A MIXED INTEGER PROGRAMMING METHOD FOR PARETO FRONT OPTIMIZATION OF DISCRETE TIME COST TRADE-OFF PROBLEM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

MERT BİLİR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CIVIL ENGINEERING

JUNE 2015

Approval of the thesis:

A MIXED INTEGER PROGRAMMING METHOD FOR PARETO FRONT OPTIMIZATION OF DISCRETE TIME COST TRADE-OFF PROBLEM

submitted by **MERT BİLİR** in partial fulfillment of the requirements for the degree of **Master of Science in Civil Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver ____________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Ahmet Cevdet Yalçıner ____________________
Head of Department, **Civil Engineering**

Assoc. Prof. Dr. Rifat Sönmez ____________________
Supervisor, **Civil Engineering Dept., METU**

Asst. Prof. Dr. S. Tankut Atan ____________________
Co-Supervisor, **Industrial Engineering Dept.,**
Işık University

Examining Committee Members:

Assoc. Prof. Dr. Ali Murat Tanyer ____________________
Architecture Dept., METU

Assoc. Prof. Dr. Rifat Sönmez ____________________
Civil Engineering Dept., METU

Asst. Prof. Dr. S. Tankut Atan ____________________
Industrial Engineering Dept., Işık University

Asst. Prof. Dr. Aslı Akçamete Güngör ____________________
Civil Engineering Dept., METU

Asst. Prof. Dr. Güzide Atasoy Özcan
Civil Engineering Dept., METU ____________________

Date: 30.06.2015

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Mert Bilir

Signature :

ABSTRACT

A MIXED INTEGER PROGRAMMING METHOD FOR PARETO FRONT OPTIMIZATION OF DISCRETE TIME COST TRADE-OFF PROBLEM

Bilir, Mert

M.S., Department of Civil Engineering

Supervisor: Assoc. Prof. Dr. Rifat Sönmez

Co-Supervisor: Asst. Prof. Dr. S. Tankut Atan

June 2015, 69 pages

There is a reverse relationship between the activity durations and costs in construction projects. In scheduling of construction projects, the project duration can be compressed (crashed) by expediting some of its activities in several ways including; increasing crew size, working overtime, or using alternative construction methods. As a result, when duration of a critical activity is decreased, its cost increases and project duration decreases. In construction projects, resources are usually available in discrete units. This trade-off between time and cost is named as Discrete Time Cost Trade-off Problem (DTCTP) in literature. DTCTP plays an important role in construction scheduling and especially during schedule acceleration. Inadequate analyses and results for the DTCTP lead to unrealistic project durations and schedule acceleration costs. Hence, development of effective methods for the DTCTP is crucial for not only determination of the right alternative for project costs, but also for setting realistic project duration and budget expectations. However, available software packages do not contain DTCTP analysis which is a drawback.

In the literature, there exist both exact and heuristic and meta-heuristic methods to solve DTCTP. However, very few researches have focused on achieving exact solutions for medium and large scale DTCTPs. In this study, a method based on mixed integer programming (MIP) is presented for mainly Pareto front optimization of the medium and large scale DTCTPs. Problem networks are generated to evaluate the performance of the proposed method. The method is mainly developed for Pareto Optimization, however is also tested for single criteria optimization of DTCTP.

Keywords: Discrete Time-Cost Trade-off Problem, Exact Methods, Mixed Integer Programming, Pareto front Curve.

ÖZ

KESİKLİ ZAMAN MALİYET ÖDÜNLEŞİM PROBLEMİNİN PARETO FRONT OPTİMİZASYONU İÇİN DOĞRUSAL TAMSAYILI PROGRAMLAMA YÖNTEMİ

Bilir, Mert

Yüksek Lisans, İnşaat Mühendisliği Bölümü

Tez Yöneticisi: Doç. Dr. Rifat Sönmez

Ortak Tez Yöneticisi: Yrd. Doç. Dr. S. Tankut Atan

Haziran 2015, 69 sayfa

İnşaat yapım projelerinde faaliyet süreleri ve maliyetleri arasında ters bir bağlantı vardır. İş programında yer alan faaliyetlerin sürelerini kısaltmak için bu faaliyete ait işgücü ve makine kaynaklarının arttırılması veya fazla mesai yapılması, ya da maliyeti yüksek yapım yöntemlerinin kullanılması gerekmektedir. Bu sebeple proje süresini belirleyen bir faaliyetin süresi kısaltılınca maliyeti artmakta, ancak faaliyetin ve projenin süresi kısalmaktadır. İnşaat yapım projelerinde çoğu zaman süre ve maliyet arasındaki bu ilişki kesikli bir fonksiyon şeklindedir. Literatürde Kesikli Zaman Maliyet Ödünleşim Problemi (KZMÖP) olarak bilinen bu zaman-maliyet problemi, inşaat yapım projelerine ait iş programı oluşturulması aşamasında ve özellikle yapım faaliyetlerine ait iş programlarının hızlandırılması esnasında kritik önem taşımaktadır. Proje faaliyetleri için zaman-maliyet seçimlerinin doğru yapılmaması, proje maliyetlerinin artmasına sebep olmaktadır. Bununla birlikte, KZMÖP’ in doğru bir şekilde analiz edilip çözülmemesi, proje sürelerinin ne kadar kısaltılabileceği ve kısaltmaların hangi maliyetlerle gerçekleşeceği konularında gerçekçi olmayan beklentilerin oluşmasına sebebiyet verebilmektedir. Bu sebeplerle KZMÖP için etkin yöntemlerin geliştirilmesi, hem proje maliyetleri için doğru tercihlerin yapılması açısından, hem de gerçekçi iş programları ve proje bütçesi beklentileri oluşturulması açısından son derece önemlidir. Fakat mevcut bilgisayar programları KZMÖP analizini içermemektedir. Bu durum bir dezavantaj oluşturmaktadır.

Literatürde KZMÖP' ü çözmek için kesin ve sezgisel ve üst sezgisel yöntemler bulunmaktadır. Fakat orta ve büyük ölçekli KZMÖP' ü çözmeyi hedefleyen çalışma sayısı çok azdır. Bu çalışmada orta ve büyük ölçekli KZMÖP' lerin başta Pareto front optimizasyonu için doğrusal tamsayılı programlama bazlı bir metot önerilmiştir. Metodun performansını değerlendirmek için örnek problem şebekeleri oluşturulmuştur. Metot başlıca Pareto front optimizasyonu için geliştirilmiş olsa da şebekeler üzerinde tek amaçlı optimizasyon da test edilmiştir.

Anahtar Kelimeler: Kesikli Zaman-Maliyet Ödünleşim Problemi, Kesin Yöntemler, Doğrusal Tamsayılı Programlama, Pareto front Eğrisi.

Dedicated to my beloved family...

ACKNOWLEDGEMENTS

First of all, I wish to express my deepest gratitude to my supervisor Assoc. Prof. Dr. Rifat Sönmez for his continuous support, guidance, criticism and encouragement throughout this study. Also, I would like to present special thanks to my co-advisor Asst. Prof. Dr. S. Tankut Atan for his special comments and suggestions.

I am deeply thankful to Prof. Dr. M. Talat Birgönül, Prof. Dr. İrem Dikmen Toker, Asst. Prof. Dr. Aslı Akçamete Güngör, Asst. Prof. Dr. Güzide Atasoy Özcan, Assoc. Prof. Dr. Yalın Arıcı and Kerem Tanboğa for their invaluable contributions.

I would like to thank Saman Aminbakhsh and Mahdi Abbasi Iranagh for their significant support throughout the study.

I wish to thank to my colleagues Hüseyin Erol, Gözde Bilgin, Görkem Eken, Beste Özyurt, Murat Altun, Abdullah Demir and Dr. Şevki Öztürk; and to my close friends Orkun Başara, Can Acar, Sercenk Hızal and Burak Ataş.

I am deeply grateful to my family, especially to my brother Çağlar and sister Özge for their continuous support.

This study has been conducted with the funding provided by The Scientific Technological Research Council of Turkey (TÜBİTAK) under the grant 213M253.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBRREVIATONS

AoA	Activity on Arrow
AoN	Activity on Node
CA	Cash Availability
CPM	Critical Path Method
CPU	Central Processing Unit
DP	Dynamic Programming
DTCTP	Discrete Time Cost Trade-off Problem
GA	Genetic Algorithm
LP	Linear Programming Solver
MAWA	Modified Adaptive Weight Approach
MIP	Mixed Integer Programming
MILP	Mixed-Integer Linear Programming Solver
MIQCP	Mixed-Integer Quadratically Constrained Programming Solver
NPV	Net Present Value
NP hard	Non-polynomial hard
PSIC	Irregular Costs Project Scheduling Problem
PSO	Particle Swarm Optimization
QCP	Quadratically Constrained Programming Solver

QP	Quadratic Programming Solver
RAM	Random Access Memory
SAM	Siemens Approximation Method
TCT	Time Cost Trade-off
TCTP	Time Cost Trade-off Problem

CHAPTER 1

INTRODUCTION

Construction projects have a certain scope, budget and schedule. These three elements are interdependent upon each other. Especially, throughout a construction project, budget and schedule have essential effects on each other. Decision makers of the project have a significant responsibility to prepare and conduct the project schedule. The schedule should be arranged in a way that planned durations, budget and resource usage are realized as much as possible.

The project duration is mostly determined by network analysis. Critical Path Method (CPM) has been one of the most widely used network analyses for scheduling in construction industry. The logic behind this method is finding the longest activity duration path in the network which will be the total project duration. This path is called as critical path and the activities on that path are depicted as critical activities. By definition, CPM deals only with the durations of the activities. If a modification is needed for the durations, it should not be done without taking budget and resource availability into consideration. A modification to be committed regardless of these parameters would not be realistic and logical in real life construction projects. Furthermore, the contractor would have financial difficulties with such kind of problematic schedule which could be evaluated as loss of prestige by the employer side. Therefore, considering all mentioned elements while preparing or modifying the project schedule is crucial.

As a general practice, it is desired to finish a construction project before or at the predetermined contract deadline by the contractor with minimum possible costs. These two objectives are adopted to be realized by both the employer and the contractor. Decision makers could mostly try to minimize the durations which will create the possibility of decreasing some parts of the costs in the project. In construction projects, costs are classified in two main categories which are direct and indirect costs. As the name implies, direct costs are immediately related to production of work, product or service. Labor costs such as carpenter, iron worker, equipment and material costs are regarded as direct costs. On the other hand, indirect costs, also called as overhead costs, are not associated with any specific work item. Examples of indirect costs could be given as office expenses, salaries of indirect personnel (e.g. project manager, timekeeper).

There are several ways to expedite the network activities such as increasing crew size, equipment or machinery amount which means increase in resources. If amount of resources are needed to increase to speed up some activities, direct and indirect costs are affected inversely from this action. While direct costs will rise due to procurement of new resources, since the project duration will be less than before, indirect costs will be decreased naturally. The aim is to finish the project before the deadline, there could be some problems which will postpone the project finish time. For example, the employer may demand changes in the scope after the project initialization. Another example could be resource availability problems at the contractor side which will increase the related activity durations. In order to avoid these kinds of delays, decision makers could decrease durations of the activities by adding extra resources. In this case, time-cost options come up with different cost and duration alternatives for each activity. These are called as modes in the literature. Moreover, reducing the durations of activities with extra resources is named as crashing. Evidently, while direct costs become more with crashing, indirect cost will decay due to duration decrease. In other words, there is intrinsically a reverse relationship between the activity durations and costs in the project which is a trade-off. This trade-off between activity durations and costs is depicted as time cost trade-off problem (TCTP) in the literature. TCTP analysis has a major aim to squeeze the total project duration which will minimize the total cost.

TCTP has been a hot topic in project management world since 1960s. There is no doubt that it is also related to construction projects in which time and cost managements have a prominent part. Most of the construction projects contain great number of activities by their nature. Also, these activities could have different mode options. Due to these facts, TCTP analysis of project networks becomes a crucial aspect for companies in today's competitive construction industry.

TCT problems could be linear or discrete according to relationship between time-cost modes of the activities. They have been both evaluated in the literature. However, since construction projects activities mostly have discrete time-cost modes, discrete version of the TCT, also called as discrete time cost trade-off problem (DTCTP) has been drawn more attention by researchers in the construction project management field.

Researchers have diversified DTCT problems to three categories. They are deadline problem, budget problem and time cost curve, also called as Pareto front curve, problem. These three types will be discussed extensively by giving examples from the literature in Chapter 2.

Several solution methods have been developed to solve DTCTP. They could be classified in two groups as exact methods and heuristic and meta-heuristic methods. As the name implies, the first one finds the global optimal solution. On the other hand, optimality of solutions is not certain for heuristic and meta-heuristic methods. They can give optimal, near-optimal solutions. Also, an output which is far away from the optimal solution could be handled as a result of heuristic and meta-heuristic procedures. Mixed integer programming (MIP), dynamic programming (DP) and branch-and-bound algorithm are widely known examples of exact methods in the literature. For heuristic and meta-heuristic procedures, Siemens' algorithm (SAM), genetic algorithm, ant colony optimization algorithm and particle swarm optimization (PSO) algorithms could be given as examples. These procedures, their advantages, disadvantages and different applications will be mentioned in Chapter 2.

After scanning the literature about DTCTP and proposed algorithms, it has been observed that the project networks used in the solutions have limited number of activities. In practice, construction projects have a lot of activities. In order to make realistic analysis, large scale project networks are needed to be investigated for DTCTP. Hence, this situation could be evaluated as a missing part in the literature. For this reason, one of the focuses of this study will be networks with high number of activities. On the other hand, researchers have not shown an intense attention to Pareto front curve problem in the literature. Especially, there is a rare effort to solve this problem with exact methods such as mixed integer programming. Since Pareto curve presents all the non-dominated solutions in DTCTP which will be discussed elaborately in Chapter 2, it becomes a crucial part for the analysis of networks. Therefore, this thesis aims to handle not only the optimal cost solutions but also the time-cost curve of the problems. In this study, mixed integer programming methodology is used in order to obtain exact solutions. On the other hand, in spite of the intense scientific literature research background, software used in real life construction projects does not mostly contain time cost trade-off analysis which is an obvious deficiency for practical applications. Such a lack in the industry could cause time and money losses for the construction companies. Another aim of this study is provide an efficient DTCTP analysis especially for large scale construction projects.

Considering the mentioned factors, a method based on mixed integer programming (MIP) is presented for the DTCTP. The method is implemented by using GUROBI optimizer. The software is one of the newest optimization programs when compared with others. An important advantage of GUROBI is that it can be used with variety of modelling languages. According to this situation, C# (.NET) programming language has been used for coding in Microsoft Visual Studio 2013 environment.

The organization of the thesis is arranged as follows. In Chapter 2, a literature review about TCTP, its problem types and solution methods studied in the literature is introduced. Chapter 3 is reserved for problem network generation which will be used as sample networks in the models. Chapter 4 is the main body including proposed mixed integer programming method, DTCTP analyses and computational results. Finally, conclusions and remarks are given in Chapter 5.

CHAPTER 2

LITERATURE REVIEW

In this chapter, the proposed solution methods in the literature for TCTP are presented. As it was mentioned in the previous chapter, the methods are investigated in two parts which are exact methods and heuristic & meta-heuristic methods.

2.1 TCTP

Finishing a construction project timely is a fundamental wish of all the parties taking part in the project. For this desire, TCT analysis aims to compress the project schedule in a manner that total cost is minimized as much as possible. However, reducing the project duration is desirable up to a certain point at which the indirect cost increase suppresses the direct cost decrease. This relation is shown in Figure 2.1. Up to that point, total cost decreases as the project duration decays. After that level, additional expenses due to acceleration of some activities start to increase the total cost. Obviously, this case explains the trade-off between cost and duration as it was mentioned in Chapter 1. The main objective behind TCT analyses is expediting some activities with choosing the optimal crashing alternatives in order to reduce the total project cost (Siemens, 1971).

TCT analyses have been worked more than 50 years in the literature. Fulkerson (1961) and Kelley (1961) proposed basic manual solution algorithms for TCT. Although TCT was analyzed with linear cost functions in the first studies, then it has been realized that the problem should be investigated as discrete since the situation that the time cost modes of the activities are discrete is more common in practice (Vanhoucke & Debels, 2007).

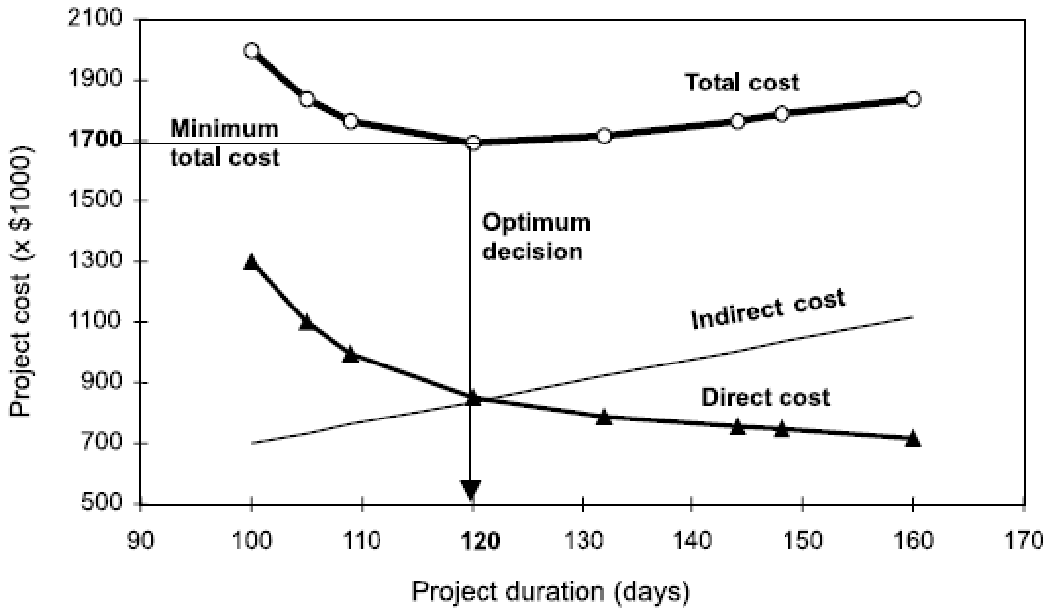


Figure 2.1: Relationship between Project Duration and Direct & Indirect Costs (Hegazy, 1999)

With respect to problem objective, there are three types of TCTP. These are deadline, budget and time-cost curve problems. Deadline problem has the objective to minimize the total cost according to predetermined project duration. On the other hand, there is an upper limit of total cost in the budget problem. Considering this limit, the aim is to minimize the total project duration. Thirdly, time-cost curve problem type is more inclusive when compared with two other types. Time-cost curve problem aims to find all the non-dominated solutions according to total project duration and total cost. Zheng et al. (2004) stated that in order that a solution is regarded as a non-dominated solution, there must be at least one solution as good as in all measures, and better in at least one of them. In other words, all the efficient solutions with respect to time and cost are found out in this problem type which develops a Time-cost curve. This problem is also depicted as Pareto curve problem in the literature which was depicted by Pareto (1906). These problems have been widely discussed in the literature with different solution methods.

