Project Schedule Compression Considering Multi-objective Decision Environment

Nazila Roofigari-Esfahan

A Thesis
In the Department
of
Building, Civil and Environmental Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of
Master of Applied Science (Building Engineering)
Concordia University
Montreal, Quebec, Canada

December 2011

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Nazila Roofigari-Esfahan

Entitled: Project Schedule Compression Considering Multi-objective Decision Environment

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

Master of Applied Science (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. T. Zayed

____ Examiner
Dr. M. Chen

____ Examiner
Dr. Z. Zhu

____Supervisor
Dr. O. Moselhi

Approved by ____
Chair of Department or Graduate Program Director

Date ____
Dean of Faculty

ABSTRACT

Project Schedule Compression Considering Multi-objective Decision Environment

Nazila Roofigari-Esfahan

This research aims to present a new method to circumvent the limitations of current schedule compression methods, which reduce schedule crashing to the traditional time-cost trade-off analysis, where only cost is considered. In this research the schedule compression process is modeled as a multi-attributed decision making problem in which different factors contribute to priority setting for activity crashing. For this purpose, a modified format of the Multiple Binary Decision Method (MBDM) and an iterative crashing process are utilized. The developed method is implemented in Visual Basic 2010 environment, with a dynamic link to MS-Project to facilitate the needed iterative rescheduling of project activities. To demonstrate the use of the developed method and to highlight its capabilities, 3 case examples drawn from literature were analyzed. When considering cost only, the generated results were in good agreement with those generated using the Harmony Search method, Genetic Algorithms and iterative crashing process used in original examples, particularly in capturing the project least-cost duration. However, when other factors in addition to cost were considered, as expected, different project least-cost and associated durations were obtained.

The novelty of the developed method lies in its capacity to allow for the consideration of a number of factors in addition to cost. Also through its allowance for possible variations in the relative importance of these factors at the individual activity level, it provides contractors with a number of compression execution plans and assists them in identifying the most suitable plan. Accordingly, it enables the integration of contractors' judgment and experience in the crashing process and permits consideration of different project environments and constraints.

AKNOWLEDGEMENT

First and foremost, I am very grateful to God for keeping me blessed and granting me the ability to follow through and achieve my goal.

This thesis would not have been possible without the generous support and encouragement of my supervisor, Professor Osama Moselhi. I have been truly lucky to have him as my mentor. He treated me as his daughter, and guided me with his keen advice, critical insight, and confidence throughout the whole dissertation process. He also inspired me with his wisdom, and his patience. I would like to express my sincerest appreciation and gratitude to him for his guidance, motivation, unconditional help and support, and for always believing in me during the last year.

I would like to take this opportunity to thank all the individuals, colleagues and friends who assisted me during this research project. I would like to acknowledge Professor Mohamed Al-Hussein form University of Alberta and Professor Mohamed Attalla from Ryerson University for helping me in distributing my questionnaire within construction firms of their region. I would like also to thank all the contractors and construction management professionals from SNC-Lavalin, Hatch, Magil Construction and other construction management firms who kindly dedicated their precious time to participate in my questionnaire survey.

Last but not least I would like to thank my parents Mohammad Mehdi and Shahnaz for their unconditional love and support and my sisters Negar and Nooshin for always being there for me. I would like to express my deepest gratitude to my family for teaching me how to live.

DEDICATION

This thesis is dedicated with love, admiration, and respect

To my dear father Mohamad Mehdi,

my kind mother Shahnaz,

and my beloved sisters Negar and Nooshin;

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

1. CHAPTER ONE: INTRODUCTION

1.1. Schedule compression

Despite project managers’ effort to deliver projects within targeted due dates and budgets, delays and cost overruns occur as a routine phenomenon at many construction projects. Considering the fact that construction industry is a highly competitive industry, these delays and cost overruns can cause inevitable disputes and tensions between owners and contractors. Considering time value for money, time saving can contribute to projects’ success and improve their expected profits. In other words, contractors and owners usually aim to establish the delicate balance between the overall cost of a project and its duration. Consequently, the topic of schedule compression, also known as project time reduction, least-cost expediting, optimized scheduling, scheduling with time constraints and time-cost trade-off has been introduced in the literature and has widely been studied.

Time-cost trade-off analysis, as described in literature, typically leads to rational estimation of project least cost duration, which is not necessarily identical to the original contractual duration. As a result, Contractors and project managers often encounter the need to expedite the execution of the project under their responsibility to meet targeted milestones imposed by owners and/or to make up for lost time due to delays experienced during execution of the project. This need also can arise from the fact that “originally estimated project duration is not necessarily the least time solution nor is the least cost schedule for the project, in spite of the fact that each activity within the project was originally planned to be done in the most efficient manner" (Hinze 2008).

According to the Construction Industry Institute (CII 1988) schedule compression is referred to as the shortening of the required time for accomplishing one or more of the engineering, procurement, construction or start-up tasks (or a total project) to serve one of the three purposes: (1) reducing total design-construction time from that considered normal; (2) accelerating a schedule for owner convenience; and (3) resolving lost time after falling behind schedule. As such, both owners and contractors may will to compress or accelerate the schedule of a construction project because of below primary reasons:

Contractors usually tend to perform schedule compression in order to: a) meet imposed contractual times b) benefit from early completion bonus and avoid related contractual penalties mentioned in contract documents c) recover from delays and/or loss of productivity experienced during execution of the project and d) avoid adverse weather conditions; Owners, on the other hand, may order accelerated delivery of their under-construction projects because of : a) monetary considerations such as project financing e.g. to meet prescribed fiscal requirements, b) to minimize the effects of change orders on project schedules, c) to recover from delays for which they were the main source such as late delivery of material and/or equipment, d) to minimize project total cost, because of stockholder pressure and e) or simply because of their desire to complete the project earlier to address market demands in case of the development of a new product or service by the owners' organization that needs to get to market as soon as possible due to rising loss-of-opportunity costs (CII 1988, Noyce and Hanna 1998). Consequently, duration of projects may have to be shortened in order to meet these schedule constraints.

Hence, expediting respective duration of projects is becoming a challenging task in management of Engineering, Procurement and Construction (EPC) projects. It aims to shorten project schedules without changing project scope of work, in order to meet schedule constraints and objectives. As such, reducing duration of construction projects while imposing least additional direct cost to them, has always been of interest to researchers and professionals alike.

1.2. Motivation for the study

When reviewing literature on project compression, certain issues appeared to have been left unanswered. First, despite the fact that contractors and project managers frequently resort to schedule crashing, as described in previous section, still there is no commercially available software that they can use to perform this important management function. This left them to relay on their own judgement and intuition. Second, although various methods are proposed in the literature to solve the schedule compression problem, also called as time-cost trade-off problem, there is limited use and uptake, if any, of these methods by contractors and project managers, in practice (Sears, et al. 2008, Moselhi and Roofigari 2011-a). This is likely because of the fact that in all these methods, schedule compression is still reduced to some form of time-cost trade-off analysis, where schedule compression is performed based on cost only. In other words, these methods do not take into account any factor beyond the additional direct cost required for acceleration of project activities.

However, early discussions with construction management professionals revealed that it is practically more feasible to consider other factors such as resource availability, complexity and logistics of the work involved, contractor’s leverage on the sub-contractor who is deemed more capable of performing the accelerated work and the risk associated with crashing of each activity in addition to cost, in queuing activities for crashing. This can be the case, particularly when owners request contractors to accelerate the delivery of their projects. In this case, cost might not be the major factor to be considered, as factors such as complexity of the work involved and availability of required resources will be of essence and, accordingly, gain more importance in setting priorities for activity crashing. Even in the case where acceleration is performed to recover contractor’s own delays, cost seems not to be the only factor to be considered, but may be assigned more importance than in the case referred to above. In the latter case, factors such as cash flow constraints and the risk associated with the work involved can be of more importance.

Further, as risk exists in all phases of EPC projects, its impact on the crashing process should not be ignored. While the main purpose of schedule compression is to perform such acceleration while bringing less possible additional direct cost to the project, the added cost associated with such crashing should account for the risk involved. The risk associated with that direct cost should be identified and quantified to help generate a realistic crashing plan.

1.3. Scope and objectives

The main objective of this research is to study schedule compression of engineering, procurement and construction project with a focus on their execution phase. And develop a structured method that meets the challenging requirements of performing realistic schedule compression of construction projects. For this purpose, a set of sub objectives are defined for this research to achieve above stated objective:

1. To study current industry practice,
2. To perform a comprehensive literature review,
3. To model the uncertainty encountered in the schedule compression process,
4. To model the schedule compression problem in a flexible and practical manner; that may allow for the consideration of multiple objectives, and
5. To implement the developed method in a proof of concept software application.

1.4. Organization of the thesis

Chapter 2 presents a review of the current literature regarding schedule compression problem along with their advantages and limitations. Chapter 3 mainly reports the finding of the questionnaire survey that was conducted to find out the real decision environment usually considered by contractors and project managers when planning to accelerate respective duration of their projects. A copy of the questionnaire of the survey is available in both English and French formats in Appendix II. Chapter 4 presents the proposed method and its components along with in detail computational procedures as well as its limitations and recommendations for future work. As well risk modeling is presented in this chapter. The computer implementation along with its validation through use of 3 numerical examples drawn from literature is described in chapters 5 and 6, respectively. Finally, chapter 7 presents a summary of the study and concluding remarks.

2. CHAPTER TWO: LITERATURE REVIEW

2.1. General

In general, success of any project could be translated to delivering its deliverables within specified time, budget and quality. Considering this fact and since any change in project time could cause its cost to differentiate from what is primarily estimated, significant attention should be given to initial scheduling of the project.

As such, essential terms of project scheduling i.e. project duration and cost should first be completely understood. Stating in brief, each activity that makes up a project have an estimated duration coming along with an associated cost, both calculated in initial scheduling stage of project, i.e. before project's execution starts. Project total duration is usually calculated using critical path method (CPM) forward and backward calculations. Subsequently, the final project cost is the summation of the cost required to complete all its activities plus lateness penalties that may be assessed if the project is not completed by the specified completion time.

Scheduling of EPC projects while satisfying all their constraints and project-dependent conditions, has always been the most challenging task for contractors and project managers of these projects. Especially when it comes to the need for accelerating durations of construction projects, as it highly impacts project cost as well as its productivity, a careful attention should be placed on how to perform this acceleration. This reduction in project duration can be carried out at strategic or tactical levels (see Figure 2.1). At the strategic level, first, the project's job logic is revised to see if the project can be done using other order of activities, i.e. by removing or changing precedence relations between project activities (Liberattore and Pollak 2006, Sakellaropoulos and Chassiakos 2004) .

But the schedule compression problem does not stop at revising the job logic and other factors at the strategic level such as the selected project delivery systems that can have an impact on schedule compression, i.e. by using fast-tracking instead of traditional or phased construction (Moselhi and Alsheibani 2011). In other words, projects can be accelerated using phased or fast-tracked project delivery systems. This way, the design and construction phases of projects are overlapped, or activities are planned to be carried out concurrently instead of in serial. Figure 2.2 schematically illustrates how using phased or fast track project deliveries result in reduction in project duration.

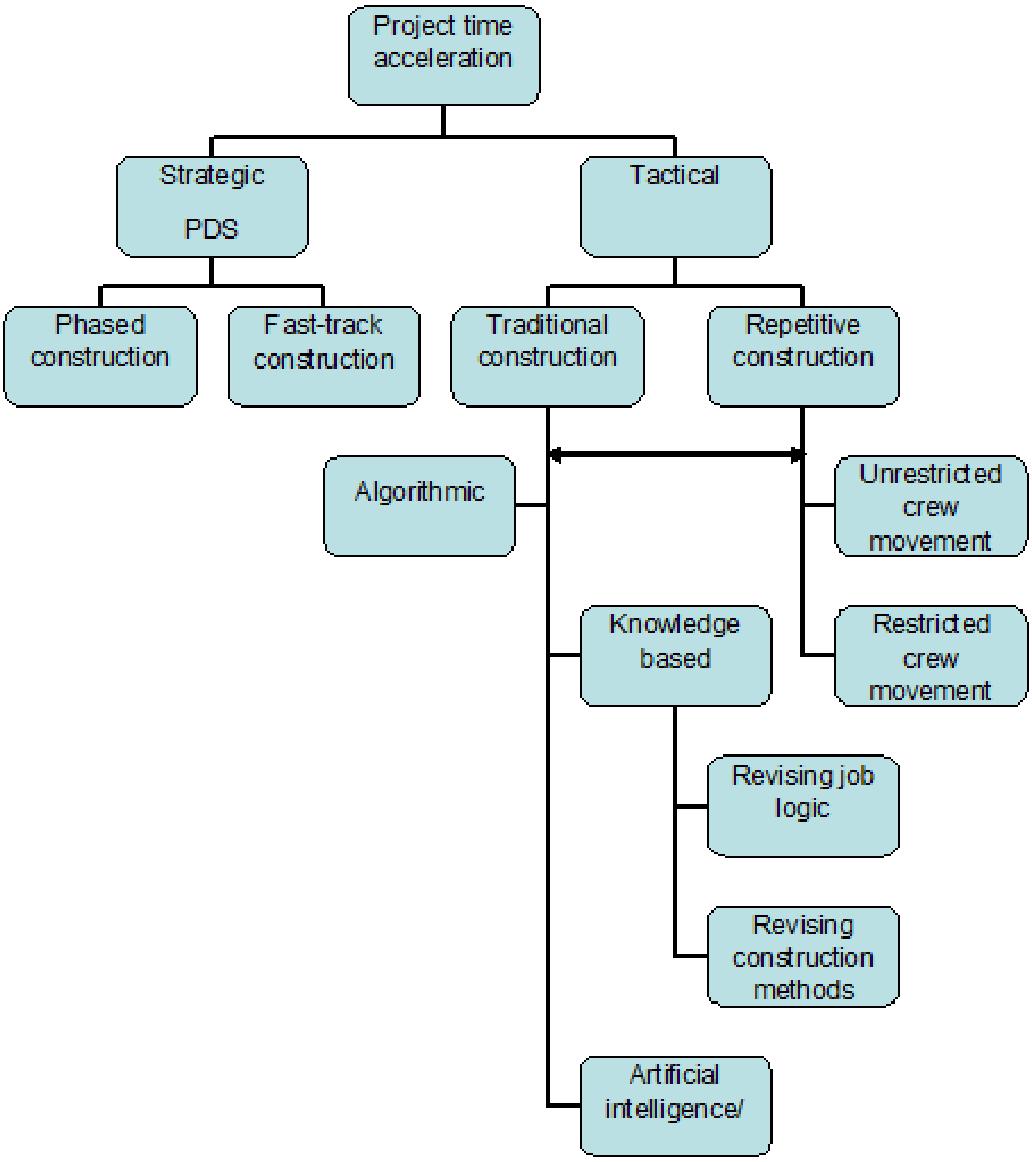


Figure 2.1: Project acceleration strategies (Moselhi and Alsheibani 2011)

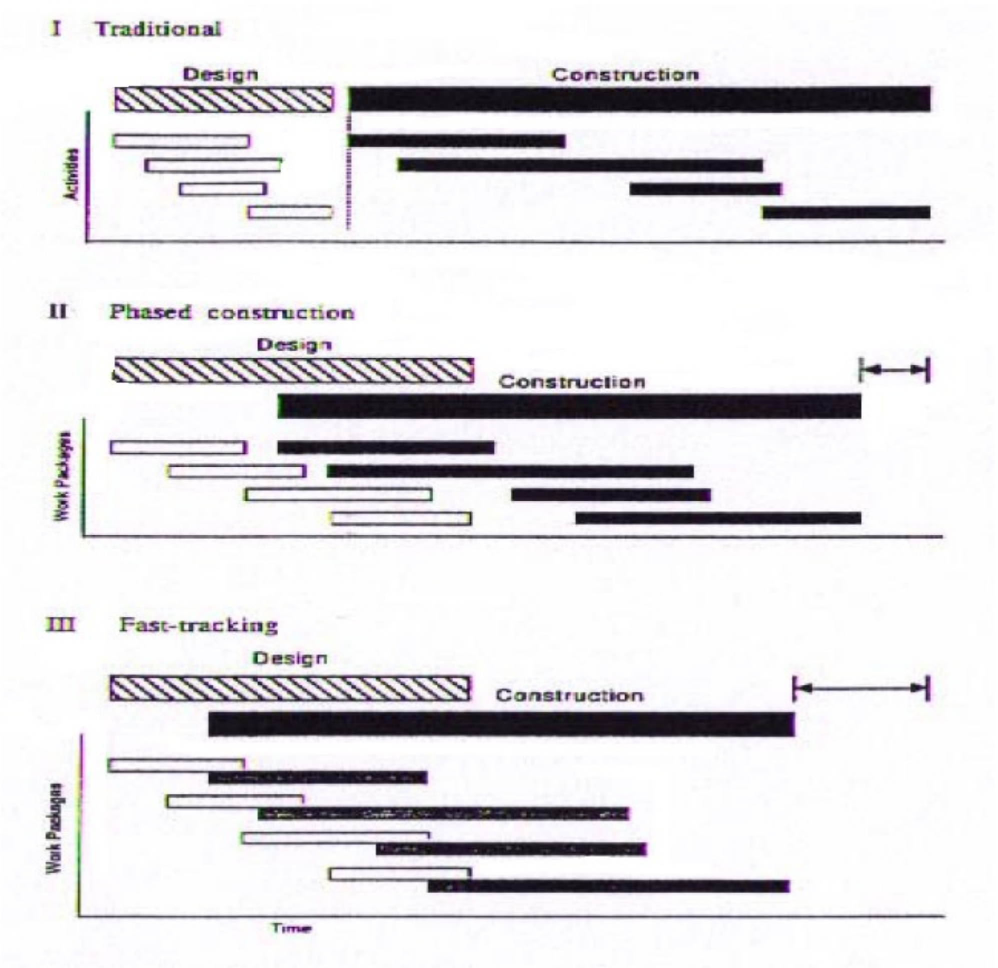


Figure 2.2: Effects of different project delivery systems on project acceleration (Fazio, et al. 1988)

The majority of the methods that is available in the literature focus on accelerating construction projects at the tactical level. These methods usually tend to reduce duration of construction projects 1) by crashing the duration of activities that form the project network's critical path or 2) by finding out the best set of activity crashed durations (i.e. the Pareto front set) that minimizes project duration while not exceeding project's pre-set budget. This thesis mainly focuses on accelerating construction projects at the tactical level. Accordingly, a review of the methods available in the literature on project schedule acceleration is presented in this chapter.

Further, because in real world project activities are subject to considerable uncertainty, efficient management of impacts of such uncertainty on the process of schedule compression has always been a concern for contractors and project managers of EPC projects. As such, the uncertainty associated with the compression process and its impacts on projects' duration and cost should carefully be undertaken.

In this chapter a quick introduction to project scheduling under constraints in general and review of recent literature on special techniques for scheduling under time constraints in particular are presented. Moreover, different available models for solving time-cost trade-off problem along with their assumptions and limitations are provided. A review of current literature on the uncertainty associated with the crashing process is also presented. Finally, the existing gaps in the literature on schedule crashing problem that are the subject of current study, are introduced and are going to be further explained in following chapters.

2.2. Scheduling under constraints

Generally, project time and resources are the two most important components of a construction project that are subjected to constraints. Considering these constraints is required not only in the initial scheduling stage, but also it should dynamically be considered during execution phase of construction projects. Otherwise, generated schedules are bounded to be unrealistic; since some resources are highly limited and also each project has timely deadlines that cannot be passed (Kim and de la Garza 2005). To address this problem, the project scheduling literature largely concentrates on the resource feasible schedules that optimize project duration.

Accordingly, many resource constrained scheduling (RCS) techniques are introduced in the literature to apply resource constraints to project schedules. These methods create schedules that contain resource dependencies between activities as well as their technological relationships (Kelly 1963, Moder, Phillips and Davis 1983, Aslani 2007, Finke 2010). Kim and de la Garza (Kim and de la Garza 2005) have further upgraded resource-constrained scheduling methods by recalculating late start and finish times of activities through a backward pass, considering both technological and resource links. This way, their method recalculates activities' total floats after their respective resources have been allocated and creates resource links if these total floats are not available due to resource constraints.

As explained previously, time is another project component that can highly be bounded to constraints. There are delays occurring frequently in construction projects, while most of the times no extra time is awarded by owners. Also owners might order accelerated delivery of their under-construction projects. As such, project scheduling under time constraints has always been of interest to academics, and has widely been studied. The rest of this chapter is devoted to scheduling under time constraints, also called as project schedule compression.

2.3. Time-cost trade-off

As stated earlier in introduction chapter, the process of accelerating completion of construction projects is also referred to as time-cost trade-off. It was originally developed by (Kelly 1961) after introduction of critical path method (CPM) for planning, scheduling and controlling projects, in late 1950's (Rehab 1986); It aims at establishing the delicate balance between the overall cost of a project and its duration, to achieve the desired overall project objectives. In literature, the process of schedule compression is also referred to as (Moselhi 1993, Evensmo and Karlsen 2008):

- Project time reduction
- Least-cost expediting
- Project compression or schedule compression
- Least-cost scheduling
- Optimized scheduling
- Scheduling with time constraints
- Project acceleration
- Project time crashing or schedule crashing

This duration reduction results in increasing the total direct cost of projects and in decreasing project indirect cost. It should be noted that direct costs are those costs related to putting the facility components in place, containing cost of all resources directly used in execution of project (e.g. materials, labor, equipment and subcontractors); likewise, indirect costs are the costs generally incurred whether or not productive work is actually accomplished, (e.g. office personnel, office services and supplies, site supervision, etc) and should be considered as long as the project is underway (Hinze 2008). In the process of schedule compression, additional resources are used to reduce the original durations of individual activities, which give rise to progressive increase of the project direct cost and steady reduction in the project indirect cost as is shown in Figure 2.3. Accordingly, the resulting relationship between project total cost (direct plus indirect costs) and its duration provides project teams with useful information. Because of the above mentioned changes in project direct and indirect costs over projects' shortened duration, the project total cost versus duration curve typically depicts a valley, which identifies the optimum project duration and its associated cost, i.e. the project's least cost duration. In other words, this curve includes the project optimum duration, which coincides with the project least total cost, as well as the total additional direct cost required to compress project's schedule to any targeted duration.

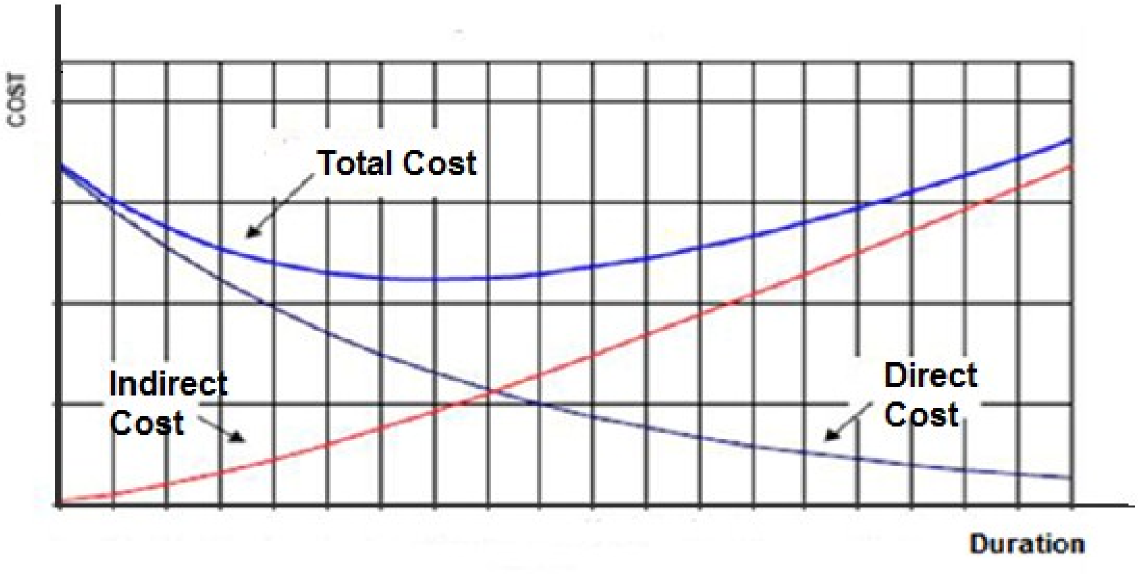


Figure 2.3: Project time-cost relation

Minimization of such increased direct cost and finding the point of least-cost duration has always been of interest to researchers and professionals alike.

Consequently, because of the importance of schedule compression process in successful management of engineering, procurement and construction (EPC) projects, considerable studies were carried out to develop methods to solve this problem. This resulted in the development of a number of models to determine the least-cost project duration and/or project least-cost associated with any targeted duration e.g. (Kelly 1961, Elmaghrabi 1993, Yang 2005, Geem 2010, Evensmo and Karlsen 2008, Ezeldin and Soliman 2009, Cheng, Huang and Cuong 2011)

Accordingly, (Noyce and Hanna 1998) have divided schedule compression to planned schedule compression which is planned before construction starts; and unplanned, that is a result of unexpected changes to planned scope of work and in the majority of the cases the need for project acceleration is due to the later. In this study, however, we focus mainly on unplanned needs for schedule compression. In other words, the proposed method deals with the situations where delays have been already occurred. In such situations, contractors and project managers will find themselves trending beyond their committed deadline date and are forced to compress schedules of their projects; since in 75 percent of the cases, no extra time is granted by owners (Noyce and Hanna 1998, Chang 2004). Also, it is applicable to the cases where during execution phase of the project, owners introduce changes in scope and/or pre-scribed project milestones. In either of these cases, contractors resort to use different compression strategies to get their projects back on track. The sooner these decisions are made, the project is more likely to be succeeded to get back on track; since in early stages of project execution there are many options to solve the problem, but toward the end, available choices dwindle.

It should be kept in mind that based on Yerkes-Dodson Law, “performance increases with cognitive arousal, but only to a certain point. Performance, however, decreases when levels of arousal become too high” (Lee 2008). Considering this fact in project schedule compression will be translated as existence of a level of schedule pressure at which performance is at a maximum. In other words, pressurizing an activity less or more than this level will lead to reduction in performance and productivity. As such, although contractors and owners can benefit from the results of accelerating a project, that can be earlier entrance of their product to the market for owners, and avoiding penalties and/or gaining early completion bonuses for contractors, productivity and quality may be sacrificed in this acceleration process. To quantify the impact of schedule compression on labor productivity and to reduce such impact, a few works are presented in the literature (e.g. Noyce and Hanna 1998, Thomas 2000, Chang 2004). Noyce and Hanna also reported the factors which has the most effect on this loss of productivity for both planned and unplanned schedule accelerations.

To start the compression process, before applying any of the project compression strategies, the relations between activities direct cost and their respective durations should be determined. In other words, firstly, it should be determined how each activity’s direct cost is changed over its crashed duration. It should be noted that these time-cost relations mainly are assigned based on contractors’ own judgment and experience or are dedicated from historical data available on a particular activity. A continuous relationship represents an activity that can be completed at any time–cost combination along the curve. In contrast, a discrete time–cost relationship appears when only specific and distinct duration values are feasible and is more appropriate than a continuous one to model engineering project activities. For example, when dealing with delivery of material and equipment that should be shipped from overseas, not any point between normal duration and completely crashed durations for an activity will be feasible. In this case, activity's direct cost versus duration only contains discrete points that are the particular feasible delivery dates that are possible (See Figures 2.4 and 2.5). It should be noted that an activity might also have hybrid of continuous and discrete time-cost relations.

