ((result[i].FINISH - Deadline_) < 0){
    result[i].BC =
    Math.max((Schedule_Bonus_Cost * (result[i].FINISH - Deadline_)),
    - Max_Schedule_Bonus__Cost)
    }

  // Budget driven projects - Penalty Cost scheme:

   if ((result[i].TC - Budget__) > 0){
    result[i].PC +=
    Math.min((Budget_Penalty_Cost * (result[i].TC - Budget__)),
    Max_Budget_Penalty__Cost )
    }
  // Budget driven projects - Bonus Cost scheme:
   if ((result[i].TC - Budget__) < 0){
   result[i].BC +=
    Math.max((Budget_Bonus_Cost * (result[i].TC - Budget__)),
    - Max_Budget_Bonus__Cost)
    }
    result[i].TC += (result[i].PC + result[i].BC)
    }
    }
    var Final_result = result.filter(function (dataRow) {
    return dataRow.TC > 0;
    });

    for (var i = 0; i < Final_result.length; i++){
    var count = 0
    for (var j = 0; j < Final_result.length; j++){
    if (Final_result[i].TC >= Final_result[j].TC  &&
    Final_result[i].FINISH >= Final_result[j].FINISH ){count +=1 }
    }
```

```
    Final_result[i].JCL = count / (Final_result.length )
    }

      var Final_result1 = Final_result.filter(function (dataRow) {
    return dataRow.JCL > 0.5 && dataRow.JCL < 0.6/* Lower and upper JCL:
    Change Numbers using App Script */;
    });

    return Final_result1
    """;

  --#####################################################################
  -- The Normalization_of_Frontier_Solutions_Function performs calculations
  -- to nomalize the cost and schedule pairs of the Frontier solutions to
  -- the range [0,1]
  --#####################################################################
    CREATE TEMPORARY FUNCTION Normalization_of_Frontier_Solutions_Function (arr ARRAY<STRING>)
    RETURNS ARRAY<STRUCT<TRADE_OFF_RUN INT64, SIMULATION_RUN INT64,
    S_Activity_ID INT64, P_Activity_ID INT64,
    U_Activity_ID INT64, START FLOAT64 ,  FINISH FLOAT64, TC FLOAT64 ,
    JCL FLOAT64,
    CHROMOSOME String ,
    Mode_Option INT64,
    Activity_Type string, P1 INT64 , P2 INT64, P3 INT64, P4 INT64,
    P5 INT64, P6 INT64, Ladder_Seq INT64, RDUR FLOAT64, RFC FLOAT64,
    RVC FLOAT64 , ROHC FLOAT64, FC FLOAT64, VC FLOAT64,
     OHC FLOAT64, PC FLOAT64,  BC FLOAT64,
     Lable STRING, Num_Of_Modes INT64 ,
     Vector FLOAT64
     >>
    LANGUAGE js AS """
    var Final_result = [];
    for (var i = 0; i < arr.length; i++){
    Final_result.push(JSON.parse(arr[i])) }
    var maxTC = Math.max.apply(Math, Final_result.map(function(v) {
    return v.TC;}));
  //  var minTC = Math.min.apply(Math, Final_result.map(function(v){
  //  return v.TC;}));
     var minTC = 0
     var maxFinish = Math.max.apply(Math, Final_result.map(function(v)
     { return v.FINISH;}));
  // var minFinish = Math.min.apply(Math, Final_result.map(function(v){
  // return v.FINISH;}));
     var minFinish = 0
    for (var i = 0; i < Final_result.length; i++){
    Final_result[i].Vector =
    Math.sqrt(Math.pow(((Final_result[i].FINISH - minFinish )/
    (maxFinish - minFinish)), 2)
    + Math.pow((Final_result[i].TC - minTC)/(maxTC - minTC),2) )
        }
    return Final_result
    """;


     CREATE TEMP FUNCTION TriDist_Sampling( P FLOAT64 ,
     a FLOAT64, m FLOAT64, b FLOAT64)
    RETURNS FLOAT64
    LANGUAGE js AS """
    var d ;   var x
    d = b - a
```

```
     if (d != 0){x = (m - a) / d} else {x = b;}
     if (P <= x){return (a + ((Math.sqrt(P * x)) * d))} else {
     return  (b - ((Math.sqrt((1 - P) * (1 - x))) * d));}
     """;

   -- END OF TEMPORARY FUNCTIONS
   --#####################################################################
   -- The below series of BigQuery SQL Sub Queries are set to perform the
   -- data input set up for the described Temporary functions above.
   --#####################################################################
     WITH

   --#####################################################################
   -- The sub query Simulation below prepares a replication of the project
   -- to a user defined number of simulation scenarios.
   --#####################################################################
     Simulation as (
     Select
       SIMULATION_RUN  , rec.*
      , RAND() as P_Dur
      , RAND() as P_FC
      , RAND() as P_VC
      , RAND() as P_OHC
      from
       ESDTCT_Database.Tasks_bq rec
        Cross JOIN
        UNNEST((GENERATE_ARRAY(1,1000) )) AS SIMULATION_RUN
   --      Change SIMULATION_RUN Number using App Script
   --    order by SIMULATION_RUN ,Activity_ID
        )
     ,
   --#####################################################################
   -- The sub query Tasks below prepares the initial network details for
   -- a typical unit and creates the global network for the user defined
   -- number of repetitive units.
   --#####################################################################
       Tasks as (
       Select
       SIMULATION_RUN, Activity_Type,
       Probability_of_Occurrence, Ladder_Seq,
       Lable, Num_Of_Modes,
       Activity_ID as U_Activity_ID  ,
       Activity_ID    as P_Activity_ID
       , (Activity_ID  + (SIMULATION_RUN-1) *
       ( Num))   as S_Activity_ID
       , P1 + (SIMULATION_RUN - 1) *
       ( Num)   as P1
       , P2  + (SIMULATION_RUN - 1) *
       ( Num)   as P2
       , P3  + (SIMULATION_RUN - 1) *
       ( Num)   as P3
       , P4  + (SIMULATION_RUN - 1) *
       ( Num)   as P4
       , P5  + (SIMULATION_RUN - 1) *
       ( Num)   as P5
       , P6  + (SIMULATION_RUN - 1) *
       ( Num)   as P6
        , P_Dur , P_FC , P_VC , P_OHC
        from Simulation rec