2.2 Solution Methods for TCTP

De et al. (1997) defined discrete time cost trade problem as strongly non polynomial hard (NP-hard). Due to this feature, there is a strong possibility that the computational time required to solve the all kinds of DTCTP could be very large. Hence, while developing solution algorithms, this situation is needed to be taken into consideration seriously.

The solution methods are grouped under two categories. The first one is exact methods solving the problems with a guaranteed optimality. Heuristic and Meta-heuristic procedures aim to solve TCTP optimally or near-optimally in short amount of time. However, there is no guarantee for optimal solution in these methods.

2.2.1 Exact Methods

As it was depicted in the previous chapter, by definition exact procedures developed to solve TCT problems find the optimal solution exactly. These methods are time consuming due to NP-hard characteristic of DTCTP. In addition, as the scale of networks enlarges, the solution time increases exponentially (Moussourakis and Haksever, 2004). On the other hand, in spite of this disadvantage, exact methods are the only algorithms solutions of which could be compared with the solutions of heuristic and meta-heuristic algorithms due to optimality guarantee. In other words, quality of heuristic and meta-heuristic algorithms could only be measured by the results of exact methods. Therefore, there is a certain need for exact algorithms to solve DTCTP.

In the literature, mixed integer programming (MIP), dynamic programming (DP) and branch-and-bound algorithms are mostly proposed as exact methods.

Meyer and Shaffer (1963), Crowston and Thompson (1967), Crowston (1970), Harvey and Patterson (1979) are the pioneer studies solving TCT with mixed integer programming algorithms.

As a more recent study, Liu et al. (1995) solved TCT problem for a network comprised of seven activities in Microsoft Excel environment. Actually, a hybrid method combining linear and integer programming is developed in this study. With the model, total time-cost curve is handled. Other than these studies, Moussourakis and Haksever (2004) presented a flexible mixed integer programming model for TCT problems. What is indicated by flexible is that with minor modifications the model has the capability to solve different kinds of time cost trade-off problems. Deadline problem was studied with a sample 7-activity project network in the study. The developed model has a basic assumption that continuous parts of the cost curve are piecewise linear. Another literature study based on mixed integer programming to solve DTCTP was prepared by Sönmez and Bettemir (2012). Actually, the authors used the model in order to measure the performance of a hybrid meta-heuristic procedure coded in Visual C++ that they developed to solve deadline problem for a 63-activity network. Moreover, Szmerekovsky and Venkateshan (2012) presented a mixed integer programming model which could solve irregular costs project scheduling problem (PSIC) which is also a trade-off problem. In this study, every activity has a cost function for different time-cost modes as it was stated in the study of Grigoriev and Woeginger (2004). With the proposed model, it was emphasized that the objective function could be adjusted both to minimize Net Present Value (NPV) of costs and to maximize Cash Availability (CA) which are main concerns of a contractor in the construction industry. In the study, networks having activities up to 90 are solved efficiently.

Dynamic programming (DP) is another exact procedure proposed in the literature to solve TCT. Basically, DP has an objective of network size reduction by combining the activities. With this reduction, the objective function decomposes and the complexity level of problem decreases. Butcher (1967) is the pioneer researcher developing dynamic programming algorithm. Budget problem was studied in this research. Robinson (1975) also presented a DP algorithm solving Budget problem. In this study, some assumptions and sufficient condition are defined to reduce the problem into One-dimensional optimization problem. However, there could be some network problems in which objective function is hard to be decomposed due to high complexity. In this case, the author suggested solving the problem as a Multidimensional optimization problem. Another DP algorithm to solve DTCTP was developed by De et al. (1995). In this study, an extensive overview of past research about this subject is provided firstly. Also, drawbacks of these studies were presented following this summary. Then, new DP procedure was introduced with correcting the algorithm of Hindelang and Muth (1979). It was stated that Hindelang-Muth algorithm is a decentralized approach with having a cumulative cost distribution to the nodes in the network in the course of execution. Using modular decomposition, a centralized procedure was developed and applied for Pareto front curve problem. Also, De et al (1997) presented another DP algorithm to solve Pareto front curve making another correction on Hindelang-Muth algorithm by discarding the pseudo-polynomial insolvability. Demeulemeester et al. (1996) proposed a method containing two algorithms which are based on dynamic programming to solve time-cost curve problem for deterministic activity-on-arrow (AoA) networks. The first algorithm aimed to convert the problem network to a series-parallel network by node reduction. In the second algorithm, computational effort of time-cost mode calculation formed by branch-and-bound procedure was desired to be minimized. Algorithms were coded in C programming software and the developed model is applied for project networks up to 45 activities to find time-cost curve.

Also, branch-and-bound method is another procedure used for exact solution of TCT problems in the literature. Demeulemeester et al. (1998) suggested a branch-and-bound model to solve DTCTP for deterministic AoA networks. A horizon-varying approach was suggested for complete time-cost curve problem. In the first step, lower boundaries were found by doing convex piecewise linear underestimations for DTCT curve of each activity. Then, vertical distance indicating the quality of underestimation was calculated for each activity. The activity with the largest vertical distance is chosen by dividing its time cost mode into two subsets. This procedure is called as branching. The algorithm was worked on networks having a scale of up to 50 activities. Furthermore, Vanhoucke et al. (2002) developed a branch-and-bound algorithm to solve deadline problem with time-switch constraints. Time-switch constraints were proposed by Yang and Chen (2000). By these constraints, it was assumed that activities could only start in a time interval of a cycle which is specified previously. This logic is different than traditional CPM procedure. The cycles are defined with a rest and work windows. In work windows, the activity could be processed while they could not be executed in rest windows. In addition, there is a leading number indicating the maximum number of cycle pair iteration. Time-switch constraints could provide a more efficient schedule by dealing with the shifts of the activities in the network. The authors also claimed that with incorporating this phenomenon project management is improved by effective budget and resource management. Based on time-switch constraints, Vanhoucke et al. (2002) solved a sample AoA network composed of 20 activities using a branch-and-bound algorithm which is coded in Visual C++. Vanhoucke (2005) improved another branch-and-bound algorithm to solve deadline problem showing better performance than Vanhoucke et al. (2002). Time-switch constraints are also used in this study. A different strategy for branching was utilized which ignores time switch constraints of some activities that causes exceeding the predetermined project deadline. This will prevent dealing with solutions which are not optimal. With this better performance, it is claimed that DTCTP in large scale project networks could be solved easily.

Benders decomposition is another exact procedure used in the literature to solve TCT problems. Hazır et al. (2010) proposed a tailored Benders decomposition to solve realistic sizes of DTCTP. Benders (1962) developed an algorithm to decompose linear and mixed integer programming models which have a large scale in terms of decision variables and constraints. The procedure basically aims to separate the main problem into two sub problems. The first problem is depicted as master problem in which the goal is to solve a relieved version of the problem and produce trial values for the variables. This part also aims to specify a lower bound for the minimization objective. The second decomposition part is called as subproblem. The subproblem is the major problem containing the values of the integer variables which are transiently fixed by master problem. In this part, an upper bound for the minimization objective is determined also. Hazır et al. (2010) claimed that the budget problem has not been studied with Benders decomposition. With some new modifications, the model is accelerated and project networks up to 136 activities were solved. Another procedure based on Benders decomposition was improved by Hazır et al. (2011). They proposed a robust scheduling algorithm to represent the uncertainties that could happen in real life construction projects as much as possible. In this study, three models were created to solve problem with cost uncertainties. Probabilistic intervals were determined for the cost uncertainties. This procedure was also able to solve networks having 136 activities effectively.

2.2.2 Heuristic and Meta-heuristic Methods

Different than exact procedures, heuristic and meta-heuristic methods do not guarantee optimality in the solution of TCTP. However, when these algorithms converge, the solution time is relatively less than most of the exact methods. Heuristic methods are dependent on problem type whereas meta-heuristic procedures are not related to problem's nature. This is the major difference between these concepts.

One of the very first heuristic studies to solve TCT was developed by Siemens (1971). In the study, Siemens Approximation Method (SAM) was proposed. It was claimed that the algorithm is adaptable both hand and computer computations due to its simple and systematic structure. The model assumes the time cost trade-off curve as a piecewise linear curve. The objective of the algorithm is to crash critical paths in the network as long as the indirect cost decrease is more than the crashing cost. A major drawback in this procedure is determination of all critical paths which would be hard in large scale networks. In addition, time-cost functions are accepted as convex functions. It was stated that the solutions handled by SAM are equal or almost the same solutions with linear programming results which shows the quality of the algorithm. On the other hand, Goyal (1975) improved another procedure which was stated as alternative to SAM. In this study, effective cost slope was redefined by determining the activities which are not crashed adequately. Moselhi

(1993) was another researcher developing a heuristic procedure using schedule compression technique of CPM.

In the literature, meta-heuristic procedures are currently popular with their applicability to solve TCT problems. One of the mostly used meta-heuristic procedures is genetic algorithm (GA) which was developed by Holland (1975). As the name implies, this algorithm was developed by an analogy based on natural selection and genetic reproduction. Feng et al. (1997) applied this algorithm to solve Pareto front curve problem of TCTP. The normal modes and the crash modes of each activity are determined as two chromosomes. With the iterations and cross overs, the distances of the chromosomes to the convex hull are decided and the solution comes out accordingly. In this study, a sample project network comprised of 18 activities was used which is a very popular problem in the literature of TCTP. Another GA model was suggested by Hegazy (1999). In this study, deadline problem was solved for 18-activity network which was slightly different for one activity from the network used in Feng et al. (1997). Meanwhile, the developed mathematical model was integrated with Microsoft Project scheduling software to solve problems. Zheng et al (2005) also proposed a GA procedure to solve Pareto curve problem. The authors claimed that the model brings a modified adaptive weight approach (MAWA) which was different than traditional weight approach. Project time and total cost were incorporated into a single objective function by this concept. Kandil and El-Rayes (2006) introduced another GA which is designed to aim large-scale construction projects by making parallel processing on the network. Pareto curve was handled for project networks up to 720 activities which is one of the biggest networks studied in the literature. Nonetheless, the computation time reached 137 hours which could be regarded as a very long time for a meta-heuristic algorithm. Sönmez and Bettemir (2012) proposed a hybrid GA including simulated annealing (SA) and quantum simulated annealing (QSA) to solve DTCTP considering the possible insufficiencies of using only one Meta-heuristic algorithm. Maximum number of activities in the benchmark problems is 630 in this study. While developing the algorithm, it was considered that SA has an essential hill climbing ability which will be useful for finding the optimum value in the global space. The authors emphasized the efficiency of using hybrid algorithms by benefiting from complementary features of the algorithms.

Another Meta-heuristic procedure studied in the literature to solve TCTP is Particle Swarm Optimization (PSO) algorithm which is basically derived from swarm intelligence. Swarm intelligence could be explained as a collective behavior during migration. This situation inspires the researchers to develop an analogous algorithm. PSO was developed by Kennedy and Eberhart (1995). One of the very first applications of this algorithm on DTCTP was improved by Elbeltagi et al. (2005). Actually, the authors compared five evolutionary-based algorithms in the study which are PSO, GA, ant colony optimization (ACO), memetic algorithm (MA) and shuffled frog leaping (SFL). Out of these 5 procedures, PSO showed the best performance for solving 18-activity sample network created by Feng et al (1997). Yang (2007a) developed another PSO to study Pareto front curve problem. It was claimed that the proposed model is able to solve different kinds of time-cost functions. A real life highway project composed of 28 activities was used in order to measure the performance of the algorithm. A more recent study about PSO was utilized by Zhang and Li (2010). They developed a combined scheme based PSO to find out Pareto curve by determining the global best for each particle.

Also, ant colony optimization (ACO) which was introduced by Colorni et al. (1992) is a popular meta-heuristic method used in the literature to solve TCTP. The algorithm was developed by inspiring coordinated interactions of ant colonies. The first application of this method for TCTP was done by Ng and Zheng (2008). The aim of this study was to find out the Pareto front curve of 18-activity network generated by Feng et al (1997). Xiong and Kuang (2008) developed another ACO procedure with integrating MAWA. MAWA improves the ability of ACO while searching the optimum solution. Also, this study aimed to find Pareto front curve of an 18-activity network. Furthermore, Afshar et al. (2009) presented an archiving multicolony ant algorithm to reach non-dominated solutions of the same network. A comparison between the proposed algorithm and the model by Zheng et al. (2005) was utilized. It was observed that the proposed ACO procedure demonstrates a better performance. Moreover, Zhang and Ng (2012) utilized another study with ACO which is integrated with Microsoft Project.

Vanhoucke and Debels (2007) developed a meta-heuristic algorithm which could be called also as a hybrid algorithm since the procedure includes some heuristic procedures also. The study focused time-switch constraints, work continuity constraints and NPV maximization in DTCTP. The quality of the meta-heuristic approach was measured by the exact branch-and-bound procedure proposed by Demeulemeester (1998).

In Table 2.1 and Table 2.2, summaries of past researches about TCTP including, exact, heuristic and meta-heuristic methods are provided respectively in a chronological order. If the information about the computational time is given in these researches, the data is added to tables also.

Table 2.1: Exact Methods to Solve TCTP

Year	**Author(s)**	**Method**	**Problem Type**	**Max # of Activities**	**Computational Time**	**Notes**
1963	Meyer and Shaffer	MIP	Deadline	9	-	
1967	Crowston and Thompson	MIP	Deadline	8	-	A basic mathematical MIP model is proposed to solve the problem called as Decision CPM
1995	De, Dunne, Ghosh, and Wells	DP	Pareto front	45	-	An overview of past researches is given and a centralized approach is developed for Hindelang and Muth Algorithms.
1995	Liu, Burns, and Feng	LP/IP Hybrid	Pareto front	7	30 minutes	A hybrid model in which linear programming aims to find lower bound for the minimum direct cost curve and integer programming has a goal to find the exact solutions.
1997	De, Dunne, Ghosh, and Wells	DP	Pareto front	16	-	A DP algorithm is developed by modifying Hindelang and Muth algorithm with discarding its pseudo-polynomial insolvability.
1996	Demeulemeester, Herroelen, and Elmaghraby	DP	Pareto front	45	530.40 seconds	Two DP algorithms are proposed for deterministic networks.
1998	Demeulemeester, De Reyck, Foubert, Herroelen, and Vanhoucke	Branch-and-Bound	Pareto front	50	200 seconds	A branch-and-bound algorithm determining lower bound by convex piecewise linear underestimations.

Table 2.1: Exact Methods to Solve TCTP (*Continued)*

Year	Author(s)	Method	Problem Type	Max # of Activities	Computational Time	Notes
2004	Moussourakis and Haksever	MIP	Deadline	7	-	A flexible MIP model developed which can solve different time/cost functions with some modifications.
2010	Hazır, Haouari, and Erel	Benders Decomposition	Budget	136	-	With making enhancements, slow convergence feature of Benders Decomposition is discarded.
2011	Hazır, Erel, and Günalay	MIP	Cost Uncertainty Problem	136	19139.61 seconds	An MIP model is developed for uncertain cost parameters by assuming probabilistic intervals.
2012	Szmerekovsky and Venkateshan	MIP	Irregular Cost Problem	90	206 seconds	A new MIP model is developed in which the objective function could be adaptable to minimization of NPV and maximization of CA.

Table 2.2: Heuristic and Meta-heuristic Methods to Solve TCTP

Year	Author(s)	Method	Problem Type	Max # of Activities	Computational Time	Notes
1971	Siemens	Heuristic(SAM)	TCT	5	-	A systematic heuristic approach that will expedite the activities with minimum possible cost.
1997	Feng, Liu, and Burns	GA	Pareto front	18	-	The first GA proposed to solve DTCTP.
1999	Hegazy	GA	Deadline	18	390 seconds	A GA which aims to minimize total cost with considering project-based time limitations.
2005	Zheng, Ng, and Kumaraswamy	GA	Pareto front	18	-	A GA including Niche formation is developed to prevent genetic drift, manage selection order, and increase diversity range.
2005	Elbeltagi, Hegazy, and Grierson	PSO	TCT	18	10 seconds	The prominent features of PSO are highlighted by comparing with four other Meta-heuristic methods.
2006	Kandil and El-Rayes	GA	Pareto front	720	137 hours	One of the sparse studies with a large scale network. However, there is a solution time problem.