Traditionally, a linear continuous relation is assumed between activities time and their respective direct cost. In this kind of relation, by decreasing activity duration from its normal duration to its crashed point, i.e. the point in which activity reaches its most compressive duration and cannot be further crashed, its associated direct cost will increase linearly (see Figure 2.4). This kind of time-cost relation lets the problem to be formulated as a linear programming problem which leads to exact solutions (Liberattore and Pollak 2006) and as such has frequently been used to model the compression problem (Ammar 2011). However, this assumption might not be realistic enough for a number of construction activities. Subsequently, to better present each activity's special direct cost change over its crashed duration, a number of time-cost relations are introduced in the literature (Ahuja, Dozzy and AbouRizk 1994, Hinze 2008, Moussourakis and Haksever 2004, Meredith and Mantel 2006, Tareghian and Taheri 2006):

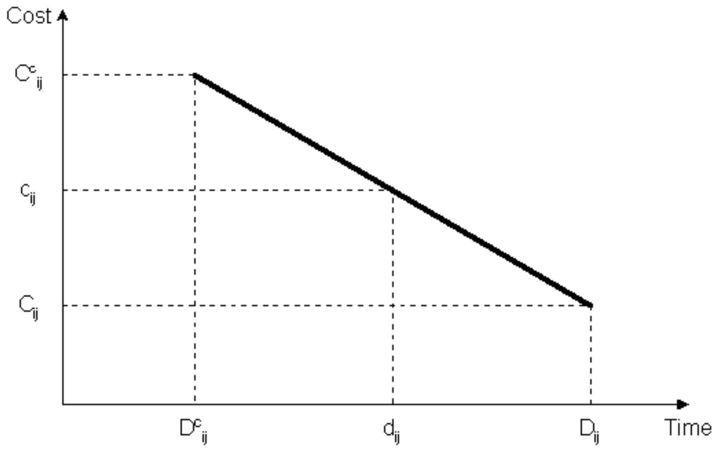


Figure 2.4: Linear activity time-cost relation

- Multi-linear (Piecewise linear and linear with gaps in between which could be attributed to the use of different technologies). This kind of time-cost relation is also used frequently in the methods presented in the literature since the linearity can approximate the true cost variation without much error. Also, linear relationships allow the application of linear programming (LP) techniques, which are efficient and can guarantee a global optimal solution and finally, nonlinearity of time-cost relationships can be circumvented by piecewise linearization (Yang 2005).
- Discrete
- Curve –linear, concave or convex (could be converted to piecewise linear) and etc.

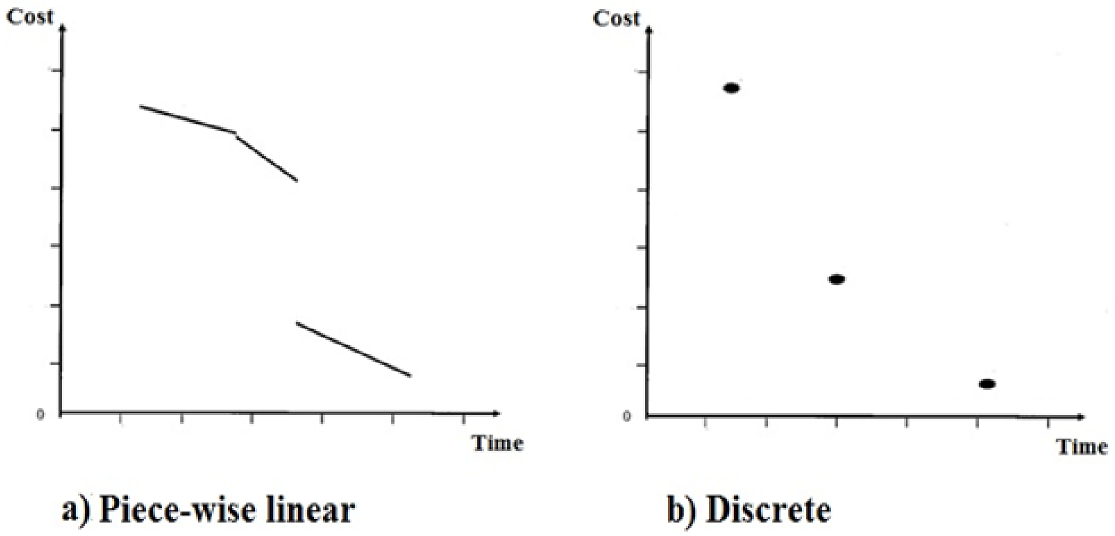


Figure 2.5: a) Piece-wise linear and b) discrete activity time-cost relations

After each activity's time cost relations has been determined, the process of schedule compression should be started. Typically, the time-cost trade-off problem in the literature is expressed in one of the two ways (Liberattore and Pollak 2006, Yang 2007): to minimize the time required to complete the project, also called as make-span, subject to a budget constraint (the budget problem) or to minimize the cost required to finish the project subjected to time constraints (deadline problem). According to Vanhoucke and Debels (Vanhoucke and Debels 2007) there exists also a third objective for solving time-cost trade-off problem that is to construct the complete and efficient time/cost profile over the set of feasible project durations.

Contractors are then to select a method to find the best activity or the best set of activities that should be accelerated to optimize project schedule. This is usually done by means of using additional resources such as working overtime and double shifts (additional hours from existing workers), bringing expert crew or subcontracting the work, utilizing more productive equipment, using different construction methods (Senouci and El-Rays 2009, Liu, Burns and Feng 1995, Mitchell 2005). Evensmo and Karlsen (Evensmo and Karlsen 2008) also did have a closer look at break down of the cost needed to accelerate the activities involved.

As such, different methods are proposed in the literature to address these three problems stated above. These methods can mainly be divided to optimization and heuristic methods as will be gone through in the next sections.

2.3.1. Optimization methods

Various optimization models are presented in the literature to solve the time-cost trade-off problem. These methods can mainly be categorized to two categories; first group are optimization methods that use different mathematical and artificial intelligence techniques to solve the time-cost trade-off problem. These methods provide good optimum or near optimum solutions but are difficult to apply and require considerable computational effort (Moselhi 1993, Que 2002). A review of these two categories of optimization methods that are presented in the literature is shown in following sections.

2.3.1.1. Mathematical programming

Mathematical approaches convert the project time-cost trade-off problem to mathematical problems. In other words, they convert project CPM network and its precedence and time-cost relationships into constraints and objective functions. Subsequently, they use linear programming, integer programming, dynamic programming or a hybrid of these methods to optimize the trade-offs between construction time and cost (Elmaghrabi 1993).

Kelly (Kelly 1961) was the first to formulate the time-cost trade-off problems using linear programming. He assumed linear relationship between time and cost for construction activities. His objective then was to find “least costly schedule for any given feasible earliest project completion time”. His work paved the way for other scholars to apply different mathematical optimization methods to solve time-cost trade-off problem. Liu et al. (Liu, Burns and Feng 1995) have used a hybrid of linear programming with integer programming, to not only getting minimum direct cost for the project efficiently, but also to get exact solutions; since only integer durations are deemed feasible in this domain. Likewise, Moussourakis and Haksever (Moussourakis and Haksever 2010) applied mixed integer linear programming models to “assist project managers in making decisions to compress project completion time under realistic activity time-cost relationship assumptions”. As such, their model is designed to consider highly complex, but realistic continuous activity time-cost relationships. These types of time-cost relationships are then approximated with piece-wise linear relations and are used to model three time-cost trade-off problems. The main model that focuses on completing the project as early as possible and under a crash budget constraint; and two other versions of the main model that deal with the two main time-cost trade-off concerns: the budget and deadline problem as stated previously. However, integer programming requires a lot of computational effort once the number of options to complete an activity becomes too large or the network becomes too complex (Feng, Liu and Burns 1997).

However, the basic assumption made in linear programming methods that is the consideration of linear cost-duration relationships for project activities, i.e. based on the normal and crashed points only, makes the solutions obtained through these methods to be only usable as approximate starting points rather than actual optima. These methods also fail to solve those with discrete time-cost relationships (Feng, Liu and Burns 1997).

Similarly, Deckro et al. (Deckro, et al. 1995) have Developed a quadratic programming model as well as a goal programming formulation. This non-linearity assumption made in their method avoids the piecewise approximation used in Moussourakis and Haksever's method. However, although these methods assist project managers by providing them with the possibility of realistic activity time-cost relationship assumptions, still the continuous nature of the nonlinear time/cost trade-offs may not represent reality. This way, these methods discard non-continuous relations (e.g. discrete functions) that are very common in construction projects. Further, like all other available methods, they consider cost as the only factor in the process of schedule compression.

Elmaghrabi (Elmaghrabi 1993) used dynamic programming to minimize the project completion time while also allocating required resources to project activities. Dynamic programming is the process of making a sequence of inter-related decisions. The procedure starts with a small portion of the original problem and finds the optimal solution for this smaller problem. It then enlarges the problem finding the current optimal solution from the preceding one until the original problem is solved entirely. However, because of characteristics of dynamic programming such as its number of functional constraints, its applicability to complicated multi-variable problems such as large scale construction projects becomes limited.

These mathematical algorithms mainly are used to obtain the optimal solutions for the time-cost trade-off problem. The main advantages of mathematical approaches include their efficiency and accuracy. However, as stated previously, formulating constraints and objective functions is time-consuming and prone to errors (Ammar 2011). Furthermore, having mathematical programming knowledge is necessary to formulate these mathematical models correctly, while few construction planners are trained to perform this type of formulation, especially for large networks. These models can increase in size very rapidly and large problems may not be computationally tractable in reasonable time frames. Because of these reasons, the application of these models is limited as they are not efficient in optimizing large-scale construction projects.

2.3.1.2. Near optimum solutions

With the fast growth in computer technology and advances in artificial intelligence applications, computational optimization techniques were used more and more to solve the schedule compression problem. In contrast with mathematical methods, these approximate methods perform well over a variety of problems. These methods are simple and easy to use, but may lead only to near optimum solutions (Liu, Burns and Feng 1995, Ammar 2011). Approximate methods utilize different techniques to carry out the schedule compression process such as Genetic Algorithm (Que 2002, Zheng, Thomas and Kumaraswamy 2004, Cheng, Huang and Cuong 2011), analogy with the direct stiffness method for structural analysis (Moselhi 1993)Particle Swarm Optimization (Yang 2007),Harmony Search (Geem 2010), and iterative crashing process (e.g. Ahuja, Dozzy and AbouRizk 1994, Meredith and Mantel 2006).

Genetic Algorithms (GAs) are search algorithms first developed by Holland (Holland 1975). These algorithms use the mechanics of natural selection and genetics to search through decision space for optimal solutions. In evolution, the problem that each species is dealt with is searching for beneficial adaptations to the complicated and changing environment. In other words, in order to survive in the living world, each species should change its chromosome combination. “In GAs, a string represents a set of decisions (chromosome combination), a potential solution to a problem. Each string is evaluated on its performance with respect to the fitness function (objective function). The ones with better performance (fitness value) are more likely to survive than the ones with worse performance. Then the genetic information is exchanged between strings by crossover and perturbed by mutation. The result is a new generation with (usually) better survival abilities. This process is repeated until the strings in the new generation are identical, or certain termination conditions are met”. (Feng, Liu and Burns 1997)

During the last decade, GAs has been widely used by researchers as a novel approach for solving construction planning problems. They repeated showing success in attacking large size, complex problems. Further, as they do not rely on heuristic rules, they are deemed more robust in tackling such problems (Que 2002). As such, different authors used GAs to solve the time-cost trade-off problem (e.g. Feng, Liu and Burns 1997, Hegazy 1999, Que

2002, Zheng, Thomas and Kumaraswamy 2004, Cheng, Huang and Cuong 2011).

In these models, a solution to the time-cost optimization problem is simply a specific combination of possible durations for the activities. Hence, only activities that are crashable will take part in optimization. Solutions are represented as chromosomes: Each box (gene) in the chromosome string corresponds to an activity. There are as many genes in the chromosome as there are activities. The sequence of the activities in the chromosome corresponds to the sequence of the activities in the project activity network. The content of the box corresponds to the duration of its corresponding activity. Each solution, therefore, defines a certain set of gene values for its chromosome. The generated GAs model minimizes the total project cost as an objective function while other project specific constraints on time and cost are also accounted for (Hegazy 1999).

Senouci and El-Rays (Senouci and El-Rays 2009) used Genetic Algorithms and presented a robust multi-objective method. Their method aims at generating and evaluating optimal construction resource utilization and scheduling plans that establish optimal trade-offs between project time and profit. As such, their method searches for the best set of activity crew formation that minimizes project time while maximizing contractors' profit.

Recently, Cheng et al. (Cheng, Huang and Cuong 2011) have implied K-Means and Chaos clustering approach to Genetic Algorithms to assure optimality of the results generated for time-cost trade-off problem through application of GAs. However, although application of GAs leads to generation of good near optimum solutions, because they consider all the activities within projects' network, and not only critical ones, their application will be time consuming for large scale projects.

Yang (Yang 2007) has also developed an evolutionary computation technique, particle swarm optimization, to solve project compression problem. His method aims at developing an optimization algorithm to find the complete time-cost time cost profile (called Pareto front set) considering all types of activity cost function (i.e. linear, Piece-wise linear, discrete, etc.). It should be noted that so called Pareto front is considered as a non-dominated solution in the solution space which is not dominated by any other solutions in that space. Hence, these solutions have the least objective conflicts of any other solutions, providing the best alternative for decision making.

A classical Particle Swarm Optimization (PSO) procedure maintains a population of individual particles, each of which represents a decision vector (or solution) in the search space. During the optimization process, every particle is moved in a multidimensional search space toward its own best experiences (personal best) and also toward the best individual found so far by the entire swarm (global best). Similar to GAs, a PSO algorithm employs the concept of population and a measure of performance (fitness) to conduct the iterative search. But while GA evolves the entire population as a group as chromosomes sharing information with each other, PSO moves each particle based on its best experience and only the best among the entire swarm, not all the other particles (Yang 2007).

The Harmony search method (Geem 2010) is one of the other approximate methods that has been used to solve the time-cost trade-off problem. It tends to find the best set of project activity alternatives through generation of harmony memory matrix and updating this matrix by replacing each activity's alternative with another alternative that better satisfies the objective of gaining less total project cost. The analysis typically ends with detecting the Pareto set that is the best set of activities' alternatives which minimizes project cost. All these methods, which search for the Pareto front set within project network, consider all activities within project and not only critical ones. This renders the application of these methods to be time consuming for large scale project, although they may lead to good near optimum solutions.

Other near optimum methods was also proposed in the literature. E.g. Liu and Rahbar (Liu and Rahbar 2004) have innovatively modeled the project network with a pipeline network. In their model, the objective is to maximize the flow that is passed through nodes through using Maximum Flow, Minimum Cut theory. Unlike most of the other near optimum solutions, this method deals only with critical activities, which reduced the time needed to perform the acceleration.

There are also methods that tend to minimize project total cost and accordingly maximize project benefit, while taking into account time value for money using discounted cash flow and net present value methods (Sunde and Lichtenberg 1995, Ammar 2011). Icmeli and Erenguc (Icmeli and Erenguc 1996) also proposed a method which considers not only the time value for money, but also the resource constraints that may exist in the process of schedule compression.

These optimization methods performed well over a variety of problems as they are simple and easy to use and need less computational effort, although may lead only to near optimum solutions. Also, as explained earlier, application of these methods is time-consuming for large scale projects. Because of these reasons, heuristic methods have been introduced and used widely to solve time-cost trade-off problem.

2.3.2. Heuristic methods

The second group of project schedule compression methods presented in the literature are the heuristic methods that are mainly based on rules of thumb. Although these methods are easier to model and apply, which renders them more practical for large scale projects, they do not guaranty optimal solutions (Senouci and El-Rays 2009, Feng, Liu and Burns 1997).

Prager (Prager 1963) and Moslehi (Moselhi 1993) have proposed heuristic methods that converted the time-cost trade-off problem to structural models that are more familiar for civil engineers and construction managers. Prager represented each activity within project network as a structural member that consists of a rigid sleeve containing a compressible rod of the natural length and a piston at its protruding end as shown in Figure 2.6. Then these members are subjected to a gradually increasing compressive force until reaching their un-compressible duration. Similarly, Moselhi have used the same modeling while members are subjected to imposed displacements. In other words, it establishes the analogy between project CPM network and the geometry of the equivalent structure (see Figures 2.7 and 2.8). Subsequently, it performs a nonlinear static analysis under imposed displacement (equivalent to the magnitude of schedule compression). This way, the sum of all members' axial forces represents the added cost required to crash the project schedule with a time equal to the imposed displacement.

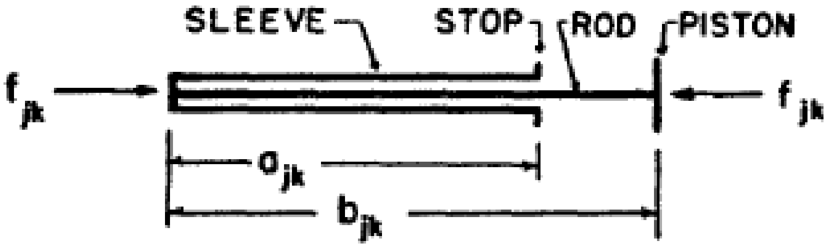


Figure 2.6: Equivalent structural member (Prager 1963)

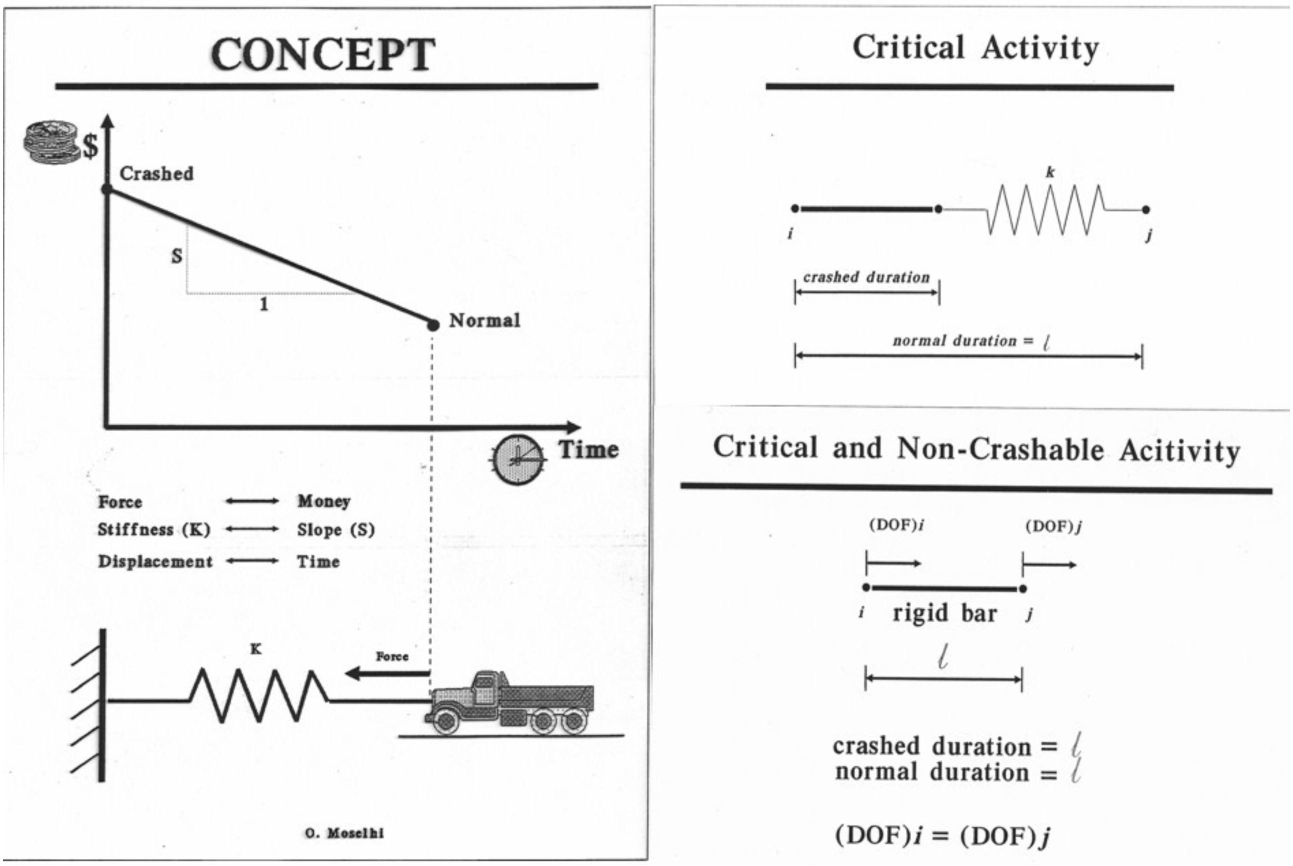


Figure 2.7: Concept of analogy with direct stiffness method (Moselhi 1993)

Table 2.1: Analogy conditions (Moselhi 1993)

No	Project compression	Structural analysis
1	Project	Structure
2	Project CPM network	Structure's geometry
3	Project activity	Structural member
4	Project event	Structural node
5	Activity cost slope	Member stiffness
6	Event early start times	Nodal coordinates
7	Activity available time	Member length
8	Activity compression	Member axial displacement
9	Crashed activity cost	Member axial force

The iterative crashing method is one of the other approximate methods, which is described in many text books for project acceleration (Ahuja, Dozzy and AbouRizk 1994, Meredith and Mantel 2006, Hinze 2008, Sears, et al. 2008). It was first introduced by Siemens (Siemens 1971) as an effective cost slope model named SAM (Siemens Approximation Method). this method commonly considers linear, piecewise linear, discontinuous, hyperbolic or discrete relations between activity’s direct cost and its duration (Yang 2007, Evensmo and Karlsen 2008)and tends to shorten project total duration by crashing the activity with the lowest cost slope on the critical path one unit of time in each iteration. In other words, this is done by selectively crashing specific activities to shorten project duration and then incrementally crashing (i.e., shave a day off of) the selected activity where that is possible. Then it keeps track of the activity-based (direct) cost of crashing selected activity (or activities) and indirect cost savings associated with reducing overall project duration while recalculating the forward pass and check for changes in network critical path(s).

The procedure of crashing ends up by reaching project least-cost duration, reaching the targeted project duration or until no further crashing is possible. Iterative crashing procedure has also been used to accelerate linear projects such as highways and pipelines (Hassanein and Moselhi 2005). These heuristic methods performed well over a variety of problems. However, the solutions obtained by these heuristic methods do not provide the range of possible solutions, making it difficult to experiment with different scenarios for what-if analyses. Still these heuristic methods can find good solutions with far less computational effort than optimization methods.

Regardless of being heuristic or optimization based, leading to optimum or near optimum solutions, none of the methods cited above take into account any factor beyond the additional direct cost required for acceleration of project activities. This has been attributed to the limited uptake and use of these methods by contractors. In fact, the lack of consideration of such factors has been attributed to the limited use, if any, of these methods in practice (Sears, et al. 2008). Attributed to that as well is the lack of commercial software systems that can be used for automated schedule compression.

2.4. Time-cost trade-off considering risk

Uncertainties are very common in construction projects. As such, because uncertainty and risk exists in all phases of EPC projects, it is important to identify, quantify and manage the risk during the execution phase of these projects. This has led to generation and use of Project Evaluation and Review Technique (PERT) since 1950’s. PERT is a probabilistic approach that considers uncertainty in activity durations to determine the completion time of the project, and that can be used to estimate the probability to complete the project by a given time (Tolentino Pena 2009). However, this method reduces the probabilistic model to a deterministic Critical Path Method (CPM) by simply using activity time means in calculations (Haga and Marold 2004). This way, it ignores the stochastic nature of activity completion times and also the fact that crashing duration of some activities may have more effect on the mean project completion time than others. Also, when considering schedule compression, PERT ignores the factors that affect probabilistic compression decisions such as the effects of competing probabilistic paths and the complex interactions created by dependent sub-paths, because it unrealistically reduces the solution space to a single path through a network (Bregman 2009).

Specifically, Network crashing, which is done by bringing in additional resources to reduce the activity completion times of activities along the critical path, can met only with limited success when ignoring the stochastic nature of the critical path of the project (Haga and Marold 2004). As a result, the uncertainty and risk should be considered when minimizing project cost and duration, which leads to the so-called stochastic time-cost trade-off problem. The generalized stochastic time-cost trade-off problem focuses primarily on the projects in which the activities may have several alternatives each with an associated cost and stochastic duration. The objective of these methods is then to determine the best configuration of alternatives that minimizes the expected project cost. The method designed to solve this problem should be capable of providing an optimal configuration of alternatives before the start of the project as well as dynamically re-evaluating the project throughout its execution (Tolentino Pena 2009). As such, Wollmer (Wollmer 1985) has developed a stochastic version of the deterministic time-cost tradeoff problem and Gutjahr et al. (Gutjahr, Strauss and Wagner 2000) have demonstrated a stochastic branch-and-bound approach for the static probabilistic version of the discrete selection process.

Consideration of deterministic activity duration and costs, i.e. considering their mean values, resulted in ignoring significant overlaps between distributions of both durations and cost of activity alternatives (Feng, Liu and Burns 2000). To address such limitation and to consider correlations between project activities, Feng et al. have proposed a method that uses a hybrid of Genetic Algorithms (GAs) and simulation techniques to solve the time-cost trade-off problem under uncertainty. This way, their method finds the best combination of activity alternatives which minimizes project duration and cost while considering uncertainties associated with durations and costs of activity alternative.

Chance-constrained programming was also proposed by Charnes et al (Charnes, Cooper and Thompson 1964) as an alternative approach to evaluate probabilistic activity networks. Kress (Kress 1984) later expanded the method by establishing upper and lower bounds for the chance-constrained critical path. Laslo (Laslo 2003) proposed a chance-constrained method for estimating activity expediting costs when costs vary with activity time. However, the use of chance-constrained programming for evaluating project activity networks was criticized by Elmaghraby et al. (Elmaghraby, Soewandi and Yao 2001) because of not adequately capturing the interdependence among network paths (Bregman 2009).

As such, simulation has been used as a method for evaluating probabilistic activity networks based on standard sampling procedures. Bregman (Bregman 2009) has proposed a matrix-based simulation method to incorporate activity duration uncertainty into the project expediting decision process. His method dynamically re-evaluates expediting options that are available for project and is meant to suit large scale projects.

Tolentino Pena (Tolentino Pena 2009) also presented a dynamic, simulation-based optimization method to minimize the expected project cost due to lateness penalties and the activity alternatives selected. In his method, project activities are considered to have uncertain durations following a stochastic probability distribution. Its objective is then to find some combination of activity alternatives that minimizes project cost. In other words, his method aims at determining the set of activity alternatives associated with the point where the total cost will be minimized. However, his method takes all the activities within the project network into account when finding the best set of activity alternatives. Also, although the method tends to consider the uncertainty, still this uncertainty is only considered for estimated duration (i.e. original duration) of each activity, and not necessarily through the compression process.

However, the use of probabilistic models has been discredited by Balasubramanian and Grossmann (Balasubramanian and Grossmann 2003) since these models require detailed information about probability distribution functions as well as high computational expense. Subsequently, fuzzy set approach has been used to represent uncertainties associated with activity durations e.g (Balasubramanian and Grossmann 2003, Lin 2008, Long and

Ohsato 2008). These methods recommend the use of fuzzy numbers to model uncertainty associated with activity durations rather than stochastic variables. In other words, these methods make use of membership functions, based on possibility theory, instead of probability distributions (Herroelen and Leus 2005).

While project cost is one of the most important aspects of a project, a crucial importance should be placed on the risk associated with it. Likewise, when planning to crash the duration of a project, the added cost associated with such crashing should account for the risk involved. Considering that the main purpose of time-cost trade-off analysis is to find the least additional cost required to crash project schedules into a targeted duration, the risk associated with that cost should be identified and quantified to help generate a realistic crashing plan.

In spite of all these methods that only consider the uncertainty associated with estimated activity durations in the process of schedule crashing, Yang (Yang 2005) has considered the uncertainty associated with project budget. He has questioned the underlying assumption in the available methods that actual funds to support the project would never deviate from the original estimate. In other words, he has considered the project budget to be stochastic, following a probability distribution. Subsequently, he used linear programming to minimize project total cost subjected to the constraint of not exceeding this uncertain budget that he also called as “financial constraint”. Although these methods tend to address the risk associated with crashing process, still the uncertainty associated with the estimated crashed costs for critical activities remains untouched. In other words, none of the methods cited above accounts for modeling the uncertainty and quantifying the risk associated with crashing cost of the critical activities involved.