                Cross JOIN
```

```
    UNNEST((GENERATE_ARRAY(5/* No of Activities in a Typical unit */,
         5/* No of Activities in a Typical unit */ ) ))
         AS Num /*: Change Number using App Script */

     --   order by S_Activity_ID
          )
   ,
   --######################################################################
   -- The sub query Modes below reads the user parameter inputs for each
   -- activity admitted modes
   --######################################################################
     Modes AS (
     SELECT * from `ESDTCT_Database.Modes_bq`
     )
     ,
   --######################################################################
   -- The sub query Experiment_run below prepares the activity modes input
   -- to activities.
   --######################################################################
     Experiment_run_1 AS (
     SELECT TRADE_OFF_RUN, t1.Activity_IID, t1.Mode_Option as MODE , Rand() as filter
     FROM Modes  t1
     CROSS JOIN
     UNNEST((GENERATE_ARRAY(1, 10000))) as TRADE_OFF_RUN

     JOIN Modes t2 ON t2.Activity_IID = t2.Activity_IID and t1.Mode_Option = t2.Mode_Option


   group by TRADE_OFF_RUN, t1.Activity_IID, t1.Mode_Option
   Order by TRADE_OFF_RUN, t1.Activity_IID
     )
     ,

     Experiment_run AS (
   SELECT TRADE_OFF_RUN, STRING_AGG(Cast(MODE as string) ORDER BY Activity_IID) as CHROMOSOME
   FROM (
     SELECT
       TRADE_OFF_RUN, Activity_IID, MODE,
       RAND() AS rnd, ROW_NUMBER() OVER(PARTITION BY TRADE_OFF_RUN, Activity_IID ) AS pos
     FROM Experiment_run_1
   )

   WHERE pos <= 1
   group by TRADE_OFF_RUN
   -- ORDER BY TRADE_OFF_RUN

     )
   ,
   --######################################################################
   -- The sub query SETUP below joins the data input tables and assign a
   -- random cost and duration values based on the trinagular probability
   -- distribution function.
   --######################################################################
     SETUP as (
     SELECT TRADE_OFF_RUN, SIMULATION_RUN, S_Activity_ID ,
     U_Activity_ID, P_Activity_ID,
     Probability_of_Occurrence , Mode_Option,
```

```
     Activity_Type, P1, P2, P3, P4, P5, P6, Ladder_Seq, 0.0 START,
     0.0 FINISH, 0 JCL, 0 IDLE,
     TriDist_Sampling( P_Dur, MinDur, MLDur, MaxDur) as RDUR,
     TriDist_Sampling( P_FC, MinFC , MLFC , MaxFC ) as RFC,
     TriDist_Sampling( P_VC, MinVC , MLVC , MaxVC ) as RVC,
     TriDist_Sampling( P_OHC, MinOHC , MLOHC , MaxOHC ) as ROHC,
     0 as TC , 0 as FC, 0 as VC, 0 as OHC, 0 as PC,  0 as BC,
     CHROMOSOME , Lable,  Num_Of_Modes
     FROM
     Experiment_run , UNNEST(SPLIT( CHROMOSOME )) oid WITH OFFSET tid
     JOIN Tasks t1 ON t1.P_Activity_ID = tid + 1
     JOIN Modes t2 ON t2.Activity_IID = t1.U_Activity_ID
      and  CAST(REPT_UNIT AS STRING)  =
      cast(Ceil ((tid + 1)/5/* No of Activities in a Typical unit */)
      as string)
     AND CAST(Mode_Option AS STRING) = oid
   -- order by TRADE_OFF_RUN, S_Activity_ID
     )
     ,
   --####################################################################
   -- The sub querys JCL_Calc and JCL_Calc2 calls the defined temporary
   -- functions to determine the supreme chromosome solution.
   --####################################################################
     JCL_Calc as (
       SELECT  rec.*
     FROM (  SELECT ARRAY_AGG(TO_JSON_STRING(t)) AS data
     FROM SETUP as t
     GROUP BY TRADE_OFF_RUN
     ) as t
    , UNNEST(Simulation_and_JCL_Calculation_Function(data)) AS rec
    )

     SELECT  TRADE_OFF_RUN, FINISH , TC  , Vector ,
      CHROMOSOME , JCL
     FROM (
     SELECT
     ARRAY_AGG(TO_JSON_STRING(t)) AS data
     FROM JCL_Calc as t
     ), UNNEST(Normalization_of_Frontier_Solutions_Function(data)) AS rec

```

c) RP-ESDTCT^{EXP}