Table 2.2: Heuristic and Meta-heuristic Methods to Solve TCTP (*Continued)*

Year	Author(s)	Method	Problem Type	Max # of Activities	Computational Time	Notes
2007	Vanhoucke and Debels	Hybrid(Heuristic and Meta-heuristic)	TCT	50	1.605 seconds	A hybrid method is developed with neighborhood search and diversity steps. The results of the hybrid method are compared with the exact procedure by Vanhoucke (2005).
2007	Yang	PSO	Deadline	28	600	A PSO algorithm based on real numbers.
2008	Ng and Zhang	ACO	Pareto front	18	-	One of the first applications of ACO on TCTP. The algorithm aims to solve Pareto curve.
2008	Xiong and Kuang	ACO	Pareto front	18	-	An ACO algorithm integrated with MAWA which enhances the possibility of finding optimum value.
2009	Afshar, Ziaraty, Kaveh, and Sharifi	ACO	Pareto front	18	-	In the algorithm, different ant colonies are assigned to different objectives. The results are evaluated between the colonies and objectives.
2010	Zhang and Li	PSO	Pareto front	18	205.08 seconds	A combined scheme based PSO aiming to determine global best for each particle
2012	Sönmez and Bettemir	Hybrid GA	TCT	630	-	A hybrid GA algorithm integrated with SA and QSA.
2012	Zhang and Ng	ACO	Pareto front	18	-	An ACO algorithm integrated with MAWA to combine time and cost objectives.

After a general look to the literature, there are some points which could be criticized. One of the issues is the number of activities in the sample networks. Most of the researchers use limited number of activities. The 18-activity network which is a small scale network developed by Feng et al. (1997) is one of the mostly used networks in the studies. As the scale of the network is decreased, practical usage of the algorithms for construction projects becomes unrealistic. Kandil and El-Rayes (2006) used a network comprised of 720 activities which has a similar scale to real life construction projects. However, they had a problem with the computational time which is really long. In other words, in the existing studies it is an obvious fact that as the scale of the network enlarges, computational time increases for the analysis of TCTP. Within the context of limitation of existing methods, this study focuses on generation and solving of medium and large scale DTCTP instances.

Nevertheless, most of the commercial software packages do not consider TCT analysis. In real life construction projects, schedules prepared without the analysis which could cause time and money losses for both the employer and contractor. Actually, there are some studies in the literature aiming to integrate a model to the software such as MS Project but these attempts are not widely utilized in real construction project schedules.

Among the three types of TCTP, Pareto front has an essential part since when the full curve is handled; all the feasible solutions come out. Having the feasible non-dominated solutions is a very crucial issue especially for the networks with large scale. In the scope of this thesis, a significant devotion is made for Pareto front solution in the sample networks.

Furthermore, it is evident that exact procedures are not popular in the last years when compared with heuristic and meta-heuristic procedures. The reason could be the NP-hard feature of the DTCTP as the scale of network gets larger. However, exact methods have no alternative since they guarantee optimality. Mixed integer programming is one of the stable exact procedures. Modelling logic of MIP is straightforward. For these reasons, MIP is studied in this thesis. On the other hand, Liu et al. (1995) is the only study aiming to solve Pareto front curve with LP/IP hybrid method. The hybrid method is very similar to MIP logic also. However, the sample 18-activity network is very limited for real life projects. Also, Demeulemeester et al. (1998) solved a 50-activity network for Pareto front curve problem. However, the size of these networks is small. In other words, there is a lack about medium and large scaled benchmark test instances in the literature for DTCTP. The developed test instances in this study are very significant for this respect. They could be used for future research.

Within this context, a method including efficient MIP models in GUROBI optimizer is developed in this study. The aim is to find optimal solution and Pareto front curve. Also, single objective optimization aiming to find the optimal cost of the problem networks is utilized with the MIP models.

CHAPTER 3

PROBLEM NETWORK GENERATION

3.1 Generation of Problem Instances

In this chapter, benchmark problem generation is firstly explained. As it was mentioned, there are popular benchmark instances for DTCTP in the literature. The network generated by Feng et al (1997) is one of them. However, problem includes only 18 activities. Hence, to test the performance of the exact methods first medium and large scale problem instances are generated in this thesis. Initially, a random network generator developed by Demeulemeester et al. (2003) called as RANGEN is tried. However, since the study aims to solve DTCTP for large scale networks, it is observed that RANGEN has problems for generating high number of activities. Therefore, ProGen/max random network generator is chosen which is developed by Schwindt (1995). ProGen/max showed better performance while creating large scale project networks. An interface view of ProGen/max is given in Figure 3.1.

Project networks are developed with four different complexity indexes. In ProGen/max, complexity index is represented by Thesen restrictiveness coefficient. Accordingly, four different coefficients which are 0.2, 0.4, 0.6, and 0.8 have been chosen for networks. For these coefficients, networks are generated with 50,100,200,500 and 1000 activities. Details of the problem sets could be seen in Table 3.1.

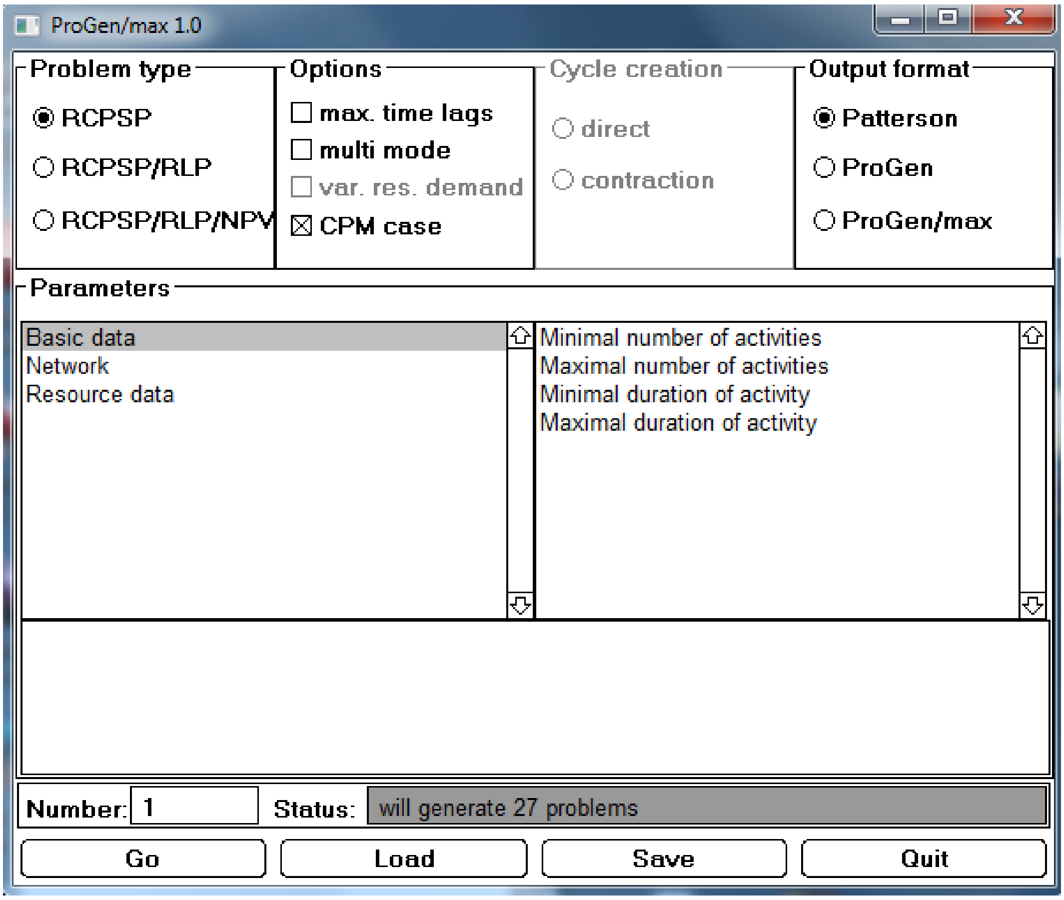


Figure 3.1: An Interface View of ProGen/max Programme

ProGen/max is actually designed to generate problem networks for resource constraint project scheduling. The network generator does not create time-cost modes for DTCTP. Hence, it is utilized only for formation of networks. The network generator develops different networks with different successor and predecessor relationships with the defined parameters.

Time-cost modes for the created networks are prepared in Microsoft Excel 2010. Initially, number of time cost-modes is determined by RANDBETWEEN function. For the time-cost modes, four intervals are chosen which are (2-5), (6-10), (11-15), (16-20). Determination of the modes is done according to Akkan et al. (2005). According to this study, duration of each time-cost mode is selected between 3 days and 123 days. This interval is divided to the number of modes accordingly. For instance, if number of modes for an activity is five:

Table 3.1: Parameter Inputs Entered to ProGen/max for Different Number of Activities

Parameter/# of Activity	50 Activities	100 Activities	200 Activities	500 Activities	1000 Activities
Minimal # of Initial Activities	1	1	1	1	1
Maximal # of Initial Activities	12	20	20	20	20
Minimal # of Terminal Activities	1	1	1	1	1
Maximal # of Terminal Activities	12	20	20	20	20
Maximal # of Predecessor Activities	12	20	20	20	20
Maximal # of Successor Activities	12	20	20	20	20
Degree of Redundancy	0.1	0.1	0.1	0.1	0.1

- duration of 1st mode is chosen between 99 days and 123 days,
- duration of 2nd mode is chosen between 75 days and 98 days,
- duration of 3rd mode is chosen between 51 days and 74 days,
- duration of 4th mode is chosen between 27 days and 50 days,
- duration of 5th mode is chosen between 3 days and 26 days

by using RANDBETWEEN function. Also, for the 1st direct cost alternative, the amount is determined randomly with the same function based on an acceptable interval for the real life construction projects which changes between 100 USD and 50.000 USD. Costs for the remaining modes are determined according to following formula used by Akkan et al. (2005).

$$c_{k+1} = c_k + s_k * (d_k - d_{k+1}) \tag{3.1}$$

Where,

$c_{k+1}: cost\ value\ for\ the\ time - cost\ mode\ k+1$

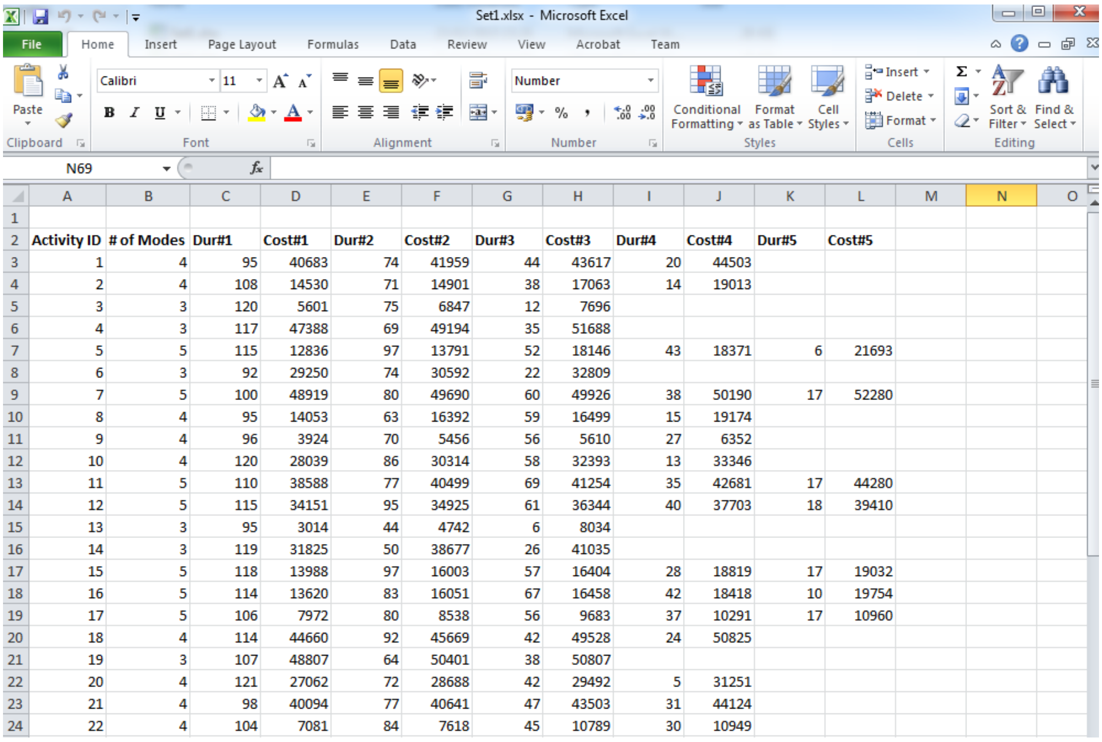
$c_k: cost\ value\ for\ the\ time - cost\ mode\ k$

$s_k: randomly\ generated\ time - cost\ slope\ value$

$d_k: duration\ value\ for\ time - cost\ mode\ k$

$d_{k+1}: duration\ value\ for\ time - cost\ mode\ k+1$

Randomly generated time-cost slope value determined between 10 and 100 with respect to literature and general construction project activities by RANDBETWEEN function. A Microsoft Excel spreadsheet view is given in Figure 3.2 in which time-cost modes of a sample network is given. In all the created networks, there are initial and final dummy activities representing the start and finish of the project. Therefore, they do not have a duration and cost.



Activity ID	# of Modes	Dur#1	Cost#1	Dur#2	Cost#2	Dur#3	Cost#3	Dur#4	Cost#4	Dur#5	Cost#5
1	4	95	40683	74	41959	44	43617	20	44503		
2	4	108	14530	71	14901	38	17063	14	19013		
3	3	120	5601	75	6847	12	7696				
4	3	117	47388	69	49194	35	51688				
5	5	115	12836	97	13791	52	18146	43	18371	6	21693
6	3	92	29250	74	30592	22	32809				
7	5	100	48919	80	49690	60	49926	38	50190	17	52280
8	4	95	14053	63	16392	59	16499	15	19174		
9	4	96	3924	70	5456	56	5610	27	6352		
10	4	120	28039	86	30314	58	32393	13	33346		
11	5	110	38588	77	40499	69	41254	35	42681	17	44280
12	5	115	34151	95	34925	61	36344	40	37703	18	39410
13	3	95	3014	44	4742	6	8034				
14	3	119	31825	50	38677	26	41035				
15	5	118	13988	97	16003	57	16404	28	18819	17	19032
16	5	114	13620	83	16051	67	16458	42	18418	10	19754
17	5	106	7972	80	8538	56	9683	37	10291	17	10960
18	4	114	44660	92	45669	42	49528	24	50825		
19	3	107	48807	64	50401	38	50807				
20	4	121	27062	72	28688	42	29492	5	31251		
21	4	98	40094	77	40641	47	43503	31	44124		
22	4	104	7081	84	7618	45	10789	30	10949		

Figure 3.2: Time-Cost Modes for Activities Created in Microsoft Excel 2010

Afterwards, the networks created in ProGen/max and the time-cost modes prepared in Microsoft Excel 2010 are combined as txt file formats as shown in Figure 3.3. In this way, it is considered that the file will be easily read by the prepared codes. In this figure, the first number at the top of the file represents the number of activities in the problem network. If the network is created as a 50-activity network, the number seems 52 with the initial and final dummy activities. The file is formed of two parts. In the first and second parts, the first column shows the activity ID numbers. The second column demonstrates the number of predecessors of that activity in the first part. Predecessors of the activities could be seen in the rest of the columns. Number of time-cost modes is displayed in the second column of the second part. Then, duration (days) and costs (USD) of the associated modes are included respectively.

52												
1	11	5	9	12	3	4	2	8	7	11	6	10
2	1	47										
3	5	29	44	35	14	24						
4	3	43	16	49								
5	1	48										
6	2	14	51									
7	1	25										
8	1	32										
9	1	27										
10	1	46										
11	1	45										
12	1	42										
13	3	15	23	37								
14	1	50										
15	2	28	41									
16	1	49										
17	1	13										
18	1	24										
19	2	44	18									
20	1	41										
21	1	33										
22	1	31										
23	2	21	37									
24	1	44										
25	1	30										
26	1	29										
27	1	17										
28	1	38										
29	5	19	44	31	14	22						
30	1	48										
31	1	19										
32	2	26	14									
33	1	49										
34	2	30	20									
35	1	39										
36	3	33	21	49								
37	2	33	36									
38	2	34	41									
39	1	40										
40	1	14										
41	2	30	23									
42	1	23										
43	2	49	16									
44	1	52										
45	1	52										
46	1	52										
47	1	52										
48	1	52										
49	1	52										
50	1	52										
51	1	52										
52	0											
0	0											
1	3	115	45623	61	48453	23	51533					
2	4	112	18566	63	20505	38	21777	32	22000			
3	5	104	5426	76	5749	67	6412	46	7721	17	8462	
4	4	94	13089	64	14800	57	15015	30	15365			

Figure 3.3: An Example of the Created Sample Problem Networks

In the figure, it stands out that the first part starts with Activity ID 1 whereas the second part's first ID number is 2. This comes from the output format of ProGen/max which does not have any effect on the results.

A crucial point of the problem networks is delay penalty in this study. Generally, construction projects are prone to delay. Therefore, a delay penalty is needed in order to approximate the model to the real cases in the construction projects. The project deadline and delay penalty is determined by:

$$Deadline = \frac{CPMMax - CPMMin}{2} + CPMMin \tag{3.2}$$

$$Cost\ of Delay\ Penalty = Indirect\ Cost \times 2 \tag{3.3}$$

where

$CPMMax$: $maximum\ CPM\ duration\ calculated\ by\ taking\ longest\ duration\ in\ time - cost\ modes\ of\ the\ activites\ on\ the\ critical\ path$

$CPMMin$: $minimum\ CPM\ duration\ calculated\ by\ taking\ shortest\ duration\ in\ time - cost\ modes\ of\ the\ activites\ on\ the\ critical\ path$

$Deadline$: $project\ deadline\ after\ which\ delay\ penalty\ is\ started\ to\ be\ paid$

With this model, delay penalty is added to project networks. Cost of delay penalty is determined as double of indirect for each network. If the project duration exceeds the defined deadline, delay penalty is started to be added to total cost daily.