2.5. Summary

When reviewing the literature on project compression, potential areas of expansion were found. In other words, while the research studies presented in this chapter have provided significant contributions to this important research area and to solve time-cost trade-off problem there are still certain gaps in the literature on schedule compression that still remain void. First, as stated before, in all of the various methods that are proposed in the literature, schedule compression is still reduced to some form of time-cost trade-off analysis, where schedule compression is performed based on cost only. This way, these methods discard other factors that are intuitively considered by contractors and project managers when they plan to crash respective duration of their projects. Second, there has been little or no reported research focusing on studying and optimizing the collective impact of the uncertainty and risk associated with crashing cost of project activities.

To address above needs, the current study proposes a method to consider a multi-attributed decision making environment for the schedule compression problem. As well, the uncertainty associated with crashing cost of activities is accounted for, in the developments made in this study. A detail description of the proposed method as well as the results of a recently conducted questionnaire survey is provided in following chapters.

3. CHAPTER THREE: CURRENT PRACTICE

3.1. Motivation

A questionnaire survey has been carried out to better understand the nature of the decision environment in which schedule compression is performed. In other words, it was designed to find out whether or not contractors consider only the added direct cost required to accelerate each activity, when selecting and prioritizing activities for crashing in order to accelerate project delivery. The survey also aims at finding out factors that contractors usually consider in order to accelerate their projects in the most efficient and practical manner.

3.2. Questionnaire design and distribution

The questionnaire of the survey was prepared in both paper and web-based formats in English and French languages. A copy of web-based format of the questionnaire is included in Appendix II. The questionnaire was sent to 60 contractors and construction management professionals in Canada, United States and Meddle East via email. And it was placed online on the worldwide web. 53 completed questionnaires were received from twenty-one contractor and construction management firms. The list of participating firms along with number of respondents from each firm is included in Appendix III.

In order to ascertain the need for schedule compression in practice, the respondents of the questionnaire survey were asked about the frequency of encountering the need to accelerate projects under their responsibility. The results show that only 5 percent of the respondents did not encounter such a need (see Figure 3.1). These results also show that the majority of the respondents (42 percent) encounter this need in 30 to 70 percent of the projects under their responsibility. This clearly shows the practical importance of the schedule compression as a critical management function.

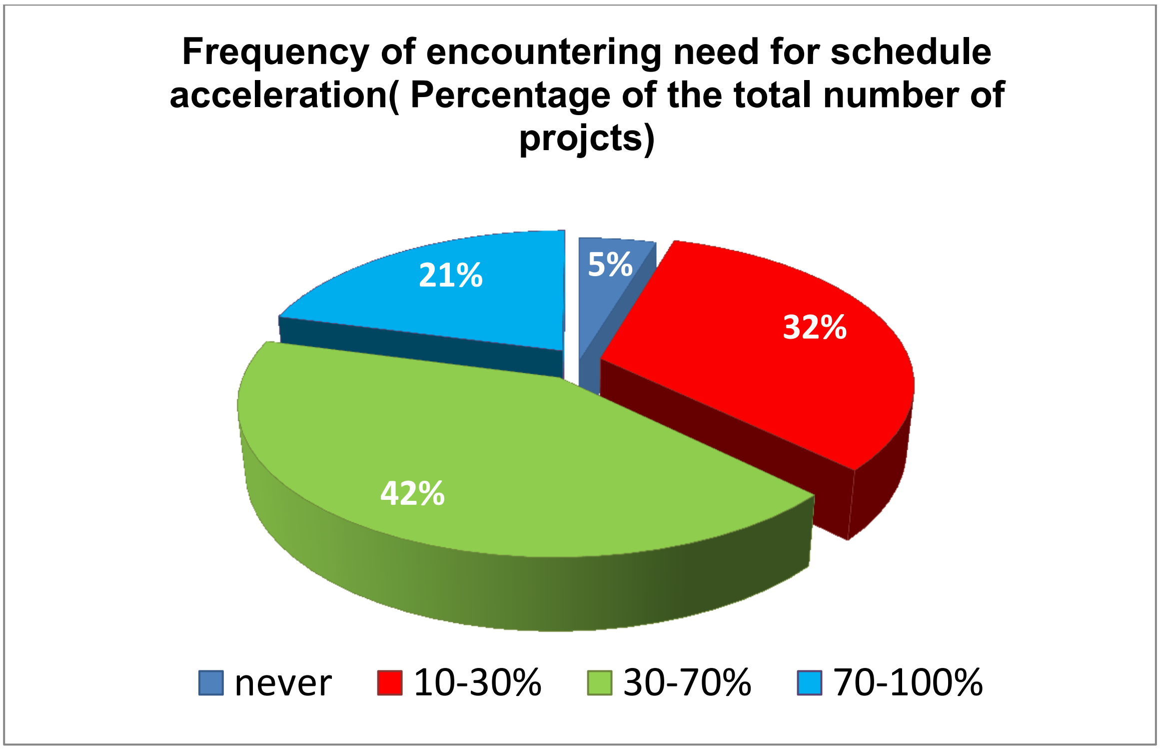


Figure 3.1: Frequency of encountering the need for schedule acceleration

Strategies used for schedule compression, the factors that are considered in such crashing process and availability of a commercial software system that professionals can use for schedule compression for schedule compression were also part of this survey. In this regard, a number of factors were provided to the participants to select from and to add to them; if needed.

3.3. Survey participants

The participants of the survey were seasoned contractors and professional working on building and industrial projects, oil and gas capital projects, power plants facilities and heavy infrastructure projects. Their experience in management of these types of projects ranged from 5 to 55 years (see Figure 3.2). Their typical job size ranges in value from one million to three billion dollars. Twenty-five percent of the respondents were professionals working for general contractors of construction projects, while forty percent worked in construction management firms and the remaining participants were professionals working as both contractor-construction managers.

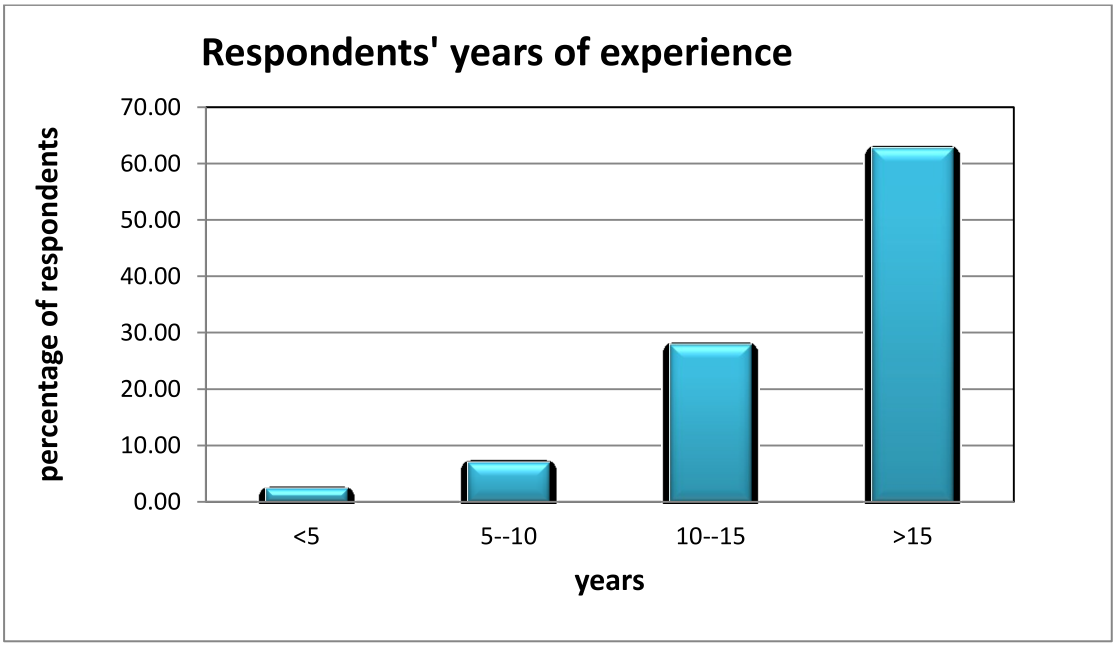


Figure 3.2: Respondents' years of experience

3.4. Findings

The survey results revealed that contractors and construction management professionals consider a wide range of factors when making decisions pertinent to shortening project durations (Moselhi and Roofigari 2011-b). These results also indicate that the top seven commonly considered factors in schedule compression are: 1) resource availability, 2) contractors' leverage on subcontractors who are selected to carry out the accelerated work, 3) additional direct cost required to crash each activity from its normal duration state, 4) risk, 5) complexity and logistics of the work involved, 6) number of successor of the activities and 7) cash flow constraints (see Figure 3.3).

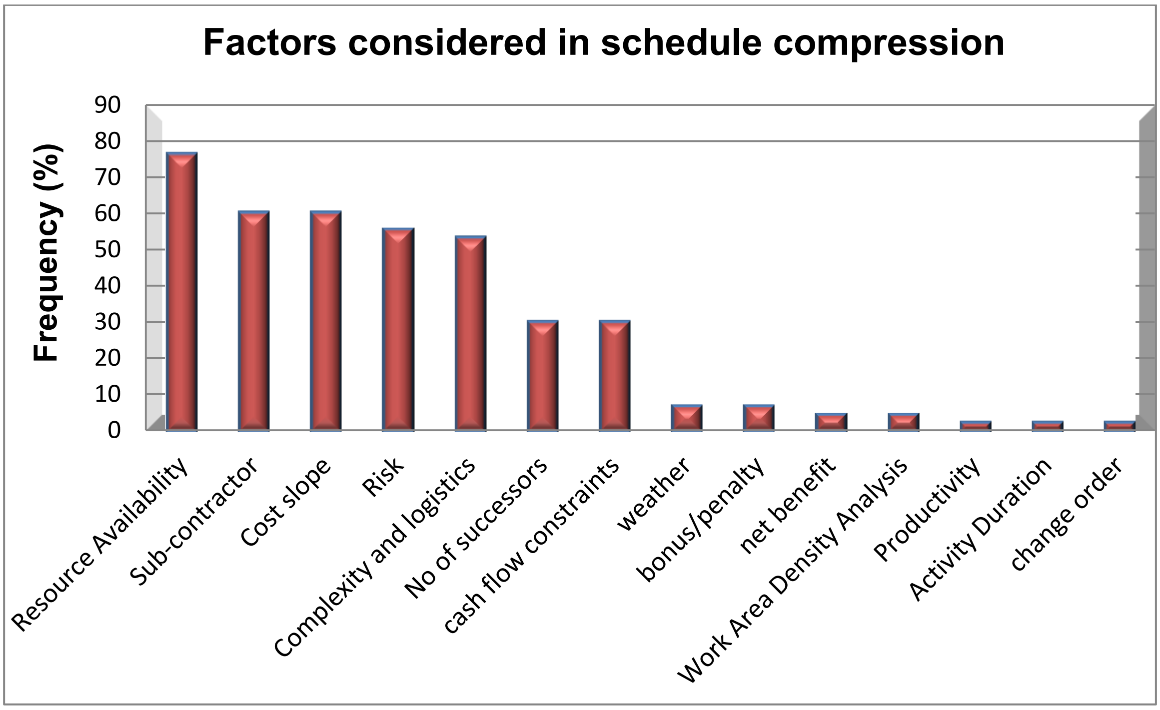


Figure 3.3: Factors considered in schedule compression

As can be seen from Figure 3.3, the results also show that factors such as resource availability and contractor's leverage on sub-contractors who are deemed more capable of performing the accelerated work where found to be even more important than the project additional cost needed for crashing activity durations. This can be the case, particularly when owners request contractors to accelerate the delivery of their projects. In this case, cost will not be the major factor to be considered; as factors such as complexity of the work involved and availability of required resources will be of essence and, accordingly, gain more importance in setting priorities for activity crashing. Even in the case where acceleration is performed to recover contractor's own delays, cost still is not the only factor to be considered, but may be assigned more importance than in the case referred to above. In the latter case, factors such as cash flow constraints and the risk associated with the work involved can be of more importance. This, perhaps, explains the limited use of existing methods that consider only cost, in practice.

Also it is interesting to note that as shown in Figure 3.4, the order of importance of these factors is different among contractors and construction management professionals working for owners or for owners and contractors. As it can be seen from that figure, while contractors put more emphasis on execution factors such as sub-contractors who are deemed to perform the accelerated work, construction managers look more into overall project conditions such as resource availability. In addition, participants who worked for both contractors and construction management firms seem to consider wider range of factors including both overall executions related and project dependent factors.

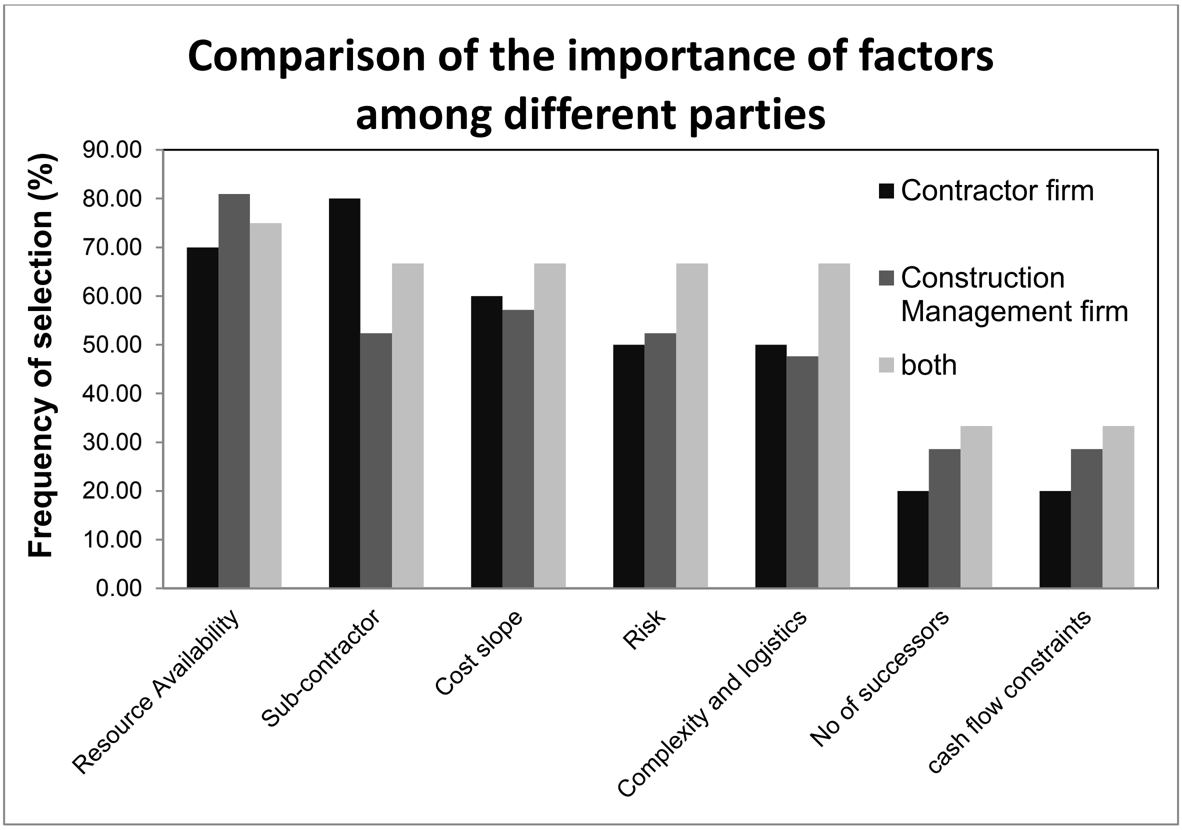


Figure 3.4: Comparison of the frequency of selection of factors among different parties

With respect to strategies being used by the participants in this survey for schedule compression, it was found that CPM crashing and the generation of realistic project baselines are common elements of their strategies. Figure 3.5 illustrates these strategies as well as their respective frequencies. The results depicted in Figure 3.5 indicate that forty-seven percent of the participants consider CPM crashing (i.e. crashing activities on the project's critical path) as the strategy for schedule compression. Still 34 percent of participants selected to check if project's baseline is realistic. It is important here to emphasize the fact that the purpose of this survey was to find out current industry practice for schedule compression during construction, which is also known as unplanned schedule compression (Noyce and Hanna 1998). As to the assessment of the base-line whether it is realistic or not, it could be useful, particularly before commencement of construction. In that case it could be useful in what is known as planned schedule compression.

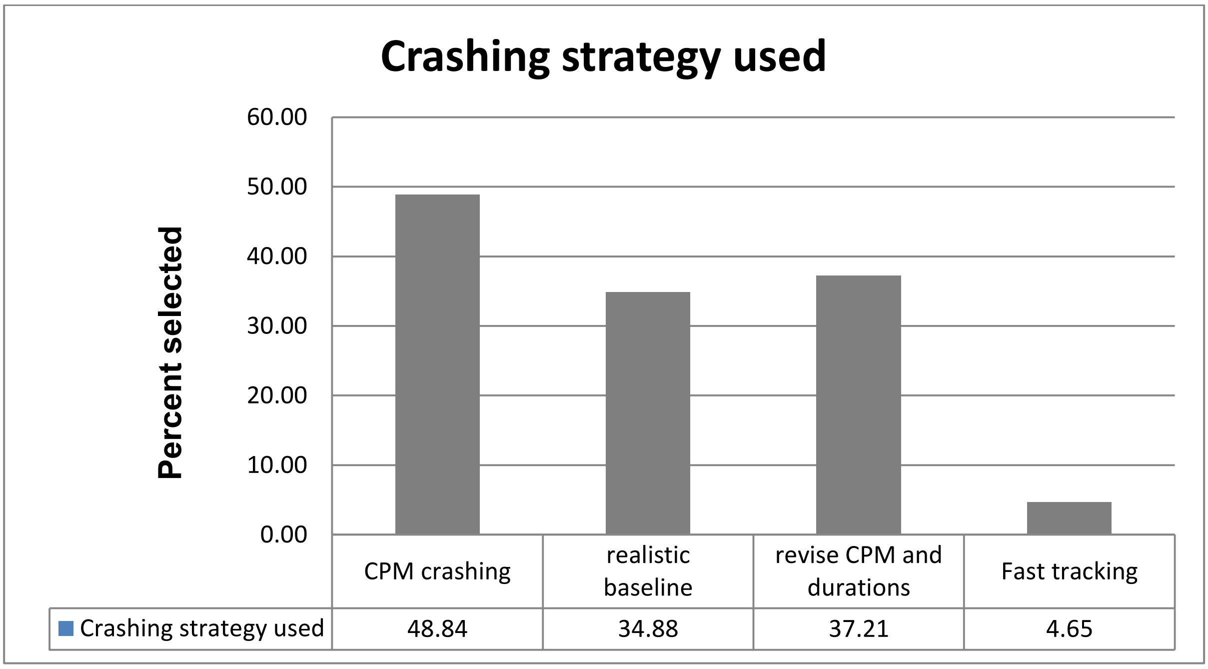


Figure 3.5: Strategies used for schedule compression

3.5. Summary

An industry-wide questionnaire survey was carried out to understand and capture the nature of current practice pertinent to the decision environment, including factors and methods used, for schedule compression in practice. The results of this survey highlighted the importance of the schedule compression problem, in practice. As well, these results revealed that despite the wide range of methods, which are available in the literature, none of the respondents refer to the use of these methods. These results also show that factors such as resource availability, complexity and logistics of the work involved, contractor's leverage on sub-contractors who are deemed more capable of performing the accelerated work and the risk associated with crashing of each activity were found to be more than or equally important to project cost, in queuing activities for crashing. The results also indicated a need for commercial software that can be used by professionals for automated schedule compression, with relative ease.

4. CHAPTER FOUR: PROPOSED METHODOLOGY

4.1. General

As stated in previous chapters, solving the traditionally defined time-cost trade-off problem involves identifying the activities whose duration is to be reduced and the amount of the reduction (referred to as the crashing configuration). The purpose of this study is to capture such crashing configuration while considering a set of objectives and constraints.

In other words, this study aims to circumvent the limitations of current methods which implicitly look only into time and cost when planning to perform schedule compression. As a result, these methods ignore other factors that emanate from each project's environment and operational constraints and are considered intuitively by contractors in practice. Accordingly, the proposed method is capable of accounting not only for cost, but also for a set of additional factors. As such, in this study schedule crashing is studied as a multi-attributed decision making problem in which different factors contribute to the priority setting for crashing critical activities. In other words, the method accounts for factors beyond cost (e.g. resource availability, risk, complexity and logistics of the work, and other factors that found to be used by contractors and project managers) and the risk associated with these factors in activity level. For this purpose, the proposed method utilizes a modified format of the Multiple Binary Decision Method "MBDM" (Marazzi 1985) along with iterative crashing process to model such decision environment as will be explained subsequently. Such risk assessment is also modeled using the “probability–impact matrix” as will be shown and described later in this chapter.

While other multi criteria decision support methods such as the Analytical Hierarchy Process (AHP) (Saaty 1980) can also be used for the purpose of modeling the decision environment and setting priorities for activity crashing, MBDM is used here because of its less exposure to subjectivity in view of its binary comparisons. Also, because of its structured and well organized pair-wise comparisons process that encourages decision makers to study and evaluate the relative importance of the attributes considered in the crashing process, it is suited to model this multi-attributed decision environment. This is particularly true in modeling the schedule compression problem, where contractors’ intuitive judgment and perception of the problem constitute major consideration in priority setting for activity crashing. In this respect MBDM, unlike AHP, does not need to calculate consistency ratio consider to remedy inconsistencies that may arise from the pair-wise comparisons (Moselhi and Roofigari 2011-c).

Iterative crashing process has been used in view of its practicality and simplicity. As well, unlike other approximate methods such as GA’s and Harmony Search which consider all activities within a project network, the used iterative process only deals with critical activities; making it more practical and suitable for large project networks. The developed methodology has been implemented in Visual Basic environment, with a dynamic link to MS-Project in order to facilitate the transfer of scheduling data needed to perform the analysis as well as the needed rescheduling of project in each incremental schedule crashing. Figure 4.1 illustrates the flowchart of the developed algorithm and depicts the sequential relationships among its various steps.

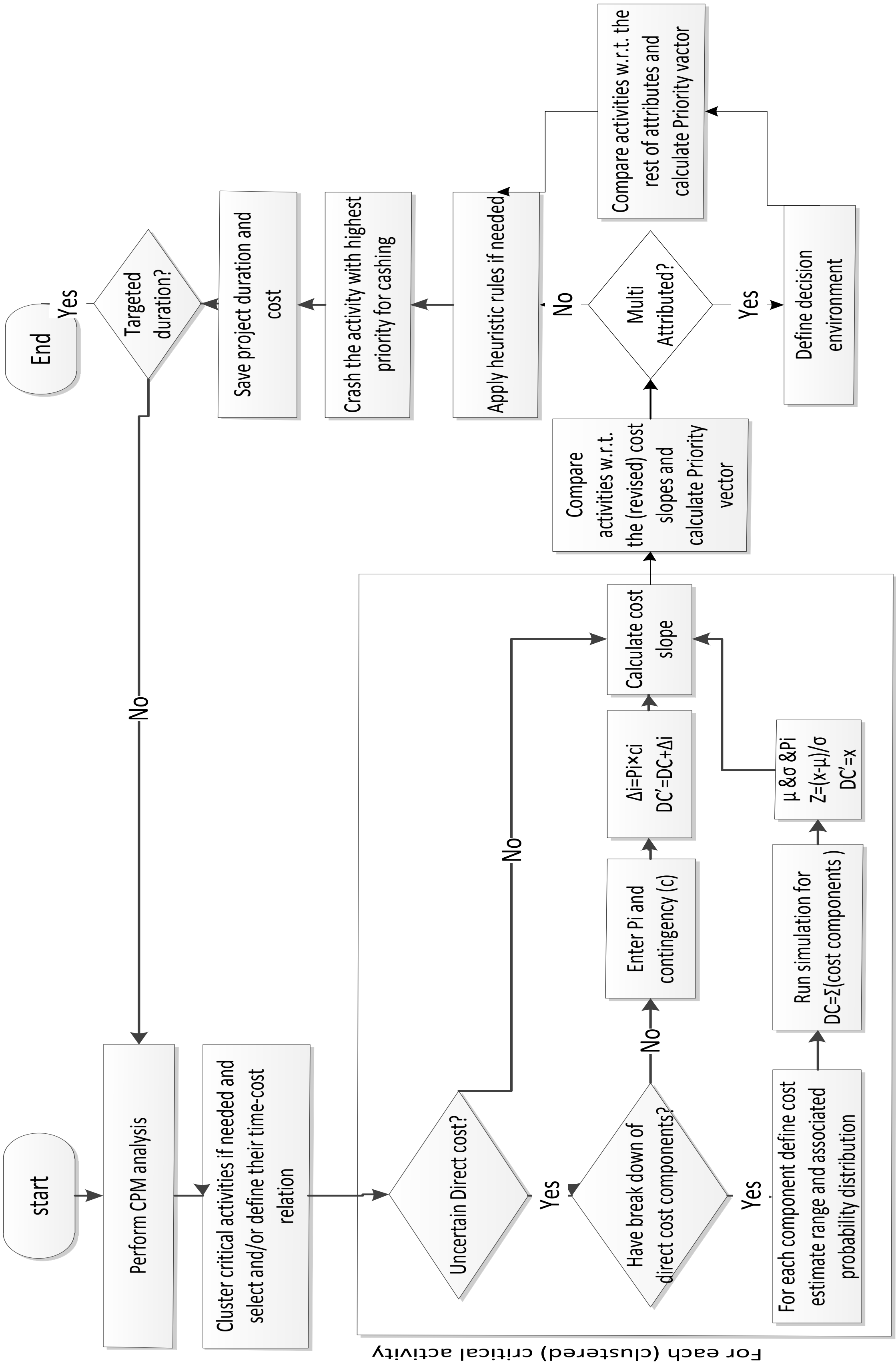


Figure 4.1: Flowchart of the proposed method

4.2. Modeling of decision environment

In the iterative crashing process, reducing project duration requires the reduction of the longest path of its schedule network (i.e. its critical path). As such, assigning additional resources to expedite haphazardly-selected set of activities may not help in achieving this objective. The objective can rather be achieved by reducing the durations of one or more of the activities that form the critical path. This results in increasing the direct cost of the expedited activities and in reducing the project indirect cost (as illustrated previously in Figure 2.3). It should be noted that each project has an optimum duration, that corresponds to the least or minimum overall total cost for the project. Any deviation from the optimum condition results in higher project total cost. The challenge here is to perform such compression while satisfying a set of objectives and constraints either imposed by contractual agreements between owners and contractors or operational constraints of the contractor's organization.

To address this need, the proposed method accounts not only for the additional direct cost needed for reducing activity durations, i.e. their cost slope, but also for other factors such as cash flow constraints, logistics and complexity of the work involved, the risk associated with compression of an activity duration, contractor's leverage on the subcontractor who is expected to perform the accelerated work, and the number of successors of the activity being considered for compression. In other words, the method provides users with the flexibility of considering factors that account for their own organizations' financial and technical constraints as well as those emanating from project specific conditions. As such, a more doable execution plan will be generated which will be more practical and likely to be successful in its implementation. As well, in the proposed method, contractors can generate more than one execution plan; giving them more flexibility to make enlightened decisions in the crashing process.

A multi-attributed schedule crashing algorithm (C-Schedule) has been developed (see Figure 4.1). The algorithm has two essential, yet integrated, processes: first, priority setting for activity crashing; i.e. queuing activities for shortening their respective durations and second, iterative schedule compression which progressively reduces project duration in search for its least cost duration. The first is carried out using a modified format of the Multiple Binary Decision Method (MBDM) developed in Visual Basic environment and the second is achieved by dynamically linking the developed algorithm to CPM-type scheduling software system.

The schedule compression process cannot start until crashing priorities are established for all critical activities. This is performed by, first, transferring critical path scheduling data from the scheduling software used by project team to the developed computer application. The user is then required to provide activity crashing data such as activity direct cost versus its duration as shown in Figure 2.4 and Figure 2.5 parts (a) and (b). The priorities for crashing individual activities are then established using the following procedure:

1. The attributes to be considered in the crashing process at the project level should first be defined. These attributes can either be selected from a check list in the developed computer application or entered to add to those

factors directly by the user via an interactive user-friendly menu as shown later in Chapter 5.