```
#standardSQL


--  RE-ESDTCT_EXP Model

--####################################################################
-- This code is developed using BigQuery SQL statements with Temporary
-- functions having an in-line Java script programing. Readers of this
-- code need advanced programing skills.
-- The input to this script is two BigQuery tables namely (Tasks_bq
-- , Modes_bq and Experiments). The input tables are created in Google
-- sheets and uploaded to BigQuery using the Google BigQuery API in
-- Apps Script.
-- (Note: This is an advanced service that must be enabled before use.
-- Reference:
-- https://developers.google.com/apps-script/guides/services/advanced)
-- Tasks_bq table contains data input describing the network
-- activities and logic relationship for a typical unit in a repetative
-- project.
-- Modes_bq table contains data input describing the admitted modes to
-- each activity and probabilistic estimates parameters for the cost
-- and duration data.
-- Experiments table contains the cartisian product for the enummeration
-- of primary activities
-- A Number of code parameters, where noted in the script, are modified
-- using the Apps Script before executing the API command.
--####################################################################
-- The Simulation_and_JCL_Calculation_Function performs a Monte Carlo
-- simulation and identify the frontier solution satisfying the
-- user defined JCL.
--####################################################################

CREATE TEMPORARY FUNCTION Simulation_and_JCL_Calculation_Function (arr ARRAY<STRING>)
RETURNS ARRAY<STRUCT<TRADE_OFF_RUN INT64, SIMULATION_RUN  INT64,
REP_UNIT INT64,  P_Rep INT64, S_Activity_ID INT64,
P_Activity_ID INT64, U_Activity_ID INT64,  START FLOAT64 ,
FINISH FLOAT64, TC FLOAT64 , JCL FLOAT64, IDLE FLOAT64,
IDLE_Combination_String ARRAY<INT64>,
Lable_Combination_String string , Interruption FLOAT64 ,
CHROMOSOME String , Opt_Idle_Time_Combination_String string ,
Mode_Option INT64,Shift_Combination_String string,
Activity_Type string, P1 INT64 , P2 INT64, P3 INT64, P4 INT64,
P5 INT64, P6 INT64, Ladder_Seq INT64,  RDUR FLOAT64, RFC FLOAT64,
RVC FLOAT64 , ROHC FLOAT64, FC FLOAT64, VC FLOAT64, OHC FLOAT64,
PC FLOAT64,  BC FLOAT64,
Lable STRING, Num_Of_Modes INT64 ,Opt_Idle_Time Int64
>>
LANGUAGE js AS """
var result = [];
for (var i = 0; i < arr.length; i++){
result.push(JSON.parse(arr[i])) }
for (var i = 0; i < arr.length; i++){
if (result[i].Probability_of_Occurrence < Math.random()){
result[i].RDUR ==  0
result[i].RFC ==   0
result[i].RVC ==  0
```

```
}}

result.sort(function (a,b) {
  if (a.SIMULATION_RUN > b.SIMULATION_RUN) return  1;
  if (a.SIMULATION_RUN < b.SIMULATION_RUN) return -1;
  if (a.REP_UNIT > b.REP_UNIT) return  1;
  if (a.REP_UNIT < b.REP_UNIT) return -1;
  if (a.Ladder_Seq > b.Ladder_Seq) return  1;
  if (a.Ladder_Seq < b.Ladder_Seq) return -1;
  return 0;
});
var arry = []
for (var i = 0; i < result.length; i++) {
arry.push (result[i].S_Activity_ID)
}
for (var i = 0; i < result.length; i++) {

var x1 = arry.indexOf(result[i].P1);
var x2 = arry.indexOf(result[i].P2);
var x3 = arry.indexOf(result[i].P3);
var x4 = arry.indexOf(result[i].P4);
var x5 = arry.indexOf(result[i].P5);
var x6 = arry.indexOf(result[i].P6);
var xR = arry.indexOf(result[i].P_Rep);

 if (x1 == -1) {var FX1 = 0} else {var FX1 = result[x1].FINISH}
 if (x2 == -1) {var FX2 = 0} else {var FX2 = result[x2].FINISH}
 if (x3 == -1) {var FX3 = 0} else {var FX3 = result[x3].FINISH}
 if (x4 == -1) {var FX4 = 0} else {var FX4 = result[x4].FINISH}
 if (x5 == -1) {var FX5 = 0} else {var FX5 = result[x5].FINISH}
 if (x6 == -1) {var FX6 = 0} else {var FX6 = result[x6].FINISH}
 if (xR == -1) {var FXR = 0} else {var FXR = result[xR].FINISH}

result[i].START = Math.max( FX1, FX2,FX3,FX4,FX5,FX6,FXR)
+ 0/*continuum_time_calculations*/;
result[i].FINISH = result[i].START + result[i].RDUR
- 0/*continuum_time_calculations*/;
}
result.sort(function (a,b) {
  if (a.S_Activity_ID > b.S_Activity_ID) return  1;
  if (a.S_Activity_ID < b.S_Activity_ID) return -1;
  return 0;
});
    var Num_of_Project_Tasks = 5/* No of Activities in a Typical unit */
    * 4/* number of units : Change Numbers using App Script */
      var arry2 = []
    for (var i = 0; i < result.length; i++) {
    arry2.push (result[i].S_Activity_ID)
    }
    var TInt = 0
    var TTC = 0
    var IDLE_Combination = [0,0,0,0,0]
    var Lable_Combination = []
    var Lable_Combination_String = ""
    var Opt_Idle_Time_Combination = []
    var Opt_Idle_Time_Combination_String = ""

    for (var i = 0; i < result.length; i++) {

    if( result[i].REP_UNIT > 1 ){  result[i].IDLE  =
    parseInt(result[i].START -
```