As it was mentioned, there are four different complexity indexes and 4 different mode intervals for project networks having 50, 100, 200, 500, and 1000 activities. 10 instances are prepared for each set. Details of the number of sets could be seen in Table 3.2. Totally, 400 test instances are prepared which is a significant number when compared to previous benchmark instances. In Table 3.2, it could also be seen that daily indirect cost is chosen as 250 USD for networks having 50 activities. For the rest of the networks, daily indirect cost is determined as 500 USD.

Table 3.2: Number of Problem Sets Prepared for DTCTP Analyses

# of Activities	# of Time-Cost Modes																Daily Indirect Cost (USD)
	2--5				6--10				11--15				16--20				
	Thesen Restrictiveness Coefficient								Thesen Restrictiveness Coefficient								
	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8	
	# of Test Instances				# of Test Instances				# of Test Instances				# of Test Instances				
50	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	250
100	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	500
200	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	500
500	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	500
1000	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	500

In addition to these networks, in order to utilize a comparison with the studies in the literature, the network developed by Feng et al. (1997) is used to evaluate performance of the exact methods. As it was depicted, the network is comprised of 18 activities. In this study, a 180- activity network is formed by combining the original 18-activity networks consecutively for the analyses.

CHAPTER 4

MIXED INTEGER PROGRAMMING MODEL AND EMPIRICAL ANALYSES

As it was mentioned, a method based on MIP is presented for Pareto front optimization of DTCTP. In this chapter, the proposed MIP model is initially introduced. Then, a brief introduction about GUROBI software is presented. Finally, the proposed method for Pareto front optimization is explained.

4.1 Mixed Integer Programming Model

In order to minimize the total cost of the projects comprised of direct and indirect costs, the following model based on De et al. (1995) is proposed.

4.1.1 Sets

$Pd_j: predecessors\ of\ activity\ j$

$S: activities\ in\ the\ network\ (excluding\ dummy\ activities)$

4.1.2 Parameters

$dc_{jk}: cost\ of\ activity\ j\ for\ time - cost\ mode\ k$

$i_c: daily\ indirect\ cost$

$d_{jk}: duration\ of\ activity\ j\ for\ time - cost\ mode\ k$

$m(j): number\ of\ time - cost\ modes\ for\ acitivity\ j$

$d_p: daily\ delay\ penalty\ cost$

4.2.2 Variables

$Ft_j: finish\ date\ of\ activity\ j$

$Ft_h: finish\ date\ of\ activity\ h$

$x_{jk}: binary\ variable\ \ which\ is\ 1\ if\ time - cost\ mode\ k\ is\ chosen\ to\ realize\ activity\ j, if\ not\ 0.$

$D: total\ duration\ of\ the\ project$

$D_{delay}: amount\ of\ delay$

$D_{deadline}: project\ deadline$

4.2.3 Model

$$\min \textstyle\sum_{j=1}^{S} \sum_{k=1}^{m(j)} dc_{jk}\, x_{jk} + Di_C + d_p D_{delay} \qquad (4.1)$$

Constraints

$$\sum_{k=1}^{m(j)} x_{jk} = 1\ , \qquad \forall j \in\ S \qquad (4.2)$$

$$Ft_j \geq\ Ft_h + \sum_{j=1}^{S} \sum_{k=1}^{m(j)} d_{jk}\, x_{jk}, \qquad \forall h\ \in Pd_j\ and\ \forall j\ \in\ S \qquad (4.3)$$

$$D \geq Ft_{S+1} \qquad (4.4)$$

$$Ft_0 = 0 \qquad (4.5)$$

$$D - D_{delay} \leq D_{deadline} \qquad (4.6)$$

$$D \geq 0 \qquad (4.7)$$

$$D_{delay} \geq 0 \qquad (4.8)$$

$$x_{jk} \in \{0,1\}, \forall j \in\ S, and\ \forall k \in\ m(j) \qquad (4.9)$$

$$Ft_j \geq 0, \forall j \in\ S \qquad (4.10)$$

Objective function (4.1) aims to minimize the total cost of the project which is equal to summation of direct and indirect costs. Constraint (4.2) dictates only one time-cost mode is chosen for each activity. For instance, if 2^{nd} mode of activity 3 is chosen in a sample network, x_{32} value is equal to 1 and $x_{31}, x_{33}, x_{34}, \ x_{35}$ are equal to 0. Since these values are multipliers of activity costs, only the selected time-cost mode will affect the total project duration. There will not be any contribution from the remaining unselected mode alternatives. Constraint (4.3) represents that an activity could not finish before the duration of the activity elapses after all of the predecessors of that activity are completed. Here, duration of the activity for the selected mode alternative will be multiplied by 1. The rest of the modes have no effect on finish dates. Besides, it is indicated that the project could not finish before the finish of final dummy activity in constraint (4.4). In constraint (4.5), finish date of initial dummy activity is set to 0. In other words, it means the initial dummy activity has duration of 0 days. As it was mentioned before, a deadline is decided for each benchmark test instance. Constraint (4.6) represents the relation between the deadline and amount of delay. The next two constraints (4.7), (4.8) explain that the total project duration and amount of delay should be positive values respectively. (4.9) indicates that x_{jk} is a binary variable. The last constraint (4.10) shows that the finish dates of the activities must be larger than 0. Additionally, the dummy activities (activity 0 and activity S+1) have only one mode in which duration and cost are 0.

4.2 Empirical Analyses and GUROBI Optimizer

GUROBI optimizer is used to solve the MIP model presented. GUROBI is one of the powerful and fastest software for linear programming. Additionally, the optimizer has a very powerful library which can work with different programming languages such as C++, C# (.NET), Java and MATLAB. Evidently, this is a very effective feature for different kinds of users throughout the world. On the website of GUROBI (www.gurobi.com), free academic license is also available which is another reason of choice when compared with other optimization programmes in the market.

Another advantage of GUROBI is that it uses the most improved implementations of mathematical programming algorithms. The optimizer contains

- Linear programming solver (LP)
- Mixed-integer programming linear programming solver (MILP)
- Mixed-integer quadratic programming solver (MIQP)
- Quadratic programming solver (QP)
- Quadratically constrained programming solver (QCP)
- Mixed-integer quadratically constrained programming solver (MIQCP)

which are commonly used mathematical programming algorithms. Also, it is stated that GUROBI shows the best performance in public benchmark results of LP, MIP and QP problems in the website with statistics of other optimizers.

Due to these reasons, GUROBI optimizer version 5.6.3 is selected to solve the proposed model. The coding has been done via C# (.NET) language in Microsoft Visual Studio 2013 environment. The analyses and solutions are utilized with a desktop computer running Windows 7 Professional Edition (64-bit) operating system having Intel Core i5 3.10 CPU GHz and 4.00 GB random access memory (RAM).

4.2.1 LP Format

The problem sets prepared as txt files are converted to LP formats by which GUROBI will read the files in an easier way. Thus, with a C# (.NET) code the txt files are transformed to LP files. An LP file format example is given in Figure 4.1 retrieved from GUROBI website. In the first row, it is seen that the objective function aims to maximize summation of x, y, and z decision variables. Then, constraints part starts with “Subject to” expression. The statements “c0, c1, qc0” at the beginning of the constraint only represent the number of the constraints. These constraints could be written without these statements. In “Bounds” section, constraints for every decision variable are defined. Then, the type of decision variable is defined with “Generals” expression. This means that every decision variable is an integer. If binary variable are needed in the model, “Binary” statement should be written to the file. Finally, to finish the model “End” is put to the last row. A crucial issue which should be taken into consideration in LP format is that there is a space between every character. The solver will perceive each variable and multiplier according to these spaces. An LP file prepared for a 50-activity network could be seen in Figure 4.2.

```
Maximize
  x + y + z
Subject To
  c0: x + y = 1
  c1: x + 5 y + 2 z <= 10
  qc0: x + y + [ x ^ 2 - 2 x * y + 3 y ^ 2 ] <= 5
Bounds
  0 <= x <= 5
  z >= 2
Generals
  x y z
End
```

Figure 4.1: A Simple Problem Prepared in LP Format (www.gurobi.com)

T50_1.lp - Notepad

File Edit Format View Help

```
Minimize
 45623 A2M1 + 48453 A2M2 + 51533 A2M3 + 18566 A3M1 + 20505 A3M2 + 21777 A3M3 + 22000 A3M4 + 5426 A4M1 + 5749 A4M2 + 6412 A4M3 + 7721 A4M4 + 8462 A4M5 + 13089 A5M1 + 14800
1 + 42103 A24M2 + 44434 A24M3 + 28977 A25M1 + 29876 A25M2 + 33722 A25M3 + 1679 A26M1 + 3744 A26M2 + 6222 A26M3 + 7670 A26M4 + 10317 A26M5 + 8987 A27M1 + 9938 A27M2 + 1806
+ 18243 A45M3 + 20297 A45M4 + 21215 A45M5 + 25842 A46M1 + 26472 A46M2 + 28622 A46M3 + 44148 A47M1 + 45153 A47M2 + 29088 A48M1 + 33237 A48M2 + 31448 A49M1 + 34379 A49M2 +
Subject To
 A2 - A1 - 115 A2M1 - 61 A2M2 - 23 A2M3 >= 0
 A3 - A1 - 112 A3M1 - 63 A3M2 - 38 A3M3 - 32 A3M4 >= 0
 A4 - A1 - 104 A4M1 - 76 A4M2 - 67 A4M3 - 46 A4M4 - 17 A4M5 >= 0
 A5 - A1 - 94 A5M1 - 64 A5M2 - 57 A5M3 - 30 A5M4 >= 0
 A6 - A1 - 101 A6M1 - 45 A6M2 >= 0
 A7 - A1 - 84 A7M1 - 63 A7M2 - 9 A7M3 >= 0
 A8 - A1 - 115 A8M1 - 77 A8M2 - 69 A8M3 - 48 A8M4 - 10 A8M5 >= 0
 A9 - A1 - 115 A9M1 - 67 A9M2 - 35 A9M3 - 21 A9M4 >= 0
 A10 - A1 - 96 A10M1 - 72 A10M2 - 48 A10M3 - 17 A10M4 >= 0
 A11 - A1 - 72 A11M1 - 47 A11M2 >= 0
 A12 - A1 - 98 A12M1 - 9 A12M2 >= 0
 A13 - A17 - 120 A13M1 - 78 A13M2 - 24 A13M3 >= 0
 A14 - A3 - 107 A14M1 - 86 A14M2 - 60 A14M3 - 36 A14M4 - 25 A14M5 >= 0
 A14 - A6 - 107 A14M1 - 86 A14M2 - 60 A14M3 - 36 A14M4 - 25 A14M5 >= 0
 A14 - A29 - 107 A14M1 - 86 A14M2 - 60 A14M3 - 36 A14M4 - 25 A14M5 >= 0
 A14 - A32 - 107 A14M1 - 86 A14M2 - 60 A14M3 - 36 A14M4 - 25 A14M5 >= 0
 A14 - A40 - 107 A14M1 - 86 A14M2 - 60 A14M3 - 36 A14M4 - 25 A14M5 >= 0
 A15 - A13 - 98 A15M1 - 69 A15M2 - 57 A15M3 - 10 A15M4 >= 0
 A16 - A4 - 86 A16M1 - 78 A16M2 - 29 A16M3 >= 0
 A16 - A43 - 86 A16M1 - 78 A16M2 - 29 A16M3 >= 0
 A17 - A27 - 114 A17M1 - 46 A17M2 - 22 A17M3 >= 0
 A18 - A19 - 100 A18M1 - 39 A18M2 >= 0
 A19 - A29 - 118 A19M1 - 79 A19M2 - 43 A19M3 - 13 A19M4 >= 0
 A19 - A31 - 118 A19M1 - 79 A19M2 - 43 A19M3 - 13 A19M4 >= 0
 A20 - A34 - 112 A20M1 - 93 A20M2 - 57 A20M3 - 47 A20M4 - 21 A20M5 >= 0
 A21 - A23 - 122 A21M1 - 5 A21M2 >= 0
 A21 - A36 - 122 A21M1 - 5 A21M2 >= 0
 A22 - A29 - 99 A22M1 - 89 A22M2 - 68 A22M3 - 45 A22M4 - 7 A22M5 >= 0
 A23 - A13 - 83 A23M1 - 39 A23M2 >= 0
 A23 - A41 - 83 A23M1 - 39 A23M2 >= 0
 A23 - A42 - 83 A23M1 - 39 A23M2 >= 0
 A24 - A3 - 110 A24M1 - 78 A24M2 - 26 A24M3 >= 0
 A24 - A18 - 110 A24M1 - 78 A24M2 - 26 A24M3 >= 0
 A25 - A7 - 113 A25M1 - 73 A25M2 - 28 A25M3 >= 0
 A26 - A32 - 123 A26M1 - 98 A26M2 - 61 A26M3 - 40 A26M4 - 4 A26M5 >= 0
 A27 - A9 - 75 A27M1 - 58 A27M2 >= 0
 A28 - A15 - 105 A28M1 - 89 A28M2 - 70 A28M3 - 30 A28M4 - 10 A28M5 >= 0
 A29 - A3 - 106 A29M1 - 97 A29M2 - 54 A29M3 - 50 A29M4 - 8 A29M5 >= 0
 A29 - A26 - 106 A29M1 - 97 A29M2 - 54 A29M3 - 50 A29M4 - 8 A29M5 >= 0
 A30 - A25 - 99 A30M1 - 67 A30M2 - 31 A30M3 >= 0
 A30 - A34 - 99 A30M1 - 67 A30M2 - 31 A30M3 >= 0
 A30 - A41 - 99 A30M1 - 67 A30M2 - 31 A30M3 >= 0
 A31 - A22 - 110 A31M1 - 79 A31M2 - 55 A31M3 - 44 A31M4 - 16 A31M5 >= 0
 A31 - A29 - 110 A31M1 - 79 A31M2 - 55 A31M3 - 44 A31M4 - 16 A31M5 >= 0
 A32 - A8 - 123 A32M1 - 55 A32M2 >= 0
 A33 - A21 - 110 A33M1 - 80 A33M2 - 38 A33M3 - 28 A33M4 >= 0
 A33 - A36 - 110 A33M1 - 80 A33M2 - 38 A33M3 - 28 A33M4 >= 0
 A33 - A37 - 110 A33M1 - 80 A33M2 - 38 A33M3 - 28 A33M4 >= 0
 A34 - A38 - 123 A34M1 - 60 A34M2 - 42 A34M3 >= 0
 A35 - A3 - 94 A35M1 - 69 A35M2 - 20 A35M3 >= 0
 A36 - A37 - 111 A36M1 - 35 A36M2 >= 0
 A37 - A13 - 108 A37M1 - 75 A37M2 - 46 A37M3 - 6 A37M4 >= 0
 A37 - A23 - 108 A37M1 - 75 A37M2 - 46 A37M3 - 6 A37M4 >= 0
 A38 - A28 - 120 A38M1 - 90 A38M2 - 51 A38M3 - 24 A38M4 >= 0
```

Figure 4.2: An LP File Prepared for a 50-activity Network

In this example, number of time-cost modes change between two and five. As it is emphasized in Table 3.2, indirect cost for 50-activity networks is determined as 250 USD/day. Indirect cost is added to objective function after the last mode of the last activity by multiplying with "DUR" variable which represents total project duration. In the LP file, "A2M1" expression is the binary variable (i.e. x_{jk}) representing the first mode of the second activity in the network. The number near this expression is the cost value in USD for that specific time-cost mode. As it could be seen in Figure 4.2, the objective function starts from activity 1. The reason behind this is that there is an initial dummy activity 0 which does not have any duration and cost. Thus, there is no need to put this activity to the objective function