2. The weights which reflect the relative importance of each attribute in setting priorities for crashing critical activities are calculated automatically, based on the procedure described in Step 3 (see Figure 4.1). Also in case of large networks, an option is provided to decision makers to cluster activities in groups and queue them for the crashing process.

3. The decision maker has to compare the attributes defined in Step 1 based on their relative importance, through a process of pair-wise comparisons. In other words, each attribute is individually compared to other attributes. A decision matrix of the order n×n is then generated where "n" represents the number of attributes considered for setting activity crashing priorities. Figure 4.2 depicts the decision matrix generated from a pair-wise comparisons process. In that matrix, a_{ij} represents whether attribute A_i is more important or less important than attribute A_j. According to the original MBDM method (Marazzi 1985), if A_i is more important than A_j, a_{ij} is set equal to 1. Otherwise, it will be set equal to 0. It follows that if a_{ij} is set equal to 1, then a_{ji} must be equal to 0. In the original method also, diagonal elements are set equal to 0. The original method, in this case, discards cases were two attributes or alternatives are of equal importance. Also, by assigning zero values to diagonal elements, two problems arise; first, each activity is assumed to be of less important than itself and second, the attribute or alternative that has the lowest importance is, accordingly, eliminated. To overcome these problems, a modified format of the MBDM method is introduced and utilized in the developments presented in this study. In this

modified format, in cases where one attribute (or proposal) is neither more important nor less important than another attribute, but rather is of equal importance, a value of 0.5 is assigned to their representative elements in the decision matrix. Accordingly, the diagonal elements in the decision matrix are assigned a value of “0.5” instead of the “0.0” value used in the original method.

$$\begin{array}{c|cccc|cc}
 & A_1 & A_2 & \cdots & A_n & \text{weights} & \text{Normalized weights}(W_i) \\
\hline
A_1 & a_{11} & a_{12} & \cdots & a_{1n} & w_1 & W_1 \\
A_2 & a_{21} & a_{22} & \cdots & a_{2n} & w_2 & W_2 \\
\vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\
A_n & a_{n1} & a_{n2} & \cdots & a_{nn} & w_n & W_n \\
\hline
 & & & & & S & 1.00
\end{array}$$

Figure 4.2: Sample attributes’ decision matrix and the related normalized vector

Upon constructing the decision matrix, the relative weights are calculated by summing up the elements of each row and then normalizing the values of each element of the summation vector to generate the weight vector (Marrazi, 1985). It should be noted that these weights reflect the relative importance of each attribute in setting priorities for activity crashing and are calculated using following Equations:

$$w_i = \sum_{j=1}^{n} a_{ij} \tag{4.1}$$

$$S = \sum_{i=1}^{n} w_i \tag{4.2}$$

$$W_i = {w_i}/{S} \tag{4.3}$$

Where:

w_i =non-normalized weights;

a_{ij}= importance indicator of attribute A_i over attribute A_j;

S= Sum of elements of the vector w_i;

W_i= normalized weights;

This way, the decision environment that is to be considered in the crashing process, i.e. the decision attributes and their relative importance in setting priorities for crashing critical activities is established. Subsequently, activities on the project network's critical path are compared with respect to each of these decision attributes as will be described in next sections.

4.3. Modeling the risk associated with activities' crashed cost

As stated previously, project cost is one of the most important aspects of a project. As a result, a crucial importance should be placed on the risk associated with it. Likewise, when planning to crash the duration of a project, the added cost associated with such crashing should account for the risk involved. Considering that the main purpose of time-cost trade-off analysis is to find the least additional cost required to crash project schedules into a targeted duration, the risk associated with that cost should be identified and quantified to help generate a realistic crashing plan. To circumvent this need, risk impact on activities crashed cost is considered in the developed method presented in this study. This risk is quantified here in two different ways: (1) for the cases where details on different resources of an activity are not available and only a lump sum direct cost is estimated for its crashing, such risk is quantified utilizing the probability impact matrix as will be described subsequently in section 4.3.1; (2) on the condition that details are provided for each resources and sub-resources, Monte Carlo simulation is utilized to quantify the risk associated with the crashing cost of each activity.

4.3.1. Lump sum direct cost

As stated before, in the cases where details on different resources of an activity are not available and only a lump sum direct cost is estimated for its crashing, the risk is quantified utilizing the probability impact matrix (Stackpole 2010); applied in a manner similar to the itemized probabilistic method for contingency estimating (Moselhi 1997). The developed method is also able to account for different risks associated with the crashed cost of required resources needed to perform each critical activity, individually.

In the iterative crashing process used in the developed methodology, activities are queued for crashing based on the additional direct cost needed for crashing their duration one unit of time, i.e. their cost slope. As such, the activity with lowest cost slope gains the highest priority for crashing. It should be noted that to accelerate an activity, additional resources (i.e. material, labour and equipment) should be assigned to that activity. The term “cost slope” is defined as the additional direct cost required for crashing the duration of each critical activity by one unit of time and is calculated using Equation 4.4:

$$Cs_i = \frac{CC_i - NC_i}{ND_i - NC_i} \tag{4.4}$$

Where:

Cs_i= Cost slope of activity i;

CC_i= Crashed cost of the activity i;

NC_i= Normal cost of the activity i

ND_i= Normal duration of the activity i;

CD_i= Crashed duration of the activity i

The cost slope calculated for each activity should then be revised to account for the risk associated with the crashing cost of each activity. Therefore, the risk associated with each activity is represented by the contingency associated with its crashed cost (Δ), which represents the severity of the risk impact (see Figure 4.3), as well as the probability of not exceeding its contingency.

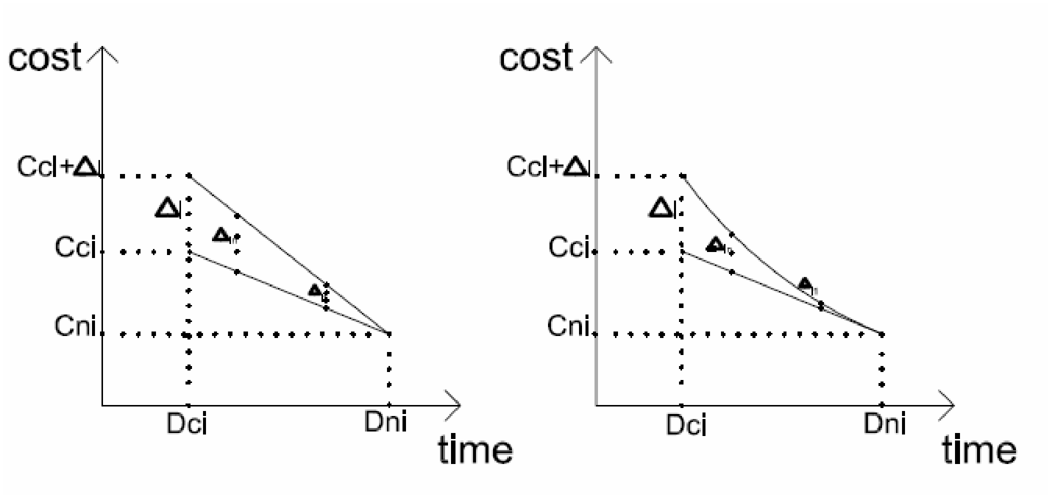
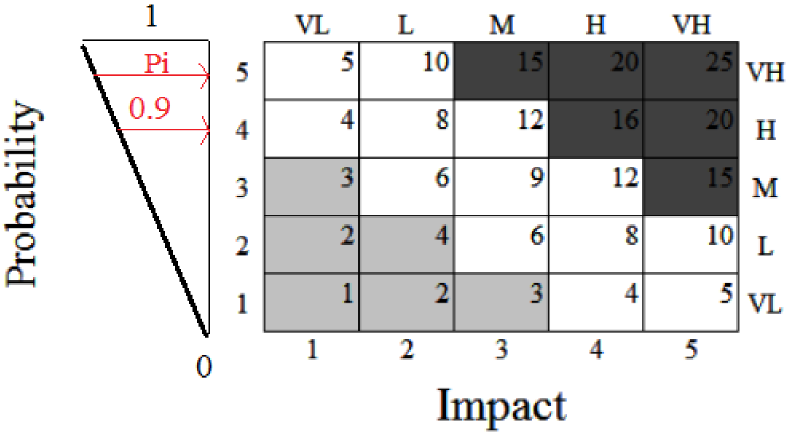


Figure 4.3: Linear Δ and (b) Varying Δ (activity level)

Figure 4.4 illustrates the probability-impact matrix along with how its probabilities are converted to 0-1 probability scale in the proposed method.



	VL	L	M	H	VH	
5	5	10	15	20	25	VH
4	4	8	12	16	20	H
3	3	6	9	12	15	M
2	2	4	6	8	10	L
1	1	2	3	4	5	VL
	1	2	3	4	5	

VL=Very Low
L=Low
M=Medium
H=High
VH=Very High

Figure 4.4: Probability-Impact matrix

Having defined the impact and probability of the risk associated with each activity, Δ which represents the quantitative amount of risk associated with crashing cost (Cci) of each activity is calculated using Equation 4.5.

$$\Delta_i = Cc_i \times (\alpha_i / 100) \times P_i \tag{4.5}$$

Where:

Cc_i= Estimated crashed cost for activity i

α_i= The contingency expressed as percentage of the crashed cost (i.e. the severity of risk impact on crashing cost of activity i);

P_i=Probability of not exceeding the estimated contingency

In availability of the lump sum crashing cost for critical activities as stated above, the risk associated with each of these resources can be calculated in general. In other words, these risk measurements can be applied to calculate the risk associated with respective crashed cost of the individual resources; namely: material, equipment and labor. It should be noted that, when taking into account the risk associated with resources forming each activity, the value of Δ is to be calculated for each of these resources separately. As such, first, time-cost relation for each of these resources is defined and the impact (contingency) and probability of the risk associated with their crashed cost is to be assigned in the same manner it is done in activity level; e.g. the risk associated with crashed cost of material required to execute activity (i) is defined by estimating its contingency (α_{mi}) and probability of not exceeding that contingency (P_{mi}). Hence, the resulted Δ will illustrate the quantitative risk associated with crashing cost of each resource; e.g. Δ_{mi} is the risk associated with estimated crashed cost of materials needed to execute activity i. However, Δ associated with each of these resources can be zero should their crashed cost is assumed to be estimated with certitude.

Having defined the risk for each resource in this case, total Δ_i for each activity is to be calculated by adding the resources' Δ as shown in Equation 4.6.

$$\Delta_i = \Delta_{mi} + \Delta_{li} + \Delta_{ei} \tag{4.6}$$

Where:

Δ_{mi}= Quantitative amount of risk associated with crashed cost of materials for activity i
Δ_{li}= Quantitative amount of risk associated with crashed cost of labour for activity i
Δ_{ei}= Quantitative amount of risk associated with crashed cost of equipment for activity i

The amount Δ for each activity (or each resources) can be distributed linearly over activity's cost-time curve; i.e. the same percentage of contingency is considered and added to the crashed cost of the activity at each increment of crashed duration (see Figure 15(a)) or can vary over the crashed duration as shown in Figure 4.3(b). Figure 4.3 illustrates the time-cost curve at the activity level (a linear relation is assumed between activity's duration and its direct cost). However, the time-cost relation shown in this figure needs not be linear or even continuous in the developed method, meaning that each of the nonlinear, linear, piece-wise linear and discrete time cost relations can be considered for activities and/or resources. If Δ is assumed to be distributed linearly over the crashed duration of the activity, it is to be entered at the final crashed cost of the activity and the interim Δ's for each crashed unit of time is calculated proportionally. Else, the amounts of α and P are to be entered at each incremental unit of crashed time of activity duration.

After calculating Δ for all critical activities, the revised cost slopes (Cs') which accounts for risk is calculated using Equation 4.7.

$$Cs'_j = \left\{ \frac{Cc_j + \Delta c_j - Cn_j}{Dn_j - Dc_j} \right\} \tag{4.7}$$

Where:

Cc_j: Crashed cost of activity j

Cn_j: Normal cost of activity j

Dn_j: Normal duration of activity j

Dc_j: Crashed duration of activity j

Δc_j: Contingency associated with the crashed cost of activity j

Revising the activities' cost slopes with due consideration of risk may result in changing the priorities by which activities are queued for crashing. These revised cost slopes are then used in carrying out pair-wise comparisons among the critical activities to be crashed. In this case, the activity that has lower cost slope is considered more important than the one with higher cost slope. As such, priorities are regenerated based on revised cost slopes.

4.3.2. Direct cost break down

For the activities for which a number of resources and sub-resources are to be assigned, the use of itemized contingency estimating explained in previous section might not be precise enough. In such situations the risk associated with crashing cost of these activities is better quantified by using probabilistic sampling and Monte Carlo simulation. In such uncertain conditions, breaking down an activity's crashed cost into several crashing costs required to perform each of its independent resources will provide more precise cost slopes. Further, such breaking down will also help in efficient execution and acceleration of large tasks that are composed of different resources and sub-resources whose estimated crashed cost is highly uncertain.

Monte Carlo methods are those in which properties of the distributions of random variables are investigated by use of simulated random numbers (Gentle 1985). In Monte Carlo simulations, a model is run repeatedly, each time using different values for each of the uncertain parameters. The values of each of the uncertain parameters are drawn from its probability distribution (Baccou, et al. 2008).

The approach here is to consider the cost required to crash each of the resources required to execute an activity as random variables with known distributions. As such, requires one to provide a probability distribution function (pdf) for each uncertain parameter, i.e. resources and sub-resources' crashed costs. The problem of determining the distributions for each resource and/or sub-resource is reduced by limiting the used distributions to normal, beta and triangular distributions as the three probability distributions that are considered for the purpose of this study. These distributions presented good performance in representing the risks associated with construction projects. Also these parameters and their respective probability distributions are assumed to be independent in order to reduce the complexity of the problem.

To start the simulation, first the resources and sub-resources required to execute each activity should be determined. For each of these resources, the variation of their cost over the crashed duration of the activity, i.e. its time-cost relation is to be selected. As these resources are selected to have uncertainty associated with their crashing cost, a range and a probability distribution is to be assigned to represent this uncertainty. If a continuous relation is considered for each of the (sub)-resources, this range is to be identified at the completely crashed cost, and interior ranges are calculated proportionally. Otherwise, such range is to be determined at each feasible crashing point within its time-cost chart. The ranges are defined by their percentage variation from the crashed cost at any given point (see Figure 4.5); e.g. if the estimated crashed cost is probable to be α% more than the originally estimated crashed cost (CC) and β% less, the range over which random values are to be selected will be $((CC\times (1-\beta/100)), (CC\times (1+\alpha/100))$.

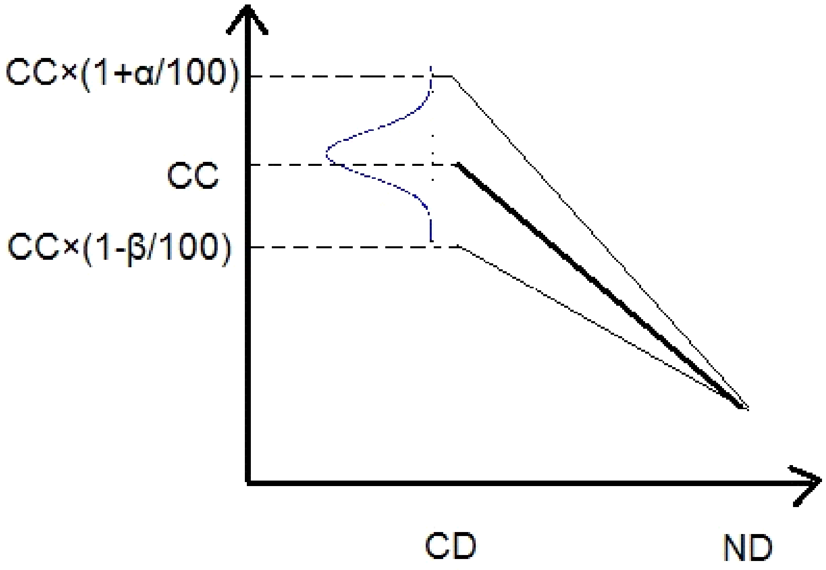


Figure 4.5: Risk assessment in resource level (linear time-cost relation)

After determining the range over which each resource’s crashed cost varies and the probability distribution associated with that range, Monte Carlo simulation is to run for the equation of total crashed cost of the activity; such total crashed cost will be summation of the crashed cost required for each of the resources and their sub-resources at any given duration as shown in Equations 4.8- 4.12:

$$CC_i = CCm_i + CCe_i + CCl_i + CCs_i \tag{4.8}$$

$$CCm_i = \sum_{j=1}^{n} CCm_{ji} \tag{4.9}$$

$$CCe_i = \sum_{j=1}^{n} CCe_{ji} \tag{4.10}$$

$$CCl_i = \sum_{j=1}^{n} CCl_{ji} \tag{4.11}$$

$$CCs_i = \sum_{j=1}^{n} CCs_{ji} \tag{4.12}$$

Where:

Cc_i= Crashed cost of the activity i

Cc_{li}=Crashed cost of labour required for activity i

Cc_{mi}=Crashed cost of material required for activity i

Cc_{si}=Crashed cost of sub-contractor required for activity i

Cc_{ei}=Crashed cost of equipments required for activity i

Cc_{eji}=Crashed cost of equipment j required for activity i

n=number of sub resources required for each resource

After running Monte Carlo simulation for equations above, the resulted crashed costs are fitted to a probability distribution. Such distribution shows the variations of crashed cost of each activity.

Having established the final crashed cost distribution for each activity, the most probable value of the distribution, i.e. its mean is used as the revised crashed cost for that activity. Subsequently, the revised cost slope for each activity is calculated in the same manner it was calculated for itemized uncertainty consideration, i.e. through using Equation 4.7. The activities are then queued for crashing based on these revise cost slopes as explained in section 4.3.1.

Project’s total cost is then calculated to generate the time-cost curve at the project level. It should be noted that in generating project total cost, the indirect cost is considered without uncertainty for it consists, typically, of a set of known cost items (e.g. overhead costs, site supervision, etc.), which are estimated with good level of certainty.

Further, the probability (P_i) of not exceeding considered risk impacts (α_i) is assumed to be over 90 percent for each activity that is deemed to have risk. It is also assumed that the crashed cost of the activities that are not deemed risky will not be exceeded, by 100 percent certitude. As such, calculated total cost for the project will not be exceeded by over ninety percent certitude.

As well, the probability of not exceeding the generated project total cost is calculated using the weighted formulation depicted in Equation 4.13. The variation of this probability over the crashed range is also shown in Figure 4.6. It should be noted that in generating these probabilities, it is assumed that the contractor is confident about the estimated normal cost of each activity and as such, the probability of not exceeding project’s normal cost is considered to be 100%. Because of the probabilities considered in activity level (P_i), as stated previously in risk modeling, probability of not exceeding project total cost (PP_i) will be over 90 percent throughout crashing process.

$$PP_i = \frac{1}{(IC_i + C_i + \sum_{i=1}^{n} \Delta c_i)} \times (IC_i + C_i + \sum_{i=1}^{n} (\Delta c_i \times P_i)) \tag{4.13}$$

Where:

PP_i: Probability of not exceeding project total cost at the i^{th} crashing iteration

C_i: Initial project direct cost at i^{th} crashed duration (without risk consideration)

IC_i: Project indirect cost at i^{th} crashed duration

n: Total number of crashing iterations

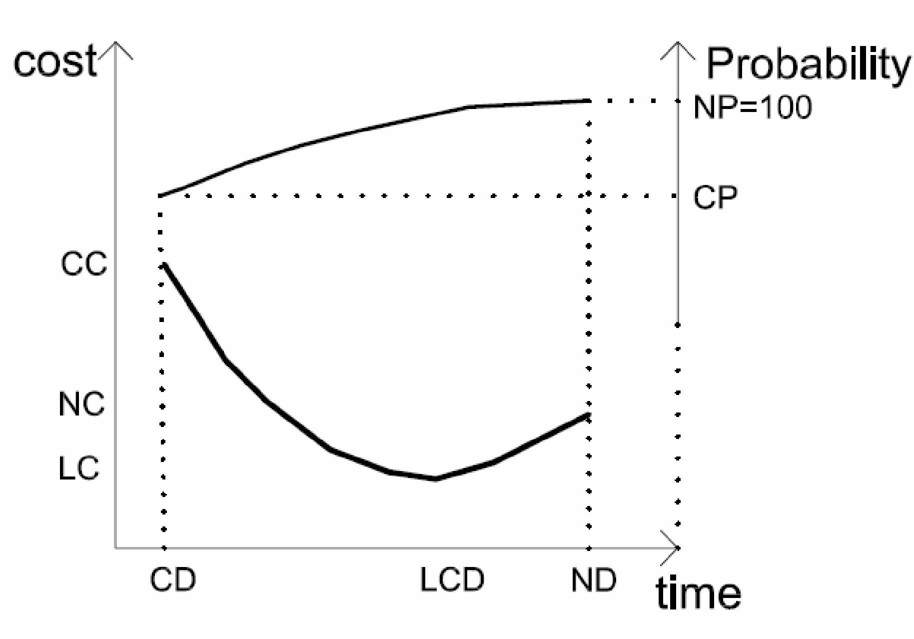


Figure 4.6: Project time-cost curve and the associated probability

4.4. Queuing activities for crashing

After defining the attributes to be considered in the schedule crashing process and their relative weights {W}, revising critical activities' cost slopes based on consideration of the uncertainty and risk associated with their crashing cost and prioritizing these activities with respect to their revised cost slopes, priority setting for crashing individual project activities with respect to all the decision attributes can then be carried out using the steps described subsequently.

Accordingly, the activities on the critical path of project network are queued for crashing based on the priority vector {P}, which accounts for their relative importance with respect to each attribute. As it is well known in construction management, the critical path is defined as the sequence of activities that must be completed on schedule for the entire project to be completed on schedule. This is the longest duration path through the project activity network. As such, if an activity on the critical path is accelerated by one day, then project total duration will be reduced by one day.

In order to queue critical activities, the decision maker is to carry out pair-wise comparisons among the activities considered for crashing on each critical path, with respect to each of the attributes. A square evaluation matrix of the order m×m will then be formed for each attribute (i.e. “n” evaluation matrices of size m×m are generated) in which “m” represents the number of activities considered for crashing on each critical path or the number of clustered activities on that critical path as described earlier in defining decision environment explained in Step 2 of section 4.2. These m×m evaluation matrices are generated in a similar manner to the decision matrix described in Step 3 above. For example in the m×m Evaluation matrix with respect to attribute A1 , e_{ij} is assigned a value of 1 only when Activity i (Ac_i) is more important than Activity j (Ac_j) with respect to decision attribute A1 (see Figure 4.7).

$$\begin{array}{c|cccc|} & Ac_1 & Ac_2 & \cdots & Ac_m \\ \hline Ac_1 & e_{11} & e_{12} & \cdots & e_{1m} \\ Ac_2 & e_{21} & a_{22} & \cdots & e_{2m} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ Ac_m & e_{m1} & e_{m2} & \cdots & e_{mm} \\ \hline \end{array} \qquad \begin{array}{c} (E_{ji}) \\ \left\{\begin{array}{c} E_{1i} \\ E_{2i} \\ \cdot \\ \cdot \\ \cdot \end{array}\right\} \\ \underline{\quad E_{mi} \quad} \\ 1.00 \end{array}$$

Figure 4.7: Sample activities' evaluation matrix and the related normalized vector

The emphasis vectors {E_{ji}} which represents how important each activity is with respect to a given decision attribute, is then generated using the same procedure described for calculating the relative weights of the attributes, i.e. using Equations 4.1 to 4.3. It should be noted that each of these "n" emphasis vectors is of m×1 order. Accordingly, the priority vector can be calculated using Equation 4 and transferred to the third process of the developed method, i.e. the iterative schedule compression.

The priorities for crashing the critical activities being considered (P_j) can then be generated as follows (Marazzi 1985):

$$P_j = \sum_{i=1}^{n} (E_{ji} \times w_i) \tag{4.14}$$

Where

E_{ji}= emphasis coefficient representing importance of Activity j (Ac_j) with respect to attribute i (A_i);

Or simply by using the matrix formulation bellow:

$$\left[\left\{E_{j1}\right\}\left\{E_{j2}\right\}\ldots\left\{E_{jn}\right\}\right]_{m\times n}\times\left\{W\right\}_{n\times 1}=\left\{P\right\}_{m\times 1} \tag{4.15}$$

The emphasis vectors described above can further be modified to account for the uncertainty and risk associated with the individual attributes at the activity level, using the developed methodology described below:

Risk assessment is modeled the same manner it was used previously in itemized risk assessment for activity crash costs; i.e. by using the “probability–impact matrix” shown in Figure 16. The risk associated with each attribute is quantified based on its severity of impact and probability of occurrence. For this purpose, the decision maker is to assign values for these severities and probabilities using a qualitative scale from very low to very high (i.e. very low (VL), low (L), medium (M), high (H) and very high (VH)). The severity here represents the level of impact of the risk being considered on crashing individual activities. The qualitative scale described above is mapped to a numerical scale ranging from 1 to 5 (e.g. 1 for very low, 2 for low and 5 for very high). The probability-impact matrix will then be generated for each activity with respect to each attribute as shown that figure. The degree of risk associated with each activity with respect to each attribute is then calculated by multiplying the probability of occurrence of that risk by its severity of impact (Stackpole 2010) and normalizing it using Equations 4.16 and 4.17. The modified emphasis vectors can then generated using Equations 4.18 and 4.19.

R_{ji}= Risk probability × Risk impact (4.16)

$$\alpha_{ji} = (R_{ji}/25) \tag{4.17}$$

$$[R_i] = \begin{bmatrix} 1-\alpha_{1i} & 0 & \dots & 0 \\ 0 & 1-\alpha_{2i} & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \dots & 1-\alpha_{ji} \end{bmatrix}_{\mathrm{m \times m}} \tag{4.18}$$

$$\{E*_{ji}\} = [R_j] \times \{E_{ji}\} \tag{4.19}$$

Where:

R_{ji}= Degree of risk calculated for activity j (Ac_j) w.r.t. attribute i (A_i);

α_{ji} = Normalized risk associated with activity j (Ac_i) w.r.t. A_i;

[R_i]= The risk matrix for the activities w.r.t. attribute i (A_i);

{E^*_{ji}}= Modified emphasis vector w.r.t. attribute i based on considering risk;

It should be noted that each element in the modified emphasis vectors "e_{ji}" accounts for the risk and uncertainty associated with i^{th} attribute being considered for crashing of the j^{th} activity.

As such, priorities established before is modified using following matrix formulation:

$$\left[\{E*_{j1}\}\{E*_{j2}\}\ldots\{E*_{jn}\}\right]_{m\times n}\times\{W\}_{n\times 1}=\{P*\}_{m\times 1} \tag{4.20}$$

Where:

$\{P^*\}$ = modified Priority vector

$\{E_{ji}\}$ = Emphasis vector w.r.t. attribute i

And m = the number of activities being considered for crashing

The iterative schedule compression process is then commenced focusing on the critical path(s) of the project network schedule. Activities are crashed in the sequence defined by the queue established in the previous process one unit of time in each iteration. The activity with top most priority will be crashed until a new critical path is generated or the activity reaches its non-crashable duration. Newly formed critical path(s), which may result from the progressive compression process, are treated likewise, as described above. The process of crashing will be continued until reaching the project least-cost duration, reaching the targeted duration or until no further crashing is possible.