```
     result[arry2.indexOf(result[i].P_Rep)].FINISH)
     - 0/*continuum_time_calculations*/;
     IDLE_Combination[result[i].U_Activity_ID - 1] +=
     parseInt(result[i].IDLE)
     Lable_Combination[result[i].U_Activity_ID - 1] =
     result[i].Lable
     }
       else{ IDLE_Combination[result[i].U_Activity_ID - 1] = 0;
         Opt_Idle_Time_Combination[result[i].U_Activity_ID - 1] =
         result[i]. Opt_Idle_Time
     }
     if( result[i].P_Activity_ID == Num_of_Project_Tasks){
      result[i].IDLE_Combination_String = IDLE_Combination
    result[i].Lable_Combination_String = Lable_Combination.join(", ")
       result[i].Opt_Idle_Time_Combination_String =
       Opt_Idle_Time_Combination.join(", ")
     }
     TTC = TTC + (result[i].RFC + (result[i].RVC * result[i].RDUR))
     if ( result[i].P_Activity_ID == Num_of_Project_Tasks){
     result[i].TC = TTC + result[i].ROHC * result[i].FINISH , TTC = 0

   // Adding Penalty / Bonus Cost

   // Linear continous schemes for Penalty / Bonus Cost

   // Schedule deadline driven projects - Penalty Cost scheme:
    if ((result[i].FINISH - Deadline_) > 0){
     result[i].PC =
     Math.min((Schedule_Penalty_Cost * (result[i].FINISH - Deadline_)),
     Max_Schedule_Penalty__Cost )
     }
   // Schedule deadline driven projects - Bonus Cost scheme:
    if ((result[i].FINISH - Deadline_) < 0){
     result[i].BC =
     Math.max((Schedule_Bonus_Cost * (result[i].FINISH - Deadline_)),
     - Max_Schedule_Bonus__Cost)
     }

   // Budget driven projects - Penalty Cost scheme:

    if ((result[i].TC - Budget__) > 0){
     result[i].PC +=
     Math.min((Budget_Penalty_Cost * (result[i].TC - Budget__)),
     Max_Budget_Penalty__Cost )
     }
   // Budget driven projects - Bonus Cost scheme:
    if ((result[i].TC - Budget__) < 0){
    result[i].BC +=
     Math.max((Budget_Bonus_Cost * (result[i].TC - Budget__)),
     - Max_Budget_Bonus__Cost)
     }
     result[i].TC += (result[i].PC + result[i].BC)
     }
     }
     var Final_result = result.filter(function (dataRow) {
     return dataRow.TC > 0;
     });

     for (var i = 0; i < Final_result.length; i++){
     var count = 0
     for (var j = 0; j < Final_result.length; j++){
```

```
      if (Final_result[i].TC >= Final_result[j].TC  &&
      Final_result[i].FINISH >= Final_result[j].FINISH ){count +=1 }
      }
      Final_result[i].JCL = count / (Final_result.length )
      }

        var Final_result1 = Final_result.filter(function (dataRow) {
      return dataRow.JCL > 0.5 && dataRow.JCL < 0.6/* Lower and upper JCL:
      Change Numbers using App Script */;
      });
     var Final_result3 = [];
        for(var i = 0; i<Final_result1.length; ++i) {
            var a = Final_result1[i];
            for(var j = 0; j<result.length; ++j) {
                    var b = result[j];
                    if(a.TRADE_OFF_RUN == b.TRADE_OFF_RUN &&
                    a.SIMULATION_RUN == b.SIMULATION_RUN) {

                    Final_result3.push(b);
                    }
            }
        }
      return Final_result3
      """;

    --######################################################################
    -- The Normalization_of_Frontier_Solutions_Function performs calculations
    -- to nomalize the cost and schedule pairs of the Frontier solutions to
    -- the range [0,1]
    --######################################################################
      CREATE TEMPORARY FUNCTION Normalization_of_Frontier_Solutions_Function (arr ARRAY<STRING>)
      RETURNS ARRAY<STRUCT<TRADE_OFF_RUN INT64, SIMULATION_RUN INT64,
      REP_UNIT INT64, P_Rep INT64, S_Activity_ID INT64, P_Activity_ID INT64,
      U_Activity_ID INT64, START FLOAT64 ,  FINISH FLOAT64, TC FLOAT64 ,
      JCL FLOAT64, IDLE FLOAT64, IDLE_Combination_String ARRAY<INT64>,
      Lable_Combination_String string , Interruption FLOAT64 ,
      CHROMOSOME String , Opt_Idle_Time_Combination_String string ,
      Mode_Option INT64,Shift_Combination_String string,
      Activity_Type string, P1 INT64 , P2 INT64, P3 INT64, P4 INT64,
      P5 INT64, P6 INT64, Ladder_Seq INT64, RDUR FLOAT64, RFC FLOAT64,
      RVC FLOAT64 , ROHC FLOAT64, FC FLOAT64, VC FLOAT64,
       OHC FLOAT64, PC FLOAT64,  BC FLOAT64,
       Lable STRING, Num_Of_Modes INT64 ,Opt_Idle_Time Int64,
       Vector FLOAT64
       >>
      LANGUAGE js AS """
      var Final_result = [];
      for (var i = 0; i < arr.length; i++){
      Final_result.push(JSON.parse(arr[i])) }
      var maxTC = Math.max.apply(Math, Final_result.map(function(v) {
      return v.TC;}));
    //  var minTC = Math.min.apply(Math, Final_result.map(function(v){
    //  return v.TC;}));
       var minTC = 0
       var maxFinish = Math.max.apply(Math, Final_result.map(function(v)
       { return v.FINISH;}));
    // var minFinish = Math.min.apply(Math, Final_result.map(function(v){
    // return v.FINISH;}));
       var minFinish = 0
      for (var i = 0; i < Final_result.length; i++){
```