T50_1.lp - Notepad

File Edit Format View Help

```
 A49 - A4 - 104 A49M1 - 43 A49M2 - 11 A49M3 >= 0
 A49 - A16 - 104 A49M1 - 43 A49M2 - 11 A49M3 >= 0
 A49 - A33 - 104 A49M1 - 43 A49M2 - 11 A49M3 >= 0
 A49 - A36 - 104 A49M1 - 43 A49M2 - 11 A49M3 >= 0
 A49 - A43 - 104 A49M1 - 43 A49M2 - 11 A49M3 >= 0
 A50 - A14 - 79 A50M1 - 21 A50M2 >= 0
 A51 - A6 - 106 A51M1 - 11 A51M2 >= 0
 A52 - A44 >= 0
 A52 - A45 >= 0
 A52 - A46 >= 0
 A52 - A47 >= 0
 A52 - A48 >= 0
 A52 - A49 >= 0
 A52 - A50 >= 0
 A52 - A51 >= 0
 DUR - A52 >= 0
 D >= 0
 DUR - D <= 1024
 A2M1 + A2M2 + A2M3 = 1
 A3M1 + A3M2 + A3M3 + A3M4 = 1
 A4M1 + A4M2 + A4M3 + A4M4 + A4M5 = 1
 A5M1 + A5M2 + A5M3 + A5M4 = 1
 A6M1 + A6M2 = 1
 A7M1 + A7M2 + A7M3 = 1
 A8M1 + A8M2 + A8M3 + A8M4 + A8M5 = 1
 A9M1 + A9M2 + A9M3 + A9M4 = 1
 A10M1 + A10M2 + A10M3 + A10M4 = 1
 A11M1 + A11M2 = 1
 A12M1 + A12M2 = 1
 A13M1 + A13M2 + A13M3 = 1
 A14M1 + A14M2 + A14M3 + A14M4 + A14M5 = 1
 A15M1 + A15M2 + A15M3 + A15M4 = 1
 A16M1 + A16M2 + A16M3 = 1
 A17M1 + A17M2 + A17M3 = 1
 A18M1 + A18M2 = 1
 A19M1 + A19M2 + A19M3 + A19M4 = 1
 A20M1 + A20M2 + A20M3 + A20M4 + A20M5 = 1
 A21M1 + A21M2 = 1
 A22M1 + A22M2 + A22M3 + A22M4 + A22M5 = 1
 A23M1 + A23M2 = 1
 A24M1 + A24M2 + A24M3 = 1
 A25M1 + A25M2 + A25M3 = 1
 A26M1 + A26M2 + A26M3 + A26M4 + A26M5 = 1
 A27M1 + A27M2 = 1
 A28M1 + A28M2 + A28M3 + A28M4 + A28M5 = 1
 A29M1 + A29M2 + A29M3 + A29M4 + A29M5 = 1
 A30M1 + A30M2 + A30M3 = 1
 A31M1 + A31M2 + A31M3 + A31M4 + A31M5 = 1
 A32M1 + A32M2 = 1
 A33M1 + A33M2 + A33M3 + A33M4 = 1
 A34M1 + A34M2 + A34M3 = 1
 A35M1 + A35M2 + A35M3 = 1
 A36M1 + A36M2 = 1
 A37M1 + A37M2 + A37M3 + A37M4 = 1
 A38M1 + A38M2 + A38M3 + A38M4 = 1
 A39M1 + A39M2 + A39M3 + A39M4 + A39M5 = 1
 A40M1 + A40M2 = 1
 A41M1 + A41M2 = 1
 A42M1 + A42M2 + A42M3 + A42M4 = 1
 A43M1 + A43M2 = 1
 A44M1 + A44M2 + A44M3 + A44M4 + A44M5 = 1
 A45M1 + A45M2 + A45M3 + A45M4 + A45M5 = 1
 A46M1 + A46M2 + A46M3 = 1
 A47M1 + A47M2 = 1
 A48M1 + A48M2 = 1
 A49M1 + A49M2 + A49M3 = 1
 A50M1 + A50M2 = 1
 A51M1 + A51M2 = 1
Binary
 A2M1 A2M2 A2M3 A3M1 A3M2 A3M3 A3M4 A4M1 A4M2 A4M3 A4M4 A4M5 A5M1 A5M2 A5M3 A5M4 A6M1 A6M2 A7M1 A7M2 A7M3 A8M1 A8M2 A8M3 A8M4 A8M5 
Generals
 DUR D A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16 A17 A18 A19 A20 A21 A22 A23 A24 A25 A26 A27 A28 A29 A30 A31 A32 A33 
End
```

Figure 4.2: An LP File Prepared for a 50-activity Network *(Continued)*

For final dummy activity, this case is also valid. In the constraints, as an example "A2" statement symbolizes the finish date of the second activity (i.e. Ft_j). In the LP format, since there could not be variables at both sides of the inequality, all variables are shown on the left hand side of the equation. For the project duration constraint, "DUR" variable should be larger or equal to the finish date of final dummy activity which is demonstrated with a constraint also. After this constraint, decision variable "D" is written to LP file. As it was mentioned, amount of duration should be positive. If a delay exists, the total duration is allowed to exceed the deadline via a soft constraint. These two criteria are represented with two constraints in LP file.

Afterwards, constraints for binary variables representing the selected and unselected time-cost modes are entered. In addition, in order to specify that these variables are binary, they are defined under “Binary” heading after all the constraints. Rest of the decision variables are integer. Hence, “Generals” statement is used for them. Finally, “End” expression represents the LP model is completed for that problem.

4.2.2 Finding Optimal Cost in Project Networks

As it was mentioned, GUROBI optimizer has many mathematical programming based solvers. Mixed integer linear programming solver (MILP) is one of them. It could read LP files efficiently. In this study, DTCT analyses for the prepared networks have been utilized by this solver. A code is written for reading and solving the LP files to find optimal total cost of the project networks one by one. Besides, a time limit which is 600 seconds is put in the code. If the model could not find the desired optimal solution in 600 seconds, there will be a warning written as “TimeLimit is reached for problem ...” in the file. A screen view while GUROBI is running on the console window could be seen in Figure 4.3. Moreover, optimal solution output in txt format of a project network is shown in Figure. 4.4.

In the output file, the value “Optimal objective” expression represents the optimal total cost of the project network. Then, the outputs related to selected and unselected time-cost modes are added. For example, 1 is printed for “A2M1” variable which means that first mode of the second activity is chosen for the optimal solution. Intrinsically, 0 is written near the remaining modes of the second activity since they are not selected. Furthermore, “D” variable seems as 0 in the output which means that total project duration does not exceed the given deadline. On the other hand, the project duration is 378 days which is printed near “DUR” variable in the file. In the last part of output, the finish dates of activities are indicated.

Besides, in order to follow and keep the iteration details in GUROBI optimizer, LOG files have been created during execution of the code. An example from a LOG file could be seen in Figure 4.5.

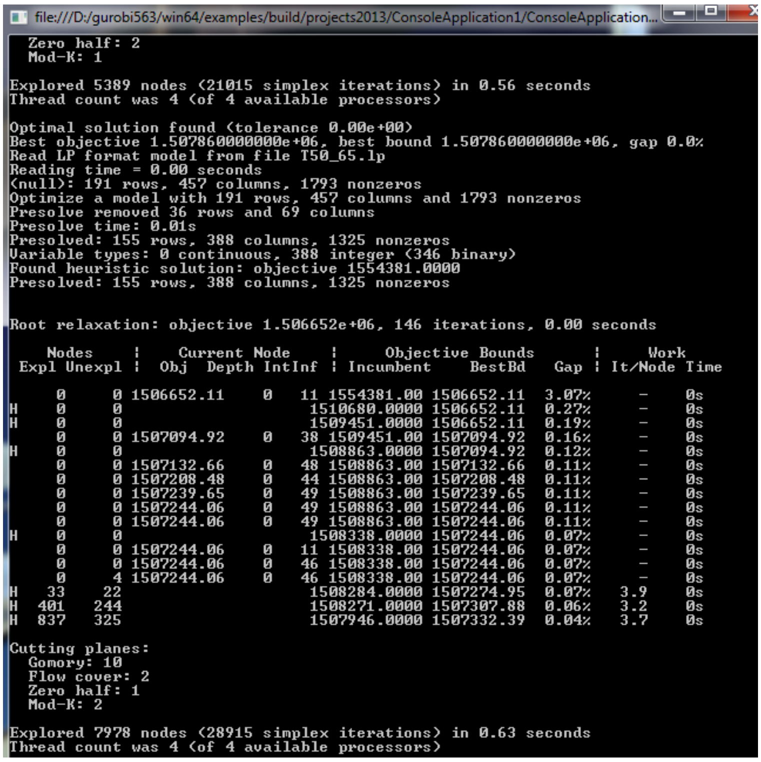
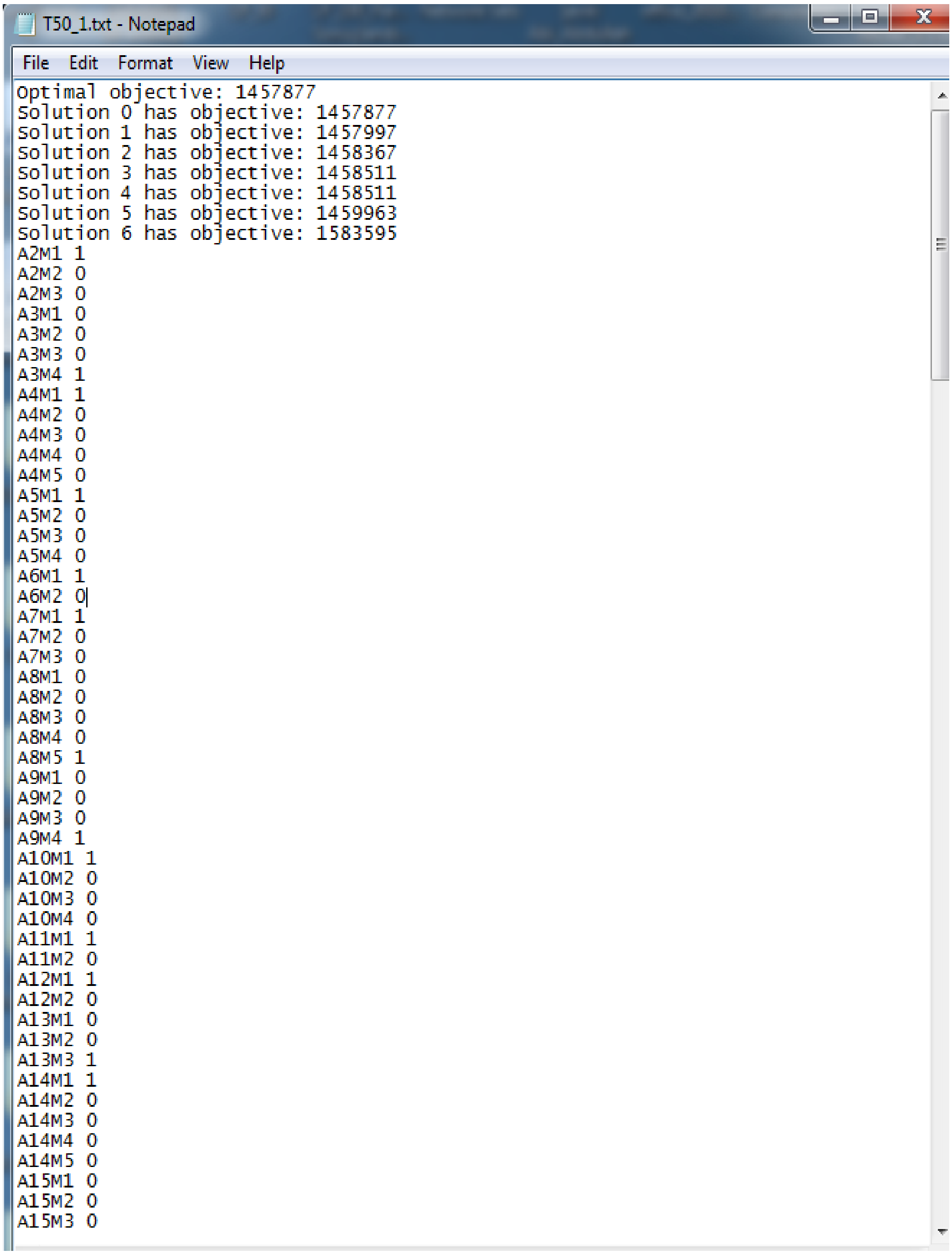


Figure 4.3: A View from Console Window while GUROBI is Running

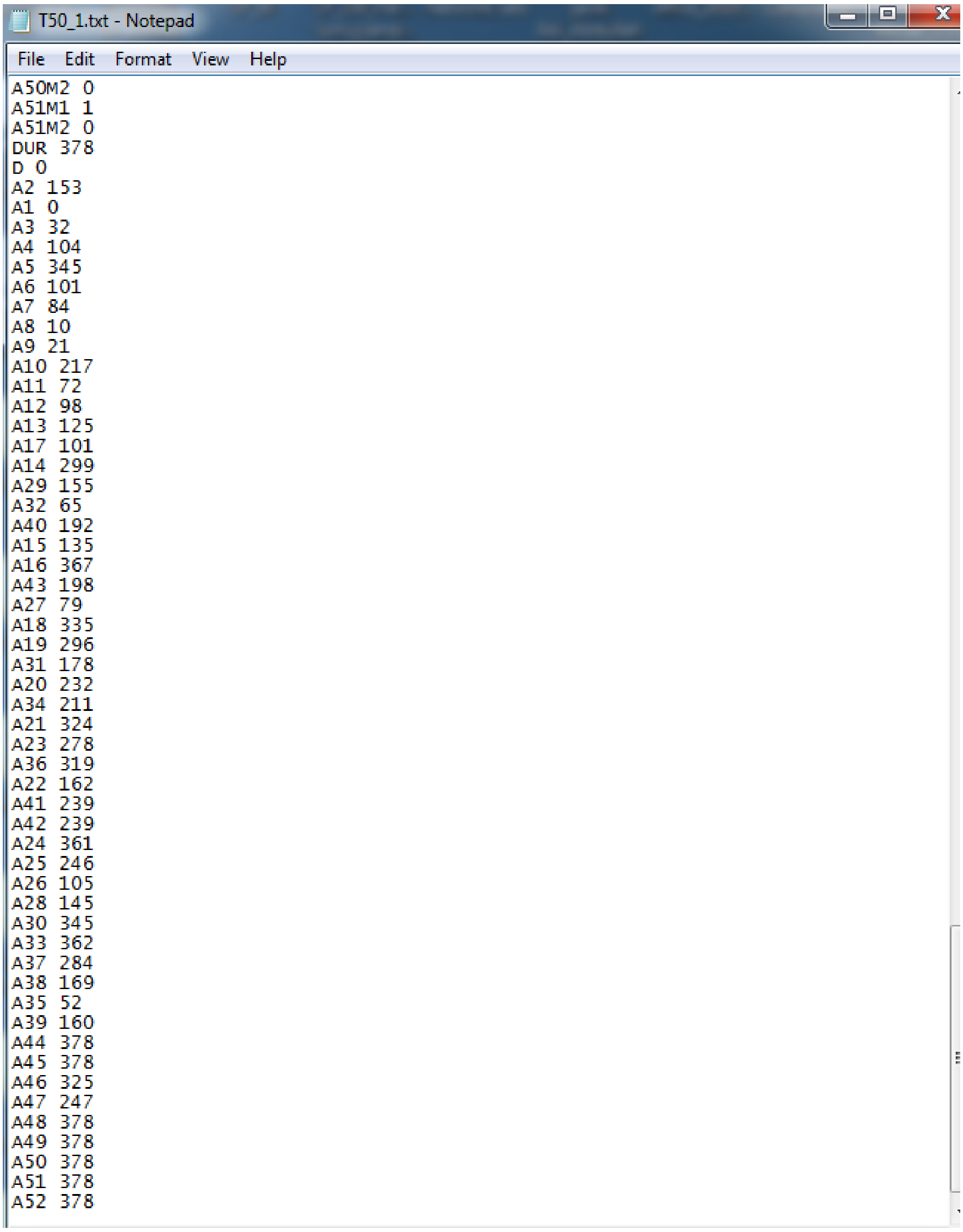


T50_1.txt - Notepad

File Edit Format View Help

```
Optimal objective: 1457877
Solution 0 has objective: 1457877
Solution 1 has objective: 1457997
Solution 2 has objective: 1458367
Solution 3 has objective: 1458511
Solution 4 has objective: 1458511
Solution 5 has objective: 1459963
Solution 6 has objective: 1583595
A2M1 1
A2M2 0
A2M3 0
A3M1 0
A3M2 0
A3M3 0
A3M4 1
A4M1 1
A4M2 0
A4M3 0
A4M4 0
A4M5 0
A5M1 1
A5M2 0
A5M3 0
A5M4 0
A6M1 1
A6M2 0
A7M1 1
A7M2 0
A7M3 0
A8M1 0
A8M2 0
A8M3 0
A8M4 0
A8M5 1
A9M1 0
A9M2 0
A9M3 0
A9M4 1
A10M1 1
A10M2 0
A10M3 0
A10M4 0
A11M1 1
A11M2 0
A12M1 1
A12M2 0
A13M1 0
A13M2 0
A13M3 1
A14M1 1
A14M2 0
A14M3 0
A14M4 0
A14M5 0
A15M1 0
A15M2 0
A15M3 0
```

Figure 4.4: Output File of a-50-activity Network in txt Format



T50_1.txt - Notepad

File Edit Format View Help

```
A50M2 0
A51M1 1
A51M2 0
DUR 378
D 0
A2 153
A1 0
A3 32
A4 104
A5 345
A6 101
A7 84
A8 10
A9 21
A10 217
A11 72
A12 98
A13 125
A17 101
A14 299
A29 155
A32 65
A40 192
A15 135
A16 367
A43 198
A27 79
A18 335
A19 296
A31 178
A20 232
A34 211
A21 324
A23 278
A36 319
A22 162
A41 239
A42 239
A24 361
A25 246
A26 105
A28 145
A30 345
A33 362
A37 284
A38 169
A35 52
A39 160
A44 378
A45 378
A46 325
A47 247
A48 378
A49 378
A50 378
A51 378
A52 378
```

Figure 4.4: Output File of a-50-activity Network in txt Format *(Continued)*

T50_1.log - Notepad

File Edit Format View Help

```
Gurobi 5.6.3 (win64) logging started 02/16/15 17:39:27

Read LP format model from file T50_1.lp
Reading time = 0.00 seconds
(null): 136 rows, 221 columns, 605 nonzeros
Optimize a model with 136 rows, 221 columns and 605 nonzeros
Found heuristic solution: objective 1.54905e+06
Presolve removed 82 rows and 119 columns
Presolve time: 0.00s
Presolved: 54 rows, 102 columns, 257 nonzeros
Variable types: 0 continuous, 102 integer (88 binary)
Presolve removed 1 rows and 1 columns
Presolved: 53 rows, 101 columns, 255 nonzeros


Root relaxation: objective 1.456721e+06, 29 iterations, 0.00 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0 1456721.04    0    7 1549054.00 1456721.04  5.96%     -    0s
H    0     0                    1463438.0000 1456721.04  0.46%     -    0s
H    0     0                    1459285.0000 1456721.04  0.18%     -    0s
     0     0 1456951.07    0    6 1459285.00 1456951.07  0.16%     -    0s
H    0     0                    1458511.0000 1456951.07  0.11%     -    0s
     0     0 1457161.22    0   17 1458511.00 1457161.22  0.09%     -    0s
     0     0 1457161.22    0    7 1458511.00 1457161.22  0.09%     -    0s
     0     0 1457161.22    0   12 1458511.00 1457161.22  0.09%     -    0s
     0     0 1457250.96    0   16 1458511.00 1457250.96  0.09%     -    0s
     0     0 1457336.87    0   15 1458511.00 1457336.87  0.08%     -    0s
     0     0 1457349.64    0   13 1458511.00 1457349.64  0.08%     -    0s
H    0     0                    1457997.0000 1457349.64  0.04%     -    0s
     0     0 1457349.64    0    7 1457997.00 1457349.64  0.04%     -    0s
     0     0 1457350.64    0   13 1457997.00 1457350.64  0.04%     -    0s
     0     0 1457357.42    0   17 1457997.00 1457357.42  0.04%     -    0s
     0     0 1457357.42    0   17 1457997.00 1457357.42  0.04%     -    0s
H   20     0                    1457877.0000 1457404.45  0.03%   2.6    0s

Cutting planes:
  Gomory: 7
  Flow cover: 17
  Zero half: 1
  Mod-K: 2

Explored 77 nodes (325 simplex iterations) in 0.06 seconds
Thread count was 4 (of 4 available processors)

Optimal solution found (tolerance 0.00e+00)
Best objective 1.457877000000e+06, best bound 1.457877000000e+06, gap 0.0%
Optimize a model with 136 rows, 221 columns and 605 nonzeros
Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    1.4578770e+06   0.000000e+00   0.000000e+00      0s

Solved in 0 iterations and 0.00 seconds
Optimal objective  1.457877000e+06
```

Figure 4.5: An Example LOG File Created for a 50-activity Network

4.2.3 A Method to Achieve the Optimal Pareto Front

Since Pareto front curve problem is more complicated than finding optimal solution, a method based on MIP is presented. In this method, minimum CPM duration is calculated by taking minimum time modes of the activities. The same procedure is utilized using maximum time modes. These values could be depicted as the lower and upper duration boundaries for the sample benchmark problems. In other words, the project duration will certainly have a value between these boundaries. After determination of these values, the proposed model and GUROBI optimizer are used to find the optimal project costs for each feasible duration between the boundaries one by one. Pareto front curve includes the non-dominated solutions in the network. Therefore, while calculating the costs, a comparison is utilized between each duration step and related optimal total cost. If there is a dominated solution (i.e. solution having larger or equal total cost for longer project duration), it is deleted and is not printed to the output file. A flowchart displaying the solution algorithm of Pareto front calculations could be seen in Figure 4.6. According to the flowchart, initially the critical path(s) of the problem network is identified. Afterwards, maximum and minimum CPM durations are determined using minimum and maximum time-cost values of activities. Starting from the minimum CPM duration, the optimal cost of the benchmark problems is calculated for each duration step. Then, two successive optimal solutions are compared. If the solution is a non-dominated solution, it is written to output file. Otherwise, the solution is taken out. This process is continued until the duration is equal to maximum CPM duration. Also, 600 seconds time limit is used for each network in Pareto front Curve calculations.