4.5. Computational Algorithm

In this process, incremental schedule compression is applied based on the priority vector calculated in the previous process. In other words, after establishing the priorities for crashing critical activities in previous section, the activity that has the highest priority will be crashed one unit of time, only if there is one critical path in the network and only one activity has the highest priority. Otherwise, in the a) eventuality of having more than one activity with the same priority for crashing and/or b) the presence of more than one critical path in project network, a set of heuristic rules are applied as described below:

a) If there is one critical path but more than one activity share the same priority, following heuristic rules are applied to select the activity with the top most priority for crashing.

  i. The activity with more priority based on contractor’s judgment will be crashed first, if still more than one activity have the same priority, then:
  ii. Activity which finishes earlier is to be crashed first.

b) Else, if there is more than one critical path in the network, the following heuristic rules will be applied:

  i. The activity which is on more than one critical path, even if it is not among the activities with the highest priority for crashing on each critical path, should be crashed first only if the cost slope of that activity is less than the sum of the cost slopes of the critical activities that have the highest priority on each critical path; If a tie exist,
  ii. The Activity with more priority based on contractor’s judgment should be crashed; if a tie still exist, then:
  iii. The activity that finishes earlier should be crashed first.

Upon identifying the activity with the highest priority for crashing, using the procedure described above, CPM-type scheduling software is utilized in an interactive manner by shortening the critical activity with top most priority one unit of time to generate, in each iteration, a revised schedule. In other words, each time an activity is crashed, its revised duration and cost are imported to the scheduling software and the project is rescheduled. This activity will be crashed until it reaches its non-crashable duration or until another critical activity on the same critical path has a lower cost slope. The latter case can only occur if the activity being crashed has a non-linear time-cost relation. This crashing process continues until other non-critical activities become critical, resulting in one or more new critical path. In that event, the heuristic rules stated above are applied. If any of these two conditions exists, the process described previously is repeated.

The process of crashing continues until reaching the least-cost duration, the targeted duration or until no further crashing is possible.

4.6. Limitations

The developed method is not applicable, in its present formulation, to what is known as “linear projects” such as construction of highways and pipeline infrastructure projects which exhibit high degree of repetitive construction. As well, the automated software developed for the implementation of the proposed method operates in Microsoft integrated environment, which accepts project schedules in MS-Project format.

5. CHAPTER FIVE: COMPUTER APPLICATION (C-SCHEDULER)

5.1. General

The methods that are designed to solve the traditional and the general stochastic time-cost trade-off problems should be easy to apply to real projects in order to facilitate their use. Consequently, the methods should be integrated into a commercially available project management tool (such as Microsoft Project) to create an interface through which the methods can be applied. This implementation can allow the users to manage their projects and to address the time-cost trade-off problem in a single application. Finally, the software tool should provide the project manager with different execution plans along with their associated costs, to aid in their time-cost trade-off decisions.

In developing the proposed method, different tools had to be considered. As such selection of the tool to be used in developing the software system should satisfy certain features of these integrated tools. These features include availability of the selected tool, ability to integrate with other software systems, ability to conduct complex computations in short time and ability to provide user friendly interfaces. Since schedule compression requires data exchange and data storing and interfacing with commercial scheduling software, the development tool should be capable of providing a powerful support for such data exchange. In addition, the memory capacity must be made available in order to accommodate the combination and integration of different softwares that have to be activated at the same time. Therefore, it is preferable for the developed system to be able to run on a personal computer with reasonable memory consumption. For the above stated reasons and because of its inter-operational relations with other Microsoft software, Visual Basic 2010 has been selected for the development of the proposed method.

As a result, the developed method described in chapter 4 was implemented as computer application in Visual Basic environment as a proof of the concept presented in previous chapters. The computer software operates in Microsoft Windows' environment. The computer application is dynamically linked to MS-Project to facilitate the needed iterative data transfer to perform the project schedule compression. The developed application is user friendly throughout the execution of the three processes of the developed method. It provides the user with menus for the selection and /or addition of attributes to be considered in the crashing process. It provides also interactive graphical user interfaces (GUI) to facilitate the direct input of data required for generating the multi-attributed decision environment; i.e. pair-wise comparisons needed for the generation of decision, evaluation and risk matrices. Further, it generates a report of the execution plan in tabular and graphical formats. Figure 5.1 depicts the input and output of the developed system. A detailed description of the computer application as well as some of its features is provided in this chapter.

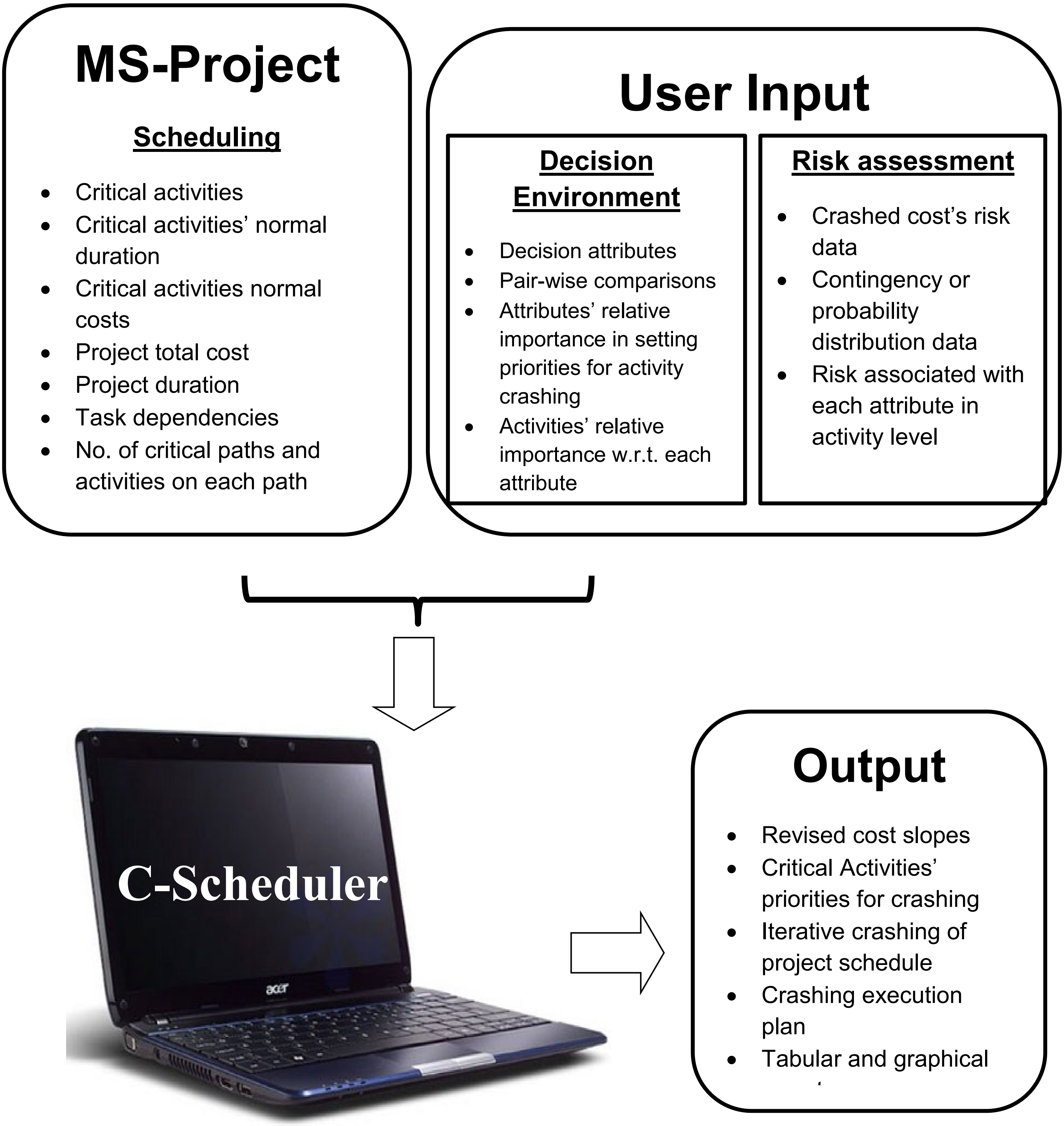


Figure 0.1: Input and output of the developed system

5.2. Scheduling

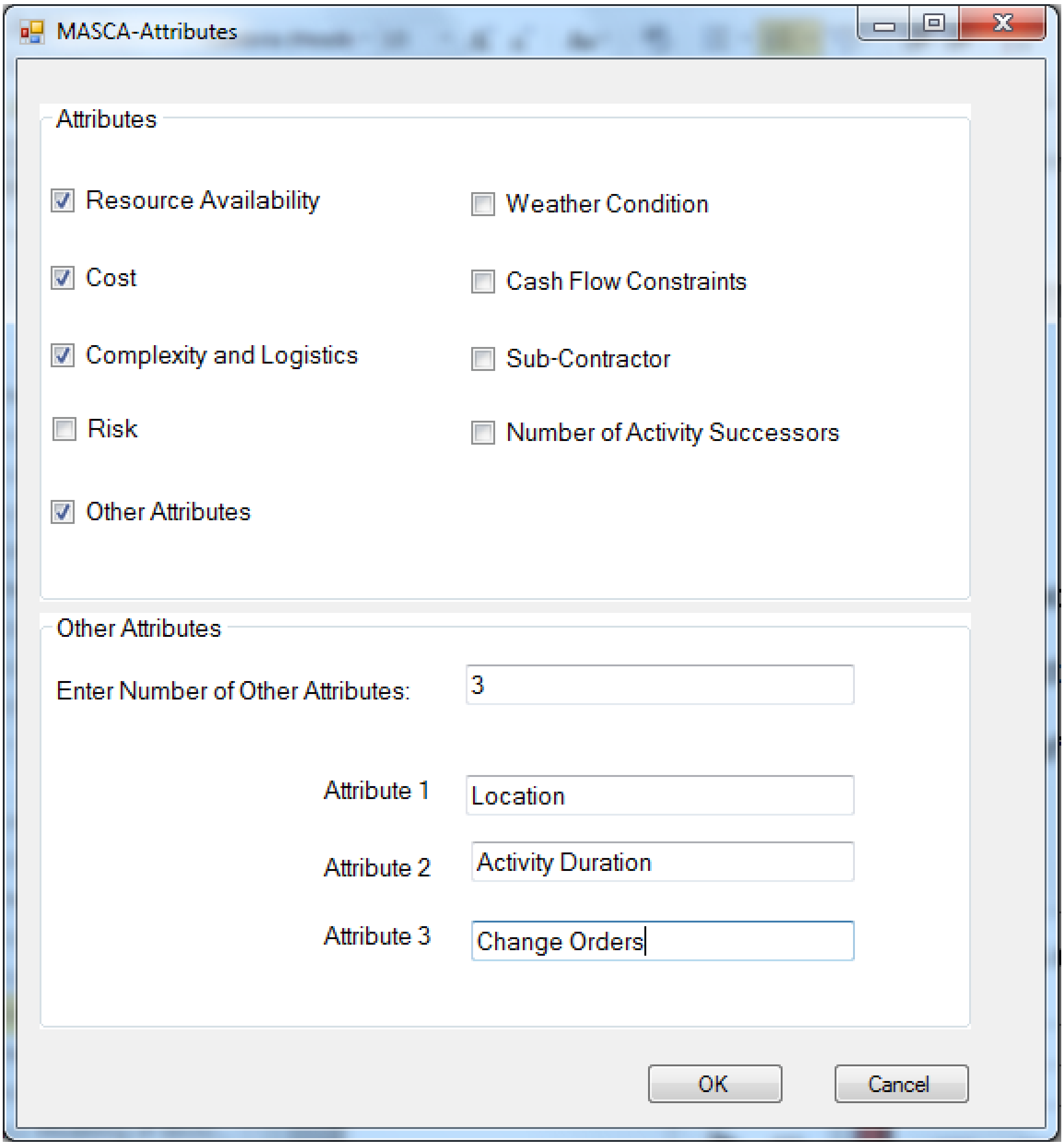
To begin the crashing process, first the critical activities along with their respective normal cost and normal duration should be identified. For this purpose, project schedule including its activities, their sequences and inter-relationships should be established. This is carried out in MS-Project environment as scheduling software. To extract the project information that is required to perform the schedule compression process, a macro is written in Visual Basic for Application (VBA) environment, Microsoft's integrated programming environment. This macro runs every time a project schedule created in MS-Project is opened through the Open menu in the computer application. It then classifies the activities based on whether or not they are critical. Numbers of critical paths that exist in the opened project as well as critical activities on each of these critical paths are then identified. For each critical path an array of size m ×4 is created in which m shows number of critical activities on it. These four columns of the array are filled out by task ID and task names of the critical activities that form this path, as well as their normal duration and cost. Further, the macro extracts project total direct cost, i.e. the summation of direct costs required to complete all project activities, as well as its total duration. These data are saved and transferred to the computer application (C-Schedule) through the dynamic link established for this purpose. Such link is established by calling the VBA macro from the computer application and saving data in the application in each iteration.

After each iteration, the updated duration of the activity that was selected to be crashed based on joint consideration of all decision attributes replaces its original duration and the project is rescheduled using that updated duration.

Subsequently, in case where as a result of iterative crashing and rescheduling of the project, new critical activities and consequently new critical paths are formed, these newly formed critical activities are identified and their respective scheduling data are transferred to the application. It should be noted that, as stated in proposed methodology, all critical paths within project network in this case are considered simultaneously in the crashing process, and then the heuristic rules are applied to find out the activities that are to be crashed on each path. Consequently, the dynamic link explained here facilitates data transfers between scheduling software (MS-Project) and the computer application; reducing the need for data entry and/or import by user.

5.3. Decision environment

To establish the decision environment for the crashing process, first its decision attributes should be identified. To facilitate this selection, a user interface (UI) is designed; in this UI, a number of decision attributes are offered to the user to select among them. These attributes are those factors that according to the results of the questionnaire survey, were found to be the most important factors considered by contractors and project managers in practice. Also, to provide more flexibility in considering all project dependent decision attributes, an option is provided to the user to add other attributes to those offered by application as shown in Figure 5.2.



Attributes

☑ Resource Availability
☑ Cost
☑ Complexity and Logistics
☐ Risk
☑ Other Attributes
☐ Weather Condition
☐ Cash Flow Constraints
☐ Sub-Contractor
☐ Number of Activity Successors

Other Attributes

Enter Number of Other Attributes: 3

Attribute 1 Location
Attribute 2 Activity Duration
Attribute 3 Change Orders

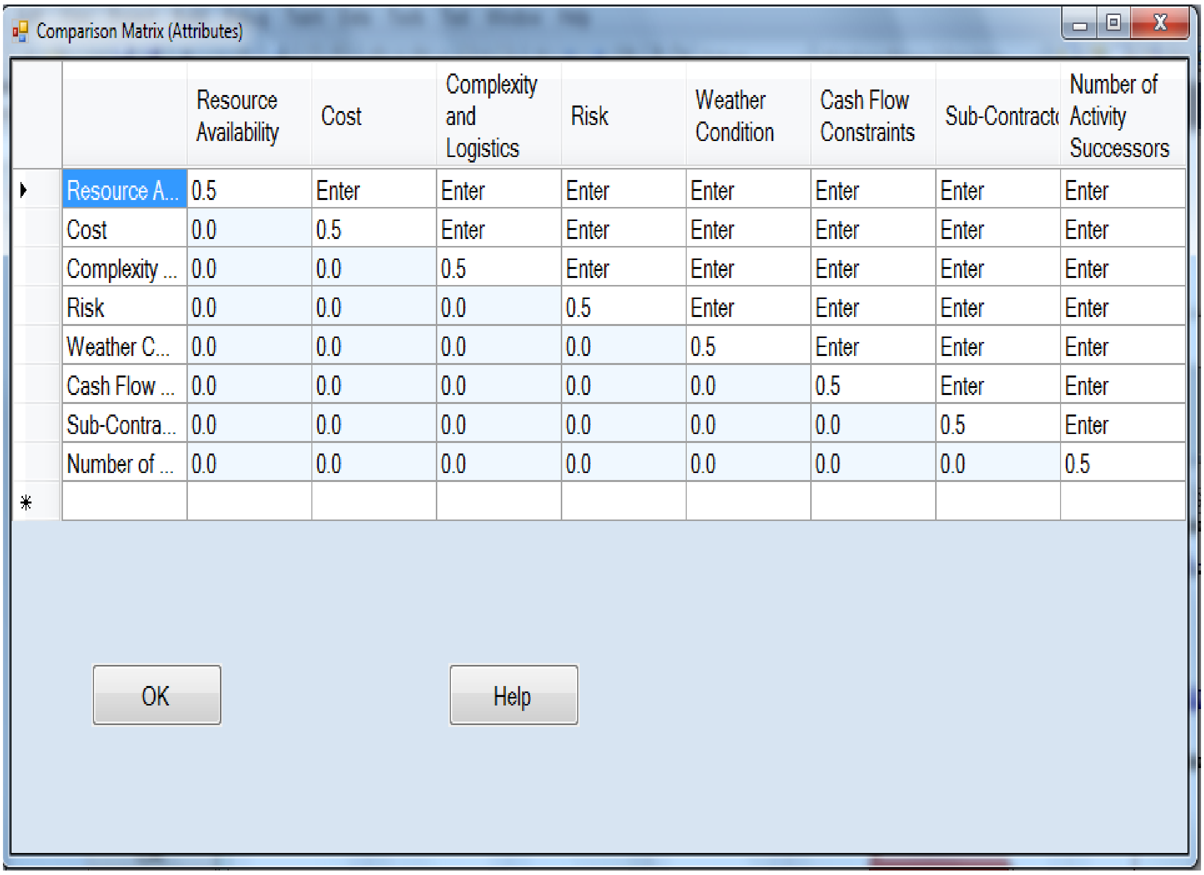
Figure 0.2: GUI to select attributes

These selected attributes are used to generate the decision matrix as explained in Chapter 4. This matrix is shown to the user in form of a user interface to perform the needed pair-wise comparison between selected attributes (see Figure 5.3). In that matrix, the diagonal elements are automatically filled out with value of 0.5 as explained previously. The user is required only to enter such comparisons for the upper triangle of the matrix.

The lower triangle is filled out automatically based on the numbers entered in matrix’s upper triangle elements; i.e. if element a_{ij} is assigned value of 1 meaning that attribute i is more important than attribute j, element a_{ji} is will be assigned value of 0.

```
Public Function MakeComparisonMatrixAct()

        Dim M As Integer = MASCA.M
        Dim CompMatrix(MASCA.N - 1) As ArrayList

        Dim WeightAct(M - 1) As Double
        Dim Sum, i, j, k, l, g As Integer
        Sum = 0
        'Create Comparison matrix from data entered to DGV
        Dim CompArrAct(M - 1, M - 1) As Double

        'Add each weight vector to CompMatrix
        For b = 0 To MASCA.N - 1

            'Copy data to a matrix
            For i = 0 To M - 1
                For j = 0 To M - 1
                    CompArrAct(i, j) =
Val(Comparison.DataGridView1.Rows(i).Cells(j + 1).Value)
                Next
            Next

            'fill lower triangle
            For g = 0 To M - 1
                For l = M + 1 To M - 1
                    If CompArrAct(g, l) = 1 Then
                        CompArrAct(l, g) = 0
                    ElseIf CompArrAct(g, l) = 0 Then
                        CompArrAct(l, g) = 1
                    ElseIf CompArrAct(g, l) = 0.5 Then
                        CompArrAct(l, g) = 0.5
                    End If
                Next
            Next

            'create weight vector
            Dim WCompAct(M - 1) As Double

            For k = 0 To M - 1
                For g = 0 To M - 1
                    WCompAct(k) = WCompAct(k) + CompArrAct(k, g)
                Next
                Sum = Sum + WCompAct(k)
            Next

            For k = 0 To M - 1
                WeightAct(k) = WCompAct(k) / Sum
            Next

            'Return Weight vector
            Return WeightAct
```

```
            CompMatrix(b).AddRange(WeightAct)
        Next
        Return CompMatrix
    End Function
```



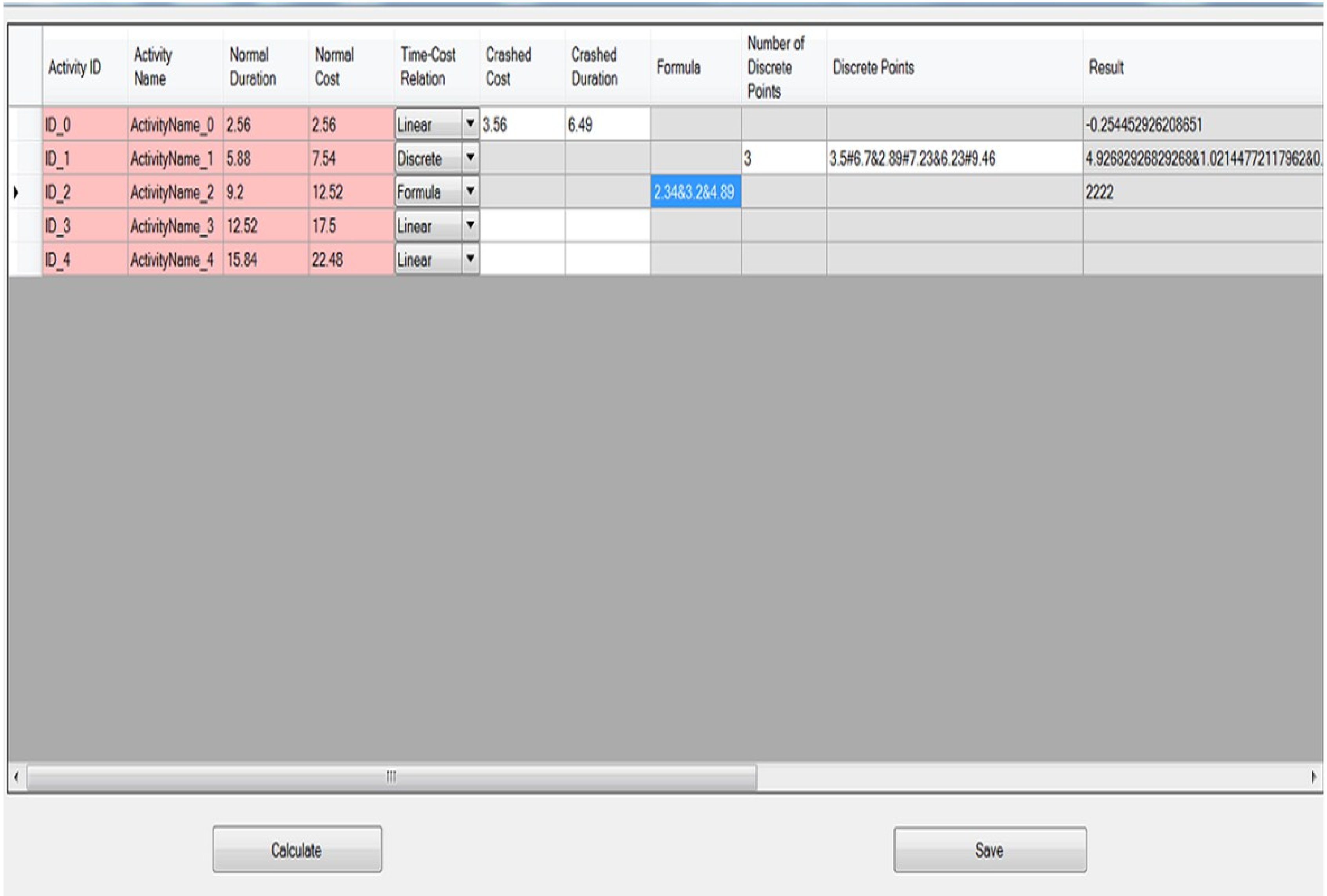
	Resource Availability	Cost	Complexity and Logistics	Risk	Weather Condition	Cash Flow Constraints	Sub-Contract	Number of Activity Successors
Resource A...	0.5	Enter	Enter	Enter	Enter	Enter	Enter	Enter
Cost	0.0	0.5	Enter	Enter	Enter	Enter	Enter	Enter
Complexity ...	0.0	0.0	0.5	Enter	Enter	Enter	Enter	Enter
Risk	0.0	0.0	0.0	0.5	Enter	Enter	Enter	Enter
Weather C...	0.0	0.0	0.0	0.0	0.5	Enter	Enter	Enter
Cash Flow ...	0.0	0.0	0.0	0.0	0.0	0.5	Enter	Enter
Sub-Contra...	0.0	0.0	0.0	0.0	0.0	0.0	0.5	Enter
Number of ...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5

Figure 0.3: GUI to perform pair-wise comparisons

5.4. Cost slope data

In that application, the user is then prompted to select imported activities' time-cost relation and to enter their respective crashed cost and crashed duration. In the case of discrete or piece-wise time-cost relations, the number of discrete points or break-points is to be identifies, and their associated cost and duration should be entered. For this purpose, another graphical user interface is designed as shown in Figure 5.4. The first 4 columns of the table, namely Activity ID, Activity Name, Normal Cost and Normal Duration are filled out automatically from the critical matrix generated using the MS-Project VBA macro. Hence, these columns are read only and pink back grounded to be distinguished easily.

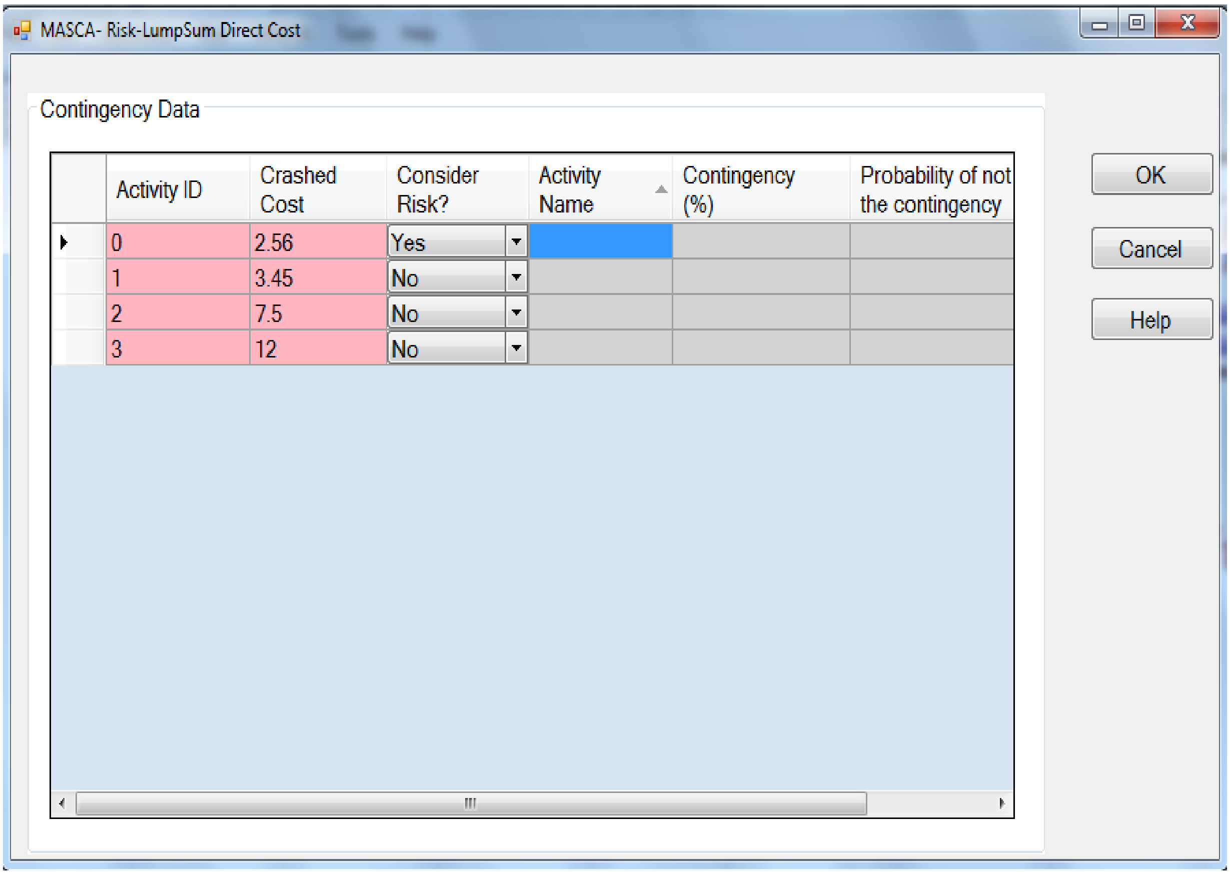
The time-cost relations and crashed data entered by user are then saved to be used in other iterations until needed. Sample code on cost slope calculations is provided in Appendix IV.