```
    Final_result[i].Vector =
    Math.sqrt(Math.pow(((Final_result[i].FINISH - minFinish )/
    (maxFinish - minFinish)), 2)
    + Math.pow((Final_result[i].TC - minTC)/(maxTC - minTC),2) )
        }
    return Final_result
    """;

  --######################################################################
  -- The Crew_Idle_Time_Optimization Function performs calculations
  -- on the supreme chromosome solution to find the optimal crew
  -- interruption times
  --######################################################################
    CREATE TEMPORARY FUNCTION Crew_Idle_Time_Optimization_Function(arr ARRAY<STRING>)
    RETURNS ARRAY<STRUCT< FINISH FLOAT64,  TC FLOAT64 , FC FLOAT64 ,
    VC FLOAT64 , OHC FLOAT64, IDLE_C FLOAT64, PC FLOAT64,  BC FLOAT64,
    Shift_Combination_String String , IDLE_Combination_ String,
    Interruption FLOAT64, Lable_Combination_String String,
    CHROMOSOME String , Vector FLOAT64
    >>
    LANGUAGE js AS """
    var result = [];     var TOarr = [];
    var Num_of_Project_Tasks = 5/* No of Activities in a Typical unit */
    * 4/* number of units : Change Numbers using App Script */

      result.sort(function (a,b) {
      if (a.S_Activity_ID > b.S_Activity_ID) return  1;
      if (a.S_Activity_ID < b.S_Activity_ID) return -1;
      return 0;
  });
    for (var i = 0; i < arr.length; i++){
    result.push(JSON.parse(arr[i]))}
   var  Lable_arr = ['N', 'N', 'N', 'P', 'P'];
  // Change Lable_arr values using App Script */
   //if (4/* number of units */ == 1) {
   //CPM(result)
   //return result}
   //else {

    for (var gen = 1; gen < 5; gen++) {
     for (var geninc = 1; geninc < 2; geninc++) {
    if(gen == 1){idle_arr2 =
    result[(Num_of_Project_Tasks - 1)].IDLE_Combination_String}
    else {idle_arr2 = Final_result[0].IDLE_Combination_String}

   for (var i = 0; i < idle_arr2.length; i++) {
    if(Lable_arr[(gen-1)][i] == 'N'){
    TOarr[i] = []; TOarr[i].push (idle_arr2[i])}
    if(Lable_arr[(gen-1)][i] == 'P'){
     TOarr[i] = [];
    if(geninc == 1){
    var Finish_Domain = parseInt(idle_arr2[i]) + 1 ;
    var Start_Domain = 0;
    var Inc_Domain = Math.max(1,((Finish_Domain - Start_Domain +1)/10))
    }
    if(geninc == 2 || geninc == 3 ){
    var Finish_Domain = parseInt(idle_arr2[i]) + 1 ;
    var Start_Domain =
    Math.max(0,(Finish_Domain-Math.ceil((parseInt(idle_arr2[i]))/10)));
    var Inc_Domain = Math.max(1,((Finish_Domain - Start_Domain +1)/10))
```

```
    }
   for (var p = 0; p < Finish_Domain ; p += Inc_Domain){
   TOarr[i].push (p)}}
    if(Lable_arr[(gen-1)][i] == 'S'){
    TOarr[i] = []; TOarr[i].push (idle_arr2[i])}
    }
     idle_Time_arr = cartesianProduct(TOarr)
    var Final_result = []
    for (var t = 0; t < idle_Time_arr.length; t++) {
     var GENERATION = 1
    for (var i = 0; i < result.length; i++) {
    result[i].Lable_Combination_String =
    Lable_arr[(GENERATION - 1)].join(", ")
    result[i].T_idle = 0
    result[i].T_idle_Sim =+ i+1

    if( result[i].REP_UNIT == 1){ result[i].T_idle  =
    parseInt(idle_Time_arr[t][result[i].U_Activity_ID - 1])}
    if( result[i].P_Activity_ID == Num_of_Project_Tasks){
    result[i].Shift_Combination_String = idle_Time_arr[t].join(", ")
        result[i].Interruption =
        result[i].IDLE_Combination_String.reduce( function(tt, ss) {
        return parseInt(tt) + parseInt(ss); } );
    result[i].IDLE_Combination_ =
    result[i].IDLE_Combination_String.join(", ")
        }
    }
   CPM(result)
  Final_result[t] = []
   Final_result[t] =
   JSON.parse(JSON.stringify(result[(Num_of_Project_Tasks-1)]));
    }
  // Activate the script below for the desired objective function
  // using Apps Script command lines.

  /* Joint Optimization
    var maxTC = Math.max.apply(Math, Final_result.map(function(v) {
    return v.TC;}));
  //  var minTC = Math.min.apply(Math, Final_result.map(function(v) {
  //return v.TC;}));
     var minTC = 0
     var maxFinish = Math.max.apply(Math, Final_result.map(function(v) {
     return v.FINISH;}));
  //   var minFinish =
  //    Math.min.apply(Math,Final_result.map(function(v) {
  //    return v.FINISH;}));
     var minFinish = 0
     var maxInterruption =
     Math.max.apply(Math, Final_result.map(function(v) {
     return v.Interruption;}));
    for (var i = 0; i < Final_result.length; i++){
    Final_result[i].Vector =
    Math.sqrt(Math.pow(((Final_result[i].FINISH - minFinish )/
    (maxFinish - minFinish)), 2)
    + Math.pow((Final_result[i].TC - minTC)/(maxTC - minTC),2) +
    Math.pow(((Final_result[i].Interruption )/(maxInterruption)),2) )
        }
     Final_result.sort(function (a,b) {
      if (a.Vector > b.Vector) return  1;
      if (a.Vector < b.Vector) return -1;
      if (a.FINISH > b.FINISH) return  1;
```