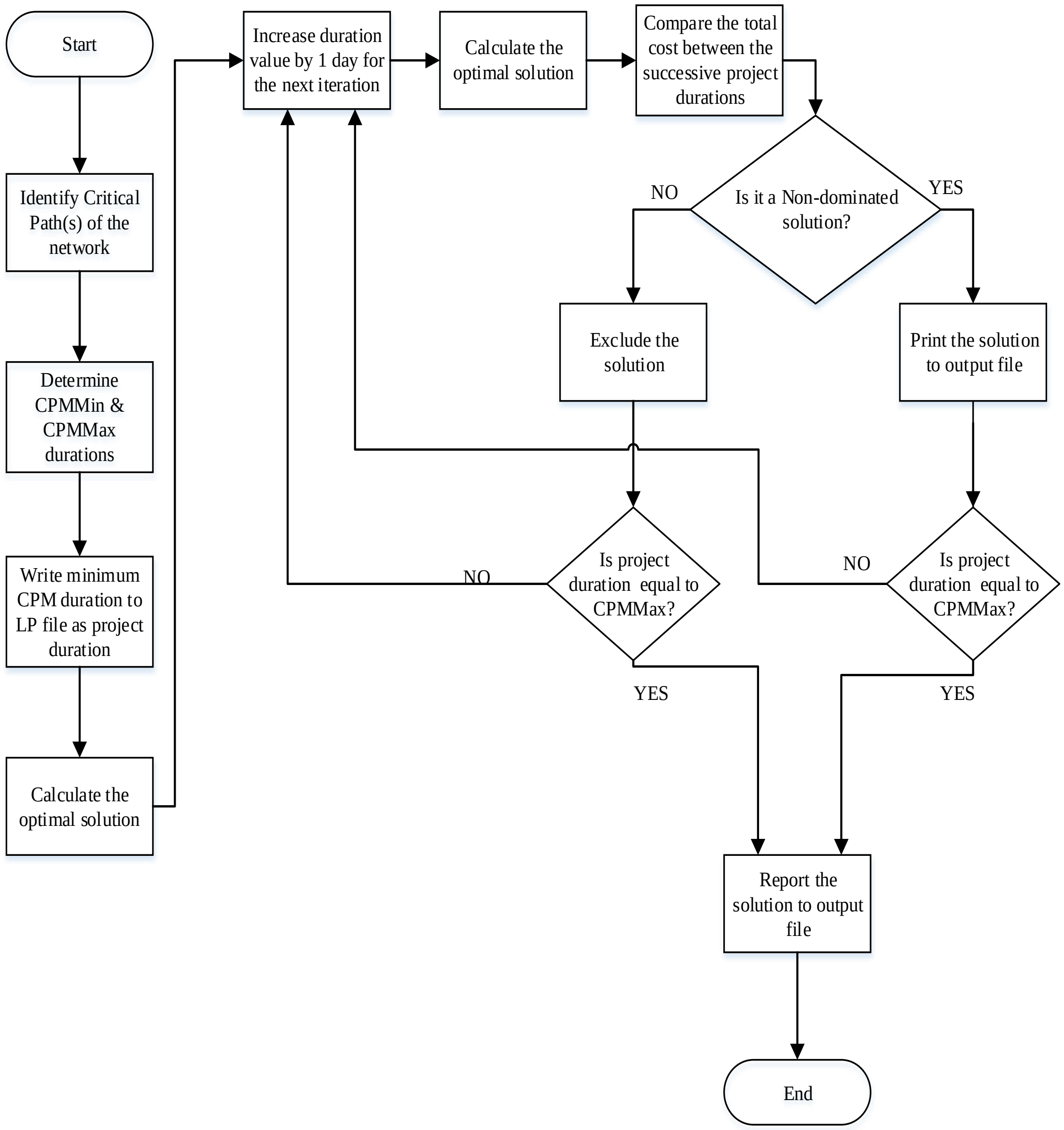


Figure 4.6: Flowchart of the Proposed Solution Algorithm for Pareto front Curve

4.3 Computational Results

4.3.1 Single Objective Problem Optimization

In this part, it is aimed to find the optimal costs of the benchmark problems which could be regarded as a single objective optimization problem. Details of optimal cost solutions in terms of number of solved instances and CPU times are given in Table 4.1 and Table 4.2 respectively. Since there is a 600-second time limit, when this limit is exceeded, the code passes to the next problem in the set. As it was mentioned before, there are eight different time-cost mode intervals containing 10 test instances for each complexity index. For optimal cost problems, networks having activities 50, 100, 200, 500 and 1000 are tried to be solved. In Table 4.1, the amount of solved test instances in the given time limit could be seen. Moreover, CPU time is calculated by taking the average of the solution time of the sets. For optimal cost solutions, all of the 50-activity networks are solved. In both Table 4.1 and Table 4.2, it is obvious that computational time gets longer with when the number of activities in the network increases. In Figure 4.7-a, this issue is stated. There are three other graphs related to computational analyses in Figure 4.7. Chart in Figure 4.7-b is drawn for the relationship between the number of solved instances and number of activities in the networks. It is evident that number of solved instances decreases as the number of activities increases. Figure 4.7-c shows CPU times increase as the maximum number of modes in the networks rises. Four sample networks having 50 activities are chosen to demonstrate this situation. Actually, this is an anticipated outcome. However, there are some deviations in the output table which is not parallel to this case. The reason behind this event is that the average CPU time is only calculated for the solved instances. If a test instance could not be solved in 600 seconds time limit, the duration of that instance is not taken into consideration for calculation of CPU time. Finally, Figure 4.7-d represents the relationship between complexity index called as Thesen restrictiveness coefficient and the computational time. As it was mentioned before, there are four different restrictiveness coefficients which are 0.2, 0.4, 0.6 and 0.8. In the drawn chart, four computational durations are chosen from 1000-activity networks. As it was expected, CPU time increases as the networks get more complex.

As it could be seen from Table 4.2, solution durations are shorter compared to similar examples in the literature. For example, Hazır et al (2011) solve the 136-activity network with an MIP algorithm in 19139.61 seconds which is even far more than the maximum CPU time in 1000-activity networks in this study. In addition, model proposed by Szmerekovsky and Venkateshan (2012) had a CPU time of 206 seconds to solve 90 activity network. This result could be regarded as a long time when compared to the solution times of the medium and large scaled benchmark problems in this study.

Furthermore, as it was mentioned before, totally 800 test instances are generated. For optimal cost solutions, all the generated benchmark problems comprised of 50 and 100 activities have been solved in the given time limit. Solution percentage of 200-activity networks is about %80. In addition, approximately 58% of the created 500-activity benchmark problems are solved. The rate is decreased to 45% for 1000-activity networks. These solution rates are very significant in the given limit.

Table 4.1: Number of Solved Instances in Optimal Cost Solutions

# of Activities	# of Time-Cost Modes																Total # of Solved Instances	Solution Percentage (%)
	2--5				6--10				11--15				16--20					
	Thesen Restrictiveness Coefficient								Thesen Restrictiveness Coefficient									
	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8		
	# of Test Instances				# of Test Instances				# of Test Instances				# of Test Instances					
50	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	160	100.0
100	10	10	10	10	9	10	5	10	10	10	10	10	10	10	10	10	154	96.3
200	10	10	10	10	9	9	5	9	6	7	5	8	8	7	7	7	127	79.4
500	10	10	8	10	5	5	3	2	5	4	6	5	6	6	4	4	93	58.1
1000	7	6	8	7	3	2	5	4	4	6	1	4	2	4	5	4	72	45.0

Table 4.2: Average CPU Time in Optimal Cost Solutions

	CPU Time(seconds)															
50	0.12	0.14	0.24	0.13	0.41	0.87	1.04	0.58	0.82	1.06	3.33	0.93	1.03	1.14	3.31	0.91
100	0.54	0.28	1.28	0.33	176.34	56.54	2.08	0.96	0.03	0.35	1.33	1.22	0.03	0.23	2.81	0.45
200	8.34	0.54	2.76	3.32	161.23	13.54	77.80	7.78	168.44	86.68	11.40	3.00	92.59	132.27	146.36	3.12
500	29.20	9.69	22.32	24.29	134.22	96.39	16.26	175.76	81.26	145.20	220.17	38.13	209.32	163.49	156.01	20.64
1000	20.75	54.02	68.28	90.38	303.93	267.66	201.60	119.37	316.39	87.62	58.37	285.16	326.09	105.95	205.70	273.48

4.3.2 Pareto Front Optimization

For time-cost curve problem, networks having activities 50, 100 and 200 are studied. As it is seen in Table 4.2, 200-activity networks have been tried to be solved up to 10 modes. In the given time limit, only one set could be solved for these networks. Hence, there is no need to try to solve networks having more than 200 activities for this model. However, it is still a significant progress to study 200-activity networks for finding Pareto front curve where the maximum activity number studied in the literature is 50 by Demeulemeester et al. (1998). Details of solved instances and average computational time of Pareto front curve solutions could be seen in Table 4.3 and Table 4.4 respectively.

Since these parameters are not independent from each other, the graphical representation may not always show the same disposition. They are drawn to shown most possible outcomes.

A comparison could be done with study of Kandil and El-Rayes (2006). In this study, non-dominated solutions set of a 720-activity network is found in approximately 137 hours with a genetic algorithm model using a single processor. Although it is a Meta-heuristic procedure, the performance is worse than the proposed MIP model in this thesis. Moreover, Liu et al (1995) solved a small sized network having 7 activities in approximately 30 minutes which is a long duration for such a network. Another comparison could be done with Demeulemeester et al. (1996). The proposed DP models solved a 45-activity network in 530.40 seconds for Pareto optimization. On the other hand, the solution times for 50 and 100 activity benchmark problems in this thesis are less than this time. Also, Demeulemeester et al. (1998) proposed a branch-and-bound model which solves a 50-activity network in 200 seconds. This time is longer than most of the 50 activity networks solution time in this thesis. Exact procedures in the literature to solve Pareto front curve problem used networks having 50 activities maximum whereas the method proposed in this study solved a 200 activity network besides 50 and 100-activity networks. Moreover, 57% of the created 50-activity networks have been solved in the given time limit. For 100-activity networks, the rate is about 52%.

Table 4.3: Number of Solved Instances in Pareto front Curve Solutions

# of Activities	# of Time-Cost Modes																Total # of Solved Instances	Solution Percentage (%)
	2--5				6--10				11--15				16--20					
	Thesen Restrictiveness Coefficient								Thesen Restrictiveness Coefficient									
	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8		
	# of Test Instances				# of Test Instances				# of Test Instances				# of Test Instances					
50	10	10	10	10	9	6	6	4	8	3	-	1	9	3	1	1	91	56.9
100	10	8	4	-	1	-	-	-	10	8	8	5	10	10	3	6	83	51.9
200	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.6

Table 4.4: Average CPU Time in Pareto front Curve Solutions

	CPU Time(seconds)															
50	37.16	79.21	134.49	204.34	198.34	351.52	414.82	414.12	202.65	331.60	-	547.30	307.81	395.17	474.60	447.04
100	183.35	436.89	475.20	-	463.90	-	-	-	61.29	121.77	199.16	340.14	73.27	214.17	259.63	294.07
200	487.27	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

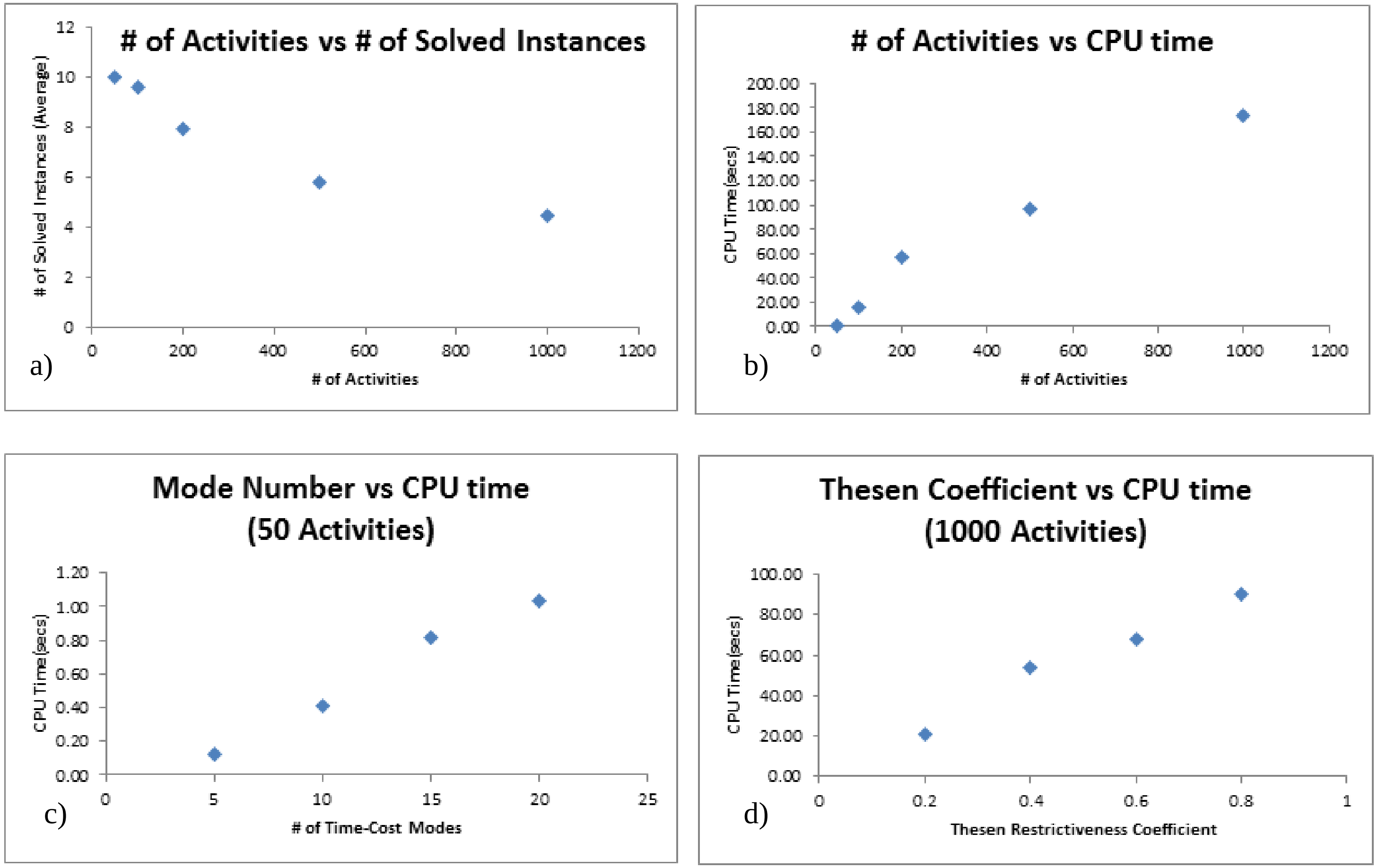


Figure 4.7: Graphical Representations Related to Computational Results

Other than the created networks, in order to have a common understanding with the literature, 18-activity network developed by Feng et al. (1997) is analyzed in this study. Activity on Node (AoN) diagram of the network could be seen in Figure 4.8. Moreover, the precedence relations and the information related to time-cost modes of the network are given in Table 4.5. The sample network is consecutively combined 10 times and a 180-activity network is handled. The created network is solved to find total time-cost curve and Pareto front curve. Total time-cost curve includes all feasible solution in the solution space. First of all, total time-cost and Pareto front curves are found out for zero daily indirect cost. A graphical representation of the Pareto front curve of the network is given in Figure 4.9. Tabulated Pareto front Curve solution outcomes are provided in Appendix A. Total Time-cost curve solutions include the entire Pareto front curve. Hence, the excluded dominated solutions are given as a separate table in the Appendix B. In addition, the network is solved with a daily indirect cost of 200 USD for total time-cost curve. A chart representing the total time-cost curve is provided in Figure 4.10. With the proposed model in GUROBI optimizer, Pareto front curve of the 180-activity network is found out in 99.70 seconds. In addition, total time-cost curve of the 180-activity network with a daily indirect cost of 200 USD is solved in 136.61 seconds. These solution durations are significantly rapid when compared with the similar examples in the literature. The values could be compared with the solution time in Kandil and El-Rayes (2006) also in which solution time of proposed GA for a 180-activity network is about four hours. Evidently, the method proposed in this study is far better than Kandil and El-Rayes.