	Activity ID	Activity Name	Normal Duration	Normal Cost	Time-Cost Relation	Crashed Cost	Crashed Duration	Formula	Number of Discrete Points	Discrete Points	Result
	ID_0	ActivityName_0	2.56	2.56	Linear ▾	3.56	6.49				-0.254452926208651
	ID_1	ActivityName_1	5.88	7.54	Discrete ▾				3	3.5#6.7&2.89#7.23&6.23#9.46	4.92682926829268&1.02144772117962&0.
▸	ID_2	ActivityName_2	9.2	12.52	Formula ▾			2.34&3.2&4.89			2222
	ID_3	ActivityName_3	12.52	17.5	Linear ▾						
	ID_4	ActivityName_4	15.84	22.48	Linear ▾						

Figure 0.4: Graphical User Interface (GUI) to enter activities' cost data

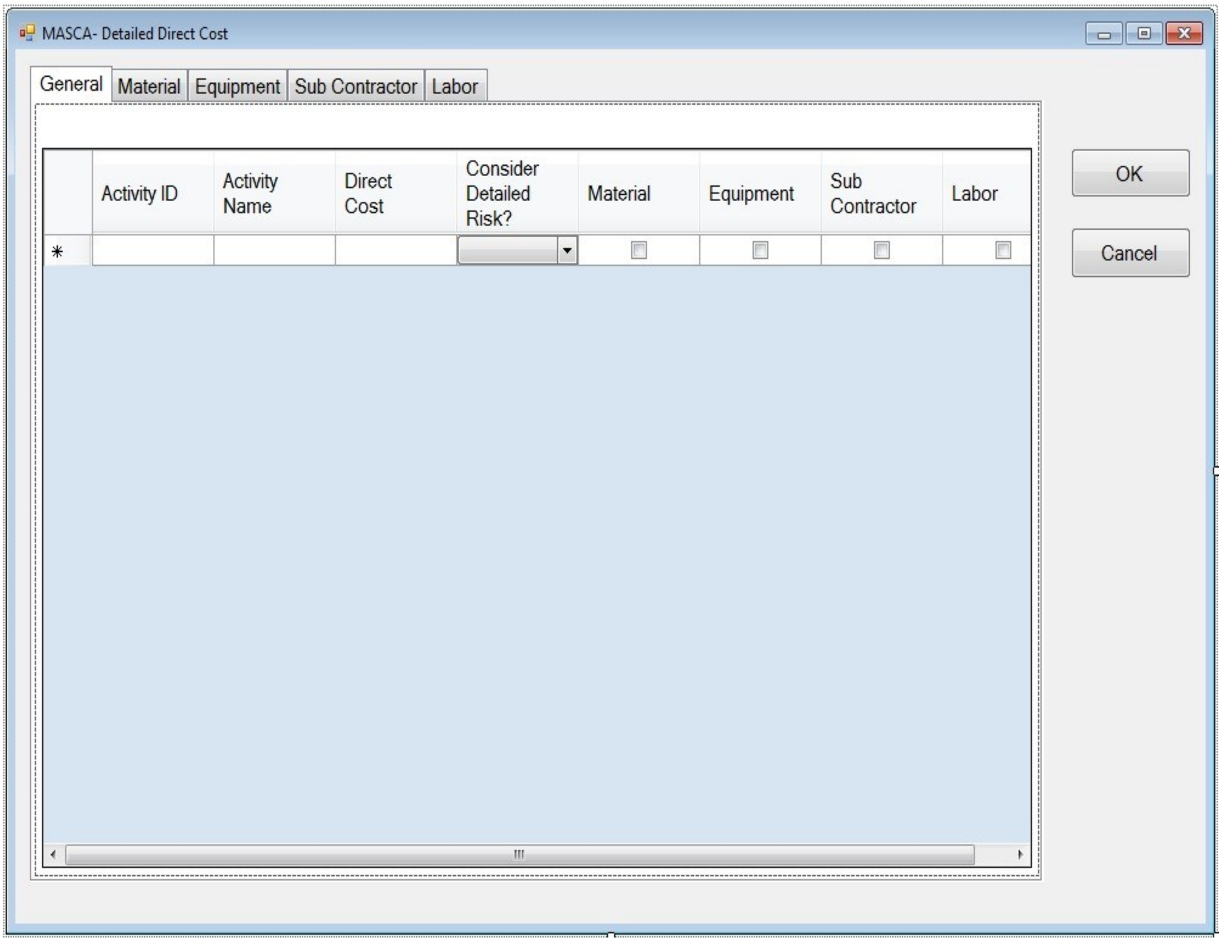
After time-cost relations are determined and cost slopes are calculated for each critical activity, the user is then to determine whether or not risk is to be considered for the crashing cost associated with each activity. For the activities for which risk is to be considered, based on whether or not detailed direct cost of the resources required to perform that activity are available or not, user is to select the method of risk assessment. In the case where lump sum risk is selected to be considered for an activity, the contingency associated with its crashed cost (α), expressed as percentage of its crashed cost, and the probability (P) of not exceeding that contingency are to be entered (Figure 5.5). Similar to the previous UI, the first 2 columns of this table are also filled from critical matrix imported from Ms-Project.



Activity ID	Crashed Cost	Consider Risk?	Activity Name	Contingency (%)	Probability of not the contingency
0	2.56	Yes			
1	3.45	No			
2	7.5	No			
3	12	No			

Figure 0.5: GUI to enter contingency data (Lump Sum Direct Cost)

In cases where detailed direct cost is selected to be considered, first user is to determine which resources (e.g. material, equipment, etc.) are considered to have uncertainty associated with their crashing cost (Figure 5.6). Subsequently, for each selected resource, the number of sub-resources as well as the crashing cost for each sub resource and the range over which the this crashing cost is to be changed and the probability distribution associated with each sub-resource along with its characteristics (i.e. mean, standard deviation, max and min, etc) are to be defined (Figure 5.7).



	Activity ID	Activity Name	Direct Cost	Consider Detailed Risk?	Material	Equipment	Sub Contractor	Labor
*					☐	☐	☐	☐

Figure 0.6: GUI to determine detailed risk components

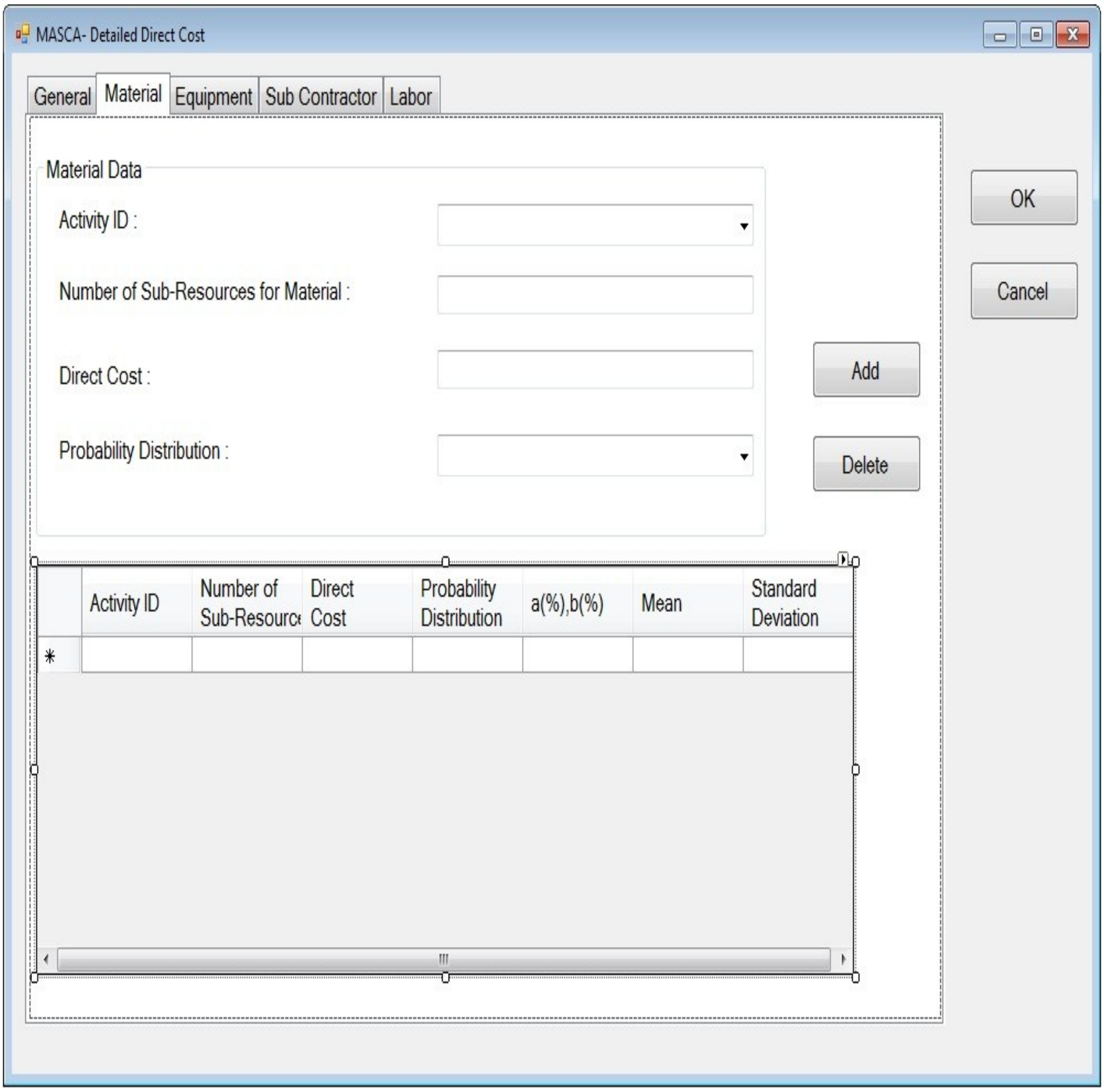


Figure 0.7: GUI to enter risk data for each resource

After these UI's have been used to gather user data for the risk associated with each activity, the cost slopes calculated previously are revised and activities are queued using these revised cost slopes.

5.5. Prioritizing

As explained in proposed methodology, after cost slopes have been calculated, the activities on each critical path are to be prioritized with respect to each of the selected decision attributes. For this purpose, the comparison matrix shown in Figure 5.2 is populated for each decision attribute and user is to compare activities on each critical path with respect to that decision attribute.

After the activity based on joint consideration of all decision attributes has been identified, its revised cost slope and duration is transferred to MS-Project (through the dynamic link). This additional cost needed for crashing that activity is then used to add a point to project total cost versus duration chart.

5.6. Crashing execution plan reports

The project total cost and the probability of not exceeding this total cost, calculated using Equation 4.12, are stored and then plotted to generate the time-cost curve shown in Figure 4.5. Sample of these graphical and tabular repots are shown in chapter 6 for the case examples. The revised duration of the activity being crashed is then transferred to the scheduling software to reschedule the project to progress with a new iteration.

The computational procedure described above is repeated until the activity reaches its crashed duration or until a new critical path is formed. In the latter case, the computational procedure explained above is applied to generate the revised cost slopes for the activities on the new critical path, which are deemed to have risk. If more than one critical path exists, these critical paths are crashed simultaneously. The added direct cost of the project is then calculated based on the summation of the least revised cost slopes of the activities selected for crashing on each path.

6. CHAPTER SIX: CASE EXAMPLES

6.1. General

To demonstrate the use of the proposed method and to illustrate its features, three case examples drawn from literature has been analysed. To illustrate how project crashing plans and their total cost will be affected by considering 1) a multi-attributed decision environment 2) increasing number of project activities, and 3) the risk associated with crashing cost of activities, each of these three has been applied to one of the examples. As such, different scenarios are considered for each example to catch variations in project total cost as well as changes in crashing execution plans.

6.2. Case example 1

Example project drawn from the literature (Geem 2010) was analyzed to demonstrate the use of the proposed method in considering multi- attributed decision environment and to illustrate its essential features. To enable a comparison, the discrete activity time-cost relationship presented by Geem (2010) was assumed linear. This assumption is reasonable since the data can be easily represented by linear relation. The project data is shown in Table 6.1. Project indirect cost equal to $1500/day has been considered to be consistent with the original example. The project network consists of 7 activities as shown in Figure 6.1. Three scenarios were generated from that example; in the first Scenario, which is referred to as base case, cost is considered as the only attribute; in Scenario 2, cost slope (CS) and contractor’s judgment (CJ), are considered and the first is deemed less important than the second; and in Scenario 3, three attributes are considered. The first two attributes are identical to those of Scenario 2 and the third is considered to be uncertainty (U) associated with estimated durations of the critical activities. In the latter scenario, importance order of U>CJ>CS is assumed in carrying out the binary pair-wise comparisons. Table 6.2 shows the execution plan for the base case. It should be noted that the contractor judgment factor used in this example accounts for availability of required resources, complexity and logistics pertinent to the work to be performed. Similarly, uncertainty accounts for the risk involved in performing the work. The results were then compared with those generated in the original example using Harmony Search method.

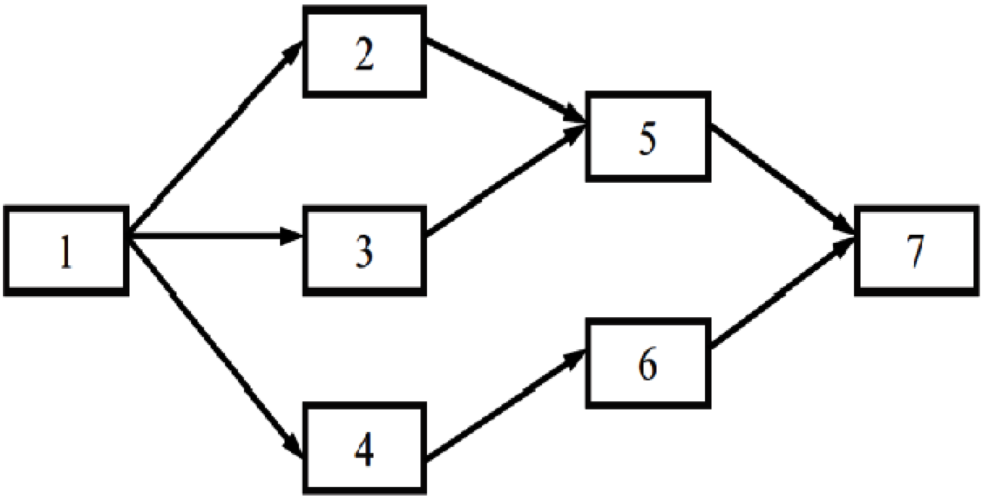


Figure 6.1: 7 activity network (Geem 2010)

Table 6.1: Project data of case example 1

Activity No	Total Float	Original Duration (days)	Crash Duration (days)	Normal Cost ($)	Crash Cost ($)	Cost Slope ($/day)	Priorities based on CS*	Priorities based on CJ**	Priorities based on U**
1	Critical	24	14	12000	23000	1100	4	1	5
2		25	15	1000	3000	200	2	3	2
3	Critical	33	15	3200	4500	72.22	1	2	1
4		20	12	30000	45000	1875	6	5	4
5	Critical	30	22	10000	20000	1250	5	4	3
6		24	14	18000	40000	2200	7	6	7
7	Critical	18	9	22000	30000	888.89	3	7	6

*The priorities for CS (cost slope) are for the first crashing cycle (e.g. Activity 3 has the highest priority for crashing with respect to cost slope in first crashing cycle)

** Priorities with respect to CJ (contractor’s judgment) and U (uncertainty) are assumptions made by author

Table 6.2: Execution plan of Scenario 1

Project duration (days)	Cost (1000$)			No of Activity compressed	Remarks
	Direct	Indirect	Total		
105	96.2	157.5	253.7	NA	
104 to 97*	96.76	145.5	242.26	3	
96 to 87	99.46	130.5	229.96	2,3	2 critical path exist, Rule No. 3 applies**
86 to 78	107.47	117	224.47	7	
77 to 68	118.47	102	220.47	1	
67	119.72	100.5	220.22	5	
66 to 60	141.6	90	231.6	4,5	Other critical path is formed

*the same activity is crashed over the indicated durations (e.g. Activity 3 is crashed from 105 days to 97 day) and the cost are for the end duration of each interval

** The activity which is on more than one critical path even if it is not among the activities with highest priority should be crashed first only if that activity would be less than sum of cost slopes for individually selected activities on each critical path

As shown in Table 6.2, after the 8th iteration (i.e. when project reaches to its 97 day duration) Activity 2 also becomes critical and another critical path that includes Activities 1, 2, 5 and 7 is formed. In this case there are three activates that are common in the two generated critical paths. As well there are 2 activities (Activities 2 and 3) that are parallel; one on each path. Heuristic rule No. 3 is applied in this case because the summation of the cost slopes of activities 2 and 3 ($272.22), is less than that of Activities 1, 5 and 7, individually paths, i.e. less than $1100, $2200 and $890, respectively. It should be noted that Activities 2 and 3 are to be crashed concurrently.

As to the second scenario, consideration of contractor's judgment and experience has led to assigning top most and least most priorities to activities 1 and 7, respectively. In this scenario, it is assumed that Activity 1 had to be done first in view of a late start of the project and that the resources of that activity were committed elsewhere by the contractor. And the seventh activity involved the use of new technology and as such was deemed risky to the contractor; resulting in postponing it's crashing to the end, i.e. after crashing the rest of project critical activities. The assumed priorities, in this scenario, are shown in Table 6.1. Table 6.3 summarizes the sequential operations of the generated execution plan in this scenario.

Table 6.3: Execution plan of Scenario 2

Project duration (days)	Cost (1000$)			No of Activity compressed	Remarks
	Direct	Indirect	Total		
105	96.2	157.5	253.7	NA	
104 to 95	107.2	142.5	249.7	1	
94 to 87	107.76	130.5	238.26	3	
86 to 77	110.46	115.5	225.96	2,3	2 critical path exist, Rule No. 3 applies
76	111.71	114	225.71	5	
75 to 69	133.585	103.5	237.085	5,4	Other critical path is formed
68	135.46	102	237.46	4	
67 to 60	142.6	90	232.6	7	

In the third Scenario, priorities are set for activity crashing based on joint consideration of uncertainties associated with activities' estimated duration and the two attributes considered in Scenario 2. As such, durations of activities 2 and 3 are considered to have the least uncertainty and, accordingly, gain higher priority for crashing. Contrary to that, activities 6 and 7 are deemed to have the highest uncertainty and were accordingly gained the least priorities. Further, according to judgment of contractor, Activity 6 should be accelerated at least 4 days (i.e. from days 75 to 72) to minimize its delayed impact on succeeding activities, although it causes project total cost to increase (see Figure 6.2). As such, it was assigned the maximum score with respect to contractor judgment during this period. Table 6.4 shows the execution plan for this scenario. The results of the analysis performed in the three scenarios along with those reported by Geem (2010) are presented in Figure 6.2. It is interesting to note that when considering only cost, i.e. in Scenario 1, the proposed method and that of Geem yielded very close project least-cost durations.

Table 6.4: Execution plan of Scenario 3

Project duration (days)	Cost (1000$)			No of Activity compressed	Remarks
	Direct	Indirect	Total		
105	96.2	157.5	253.7	NA	
104 to 97	96.776	145.5	242.276	3	
96 to 87	99.496	130.5	229.996	2,3	2 critical path exist, Rule No. 3 applies
86	100.746	129	229.746	5	
85 to 76	111.746	114	225.746	1	
75 to 72	125.546	108	233.546	5,6	Activity 6 should be accelerated 4days because of delays
71 to 63	133.556	94.5	228.056	7	
62 to 60	155.431	84	239.431	4,5	

As other factors were considered in addition to cost, i.e. in Scenarios 2 and 3, different least-cost project durations were obtained. It should be noted that because the set of factors considered for setting priorities as well as the relative importance assigned to them vary throughout the crashing process, the resulting chart is bound to have more than one local minimum (see Figure 6.2). Although consideration of other factors beyond cost may result in higher project cost, the resulting execution plan will be more practical and realistic as it accounts for actual project environments and their respective constraints as well as for the operating conditions of contractors.

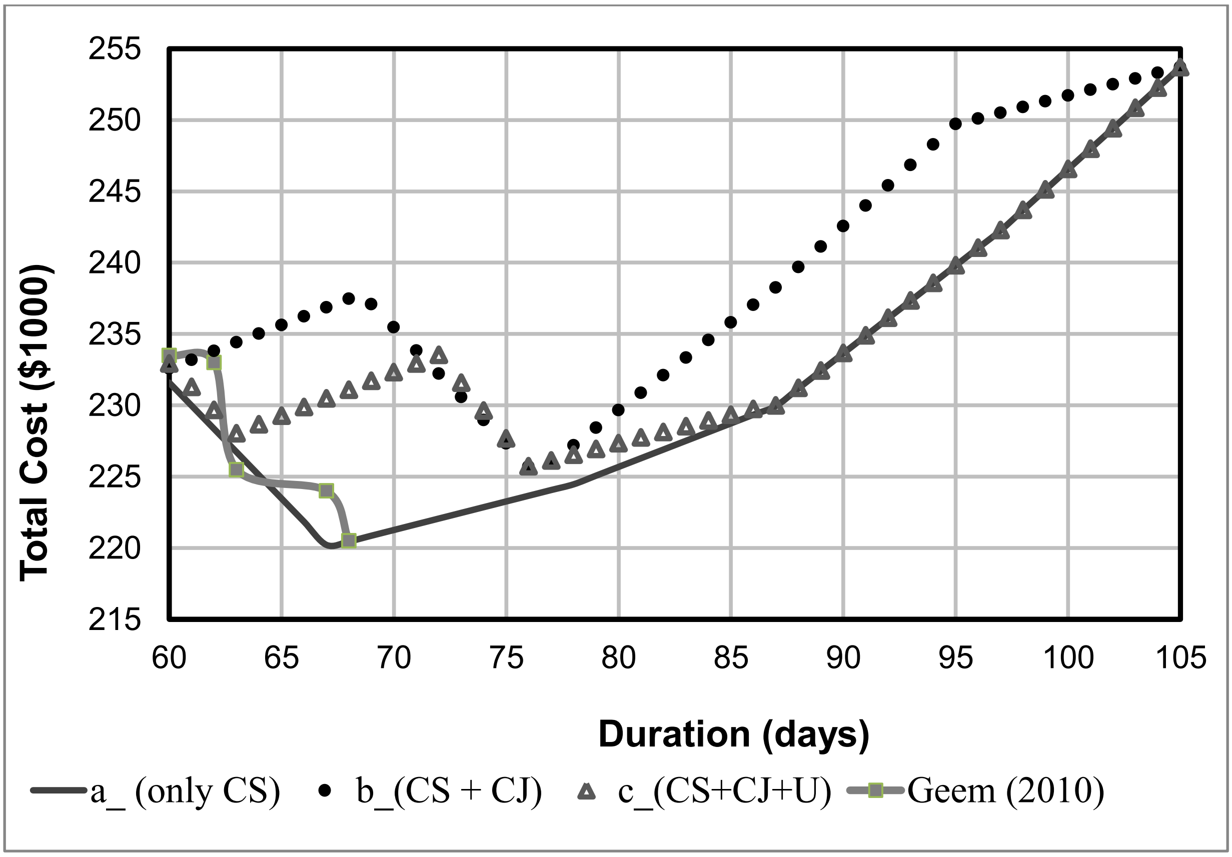


Figure 6.2: Comparison of the results

6.3. Discussion on case examples 1

As demonstrated through the numerical example, unlike other available methods, the new method presented here adds to the current literature by introducing the possibility of consideration of other important factors that contractors consider in practice. The results shown in Figure 6.2 and Table 6.5 demonstrate that when considering only cost, as in the first Scenario, the proposed method generates , in general, lower total costs compared to those generated by the HS method. For example, the project total cost at 68, 67, 62 and 60 days durations, is $220.5, 224, 233, and 233.5, respectively based on the HS method and $220.47, 220.22, 228.35 and 231.6, respectively based on the proposed method. Only in one case, where the project duration is 63 days, the proposed method resulted in slightly higher total cost of $226.72 than the $225.5 generated by the HS method.

Also, throughout the crashing process, i.e. from 105 days to 60 days, Scenarios 2 and 3 in which one and two factors are considered in addition to cost, generate more total cost for the project compared to Scenario 1 in which only cost is considered. Clearly the added cost pays for the flexibility and the consideration of additional factors, which are deemed important in the compression process. This is particularly applicable when contractors are instructed by owners and/or their agents to accelerate construction work. In that case contractors will be more concerned achieving the targeted schedule compression rather cost only, which they will be compensated for anyway.

Table 6.5: Comparison of the results

Project duration (days)	Project total cost (1000$)			
	Geem (2010)	Scenario 1 (cost only)	Scenario 2 (CS+CJ)	Scenario 2 (CS+CJ+U)
68	220.5	220.47	237.46	231.106
67	224	220.22	236.85	230.496
63	225.5	226.72	234.41	228.056
62	233	228.345	233.8	229.681
60	233.5	231.595	232.58	232.931

6.4. Case example 2

Two other project examples drawn from the literature (Stevens 1990 and Ahuja 1994) were also analyzed to find out the impact of project size on the generated execution plans. The first has a project network that consists of 6 activities connected through 5 events as shown in Figure 6.3a. The project data is shown in Table 6.6. The project has a normal duration of 16 days and a direct normal cost of $3800. Indirect cost is estimated to be $100 per day. The second has a relatively larger network that consists of 13 activities (see Figure 6.3b). The project has total duration of 70 days and total cost of $8600. The indirect cost is for this example is considered to be the same. Project data are shown in Table 6.8. Both these examples were analysed in the original articles using iterative crashing process and their results were compared with those generated using proposed method while integrating iterative crashing process with multi-attributed decision making environment.

In these examples also only a linear time-cost relation was considered for all activities.

The proposed method was applied to generate 2 scenarios for each example project; the base case which is based only on activities' added cost as in the original examples; and scenario 2 in which resource availability is considered as an attribute in addition to cost for setting activity priorities for crashing. In the latter scenario, the resource availability is deemed more important than activities' cost slope when formulating the decision matrix at the project level.