```
      if (a.FINISH < b.FINISH) return -1;
      if (a.TC > b.TC) return  1;
      if (a.TC < b.TC) return -1;
      if (a.Interruption > b.Interruption) return  1;
      if (a.Interruption < b.Interruption) return -1;
      return 0;
    });
  */ // Joint Optimization

  /* Schedule Minimization
     Final_result.sort(function (a,b) {
      if (a.FINISH > b.FINISH) return  1;
      if (a.FINISH < b.FINISH) return -1;
      if (a.TC > b.TC) return  1;
      if (a.TC < b.TC) return -1;
      if (a.Interruption > b.Interruption) return  1;
      if (a.Interruption < b.Interruption) return -1;
      return 0;
  });
  */ // Schedule Minimization

  /* Cost Minimization
     Final_result.sort(function (a,b) {
      if (a.TC > b.TC) return  1;
      if (a.TC < b.TC) return -1;
      if (a.FINISH > b.FINISH) return  1;
      if (a.FINISH < b.FINISH) return -1;
      if (a.Interruption > b.Interruption) return  1;
      if (a.Interruption < b.Interruption) return -1;
      return 0;
    });
  */ // Cost Minimization
      }
    }
     return Final_result
   //  }
  //###################
  function cartesianProduct(data) {
      var current = [[]];
      for (var p in data) {
          var arr = data[p];
          var newCurrent = [];
          for (var c = 0; c < current.length; c++) {
              var baseArray = current[c];
              for (var a = 0; a < arr.length; a++) {
                  var clone = baseArray.slice();
                  clone.push(arr[a]);
                  newCurrent.push(clone);
              }
          }
          current = newCurrent;
      }
      return current;
  }
  //######################
     function CPM(result){
    result.sort(function (a,b) {
      if (a.SIMULATION_RUN > b.SIMULATION_RUN) return  1;
      if (a.SIMULATION_RUN < b.SIMULATION_RUN) return -1;
      if (a.REP_UNIT > b.REP_UNIT) return  1;
      if (a.REP_UNIT < b.REP_UNIT) return -1;
```

```
      if (a.Ladder_Seq > b.Ladder_Seq) return  1;
      if (a.Ladder_Seq < b.Ladder_Seq) return -1;
      return 0;
  });

    var arry = []
    for (var i = 0; i < result.length; i++) {
    arry.push (result[i].S_Activity_ID)
    }
    for (var i = 0; i < result.length; i++) {
    var x1 = arry.indexOf(result[i].P1);
    var x2 = arry.indexOf(result[i].P2);
    var x3 = arry.indexOf(result[i].P3);
    var x4 = arry.indexOf(result[i].P4);
    var x5 = arry.indexOf(result[i].P5);
    var x6 = arry.indexOf(result[i].P6);
    var xR = arry.indexOf(result[i].P_Rep);

     if (x1 == -1) {var FX1 = 0} else {var FX1 = result[x1].FINISH}
     if (x2 == -1) {var FX2 = 0} else {var FX2 = result[x2].FINISH}
     if (x3 == -1) {var FX3 = 0} else {var FX3 = result[x3].FINISH}
     if (x4 == -1) {var FX4 = 0} else {var FX4 = result[x4].FINISH}
     if (x5 == -1) {var FX5 = 0} else {var FX5 = result[x5].FINISH}
     if (x6 == -1) {var FX6 = 0} else {var FX6 = result[x6].FINISH}
     if (xR == -1) {var FXR = 0} else {var FXR = result[xR].FINISH}

    result[i].START = Math.max( FX1, FX2,FX3,FX4,FX5,FX6,FXR)+
    parseFloat(result[i].T_idle) + 0/*continuum_time_calculations*/ ;
    result[i].FINISH = result[i].START +
    result[i].RDUR - 0/*continuum_time_calculations*/ ;
    }
    result.sort(function (a,b) {
      if (a.S_Activity_ID > b.S_Activity_ID) return  1;
      if (a.S_Activity_ID < b.S_Activity_ID) return -1;
      return 0;
    });
    var Num_of_Project_Tasks = 5/* No of Activities in a Typical unit */
    * 4/* number of units : Change Numbers using App Script */
      var arry2 = []
    for (var i = 0; i < result.length; i++) {
    arry2.push (result[i].S_Activity_ID)
    }
    var TTC = 0; var TFC = 0 ; var TVC = 0;
    var TOHC = 0 ; var TIDLE_C = 0
    var IDLE_Combination = [0,0,0,0,0]
    for (var i = 0; i < result.length; i++) {
    if( result[i].REP_UNIT > 1 ){  result[i].IDLE  =
    parseInt(result[i].START -
    result[arry2.indexOf(result[i].P_Rep)].FINISH)
    - 0/*continuum_time_calculations*/;
    IDLE_Combination[result[i].U_Activity_ID - 1] +=
    parseInt(result[i].IDLE)
    }
      else{ IDLE_Combination[result[i].U_Activity_ID - 1] = 0;
    }
    if( result[i].P_Activity_ID == Num_of_Project_Tasks){
     result[i].IDLE_Combination_String = IDLE_Combination
     }
    TFC = TFC + (result[i].RFC)
    if ( result[i].P_Activity_ID == Num_of_Project_Tasks){
    result[i].FC = TFC , TFC = 0}
```