Besides, the optimality of the outputs is checked one by one in AIMMS optimizer.

Table 4.5: Information about the Network Created by Feng et al. (1997)

Act. ID	Predecessors	Time-Cost Modes									
		Mode #1		Mode#2		Mode#3		Mode#4		Mode#5	
		Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)
1	-	14	2400	15	2150	16	1900	21	1500	24	
2	-	15	3000	18	2400	20	1800	23	1500	25	
3	-	15	4500	22	4000	33	3200	-	-	-	-
4	-	12	45000	16	35000	20	30000	-	-	-	-
5	1	22	20000	24	17500	28	15000	30	10000	-	-
6	1	14	40000	18	32000	24	18000	-	-	-	-
7	5	9	30000	15	24000	18	22000	-	-	-	-
8	6	14	220	15	215	16	200	21	208	24	120
9	6	15	300	18	240	20	180	23	150	25	100
10	2,6	15	450	22	400	33	320	-	-	-	-
11	7,8	12	450	16	350	20	300	-	-	-	-
12	5,9,10	22	2000	24	1750	28	1500	30	1000		
13	3	14	4000	18	3200	24	1800	-	-	-	-
14	4,10	9	3000	15	2400	18	2200	-	-	-	-
15	12	16	3500	-	-	-	-	-	-	-	-
16	13,14	20	3000	22	2000	24	1750	28	1500	30	1000
17	11,14,15	14	4000	18	3200	24	1800	-	-	-	-
18	16,17	9	3000	15	2400	18	2200	-	-	-	-

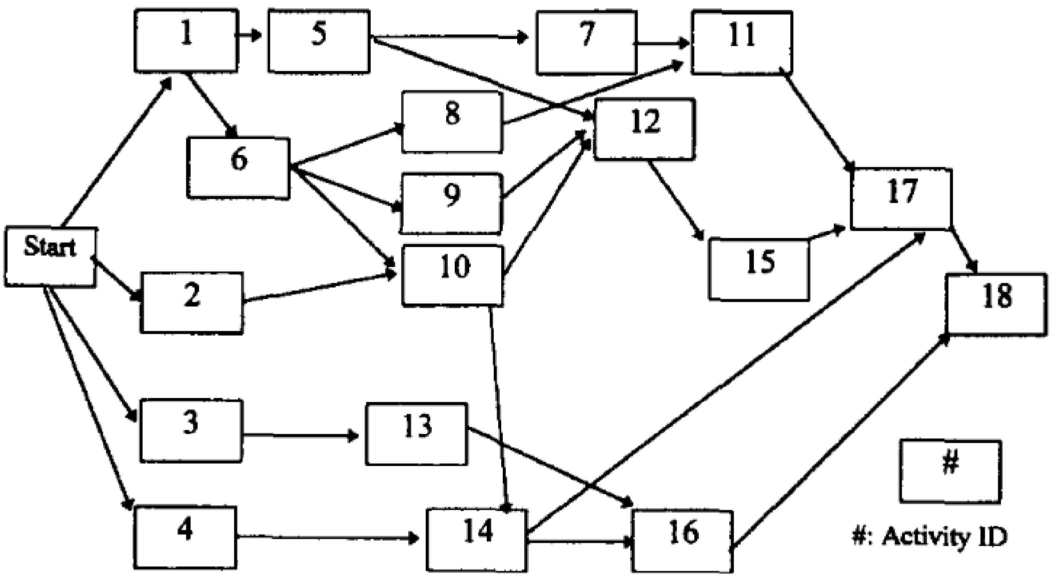


Figure 4.8: AoN Diagram of the Network Created by Feng et al. (1997).

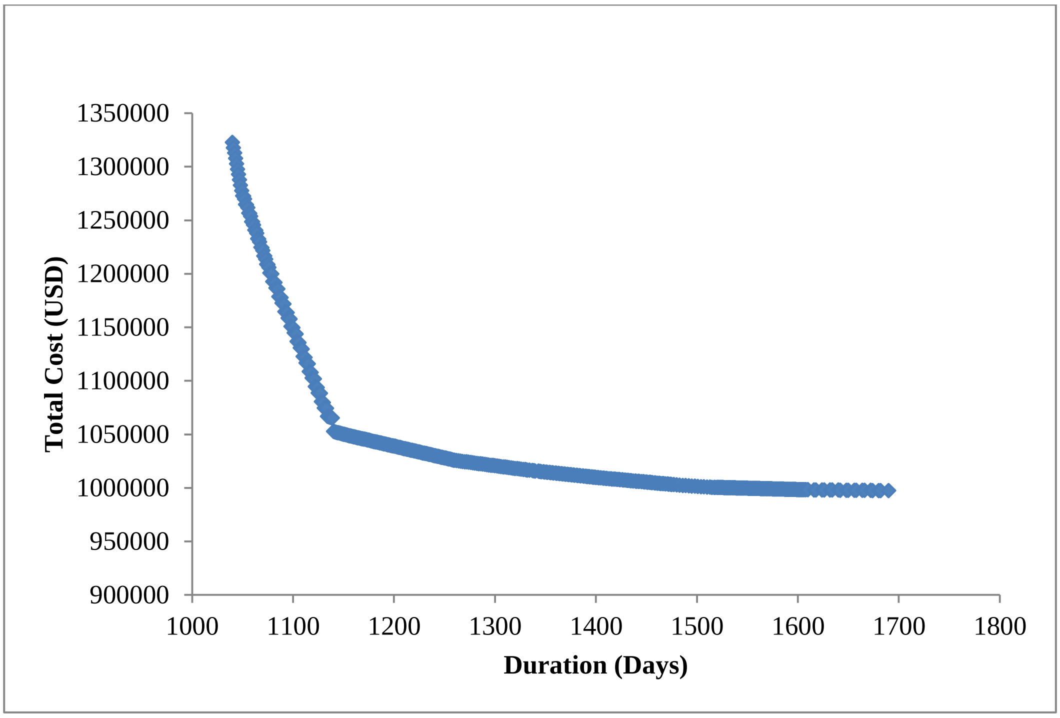


Figure 4.9: Pareto front Curve of the Created 180-activity Network (Zero Indirect Cost)

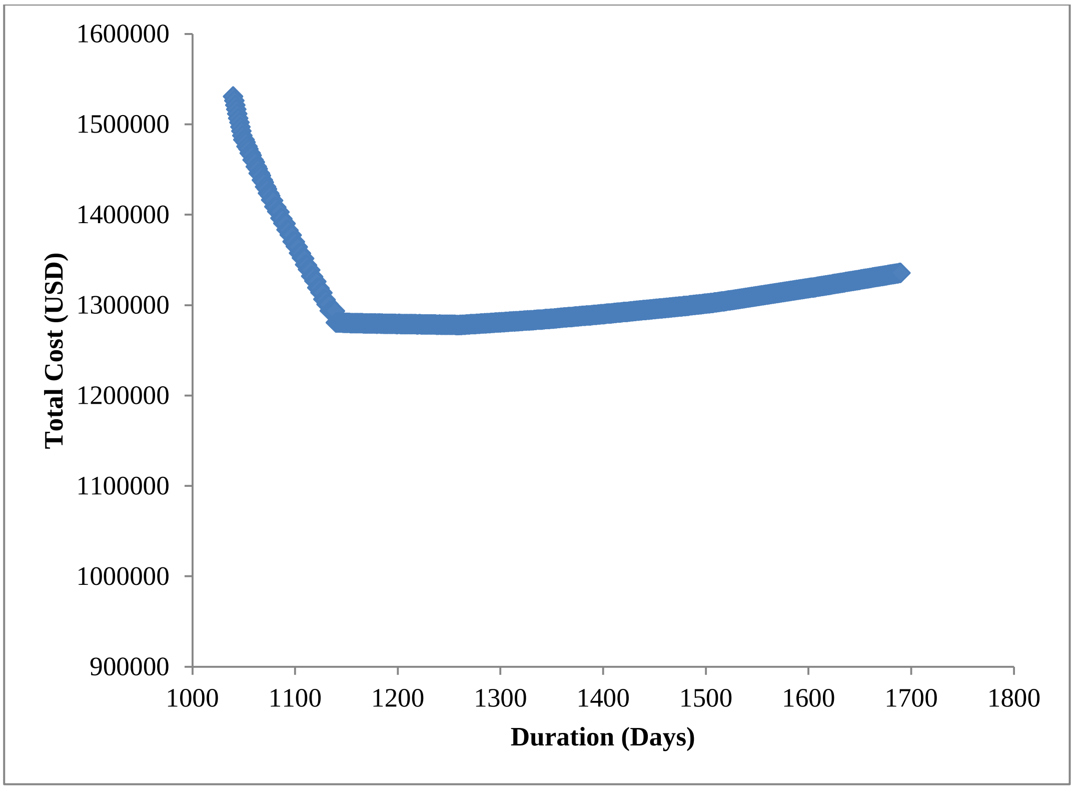


Figure 4.10: Total Time-Cost Curve of the Created 180-activity Network (Indirect Cost of 200 USD)

CHAPTER 5

CONCLUSION

This study focuses on DTCTP analyses of project networks which is a crucial issue in scheduling of construction projects. Although it has been researched in the literature, mostly small scaled benchmark problems have been studied. It is hard to find benchmark test instances having medium and large scales prepared for DTCTP. In this respect, medium and large scaled benchmark problems developed in this study become more significant. Since most of the real life construction projects have many activities, approximation to practical applications will be more efficient with the prepared test instances.

Sample networks are generated by using ProGen/max software. Four different complexity indexes are determined. With these indexes test instances are created having 50, 100, 200, 500 and 1000 activities. The scale of these networks could be regarded as medium and large scales when compared the ones studied in the literature. In fact, research containing more than 360 activities is very rare. Also, the networks are created according to four different time-cost modes. With all the parameters, a total of 800 test instances have been constructed. Besides these networks, in order to create a parallel perception with the literature, the famous 18-activity network developed by Feng et al. (1997) has been used in the analyses. However, since the problem is a small-scale problem, the network is joined 10 times in sequence and a 180-activity network is developed. The created network is solved for Pareto front and total time cost curves. A comparison has been done with the study of Kandil and El-Rayes (2006) in which a similar 180-activity network is studied for Pareto front optimization with GA. As a result of the comparison, it is seen that the solution time in this study is far shorter than Kandil and El-Rayes (2006).

Initially, an MIP model is proposed to solve single criteria optimization problem which aims to find optimal cost in the benchmark problems. Then, a method integrated with this MIP model is improved to solve Pareto front problem. The proposed model and method have been utilized in GUROBI optimizer version 5.6.3. Coding is performed via C# (.NET) language in Microsoft Visual Studio 2013.

Mainly, two problems are studied in this study. In the first problem, optimal costs of the created sample networks are tried to be found out. Then, the aim is to reach the non-dominated solution sets in the solution space of the networks which is also called as Pareto front curve. In the literature, there is an obvious lack about studying Pareto front curve problem for large-scale networks. Most of the real life construction projects have high number of activities. Thus, in order to approximate the analysis to real life conditions, studying with large scale problems is essential. Hence, this thesis tries to compensate this insufficiency with focusing on large-scale networks as well as medium-sized problems.

In the last years, researchers try to develop heuristic & meta-heuristic procedures instead of exact procedures. There is couple of reasons behind this approach. One of the main motivations could be the claim of the fast convergence capabilities of heuristic & meta-heuristic methods when compared to exact procedures. However, these procedures do not guarantee optimality which is a critical drawback. MIP models are criticized with time consumption as the scale of network gets larger while solving discrete time cost trade-off problems due to NP-hard feature. However, the comparisons in terms of CPU times done in discussions on computational results part show that the proposed MIP model applied in GUROBI optimizer presents better performance than some current Meta-heuristic approaches. Computational time is a very fundamental parameter in DTCTP analyses to measure the effectiveness of procedures.

The optimality guarantee of MIP models provides a basis for performance evaluation of Heuristic & Meta-heuristic procedures. By means of the evaluation, success of the applied procedure is measured.

Besides optimal cost solutions, the aim of this study is to emphasize the importance of Pareto front curve solutions. Set of non-dominated solutions gives an inclusive result about DTCTP of project networks. Most of the studies in the literature focus on networks up to 50 activities while solving Pareto front curve problem. Even these studies have longer computational time than the proposed model in this study. This thesis tries to find the set of non-dominated solutions up to 200-activity networks with more reasonable CPU times. This situation makes the thesis significant in which large scaled networks are used as test instances to solve Pareto front curve problem. With this study, future research may focus on solving Pareto front curve exactly for large scale problem sets.

With the mentioned reasons, this study highlights the effectiveness of MIP methods to solve DTCTP in construction projects. Also, it could be claimed that researchers will try work with large networks to realize practical real life applications in construction industry. Furthermore, the proposed model and methods could be improved by adding new constraints in future studies. MIP models are prone to be improved with small modifications which is an advantage. Additionally, parallel processing methods could be used to shorten the solution time of the networks.

REFERENCES

Afshar, A., Ziaraty, A.K., Kaveh, A. and Sharifi, F. (2009). Nondominated Archiving Multicolony Ant Algorithm in Time-Cost Trade-Off Optimization, *Journal of Construction Engineering and Management*, 135(7), 668-674.

Akkan, C., Drexl, A., and Kimms, A. (2005). Network Decomposition-Based Benchmark results for the Discrete Time-Cost Tradeoff Problem, *European Journal of Operational Research*, 165, 339-358.

Benders, J.F. (1962). Partitioning Procedures for Solving Mixed Variables Programming Problems, *Numerische Mathematik*, 4, 238-252.

Butcher, W.S. (1967). Dynamic Programming for Project Cost-Time Curves, *Journal of the Construction Division, Proceedings of the ASCE*, 93, 59-73.

Colorni, A., Dorigo, M, and Maniezzo, V. (1992). Distributed Optimization by Ant Colonies, *Toward a Practice of Autonomous Systems*, 134-142.

Crowston, W.B., and Thompson, G.L. (1967). Decision CPM: A Method for Simultaneous Planning, Scheduling and Control of Projects, *Operations Research*, 15, 407-426.

Crowston, W.B. (1970). Decision CPM: Network Reduction and Solution, *Operations Research Quarterly*, 21, 435-452.

De, P., Dunne, E.J, Ghosh, J.B, and Wells, C.E. (1995). The Discrete Time-Cost Tradeoff Problem Revisited, *European Journal of Operational Research*, 81, 225-238.

De, P., Dunne, E.J, Ghosh, J.B., and Wells, C.E. (1997). Complexity of the Discrete Time-Cost Tradeoff Problem for Project Networks, *Operations Research,* 45(2), 302-306.

Demeulemeester, E., Herroelen W.S., and Elmaghraby S.E. (1996). Optimal Procedures for the Discrete Time/Cost Trade off In Project Networks, *European Journal Of Operational Research,* 88, 50-68.

Demeulemeester, E., De Reyck, B., Foubert B., Herroelen, W., and Vanhoucke, M. (1998). New Computational Results on the Discrete Time/Cost Trade-off Problem in Project Networks, *Journal of Operational Research Society,* 49, 1153-1163.

Demeulemeester, E., Vanhoucke, M., and Herroelen, W. (2003). RANGEN: A Random Network Generator for Activity-on-the-Node Networks, *Journal of Scheduling,* 6, 17-38.

Elbeltagi, E., Hegazy, T., and Grierson, D. (2005). Comparison among Five Evolutionary-Based Optimization Algorithms, *Advanced Engineering Informatics*, 19, 43-53.

Feng C. W., Liu, L., and Burns, S.A. (1997). Using Genetic Algorithms to Solve Construction Time-Cost Trade-off Problems, *Journal of Computing in Civil Engineering*, 11, 184-189.

Fulkerson, D.R. (1961). A Network Flow Computation for Project Cost Curves, *Management Science,* 7, 167-178.

Goyal, S.K (1975). A Note on a Simple CPM Time-Cost Tradeoff Algorithm, *Management Science*, 21(6), 718-722.

Grigoriev, A., and Woeginger G.J. (2004). Project Scheduling with Irregular Costs: Complexity, Approximability, and Algorithms, *Acta Informatica,* 41(2-3), 83-97.

Gurobi Optimizer Reference Manual (2003). Retrieved from http://www.gurobi.com/documentation/, last accessed on June 2015.

Harvey, R.T., and Patterson, J.H. (1979). An Implicit Enumeration Algorithm for the Time/Cost Trade-off Problem in Project Network Analysis, *Foundations of Control Engineering*, 4, 105-117.

Hazır, Ö., Haouari, M., and Erel E. (2010). Discrete Time/Cost Trade-off problem: A decomposition-based solution algorithm for the budget version, *Computers& Operations Research*, 37, 649-655.