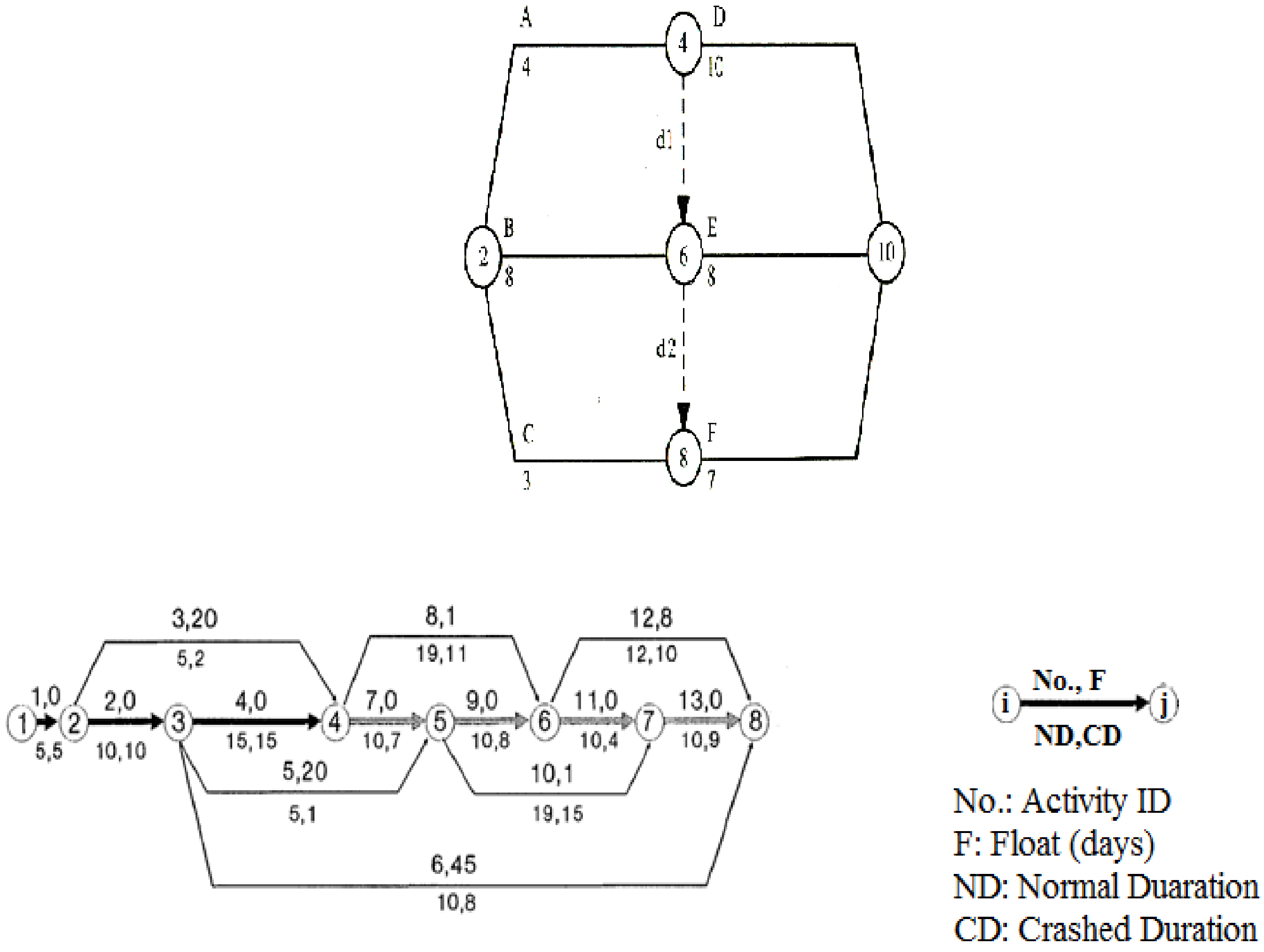


Figure 6.3: 6 and 13 activity project networks (Stevens 1990, Ahuja, Dozzy and AbouRizk 1994)

For the second scenario, the activities are queued for crashing based on considering their respective availability of resources in addition to cost. In the 6-activity project, activity E is assumed to have the needed resources for its crashing and upon generating the decision matrix at the project level and the 2-evaluation matrices at the activity level and generating the related 2 emphasis vectors, the priority vector was calculated based on the developed method. As such, activity E received the highest priority for crashing and is crashed first, although its cost slope is more than that of Activity B. likewise, activity D was found to have the least priority for crashing and so its crashing is postponed to the end. Table 6.7 shows the execution plans generated based on the two stated scenarios. Figure 6.4 depicts the change in project total cost over its crashed duration for the two scenarios. In the second example, however, the same scenario is assumed for activities 7 and 10. As such, the resources needed for crashing of activity 7 is assumed to be in hand without anticipated problems and it gains the top most priority for crashing after generation of respective evaluation matrices and emphasis vectors. Activity 10 is found to have the least priority because of anticipated difficulties in securing the needed resources for its crashing. The generated execution plans and their respective time-cost curve for this example are shown in Table 6.9 and Figure 6.5.

Table 6.6: Project data (6 activity network)

Activity	Duration		Cost $		Cost Slope
	Normal	Crash	Normal	Crash	
A	4	2	400	500	50
B	8	5	800	980	60
C	3	2	600	700	100
D	10	6	500	600	25
E	8	6	800	950	75
F	7	4	700	1000	100

Table 6.7: Execution plans (6 activity network)

Scenario 1 (base case)			Scenario 2 (Cost +Resource availability)		
Project Duration	Total Cost	Activity compressed	Project Duration	Total Cost	Activity compressed
16	5400	NA	16	5400	NA
15	5360	B	15	5375	E
14	5320	B	14	5350	E
13	5305	B,D	13	5360	A,B
12	5305	E,D	12	5370	A,B
11	5405	E,D	11	5355	B,D

Table 6.8: Project data (13 activity network)

Activity	Duration		Cost $		Cost Slope
	Normal	Crash	Normal	Crash	
1	5	5	150	150	---
2	10	10	200	200	---
3	5	2	250	310	20
4	15	15	900	900	---
5	5	1	750	1150	100
6	10	8	1000	1250	125
7	10	7	300	540	80
8	19	11	400	960	70
9	10	8	500	600	50
10	19	15	600	900	75
11	10	4	700	1210	85
12	12	10	600	800	100
13	10	9	250	300	50

Table 6.9: Execution plans (13 activity network)

Scenario 1 (base case)			Scenario 2 (Cost +Resource availability)		
Project Duration	Total Cost	Activity compressed	Project Duration	Total Cost	Activity compressed
70	8600	NA	70	8600	NA
69	8550	9	69	8580	7
68	8500	13	68	8630	7,8
67	8550	7,8	67	8680	7,8
66	8600	7,8	66	8630	13
65	8650	7,8	65	8650	8,9
64	8710	10,11	64	8845	8,9,10
63	8770	10,11	63	8905	10,11
62	8830	10,11	62	8965	1,11
61	8890	10,11	61	9025	10,11

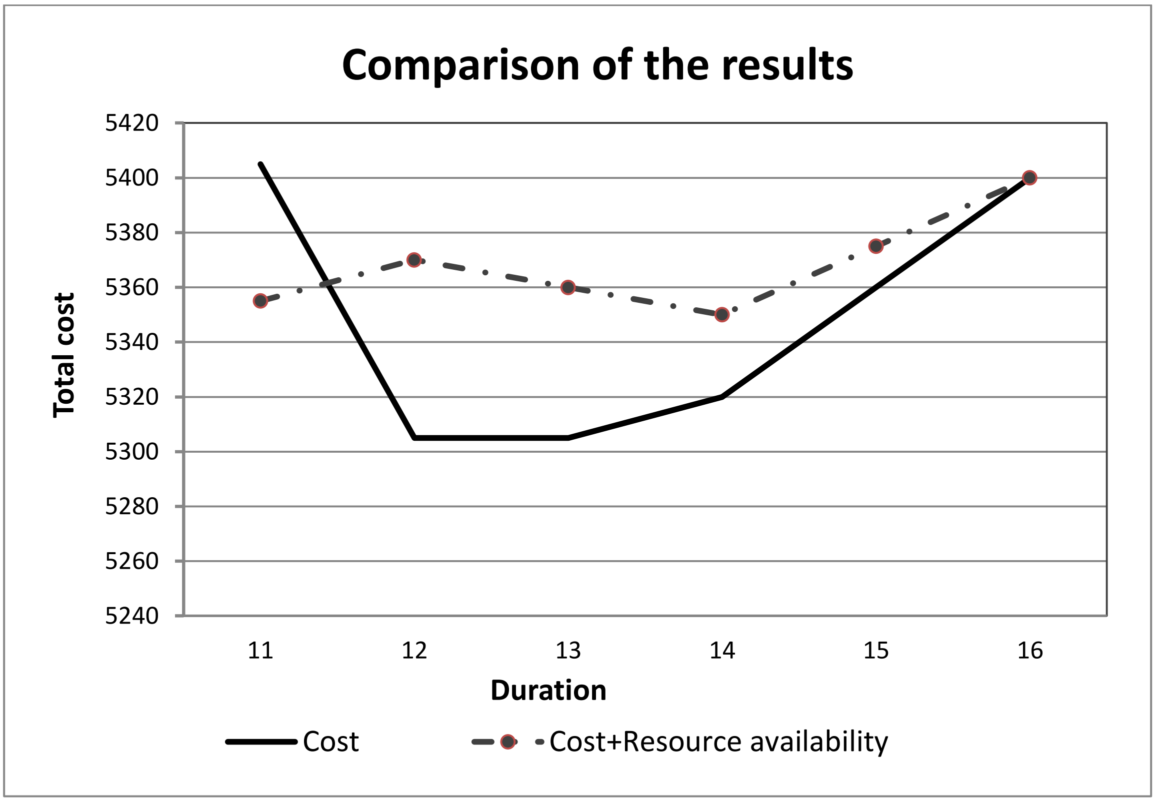


Figure 6.4: Comparison of the results for 6 activity network

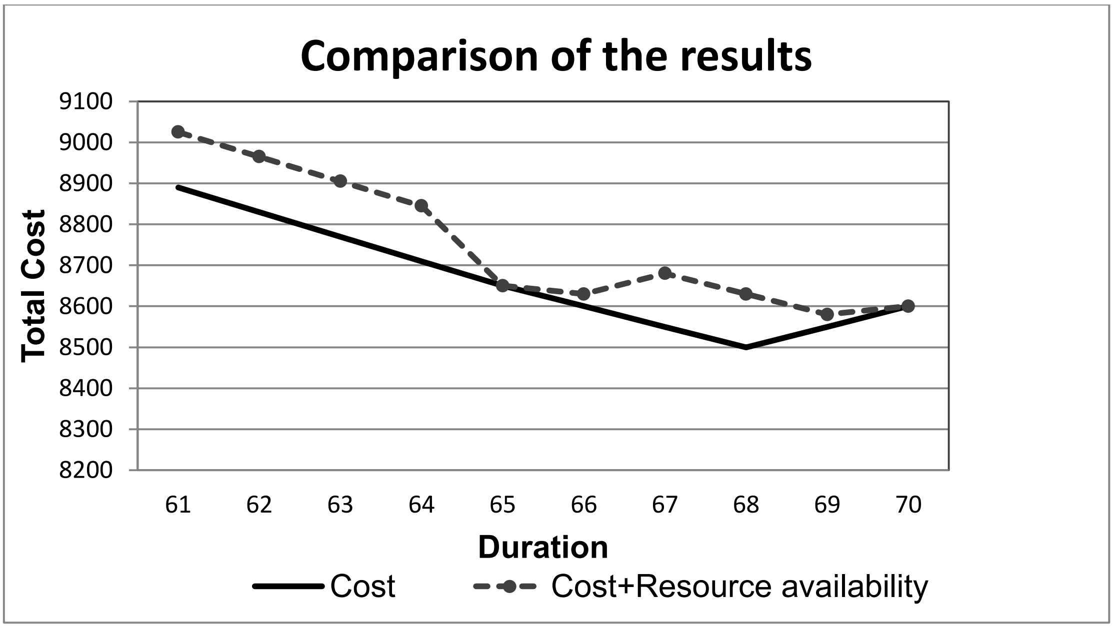


Figure 6.5: Comparison of the results for 13 activity network

6.5. Discussion on case examples 2

This case example also has shown how crashing execution plans can be affected by considering other factors in addition to cost. When considering cost only, proposed method also generated the same results as in original examples in less number of iterations and therefore in fairly less times. The results indicate that when other factor was considered in addition to cost, i.e. in scenarios 2, as expected, different least-cost project durations were obtained. It is also interesting to note that the larger the project network is, the more vivid is the impact of using multi-objective decision environment for project schedule compression will be. In such cases, even small changes in the decision factors and their relative importance largely impact the order by which activities are queued for crashing.

6.6. Case example 3

To demonstrate the impact of considering the risk associated with crashing cost of critical activities, a case example from literature (Cheng, Huang and Cuong 2011) was analyzed. The fast food outlet project network is shown in Figure 6.6. The project network consists of 14 activities as shown in that figure. The activity descriptions, precedence relationships, and time-cost functions are listed in Table 6.10. The project has a normal duration of 75 days and a normal direct cost of $94999.4. Indirect cost is estimated to be $600 per day. Three scenarios are considered in this example: (1) the base case in which priorities are set without risk consideration; (2) scenario 2 in which the severity of impact and probability of impact of the risk associated with crashing cost of the activities is considered in generating priorities for activity crashing as shown in Table 2; and (3) scenario 3 in which priorities established in scenario 2 are kept without consideration of the severity of risk impact on the crashing cost of the activity. Scenario 3, while not accounting for risk impact, is presented here to demonstrate the effect of priority-based queuing of activities for crashing. The results were then compared with those generated by other authors using Genetic Algorithms. Also, unlike previous examples in which only discrete or only linear time-cost relations were considered for activities, this example, as shown in Table 6.10, considered different time-cost relations such as linear, piece-wise linear, discrete, nonlinear and hybrid of these relations.

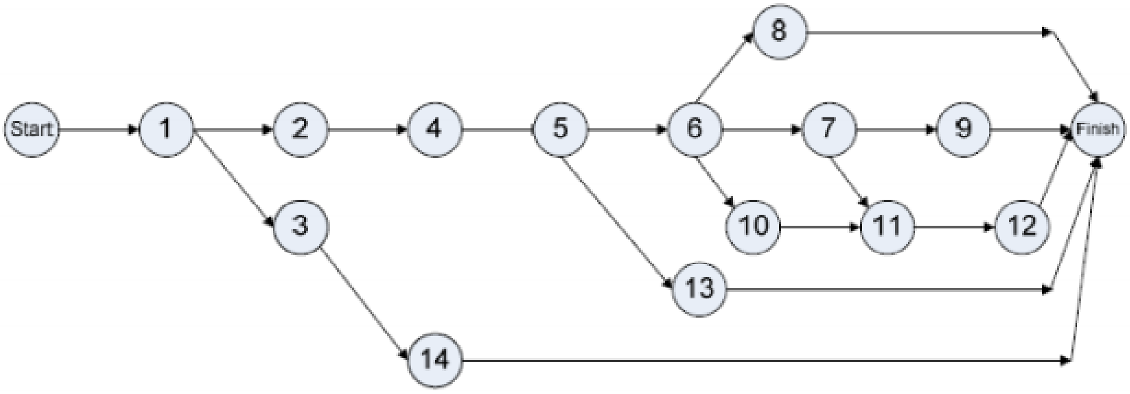


Figure 6.6: Fourteen activity project network (Cheng, Huang and Cuong 2011)

Table 6.10: Project data of case example 3

ID	Activity description	Immediate predecessors	Time-cost function	Type
1	Base slab	-	$C = 0.333D^2 - 4.8333D + 25$; $4 \le D \le 7$ $C = -0.25D^2 - 4.75D - 17$; $10 \le D \le 12$	Hybrid convex/Concave
2	Wall panels	1	$C = 0.5D^2 - 6.5D + 27$; $4 \le D \le 6$ $C = 4$ if $D = 8$	Hybrid Nonlinear/Discrete
3	Parking area	1	$C = -D + 20$; $8 \le D \le 12$ $C = -0.5D + 15$; $16 \le D \le 20$	Piecewise linear
4	Roof trusses	2	$C = -0.6667D + 7.3333$; $2 \le D \le 5$	Linear
5	Roofing	4	$C = -D + 6$; $1 \le D \le 4$	Linear
6	Windows and doors	5	$C = -D + 11$; $4 \le D \le 8$	Linear
7	Counter	6	$C = -0.4D + 6.2$; $3 \le D \le 8$	Linear
8	Walk-in refrigerator	6	$C = -0.4167D + 8.8333$; $2 \le D \le 8$	Linear
9	Counter equipment	7	$C = -1.3333D + 8.3333$; $1 \le D \le 4$	Linear
10	Kitchen equipment	6	$C = -0.4545D + 9.3182$; $4 \le D \le 15$	Linear
11	Floor coverings	7,10	$C = -0.8333 + 9.6667$; $2 \le D \le 8$	Linear
12	Furnishing	11	$C = -0.5D + 12.5$; $5 \le D \le 15$	Linear
13	Landscaping	5	$C = 10$ if $D = 3$ $C = 8$ if $D = 4$ $C = 7$ if $D = 5$ $C = 5$ if $D = 7$ $C = 4$ if $D = 9$	Discrete
14	Sign	3	$C = 5$ if $D = 3$ $C = 3$ if $D = 5$ $C = 2$ if $D = 6$	Discrete

When considering no risk associated with the crashing cost of activities (i.e. Scenario 1), activity 10 has the highest priority for crashing and will be crashed first. It is interesting to note that when considering the risk as defined above, the priorities are changed and activity 12 gains more priority than activity 10 for crashing. When comparing the results with those generated using other methods (Cheng et al. 2011), it is shown that results of considering only cost, i.e. project's minimum cost ($ 92.89×1000) and its associated duration(58 days), generated applying the proposed method are the same as those presented by Cheng et al. using GAs. Table 6.14 illustrates this comparison. as it can be shown in this table, the proposed method approaches much faster to the results and less number of iterations is needed to reach project optimum duration and cost. However, When considering the risk associated with crashing cost of the activities, project minimum cost ($ 93.89×1000), as expected, differs from those of considering only cost. Figure 6.7 illustrates the comparison of the results generated from each scenario along with the probability of not exceeding the project total cost generated from risk consideration in each crashed duration. Figure 6.7 shows clearly that considering the risk associated with crashing cost of activities leads to different project least-cost and its associated duration. Also as expected, when considering risk, the total cost of the project at any crashed duration is higher than that of the base case in which only cost is considered. Further, as we proceed to crashing the project, probability of not exceeding project total cost in each crashed duration decreases as is shown in that figure. Tables 6.12 and 6.13 show the execution plans for the 2 scenarios.

Table 6.11: Activities risk data and revised cost slopes

Activity	Csi($1000)	αi(%)	Pi	Δi($1000)	Cs'i($1000)
10	0.4545	24	0.9	1.6	0.6
12	0.5	6.7	0.9	0.6	0.56
4	0.6667	75	0.9	0.7	0.9

Table 6.12: Execution plan for Scenario 1

Scenario 1 (Base case)			
Duration	D.C.(Base)	T.C.(Base)	Compressed Activity
75	49999.4	94999.4	NA
74-68	53180.9	93980.9	10
67-58	58180.9	92980.9	12
57-55	60181	93181	4
54-49	65179	94579	11
48-45	68597	95597	10,7

Table 6.13: Execution plan for Scenario 2

Scenario 2 (with risk consideration)					
Duration	Δ	D.C.(Risk)	T.C.(Risk)	P	Compressed Activity
75	NA	49999.4	94999.4	100	NA
74-65	2418.5	55599.4	94599.4	99.99994806	12
64-58	1518.5	59699.4	94499.4	99.99980359	10
57-52	NA	64699.2	95899.2	99.99980359	11
51-49	2220.2	67399.2	96799.2	99.99968472	4
48-45	2802.2	71399.2	98399.2	99.99955276	10,7

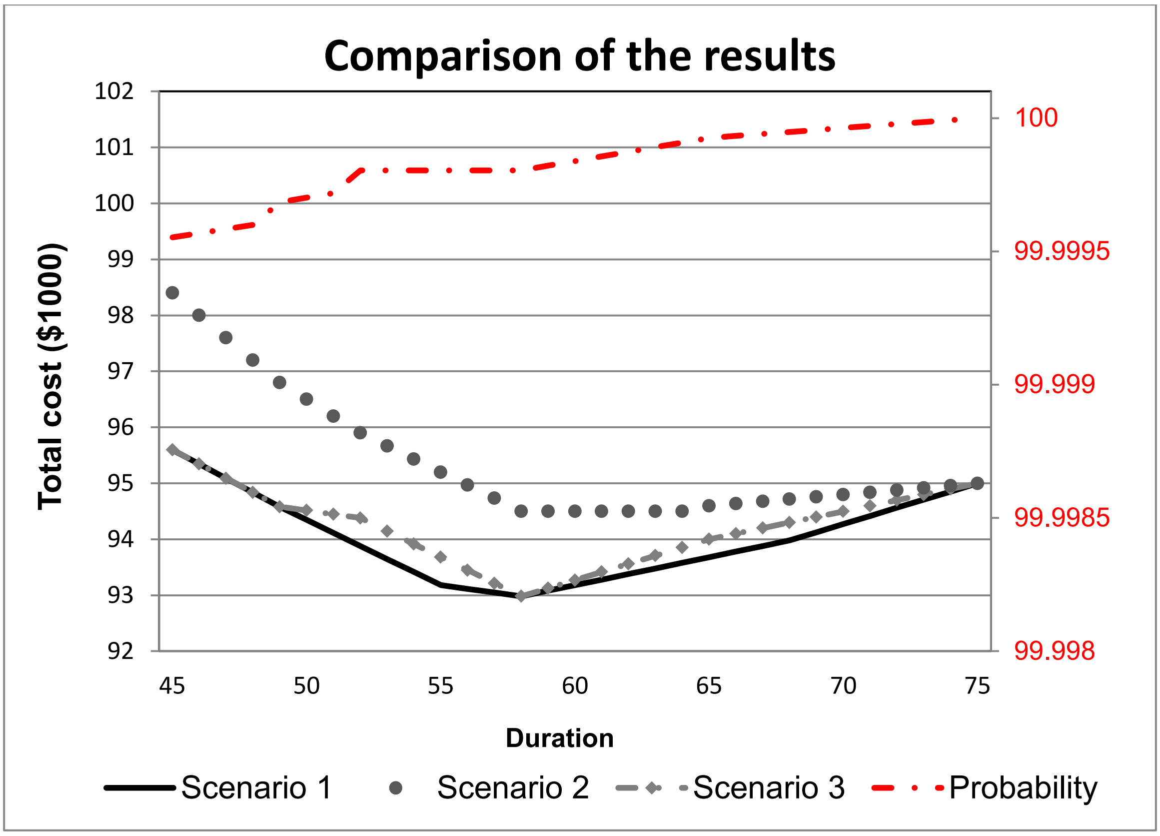


Figure 6.7: Comparison of the results

6.7. Discussion on case example 3

The first scenario in which no risk is considered yielded identical results to those of the original example. The second and third scenarios, however, demonstrated the impact of the risk associated with the crashing cost of project activities on the order by which activities are queued for crashing as well as on project total cost in each crashed duration. These results shows that considering the risk associated with crashing cost of activities has an influence on the order by which activities are queued for crashing. Consequently, such consideration causes different project total costs, at each crashed duration.

Table 6.14: Comparison of the results

	Scenario 1	Scenario 2	GA	CGA	KGA	KCGA
Min Cost (thousand $)	92.89	94.49	92.89	92.89	92.89	92.89
Least-cost duration	58	58	58	58	58	58
Iteration	17	17	82.2	62.3	52.4	47.7

6.8. Summary

Three case examples, including 4 numerical examples, drawn from literature were analyzed to demonstrate the use of the different components of the proposed method and to illustrate its essential features. These case examples were selected in a way to include different activity time-cost relations, having been analysed using different methods such as Harmony Search, iterative crashing process and Genetic Algorithms and differing in number of activities. Each case example was analysed using one component of the proposed method and the results were compared with those generated in the original example. For each example, a number of scenarios were considered taking into account different conditions. In all the examples, regardless of their original methods (i.e. Harmony search, Genetic Algorithms and iterative crashing process) and considered time-cost relations (i.e. discrete, linear, piece-wise linear, etc.), the base scenario in which only cost was considered, to be consistent with original methods, yielded same or better results than other methods in terms of project total cost and least cost duration, in relatively short time.

However, when considering other attributes as well as risk associated with crashing cost of activities, project total cost at each crashed duration, as expected, were different from that of considering only cost. The results also demonstrated that the larger the project network is the more vivid is the impact of using multi-objective decision environment for project schedule compression will be. In such cases, even small changes in the decision factors and their relative importance largely impact the order by which activities are queued for crashing.

7. CHAPTER SEVEN: SUMMARY AND CONCLUDING REMARKS

7.1. Summary

This research aims to present a new method for schedule compression construction projects with a focus on their execution phase. A comprehensive study was conducted to understand and model the schedule compression problem. As such, a thorough literature review and an industry wide questionnaire survey were conducted in an effort to understand the current industry practice. The questionnaire of the survey was prepared in both paper and web-based format and was to construction management professionals through email. 53 contractors and project managers from 21 construction management firms within Canada, United States and Middle East participated in the survey. The findings of the survey were used in the development made in this research.

The conducted literature review has shown that in all of the various methods that are proposed in the literature, schedule compression is still reduced to some form of time-cost trade-off analysis, where schedule compression is performed based on cost only. In other words, these models use different methods such as mathematical programming and/or artificial intelligence methods such as genetic algorithms to reduce the respective duration of construction projects, while considering cost and time as the only effective factors. None of these methods take into account any factor beyond the additional direct cost required for acceleration of project activities. As such, these methods overlook other factors that are likely to be of importance to contractors. In fact, the lack of consideration of such factors has been attributed to the limited use and uptake, if any, of these methods in practice (Sears, et al. 2008)

A well-structured multi-attributed method was then developed benefiting from the findings of the questionnaire survey and to address limitation of current methods. This is done in a structured and quantitative manner by generating priorities for activity crashing using a modified format of the Multiple Binary Decision Method (MBDM). Further, extension to the traditional application of MBDM is developed to account for uncertainty and risk associated with each of the attributes considered in the compression process at the activity level.

The developed method also accounts for the risk associated with crashing cost of project activities. The developed methodology was implemented in Visual Basic environment (C-Scheduler) with dynamic linkage to MS-Project to update the scheduling data needed to perform the compression in each iteration. The computer application automated activity priority setting in a multi-attributed environment considering different decision attributes. It also automates the iterative schedule compression via dynamic link with MS-Project as scheduling software.

Three numerical examples drawn from literature were analysed and was used to generate additional scenarios to present the added features of the developed method. The results were then compared with those generated by the original methods. When considering cost only, the developed method generated the same or better results than other methods in terms of project total cost. The results also indicated clear difference in the project total cost at any given compressed duration between the results generated from considering cost only and those resulting from the use of other attributes in addition to cost.

7.2. Conclusion

The findings of the conducted industry wide survey revealed that contractors and project managers consider more than one factor when planning to crash the respective duration of projects under their responsibility. These results also show that factors such as resource availability, complexity and logistics of the work involved, contractor's leverage on the sub-contractor who is deemed more capable of performing the accelerated work and the risk associated with crashing of each activity were found to be more or equally important than project cost, in queuing activities for crashing. These results further revealed that the order of importance of these factors differs for contractors and construction managers. While contractors put more emphasis on job site factors such as sub-contractor who is deemed to perform the accelerated job, construction managers look more into overall project conditions such as resource availability. In addition, the individuals who perform as both contractor and construction managers seem to consider wider range of factors including both job site related and project dependent factors.

It can be observed from the results generated form numerical examples that risk consideration while impacting the total project cost, seems not to bring major deviation in the trend of the project total cost versus the project duration for small projects. However, as the number of activities of projects increase, such risk consideration will have more influence on both project total cost and generated crashing execution plans.

It was also found from the results that even small changes in the decision environment directly impacts the generated crashing execution plan vis-à-vis the sequence of activity crashing. As such, unlike the methods referred to earlier in this thesis which implicitly assume the schedule crashing process to be the same for the operational conditions of all contractors and the same for all projects' constraints, the method presented here is capable of treating each project's unique environment and each contractor's operational conditions.

7.3. Recommendation for future work

The method presented here can also be further expanded to suit linear projects such as highways, pipe lines and construction of multi-story buildings. Also, other multi-attributed decision making methods such as Analytical Hierarchy Process (AHP) and Multi Attribute Utility Theory (MAUT) can also be used besides Multiple Binary Decision method (MBDM) used in the method presented here, to model the multi attributed decision environment considered for activity crashing.

As well, the automated software developed for the implementation of the proposed method operates in Microsoft integrated environment, which accepts project schedules in MS-Project format. Extensions to the developed software can be developed that allows for the use of schedules' data from other commercially available scheduling software systems such as Primavera P3 and P6.

REFERENCES

Ahuja, H.N., S.P. Dozzy, and S.M. AbouRizk. (1994). "Project management: Techniques in planning and controling costruction projects. " 2. New York: John Weily and Sons.

Ammar, Mohammed A. (2011). "Optimization of project time-cost trade-off problem with discounted cash flows." ASCE, Journal of Construction Engineering and Management 137(1): 65-71.

Aslani, Pooyan. (2007). "Dynamic resource-constrained scheduling." PHD Thesis, Civil Engineering, Polytechnic University.

Baccou, J., E. Chojnacki, C Mercat-Rommens, and C. Baudrit. (2008). "Extending Monte Carlo Simulations to represent and propagate uncertainties in presence of incomplete knowledge: Application to the transfer of a radionuclide in the environment." ASCE, Journal of Environmental Engineering 134, no. 5: 362-368.