```
    TVC = TVC + (result[i].RVC * result[i].RDUR)
    if ( result[i].P_Activity_ID == Num_of_Project_Tasks){
    result[i].VC = TVC , TVC = 0}
    if ( result[i].P_Activity_ID == Num_of_Project_Tasks){
    result[i].OHC = result[i].ROHC * result[i].FINISH}

    TIDLE_C = TIDLE_C +
    (result[i].RVC * parseFloat(result[i].IDLE))
    if ( result[i].P_Activity_ID == Num_of_Project_Tasks){
    result[i].IDLE_C = TIDLE_C , TIDLE_C = 0}
     TTC = TTC +
    (result[i].RFC + (result[i].RVC * result[i].RDUR)
    + (result[i].RVC * parseFloat(result[i].IDLE)))
    if ( result[i].P_Activity_ID == Num_of_Project_Tasks){
    result[i].TC = TTC + result[i].ROHC * result[i].FINISH , TTC = 0

   // Adding Penalty / Bonus Cost

  // Linear continous schemes for Penalty / Bonus Cost

  // Schedule deadline driven projects - Penalty Cost scheme:
   if ((result[i].FINISH - Deadline_) > 0){
    result[i].PC =
    Math.min((Schedule_Penalty_Cost * (result[i].FINISH - Deadline_)),
    Max_Schedule_Penalty__Cost )
    }
  // Schedule deadline driven projects - Bonus Cost scheme:
   if ((result[i].FINISH - Deadline_) < 0){
    result[i].BC =
    Math.max((Schedule_Bonus_Cost * (result[i].FINISH - Deadline_)),
    - Max_Schedule_Bonus__Cost)
    }

  // Budget driven projects - Penalty Cost scheme:

   if ((result[i].TC - Budget__) > 0){
    result[i].PC +=
    Math.min((Budget_Penalty_Cost * (result[i].TC - Budget__)),
    Max_Budget_Penalty__Cost )
    }
  // Budget driven projects - Bonus Cost scheme:
   if ((result[i].TC - Budget__) < 0){
   result[i].BC +=
    Math.max((Budget_Bonus_Cost * (result[i].TC - Budget__)),
    - Max_Budget_Bonus__Cost)
    }
    result[i].TC += (result[i].PC + result[i].BC)
    }
     }
    }
    //
    """;
     CREATE TEMP FUNCTION TriDist_Sampling(P FLOAT64 ,
     a FLOAT64, m FLOAT64, b FLOAT64)
    RETURNS FLOAT64
    LANGUAGE js AS """
    var d ;   var x
    d = b - a
    if (d != 0){x = (m - a) / d} else {x = b;}
    if (P <= x){return (a + ((Math.sqrt(P * x)) * d))} else {
    return  (b - ((Math.sqrt((1 - P) * (1 - x))) * d));}
```

```
    """;

-- END OF TEMPORARY FUNCTIONS
--######################################################################
-- The below series of BigQuery SQL Sub Queries are set to perform the
-- data input set up for the described Temporary functions above.
--######################################################################
  WITH

--######################################################################
-- The sub query Simulation below prepares a replication of the project
-- to a user defined number of simulation scenarios.
--######################################################################
  Simulation as (
  Select
    SIMULATION_RUN  , rec.*
   , RAND() as P_Dur
   , RAND() as P_FC
   , RAND() as P_VC
   , RAND() as P_OHC
   from
    ESDTCT_Database.Tasks_bq rec
     Cross JOIN
     UNNEST((GENERATE_ARRAY(1,1000) )) AS SIMULATION_RUN
--     Change SIMULATION_RUN Number using App Script
--   order by SIMULATION_RUN ,Activity_ID
     )
  ,
--######################################################################
-- The sub query Tasks below prepares the initial network details for
-- a typical unit and creates the global network for the user defined
-- number of repetitive units.
--######################################################################
    Tasks as (
    Select
    REP_UNIT , SIMULATION_RUN, Activity_Type,
    Probability_of_Occurrence, Ladder_Seq,
    Lable, Num_Of_Modes, Opt_Idle_Time,
    Activity_ID as U_Activity_ID  ,
    Activity_ID + (REP_UNIT - 1) * (Num)   as P_Activity_ID
    , (Activity_ID + (REP_UNIT - 1) * (Num) + (SIMULATION_RUN-1) *
    (Num_of_Units * Num))   as S_Activity_ID
    , P1 + (REP_UNIT - 1) * (Num) + (SIMULATION_RUN - 1) *
    (Num_of_Units * Num)   as P1
    , P2 + (REP_UNIT - 1) * (Num) + (SIMULATION_RUN - 1) *
    (Num_of_Units* Num)   as P2
    , P3 + (REP_UNIT - 1) * (Num) + (SIMULATION_RUN - 1) *
    (Num_of_Units* Num)   as P3
    , P4 + (REP_UNIT - 1) * (Num) + (SIMULATION_RUN - 1) *
    (Num_of_Units* Num)   as P4
    , P5 + (REP_UNIT - 1) * (Num) + (SIMULATION_RUN - 1) *
    (Num_of_Units* Num)   as P5
    , P6 + (REP_UNIT - 1) * (Num) + (SIMULATION_RUN - 1) *
    (Num_of_Units* Num)   as P6
     , CASE WHEN (REP_UNIT >1) THEN  Activity_ID +
     (REP_UNIT-1)*Num+(SIMULATION_RUN-1)*
     (Num_of_Units* Num) - Num   END  as P_Rep
     , P_Dur , P_FC , P_VC , P_OHC
     from Simulation rec
           Cross JOIN
 UNNEST((GENERATE_ARRAY(1,4/* number of units */) ))
```