Hazır, Ö., Erel E., and Günalay, Y. (2011). Robust Optimization Models for the Discrete Time/Cost Trade-off Problem, *Int. J. Production Economics*, 130, 87-95.

Hegazy, T. (1999). Optimization of Construction Time-Cost Trade-off Analysis Using Genetic Algorithms, *Canadian Journal of Civil Engineering*, 26(6), 685-697.

Herroelen, W., Demeulemeester, E., and De Reyck, B. (1999). A Classification Scheme for Project Scheduling Problems, In J. Weglarz(Ed.), *Handbook on Recent advances in project scheduling*, Chap. 1, 1-26. Amsterdam: Kluwer Academic.

Hindelang, T.J, and Muth, J.F. (1979). A Dynamic Programming Algorithm for the Time/Cost Tradeoff Problem in Project Network Analysis, *Operations Research*, 27-225-241.

Holland, J.H. (1975). Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, *University of Michigan Press.*

Kandil, A., and El-Rayes, K. (2006). Parallel Genetic Algorithms for Optimizing Resource Utilization in Large-Scale Construction Projects, *Journal of Construction Engineering and Management*, 132(5), 491-498.

Kelley, E.J. (1961). Critical-Path Planning and Scheduling: Mathematical Basis, *Operations Research,* 9(3), 296-320.

Kennedy, J., and Eberhart, R. (1995). Particle Swarm Optimization, *Proceedings of the IEEE International Conference on Neural Networks,* Volumes 1-6, 1942-1948.

Liu, L., Burns, S.A., and Feng C. W. (1995). Construction Time-Cost Trade-off Analysis Using LP/IP Hybrid Method, *Journal of Construction Engineering and Management*, 121(4), 446-464.

Meyer, W.L., and Shaffer, L.R. (1963). “Extensions of the Critical Path Method through the Application of Integer Programming, *Civil Engineering Construction Research Series No: 2,* University of Illinois, Urbana Champaign, III.

Moselhi, O. (1993). Schedule Compression Using the Direct Stiffness Method, *Canadian Journal of Civil Engineering*, 20, 65-72.

Moussourakis, J., and Haksever, C. (2004). Flexible Model for Time/Cost Tradeoff Problem, *Journal of Construction Engineering and Management,* 130(3), 307-314.

Ng, S.T., and Zhang Y. (2008). Optimizing Construction Time and Cost Using Ant Colony Optimization Approach, *Journal of Construction Engineering and Management,* 134, 721-728.

Pareto, V. (1906). Manual of Political Economy. *Travaux De Droit, D’économie, De Sociologie Et De Sciences Politiques.*

Robinson, D.R. (1975). A Dynamic Programming Solution to the Cost-Time Tradeoff for CPM, *Management Science,* 22,158-166.

Schwindt, C. (1995). ProGen/max: A New Problem Generator for Different Resource-Constrained Project Scheduling Problems with Minimal and Maximal Time Lags, *Institut für Wirtschaftstheorie und Operations Research Universtitat Karlsruhe,* Technical Report.

Siemens, N. (1971). A Simple CPM Time-Cost Tradeoff Algorithm, *Management Science,* 17(6), B354-B363.

Sönmez, R., and Bettemir, Ö.H. (2012). A Hybrid Genetic Algorithm for the Discrete Time-Cost Trade-off Problem, *Expert Systems with Applications,* 39(13), 11428-11434.

Szmerekovsky, J.G., and Venkateshan, P. (2012). An Integer Programming Formulation for the Project Scheduling Problem with Irregular Time-Cost Tradeoffs, *Computers & Operations Research,* 39, 1402-1410.

Vanhoucke, M., Demeulemeester, E., and Herroelen, W. (2002). Discrete Time/Cost Trade-offs in Project Scheduling with Time-Switch Constraints, *Journal of the Operational Research Society,* 53, 741-751.

Vanhoucke, M. (2005). New Computational Results for the Discrete Time/Cost Trade-off Problem with Time-Switch Constraints, *European Journal of Operational Research,* 165, 359-374.

Vanhoucke, M., and Debels, D. (2007). The Discrete Time/Cost Trade-off Problem: Extensions and Heuristic Procedures, *Journal of Scheduling,* 10(4-5), 311-326

Yang, H.H., and Chen, Y.L. (2000). Finding the Critical Path in an Activity Network with Time-Switch Constraints, *European Journal of Operational Research,* 120, 603-613.

Xiong, Y., and Kuang, Y.P. (2008). Applying an Ant Colony Optimization Algorithm-Based Multiobjective Approach for Time-Cost Trade-off, *Journal of Construction Engineering and Management*, 134(2), 153-156.

Yang, I.T. (2007). Performing Complex Project Crashing Analysis with Aid of Particle Swarm Optimization Algorithm, *International Journal of Project Management*, 25, 637-646.

Zhang, H., and Li, H. (2010). Multi-objective Particle Swarm Optimization for Construction Time-Cost Tradeoff Problems, *Construction Management and Economics*, 28, 75-88.

Zheng, D.X.M., Ng, S.T., and Kumaraswamy M.M. (2004). Applying a Genetic Algorithm-Based Multiobjective Approach for Time-Cost Optimization, *Journal of Construction Engineering Management,* 130, 168-176.

Zheng, D.X.M., Ng, S.T., and Kumaraswamy M.M. (2005). Applying Pareto Ranking and Niche Formation to Genetic Algorithm-Based Multiobjective Time-Cost Optimization, *Journal of Construction Engineering and Management,* 131(1), 81-91.

Zheng, D.X.M., and Ng, S.T. (2012). An Ant Colony System Based Decision Support System for Construction Time-Cost Optimization, *Journal of Civil Engineering and Management,* 18(4), 580-58.

APPENDIX A

A. PARETO FRONT CURVE SOLUTION OUTPUTS OF 180-ACTIVITY NETWORK

Table A.1: Pareto Front Curve Solution Outputs of 180-Activity Network

Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)
1040	1322700	1055	1261700	1070	1221700	1085	1185700
1041	1317700	1056	1256700	1071	1216700	1086	1178700
1042	1312700	1057	1256450	1072	1216450	1087	1178450
1043	1307700	1058	1253700	1073	1213700	1088	1177700
1044	1302700	1059	1248700	1074	1208700	1089	1172700
1045	1297700	1060	1248450	1075	1208450	1090	1172450
1046	1292700	1061	1245700	1076	1205700	1091	1171700
1047	1287700	1062	1240700	1077	1200700	1092	1164700
1048	1282700	1063	1240450	1078	1200450	1093	1164450
1049	1277700	1064	1237700	1079	1199700	1094	1163700
1050	1272700	1065	1232700	1080	1192700	1095	1158700
1051	1272450	1066	1232450	1081	1192450	1096	1158450
1052	1269700	1067	1229700	1082	1191700	1097	1157700
1053	1264700	1068	1224700	1083	1186700	1098	1150700
1054	1264450	1069	1224450	1084	1186450	1099	1150450

Table A.2: Pareto Front Curve Solution Outputs of 180-Activity Network

(Continued)

Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)
1100	1149700	1115	1115700	1130	1079700	1145	1051450
1101	1144700	1116	1108700	1131	1074700	1146	1051200
1102	1144450	1117	1108450	1132	1074450	1147	1050950
1103	1143700	1118	1107700	1133	1074200	1148	1050700
1104	1136700	1119	1102700	1134	1066700	1149	1050450
1105	1136450	1120	1102450	1135	1066450	1150	1050200
1106	1135700	1121	1101700	1136	1066200	1151	1049950
1107	1130700	1122	1094700	1137	1065950	1152	1049700
1108	1130450	1123	1094450	1138	1065700	1153	1049450
1109	1129700	1124	1093700	1139	1065450	1154	1049200
1110	1122700	1125	1088700	1140	1052700	1155	1048950
1111	1122450	1126	1088450	1141	1052450	1156	1048700
1112	1121700	1127	1088200	1142	1052200	1157	1048450
1113	1116700	1128	1080700	1143	1051950	1158	1048200
1114	1116450	1129	1080450	1144	1051700	1159	1047950

Table A.3: Pareto Front Curve Solution Outputs of 180-Activity Network

(Continued)

Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)
1160	1047700	1175	1044550	1190	1041100	1205	1037950
1161	1047650	1176	1044300	1191	1041050	1206	1037700
1162	1047400	1177	1044050	1192	1040800	1207	1037450
1163	1047150	1178	1043800	1193	1040550	1208	1037200
1164	1046900	1179	1043550	1194	1040300	1209	1036950
1165	1046750	1180	1043300	1195	1040150	1210	1036700
1166	1046500	1181	1043250	1196	1039900	1211	1036650
1167	1046250	1182	1043000	1197	1039650	1212	1036400
1168	1046000	1183	1042750	1198	1039400	1213	1036150
1169	1045750	1184	1042500	1199	1039150	1214	1035900
1170	1045500	1185	1042350	1200	1038900	1215	1035750
1171	1045450	1186	1042100	1201	1038850	1216	1035500
1172	1045200	1187	1041850	1202	1038600	1217	1035250
1173	1044950	1188	1041600	1203	1038350	1218	1035000
1174	1044700	1189	1041350	1204	1038100	1219	1034750

Table A.4: Pareto Front Curve Solution Outputs of 180-Activity Network

(Continued)

Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)
1220	1034500	1235	1031350	1250	1027900	1270	1024450
1221	1034450	1236	1031100	1251	1027850	1272	1024200
1222	1034200	1237	1030850	1252	1027600	1274	1023950
1223	1033950	1238	1030600	1253	1027350	1276	1023700
1224	1033700	1239	1030350	1254	1027100	1278	1023450
1225	1033550	1240	1030100	1255	1026950	1280	1023200
1226	1033300	1241	1030050	1256	1026700	1282	1022950
1227	1033050	1242	1029800	1257	1026450	1284	1022700
1228	1032800	1243	1029550	1258	1026200	1286	1022450
1229	1032550	1244	1029300	1259	1025950	1288	1022200
1230	1032300	1245	1029150	1260	1025700	1290	1021950
1231	1032250	1246	1028900	1262	1025450	1292	1021700
1232	1032000	1247	1028650	1264	1025200	1294	1021450
1233	1031750	1248	1028400	1266	1024950	1296	1021200
1234	1031500	1249	1028150	1268	1024700	1298	1020950

Table A.5: Pareto Front Curve Solution Outputs of 180-Activity Network

(Continued)

Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)
1300	1020700	1330	1017100	1352	1014500	1375	1012300
1302	1020450	1331	1016950	1354	1014400	1376	1012100
1304	1020200	1332	1016700	1355	1014300	1378	1012000
1306	1019950	1334	1016450	1357	1014100	1379	1011900
1308	1019700	1337	1016300	1358	1013900	1381	1011700
1310	1019450	1338	1016100	1360	1013800	1382	1011500
1312	1019200	1339	1015950	1361	1013700	1384	1011400
1314	1018950	1340	1015700	1363	1013500	1385	1011300
1316	1018700	1343	1015640	1364	1013300	1387	1011100
1318	1018450	1344	1015500	1366	1013200	1388	1010900
1320	1018200	1345	1015300	1367	1013100	1390	1010800
1322	1017950	1346	1015100	1369	1012900	1391	1010700
1324	1017700	1348	1015000	1370	1012700	1392	1010700
1326	1017450	1349	1014900	1372	1012600	1393	1010500
1328	1017200	1351	1014700	1373	1012500	1394	1010300

Table A.6: Pareto Front Curve Solution Outputs of 180-Activity Network *(Continued)*

Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)
1396	1010200	1418	1008200	1442	1006100	1464	1004100
1397	1010100	1419	1008100	1443	1006000	1466	1004000
1398	1010100	1421	1007900	1445	1005800	1467	1003900
1399	1009900	1423	1007800	1447	1005700	1469	1003700
1400	1009700	1424	1007600	1448	1005500	1471	1003600
1402	1009600	1426	1007500	1450	1005400	1472	1003400
1403	1009500	1427	1007400	1451	1005300	1474	1003300
1405	1009300	1429	1007200	1453	1005100	1475	1003200
1407	1009200	1431	1007100	1454	1005100	1477	1003000
1408	1009000	1432	1006900	1455	1005000	1480	1002700
1410	1008900	1434	1006800	1456	1004800	1483	1002500
1411	1008800	1435	1006700	1458	1004700	1486	1002300
1413	1008600	1437	1006500	1459	1004600	1489	1002100
1415	1008500	1439	1006400	1461	1004400	1492	1001900
1416	1008300	1440	1006200	1463	1004300	1495	1001700

Table A.7: Pareto Front Curve Solution Outputs of 180-Activity Network *(Continued)*

Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)
1498	1001500	1526	1000300	1541	999930	1556	999550
1501	1001300	1527	1000280	1542	999900	1557	999530
1504	1001100	1528	1000250	1543	999880	1558	999500
1507	1000900	1529	1000240	1544	999850	1559	999480
1510	1000700	1530	1000200	1545	999830	1560	999450
1513	1000640	1531	1000180	1546	999800	1561	999430
1515	1000580	1532	1000160	1547	999780	1562	999400
1517	1000530	1533	1000130	1548	999750	1563	999380
1518	1000500	1534	1000100	1549	999730	1564	999350
1520	1000450	1535	1000080	1550	999700	1565	999330
1521	1000440	1536	1000050	1551	999680	1566	999300
1522	1000410	1537	1000030	1552	999650	1567	999280
1523	1000380	1538	1000000	1553	999630	1568	999250
1524	1000360	1539	999980	1554	999600	1569	999230
1525	1000330	1540	999950	1555	999580	1570	999200

Table A.8: Pareto Front Curve Solution Outputs of 180-Activity Network

(Continued)

Duration (Days)	**Cost (USD)**	**Duration (Days)**	**Cost (USD)**	**Duration (Days)**	**Cost (USD)**	**Duration (Days)**	**Cost (USD)**
1571	999180	1586	998800	1601	998430	1640	997930
1572	999150	1587	998780	1602	998400	1642	997880
1573	999130	1588	998750	1603	998380	1648	997850
1574	999100	1589	998730	1604	998350	1650	997800
1575	999080	1590	998700	1605	998330	1656	997770
1576	999050	1591	998680	1606	998300	1658	997720
1577	999030	1592	998650	1607	998280	1664	997690
1578	999000	1593	998630	1608	998250	1666	997640
1579	998980	1594	998600	1610	998200	1672	997610
1580	998950	1595	998580	1616	998170	1674	997560
1581	998930	1596	998550	1618	998120	1680	997530
1582	998900	1597	998530	1624	998090	1682	997480
1583	998880	1598	998500	1626	998040	1690	997400
1584	998850	1599	998480	1632	998010		
1585	998830	1600	998450	1634	997960		

APPENDIX B

B. DOMINATED SOLUTION OUTPUTS OF 180-ACTIVITY NETWORK

Table B. 1: Dominated Solution Outputs of 180-Activity Network

Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)
1261	1025700	1291	1021950	1321	1018200	1362	1013700
1263	1025450	1293	1021700	1323	1017950	1365	1013300
1265	1025200	1295	1021450	1325	1017700	1368	1013100
1267	1024950	1297	1021200	1327	1017450	1371	1012700
1269	1024700	1299	1020950	1329	1017200	1374	1012500
1271	1024450	1301	1020700	1333	1016700	1377	1012100
1273	1024200	1303	1020450	1335	1016450	1380	1011900
1275	1023950	1305	1020200	1336	1016450	1383	1011500
1277	1023700	1307	1019950	1341	1015700	1386	1011300
1279	1023450	1309	1019700	1342	1015700	1389	1010900
1281	1023200	1311	1019450	1347	1015100	1392	1010700
1283	1022950	1313	1019200	1350	1014900	1395	1010300
1285	1022700	1315	1018950	1353	1014500	1398	1010100
1287	1022450	1317	1018700	1356	1014300	1401	1009700
1289	1022200	1319	1018450	1359	1013900	1404	1009500

Table B. 2: Dominated Solution Outputs of 180-Activity Network *(Continued)*

Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)
1406	1009300	1446	1005800	1482	1002700	1505	1001100
1409	1009000	1449	1005500	1484	1002500	1506	1001100
1412	1008800	1452	1005300	1485	1002500	1508	1000900
1414	1008600	1454	1005100	1487	1002300	1509	1000900
1417	1008300	1457	1004800	1488	1002300	1511	1000700
1420	1008100	1460	1004600	1490	1002100	1512	1000700
1422	1007900	1462	1004400	1491	1002100	1514	1000640
1425	1007600	1465	1004100	1493	1001900	1516	1000580
1428	1007400	1468	1003900	1494	1001900	1519	1000500
1430	1007200	1470	1003700	1496	1001700	1609	998250
1433	1006900	1473	1003400	1497	1001700	1611	998200
1436	1006700	1476	1003200	1499	1001500	1612	998200
1438	1006500	1478	1003000	1500	1001500	1613	998200
1441	1006200	1479	1003000	1502	1001300	1614	998200
1444	1006000	1481	1002700	1503	1001300	1615	998200

Table B. 3: Dominated Solution Outputs of 180-Activity Network *(Continued)*

Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)	Duration (Days)	Cost (USD)
1617	998170	1637	997960	1657	997770	1677	997560
1619	998120	1638	997960	1659	997720	1678	997560
1620	998120	1639	997960	1660	997720	1679	997560
1621	998120	1641	997930	1661	997720	1681	997530
1622	998120	1643	997880	1662	997720	1683	997480
1623	998120	1644	997880	1663	997720	1684	997480
1625	998090	1645	997880	1665	997690	1685	997480
1627	998040	1646	997880	1667	997640	1686	997480
1628	998040	1647	997880	1668	997640	1687	997480
1629	998040	1649	997850	1669	997640	1688	997480
1630	998040	1651	997800	1670	997640	1689	997480
1631	998040	1652	997800	1671	997640		
1633	998010	1653	997800	1673	997610		
1635	997960	1654	997800	1675	997560		
1636	997960	1655	997800	1676	997560		