Balasubramanian, J., and I. E. Grossmann . (2003). "Scheduling optimization under uncertainty: An alternative approach." Computers and Chemical Engineering Journal , 27: 469-490.

Bregman, Robert L. (2009) . "A heuristic procedure for solving the dynamic probabilistic project expediting problem." European Journal of Operational Research, 192: 125-137.

Chang, Chul-Ki. (2004). "Quantifying the impact of schedule compression on construction labor productivity." PHD thesis, Civil and Enviromental Engineering, University of Wisconsin-Madison.

Charnes, A., W. W. Cooper, and G. L. Thompson. (1964). "Critical path analysis via chance constrained and stochastic programming." Operational Research Journal 12: 460-470.

Cheng, M.Y., K.Y. Huang, and C.P. Cuong. (2011). "Genetic algorithm-based chaos clustering approach for optimizing construction time-cost tradeoff problems." International Symposium on Automation and Robotics in Construction (ISARC). Seoul, Korea, 892-897.

CII, Construction Industry Institue (CII). (1988) "Concepts and Methods of Schedule Compression." The University of Texas at Austin, Austin, TX: Construction Industry Institute Publication No. 6-7,.

Deckro, R.F., J.E. Hebert, W.A. Verdini, P.H. Grimsrud, and S. Venkateshwar. (1995). "Nonlinear time/cost tradeoff models in project management." Computers and Industrial Engineering Journal 28(2): 219-229.

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

Elmaghraby, S.E., H. Soewandi, and M. Yao. (2001). "Chance-constrained programming (CCP) in activity networks: A critical evaluation." European Journal of Operational Research 131: 440–458.

Evensmo, J., and JT. Karlsen. (2008). "Looking for the source-where do crash costs come from?" Cost Engineering Journal 50(7): 20-23.

Ezeldin, A.S., and A. Soliman. (2009). "Hybrid time-cost optimization of nonserial repetitive construction projects." ASCE, Journal of Construction Engineering and Management 136(6): 42-55.

Fazio, P., O. Moselhi, P. Theberge, and S. Revay. (1988). "Fast-tracking of construction projects: A case study." Canadian Journal of Civil Engineering 15(4): 493-499.

Feng, Chung Wei, Liang Liu, and Scott A. Burns. (1997). "Using genetic algorithms to solve construction time-cost trade-off problems." Journal of Computing in Civil Engineering 11(3): 184-189.

Feng, Chung-Wei, Liang Liu, and Scott A. Burns. (2000). "Stochastic contruction time cost trade-off analysis." Journal of Computing in Civil Engineering 14(2): 117-126.

Finke, Daniel A. (2010). "Resource and space constrained project scheduling." PHD Thesis, Industrial Engineering, The Pennsylvania State University.

Geem, Z.W. (2010). "Multi-objective optimization of time-cost trade-off using Harmony Search." ASCE, Journal of Construction Enginnering and Management 136(6): 711-716.

Gentle, J. (1985). "Monte Carlo methods". Vol. 5. New York: S. Kotz and N. J., eds, Wiley.

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

Haga, W.A, and K.A Marold. (2004). "A simulation approach to the PERT/CPM time-cost trade-off problem." Project Management Journal 35(1): 31-37.

Hassanein, A., and O. Moselhi. (2005). "Accelerating linear projects." ASCE, Journal of Construction Management and Economics 23: 377-385.

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

Herroelen, Willy , and Roel Leus. (2005). "Project scheduling under uncertainty: Survey and research potentials." European Journal of Operational Research 165: 289–306.

Hinze, J.W. (2008). "Construction planning and scheduling" . Vol 3. Upper Saddle River, NJ: Prentice Hall.

Holland, J. H. (1975). Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. Oxford, England: U Michigan Press.

Icmeli, Oya, and S. Selcuk Erenguc. (1996). "The resource constrained time/cost tradeoff project scheduling problem with sicounted cash flows." Journal of Operations Management 14: 255-275.

Kelly, J.E. (1961). "Critical-path planning and scheduling: Mathematical Basis." Operations Research Journal 9(3): 296-320.

Kelly, J.E. (1963). "The critical-path method: Resource planning and scheduling." Englewood Cliffs, N.J.: Industrial scheduling, Prentice-Hall,.

Kim, K., and J. M. de la Garza. (2005). "Evaluation of the resource-constrained critical path method algorithms." ASCE, Journal of Construction Engineering and Management 131(5): 522-532.

Kress, M. (1984). "The chance constrained critical path with location-scale distributions." European Journal of Operational Research 18: 359–363.

Laslo, Z. (2003). "Activity time-cost tradeoffs under time and cost chance constraints." Computers and Industrial Engineering Journal 44: 365–384.

Lee, Jaesung. (2008). "Dynaimc Project Management Methodology: Managing Schedule Compression." PHD Thesis, University of Florida , Florida.

Liberattore, M.J., and J.B. Pollak. (2006). "Extending project time-costanalysis by removing precedence relationships and task streaming." International Journal of Project Management 24: 529-535.

Lin, Feng-Tse. (2008). "Time-cost tradeoff in fuzzy critical path analysis based on (1- α) × 100% confidence-interval estimates." 2008 IEEE International Conference on Systems, Man and Cybernetics (SMC 2008). IEEE. 601-606.

Liu, Jinming, and Fred Rahbar. (2004). "Project time-cost trade-off optimization by maximal flow theory." ASCE, Journal of Construction Engineering and Management 130(4): 607-609.

Liu, L., S.A. Burns, and C.W. Feng. (1995). "Construction time-cost tradeoff analysis using LP/IP hybrid method." ASCE, Journal of Construction Enginering and Management 121(4): 446-454.

Long, Luong Duc , and Ario Ohsato. (2008). "Fuzzy critical chain method for project scheduling under resource constraints and uncertainty." International Journal of Project Management 26: 688-698.

Marazzi, C.A. (1985). "A value analysis method for the evaluation of telecommunication systems bid proposals." IEEE Transactions on Engineering Management 32 (2): 55-62.

Meredith, J.R., and S.J. Mantel. (2006). "Project managemnet: A managerial approach." Vol. 6. Hoboken, NJ: John Wiley, c2006.

Mitchell, Garry F. (2005). "Managing Risk in Complex Projects Using Compression Strategies." PHD Thesis, School of business administration, University of Washington.

Moder, J.J., C.R. Phillips, and E.W. Davis. (1983). "Project management with CPM, PERT, and precedence diagramming". Vol. 3. New York: Van Nostrand Reinhold.

Moselhi, O. (1997). "Risk assessment and contingency estimating." AACE Internationa Transactions: 90-95.

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

Moselhi, O., and A. Alsheibani. (2011). "Cost and experience based method for project acceleration." Proceeding of 6th International Stractural Engineering and Construction (ISEC6). Zurich, Switzerland.

Moselhi, O., and N. Roofigari. (2011-a). “Discussion of “Multiobjective optimization of time-cost trade-off using Harmony Search” by Zong

Woo Geem." ASCE, Journal of Construction Engineering and Management, 137(8): 629-630

Moselhi, O., and N. Roofigari. (2011-b). "Using the Analytical Hierarchy Process in compressing schedules of construction projects." International Symposium on the Analytical Hierarchy Process (ISAHP) proceedings. Sorrento, Naples , Italy.

Moselhi, O., and N. Roofigari. (2011-c). “Compression of project schedules using the Analytical Hierarchy Process." Journal of Multi Criteria Decision Analysis. In Press

Moussourakis, J., and C. Haksever. (2004). "Flexible model for time-cost tradeoff problem." ASCE, Journal of Construction Engineering and Management 130(3): 307-314.

Moussourakis, John, and Cengiz Haksever. (2010). "Project compression with nonlinear cost functions." ASCE, Journal of Construction Engineering and Management 136(2) : 251-259.

Noyce, D.A., and A.S. Hanna. (1998). "Planned and unplanned schedule compression: The impact on labour." Construction Management and Economics Journal 16: 79-90.

Prager, W. (1963). "A structured method of computing project cost polygons." Management Science Journal 9(3): 394-404.

Que, B.C. (2002). "Incorporating practicality into Genetic Algorithm-based time-cost optimization." ASCE, Journal of Construction Engineering and Management 128(2): 139-143.

Rehab, R. M. (1986). "Time cost trade-off modeling of related activities by using mixed integer programming." PHD Thesis, Civil Engineering, University of Michigan.

Saaty, TL. (1980). "The analytical hierarchy process: planning, priority setting, resource allocation. " London, England: McGraw-Hall International book co.

Sakellaropoulos, S., and A. P. Chassiakos. (2004). "Project time–cost analysis under generalised precedence relations." Journal of Advances in Engineering Software 35: 715-724.

Sears, S., K. Sears, G. A. Sears, and R.H. Clough. (2008). "Construction project management." Vol. 5. New York: John Wiley & Sons.

Senouci, Ahmed, and Khaled El-Rays. (2009). "Time-Profit Trade-Off Analysis for Construction Projects." ASCE Journal of Construction Engineering and Management 135(4): 718-725.

Siemens, N. (1971). "A simple CPM time-cost trade-off algorithm." Management Science Journal 17(6) : 354-363.

Stackpole, C. (2010). "A user's manual to the PMBOK guide." Hoboken, N.J.: John Wiley & Sons.

Stevens J. D. (1990). "Techniques for construction network scheduling." McGraw-Hill Inc. New York. USA

Sunde, L, and S Lichtenberg. (1995). "Net-present-value cost/time tradeoff." International Journal of Project Management 13(1): 45-49.

Tareghian, Hamed R., and S.Hassan Taheri. (2006). "On the discrete time, cost and quality trade-off problem." Journal of Applied Mathematics and Computation 181: 1305-1312.

Thomas, H. R. (2000). "Schedule acceleration, work flow and labor productivity." ASCE, Journal of Construction Engineering and Management 126(4): 261-267.

Tolentino Pena, Radhames A. (2009). "Project management: A simulation-based optimizaion method for dynamic time-cost tradeoff decisions." MS Thesis, Department of Industrial & Systems Engineering , Kate Gleason college of engineering,, Rochester, NewYork: Rochester Institue of Technology.

Vanhoucke , Mario , and Dieter Debels. (2007). "The discrete time/cost trade-off problem: extensions and heuristic procedures." Journal of Scheduling 10: 311–326.

Wollmer, R.D. (1985). "Critical path planning under uncertainty." Mathematical Programming Study Journal 25: 164–171.

Yang, I-T. (2005). "Impact of budget uncertainty on project time-cost tradeoff." IEEE Transactions on Engineering Management 52(2): 167-174.

Yang, I-T. (2007). "Using elitist particle swarm optimization to facilitate bicriterion time-cost trade-off analysis." ASCE, Journal of Construction Engineering and Management 133(7): 498-505.

Zheng, D., S. Thomas, and M. Kumaraswamy. (2004). "Applying a genetic algorithm-based multiobjective approach for time-cost optimization." ASCE, Journal of Construction Engineering and Management 130(2): 168-176.

APPENDIX I:

Sample of matrix calculation:

A sample of matrix calculations for first iteration of Scenario 3 is shown here to provide more clarity. Based on the priorities considered for decision attributes (U>CJ>CS), the following decision matrix is generated and its associated weight vector is calculated, using Equations 1 to 3.

$$
\begin{array}{cc}
 & \begin{array}{ccc} CS & CJ & U \end{array} \\
\begin{array}{c} CS \\ CJ \\ U \end{array} & \left[\begin{array}{ccc} 0.5 & 0 & 0 \\ 1 & 0.5 & 0 \\ 1 & 1 & 0.5 \end{array}\right]
\end{array}
\qquad
\begin{array}{c} \text{weights} \\ 0.5 \\ 1.5 \\ \underline{2.5} \\ S=4.5 \end{array}
\qquad
\begin{array}{c} W_i \\ \left\{\begin{array}{c} 0.11 \\ 0.33 \\ 0.56 \end{array}\right\} \\ \hline 1.00 \end{array}
$$

Evaluation matrix with respect to CS (Ac3>Ac7>Ac1>Ac5)

$$
\begin{array}{cc}
 & \begin{array}{cccc} Ac_1 & Ac_3 & Ac_5 & Ac_7 \end{array} \\
\begin{array}{c} Ac_1 \\ Ac_3 \\ Ac_5 \\ Ac_7 \end{array} & \left[\begin{array}{cccc} 0.5 & 0 & 1 & 0 \\ 1 & 0.5 & 1 & 1 \\ 0 & 0 & 0.5 & 0 \\ 1 & 0 & 1 & 0.5 \end{array}\right]
\end{array}
\qquad
\begin{array}{c} 1.5 \\ 3.5 \\ 0.5 \\ \underline{2.5} \\ 8 \end{array}
\qquad
\begin{array}{c} E_{CS} \\ \left\{\begin{array}{c} 0.1875 \\ 0.4375 \\ 0.0625 \\ 0.3125 \end{array}\right\} \\ \hline 1 \end{array}
$$

The evaluation matrix pertinent to cost slopes (CS) of critical activities is also generated from the priorities listed in Table 1 and its emphasis vector (ECS) is calculated likewise. Similarly, the emphasis vectors with respect to two other decision attributes, i.e. contractor judgment and uncertainty are calculated. As such, importance order of Ac1>Ac3>Ac5>Ac7 is considered for critical activities with respect to contractor judgment and the order of Ac3>Ac5>Ac1>Ac7 is considered with respect to uncertainty. The priority vector that shows the relative combined priority (considering the 3 crashing attributes) for crashing of each activity is then calculated using Equation 4.

$$\begin{array}{cc} E_{CJ} & E_{U} \\ \begin{Bmatrix} 0.4375 \\ 0.3125 \\ 0.1875 \\ 0.0625 \end{Bmatrix} & \begin{Bmatrix} 0.1875 \\ 0.4375 \\ 0.3125 \\ 0.0625 \end{Bmatrix} \end{array}$$

$$\{P\} = \left[\begin{Bmatrix} 0.1875 \\ 0.4375 \\ 0.0625 \\ 0.3125 \end{Bmatrix} \begin{Bmatrix} 0.4375 \\ 0.3125 \\ 0.1875 \\ 0.0625 \end{Bmatrix} \begin{Bmatrix} 0.1875 \\ 0.4375 \\ 0.3125 \\ 0.0625 \end{Bmatrix} \right] \times \begin{Bmatrix} 0.11 \\ 0.33 \\ 0.56 \end{Bmatrix} = \begin{Bmatrix} 0.2700 \\ 0.3962 \\ 0.2438 \\ 0.0900 \end{Bmatrix}$$

Consequently, Activity 3 is to be crashed first as it has the highest combined priority.

APPENDIX II:

Questionnaire of the survey-English Format

Multi Objective project acceleration

This survey serves to identify factors usually considered in crashing project scheduled durations (i.e. reducing project duration at least additional cost). It should be noted that the collected responses will remain confidential and will only be used for educational and research purposes.

Please respond to questions by placing an “x” in the relevant boxes (by double clicking on the box and selecting “checked” item).

If you have any question regarding the completion of this questionnaire, please contact [Nazila Roofigrari, Grad student, Concordia University] by phone [514-5700295] or email [n_roofig@encs.concordia.ca].

We would appreciate receiving your reply at your earliest convenience. If you would like to receive a copy of the findings of this study, please mark the appropriate box.

Note: in this study, crashing = accelerating = reducing activity or project duration.

(The following questions are required for communication purposes only and will not be disclosed)

Survey Participant

Name:

Title or position of respondent: (optional)

Company:

Type of companies business:

☐Construction management organization

☐Contractor

☐ Both (i.e. EPC/EPCM)

Contact Details

Telephone:

Email:

Would you like to receive a copy of the findings of this study? ☐ Yes ☐ No

PART 1: General Information

Specialty (main type of work)

☐ Buildings

☐ Heavy civil: ☐ Highways, ☐ Power plants, ☐ Industrial facilities)

☐ Other, Please specify

____

Years of experience

☐ < 5 ☐ 5-10 ☐ 10-15 ☐ >15 **(**Please specify**)**

____

Typical job size ($ millions)

☐ < 1 ☐ 1-10 ☐ 10-50 ☐ >50 **(**Please specify**)**

____

Indirect cost (including project overhead, general head office overhead but excluding profit) on a typical job (expressed in % of direct cost)

☐ < 10 ☐ 10-20 ☐ 20-30 ☐ >30 **(**Please specify**)**

____

PART 2: Schedule compression: General information

How frequently do you encounter the need to accelerate the schedule of projects under your responsibility?

☐ More than 7 out of 10 recent projects

☐ 3-7 out of 10 recent projects

☐ 1-3 out of 10 recent projects

☐ Never

What factors do you consider in setting priorities for activity crashing (i.e. shortening its duration)?

☐ The added direct cost needed to reduce the activity duration

☐ Availability of needed resources to accelerate the activity (i.e. reduce its duration)

☐ Work complexity and logistics related to activity acceleration (i.e. reduce its duration)

☐ Your confidence in the sub-contractor's ability to crash the activity as planned

☐ Your assessment of the risk associated with the planned crashing of the activity

☐ Number of activities that succeed the activity to be crashed

☐ Cash flow constraints

☐ Other, please specify

____

Of the factors you have identified above, please circle the 2 top most important ones.

PART 3: Current practice

How do you currently crash project schedules (i.e. shortening project duration)?

☐ By simply revisiting the CPM and examining the estimated normal durations and the floats

☐ By ascertaining the presence of a realistic baseline schedule

☐ By shortening the duration of selected activities on the project's critical path(s); If applicable, please specify on what basis you select these activities and on what basis you set the priorities for their crashing

____

Do you use any software system to perform the needed schedule crashing?

☐ Yes ☐ No

If yes, please specify:

Questionnaire of the survey-French Format

Accélération Multi-Objectif des projets

Cette étude est menée pour identifier des facteurs habituellement considérés pour raccourcir des durées programmées de projet. On affirme que les réponses rassemblées demeureront confidentielles et seront seulement employées pour des buts éducatifs et de recherches.

Veuillez répondre aux questions en plaçant un « X » dans les boîtes appropriées (double cliquez sur la boîte et choisissez « checked »).

Si vous avez des questions concernant le remplissage de ce questionnaire, veuillez contacter [Nazila Roofigrari, étudiante de deuxième cycle, Université Concordia] à [514-5700295] ou à l'adresse courriel [n_roofig@encs.concordia.ca].

Nous apprécierions de recevoir votre réponse le plus tôt possible. Si vous souhaitez recevoir une copie des résultats de cette étude, veuillez cocher la boîte appropriée.

(Les questions suivantes sont exigées pour la communication seulement et ne seront pas révélées.)

Participant

Nom :

Titre ou position du répondant : (facultatif)

Entreprise :

Domaine d'affaires :

☐ Organisation de gestion de construction

☐ Entrepreneur

☐ Les deux

Les coordonnées du contacte

Numéro de téléphone :

Adresse courriel :

Aimerez-vous recevoir une copie des résultats de cette étude? ☐ Oui ☐ Non

PARTIE 1: Informations générales

Spécialité (domaine principal de travail)

☐ Bâtiment

☐ Génie civil lourd: ☐autoroutes, ☐centrales électriques, ☐complexes industriels

☐ Autre,

spécifiez____

Années d'expérience

☐ < 5 ☐ 5-10 ☐ 10-15 ☐ >15 (SVP spécifiez)

____

Budget typique du travail (millions de $)

☐ < 1 ☐ 1-10 ☐ 10-50 ☐ >50 (SVP spécifiez)

____

Coût indirect sur un travail typique (exprimé en % de coût direct)

☐ < 10 ☐ 10-20 ☐ 20-30 ☐ >30 (SVP spécifiez)

____

PARTIE 2: Compression de programme: Informations générales

À quelle fréquence rencontrez-vous la nécessité d'accélérer les projets sous votre responsabilité ?

☐ Plus de 7 sur 10 projets récents

☐ 3-7 sur 10 projets récents

☐ 1-3 sur 10 projets récents

☐ Jamais

Quelles sont les facteurs que vous considéreriez dans l'arrangement des priorités pour réduire la durée de l'activité ?

☐ Le coût direct supplémentaire qui est lié à la réduction de la durée d'activité

☐ La disponibilité des ressources nécessaires pour accélérer la durée d'activité

☐ La complexité et la logistique de travail

☐ Votre confiance en la capacité du sous-traitant qui est responsable d'accélérer l'activité comme prévu

☐ Votre évaluation du risque lié à l'accélération d'activité

☐ Le nombre des activités qui réussissent à accélérer l'activité

☐ Contraintes de marge

☐ Autre,

spécifiez____

Des facteurs identifiés ci-dessus, entourez les deux plus importants.

PARTIE 3: Pratique actuel

Comment accélérez-vous actuellement des calendriers de projet (c.-à-d. réduire la durée du projet) ?

☐ En revisitant simplement le CPM et en examinant les durées normales prévues et flottantes

☐ En s'assurant de la présence d'un programme réaliste

☐ En raccourcissant la durée des activités choisies sur les chemins critiques du projet

Si ceci est approprié, spécifiez sur quelle base vous choisissez ces activités et sur quelle base vous avez fixé leurs priorités

____

Est-ce que vous employez un système logiciel pour réduire la durée nécessaire de projet ?

☐ Oui ☐ Non

Si la réponse est ‘oui’,

spécifiez____

APPENDIX III:

List of contractor and construction management firms that participated in the questionnaire survey:

No.	Contractor/ Construction Management Firms	Country	No of respondents
1	SNC-Lavalin	Canada(QC)	7
2	Hatch	Canada(QC)	6
3	Magil Construction	Canada(QC)	3
4	Hydro Québec	Canada(QC)	4
5	Tully Construction Co.	USA(NY)	4
6	Toronto District School Board (TDSB)	Canada(ON)	8
7	PCL Industrial Constructors Inc.	Canada(AL)	3
8	KSH Solutions Inc.	Canada(QC)	2
9	Rio Tinto Alcan	Canada(QC)	2
10	Chevron	Canada(QC)	1
11	Waiward Steel Fabricators Ltd.	Canada(AL)	2
12	Air Liquide Canada	Canada(AL)	1
13	Landmark Group of Builders	Canada(AL)	1
14	Cormode and Dickson Construction Ltd.	Canada(AL)	1
15	IGLOO Building Supplies Group	Canada(AL)	1

Continue,

No.	Contractor/ Construction Management Firms	Country	No of respondents
16	Sacs & Sons	USA(NY)	1
17	Kian Beton	Iran	2
18	Kian Pey	Iran	1
19	Talash Naghsh Jahan Co.	Iran	2
20	Reeg-e-Jonoub Co.	Iran	1
21	Parham Co.	Iran	1

APPENDIX IV:

Sample VB code to calculate activity cost slopes

```
Private Sub button1_Click(ByVal sender As Object, ByVal e As EventArgs) Handles
button1.Click
                     Dim data As DataTable = CType(dataGridView1.DataSource,
DataTable)

                     For Each row As DataRow In data.Rows
                            If row(4).ToString() = "Linear" AndAlso (Not
String.IsNullOrWhiteSpace(row(2).ToString())) AndAlso (Not
String.IsNullOrWhiteSpace(row(3).ToString())) AndAlso (Not
String.IsNullOrWhiteSpace(row(5).ToString())) AndAlso (Not
String.IsNullOrWhiteSpace(row(6).ToString())) Then
'INSTANT VB NOTE: The variable result was renamed since Visual Basic does not
handle local variables named the same as class members well:
                                   Dim result_Renamed As Double =
CalculateCostSlop(Double.Parse(row(2).ToString()),
Double.Parse(row(3).ToString()), Double.Parse(row(5).ToString()),
Double.Parse(row(6).ToString()))

                                   row(10) = result_Renamed.ToString()
                            ElseIf row(4).ToString() = "Discrete" AndAlso (Not
String.IsNullOrWhiteSpace(row(2).ToString())) AndAlso (Not
String.IsNullOrWhiteSpace(row(3).ToString())) AndAlso (Not
String.IsNullOrWhiteSpace(row(8).ToString())) AndAlso (Not
String.IsNullOrWhiteSpace(row(9).ToString())) Then
'INSTANT VB NOTE: The variable numberDiscretePoints was renamed since Visual
Basic does not handle local variables named the same as class members well:
                                   Dim numberDiscretePoints_Renamed As Integer =
Integer.Parse(row(8).ToString())
'INSTANT VB NOTE: The variable discretePoints was renamed since Visual Basic
does not handle local variables named the same as class members well:
                                   Dim discretePoints_Renamed As String =
row(9).ToString()

                                   Dim points() As String =
discretePoints_Renamed.Split("&"c)

                                   Dim arrayRows As Integer = points.Length
                                   Dim arrayCols As Integer = 2

                                   Dim discretePointsData(,) As Double

                                   discretePointsData = New Double(arrayRows -
1, arrayCols - 1){}

                                   For i As Integer = 0 To arrayRows - 1
                                          Dim coordinates() As String =
points(i).Split("#"c)

                                          For j As Integer = 0 To arrayCols - 1
                                                 If Not
String.IsNullOrWhiteSpace(coordinates(j)) Then
                                                        discretePointsData(i, j)
= Double.Parse(coordinates(j))
                                                 End If
                                          Next j
```

```
                                   Next i

                                   Dim res() As Double =
CalculateDiscreteCostSlops(Double.Parse(row(2).ToString()),
Double.Parse(row(3).ToString()), discretePointsData)

                                   row(10) = ""
                                   For i As Integer = 0 To res.Length - 1
                                          row(10) += res(i).ToString() & "&"
                                   Next i

                                   If Not
String.IsNullOrWhiteSpace(row(10).ToString()) Then
                                          row(10) =
row(10).ToString().Substring(0, row(10).ToString().Length - 1)
                                   End If


                            ElseIf row(4).ToString() = "Formula" AndAlso (Not
String.IsNullOrWhiteSpace(row(2).ToString())) AndAlso (Not
String.IsNullOrWhiteSpace(row(3).ToString())) AndAlso (Not
String.IsNullOrWhiteSpace(row(7).ToString())) Then
                                   Dim abc As String = row(7).ToString()
'INSTANT VB NOTE: The variable ABC was renamed since Visual Basic will not
allow local variables with the same name as parameters or other local
variables:
                                   Dim ABC_Renamed() As String = abc.Split("&"c)

                                   Dim A As Double =
Double.Parse(ABC_Renamed(0).ToString())
                                   Dim B As Double =
Double.Parse(ABC_Renamed(1).ToString())
                                   Dim C As Double =
Double.Parse(ABC_Renamed(2).ToString())

                                   Dim res As Double =
CalculateWithFormula(Double.Parse(row(2).ToString()),
Double.Parse(row(3).ToString()), A, B, C)
                                   row(10) = res.ToString()

                            End If
                     Next row

                     dataGridView1.Update()
                     dataGridView1.Refresh()


              End Sub

              Private Function CalculateWithFormula(ByVal normalDuration As
Double, ByVal normalCost As Double, ByVal A As Double, ByVal B As Double, ByVal
C As Double) As Double
                     ' Implement your own logic
                     Return 2222
              End Function

              Private Function CalculateDiscreteCostSlops(ByVal normalDuration
As Double, ByVal normalCost As Double, ByVal discretePointsData(,) As Double)
As Double()
                     Dim res() As Double
                     res = New Double(discretePointsData.GetLength(0) - 1){}
```

```
                    res(0) = CalculateCostSlop(normalDuration, normalCost,
discretePointsData(0, 0), discretePointsData(0, 1))

                    For i As Integer = 0 To res.Length - 2
                           res(i + 1) =
CalculateCostSlop(discretePointsData(i, 0), discretePointsData(i, 1),
discretePointsData(i+1, 0), discretePointsData(i+1, 1))
                    Next i

                    Return res
             End Function

             Private Function CalculateCostSlop(ByVal normalDuration As
Double, ByVal normalCost As Double, ByVal crashedCost As Double, ByVal
crashedDuration As Double) As Double
                    '(Crashed cost-Normal cost)/(Normal Duration-Crashed
duration)
                    Return (crashedCost - normalCost) / (normalDuration -
crashedDuration)

             End Function

             Private Sub button2_Click(ByVal sender As Object, ByVal e As
EventArgs) Handles button2.Click
                    Dim table As DataTable = CType(dataGridView1.DataSource,
DataTable)
             End Sub



       End Class
```