```
      AS REP_UNIT /*: Change Number using App Script */
               Cross JOIN
   UNNEST((GENERATE_ARRAY(5/* No of Activities in a Typical unit */,
        5/* No of Activities in a Typical unit */ ) ))
        AS Num /*: Change Number using App Script */
               Cross JOIN
   UNNEST(
   (GENERATE_ARRAY(4/* number of units */, 4/* number of units */ ) ))
        AS Num_of_Units /*: Change Number using App Script */
    --   order by S_Activity_ID
         )
  ,
  --######################################################################
  -- The sub query Modes below reads the user parameter inputs for each
  -- activity admitted modes
  --######################################################################
    Modes AS (
    SELECT * from `ESDTCT_Database.Modes_bq`
    )
    ,
  --######################################################################
  -- The sub query Experiment_run below prepares the activity modes input
  -- to repetitive activities.
  --######################################################################
    Experiment_run AS (
    SELECT TRADE_OFF_RUN ,
    STRING_AGG(Cast( CHROMOSOME as string)) CHROMOSOME
    FROM `ESDTCT_Database.Experiments` rec
      Cross JOIN
    UNNEST((GENERATE_ARRAY(1,4/* number of units */) ))
    AS REP_UNIT /*: Change Number using App Script */
    group by TRADE_OFF_RUN
    )
    ,
  --######################################################################
  -- The sub query SETUP below joins the data input tables and assign a
  -- random cost and duration values based on the trinagular probability
  -- distribution function.
  --######################################################################
    SETUP as (
    SELECT TRADE_OFF_RUN, SIMULATION_RUN, S_Activity_ID ,
    U_Activity_ID, P_Activity_ID,
    REP_UNIT, Probability_of_Occurrence , Mode_Option,
    [0,0,0,0,0] IDLE_Combination_String , "" Shift_Combination_String,
    0 as Interruption,
    Activity_Type, P1, P2, P3, P4, P5, P6, P_Rep, Ladder_Seq, null START,
    null FINISH, 0 JCL, 0 IDLE,
    TriDist_Sampling( P_Dur, MinDur, MLDur, MaxDur) as RDUR,
    TriDist_Sampling( P_FC, MinFC , MLFC , MaxFC ) as RFC,
    TriDist_Sampling( P_VC, MinVC , MLVC , MaxVC ) as RVC,
    TriDist_Sampling( P_OHC, MinOHC , MLOHC , MaxOHC ) as ROHC,
    0 as TC , 0 as FC, 0 as VC, 0 as OHC, 0 as PC,  0 as BC,
    CHROMOSOME , Lable,  Num_Of_Modes  ,Opt_Idle_Time
    FROM
    Experiment_run , UNNEST(SPLIT( CHROMOSOME )) oid WITH OFFSET tid
    JOIN Tasks t1 ON t1.P_Activity_ID = tid + 1
    JOIN Modes t2 ON t2.Activity_ID = t1.U_Activity_ID
     and  CAST(REPT_UNIT AS STRING)  =
     cast(Ceil ((tid + 1)/5/* No of Activities in a Typical unit */)
     as string)
    AND CAST(Mode_Option AS STRING) = oid
```

```
    -- order by TRADE_OFF_RUN, S_Activity_ID
      )
      ,
    --######################################################################
    -- The sub querys JCL_Calc and JCL_Calc2 calls the defined temporary
    -- functions to determine the supreme chromosome solution.
    --######################################################################
      JCL_Calc as (
        SELECT  rec.*
      FROM (  SELECT ARRAY_AGG(TO_JSON_STRING(t)) AS data
      FROM SETUP as t
      GROUP BY TRADE_OFF_RUN
      ) as t
     , UNNEST(Simulation_and_JCL_Calculation_Function(data)) AS rec
     )
        ,
      JCL_Calc2 as (
      SELECT rec.*   FROM (
      SELECT
      ARRAY_AGG(TO_JSON_STRING(t)) AS data
      FROM JCL_Calc as t
      ), UNNEST(Normalization_of_Frontier_Solutions_Function(data)) AS rec

      -- Optimization Method
    -- for  Joint cost,schedule,Interruption minimization Change the
    --        Order by to: Vector, FINISH, TC
    -- for  cost minimization Change the Order by to: TC , FINISH
    -- for schedule minimization Change the Order by to: FINISH, TC

    Order by FINISH, TC  --- 1
    Limit 1
      )
    --######################################################################
    -- The final query below calls the crew idle time optimization function
    -- the final result is the supreme chromosome with the optimal crew
    -- interruption times.
    --######################################################################
      SELECT  1 as GENERATION, rec.* Except (Vector)
      FROM (
      SELECT ARRAY_AGG(TO_JSON_STRING(t)) AS data
      FROM JCL_Calc as t
      GROUP BY TRADE_OFF_RUN , SIMULATION_RUN

      ) as t
     , UNNEST(Crew_Idle_Time_Optimization_Function(data)) AS rec

    -- Optimization Method
    -- for Joint cost,schedule,Interruption minimization Change the Order
    --              by to: Vector, FINISH, TC, Interruption ,
    -- for cost minimization Change the Order by to:
    --                            TC , FINISH , Interruption
    -- for schedule minimization Change the Order by to:
    --                            FINISH, TC, Interruption
    -- for Interruption minimization Change the Order by to:
    --                            Interruption , FINISH, TC

    Order by FINISH, TC  --- 2
      Limit 1


```