A MIXED INTEGER PROGRAMMING METHOD FOR INTEGRATED DISCRETE TIME-COST TRADE-OFF AND MANPOWER RESOURCE LEVELING PROBLEM

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

BY

ALİ CAN TATAR

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

SEPTEMBER 2016

Approval of the thesis:

A MIXED INTEGER PROGRAMMING METHOD FOR INTEGRATED DISCRETE TIME-COST TRADE-OFF AND MANPOWER RESOURCE LEVELING PROBLEM

submitted by **ALİ CAN TATAR** 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. İsmail Özgür Yaman
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:

Prof. Dr. M. Talat Birgönül
Civil Engineering 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: 01.09.2016

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: Ali Can Tatar

Signature :

ABSTRACT

A MIXED INTEGER PROGRAMMING METHOD FOR INTEGRATED DISCRETE TIME-COST TRADE-OFF AND MANPOWER RESOURCE LEVELING PROBLEM

Tatar, Ali Can

M.S., Department of Civil Engineering

Supervisor: Assoc. Prof. Dr. Rifat Sönmez

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

September 2016, 83 pages

Construction projects have to meet all of the objectives of scope, quality, schedule, budget simultaneously. These objectives, however, cannot be considered as independent of each other. For example, an increase in direct resources will usually lead to shorter activity durations. A shorter project duration results in lower indirect costs, whereas the additional resources cause an increase the project’s direct costs, in general. This phenomenon is defined as time-cost trade-off problem (TCTP). Nevertheless, in some cases supplying extra resources may increase the indirect costs, too. Hence, the need for a comprehensive approach, integrating TCTP with the optimal manpower resource utilization values, is crucial for optimizing the resources along with the cost. In the literature, however, this problem is considered as two independent sub-problems as TCTP, and resource leveling problem (RLP).

This study introduces an integrated approach considering TCTP and RLP, simultaneously. In this context, a mixed integer programming (MIP) model is presented for solving the discrete time-cost trade-off problem (DTCTP) and resource leveling problem simultaneously. Since there are no benchmark problems for the proposed problem, 1215 benchmark instances are generated for the integrated discrete time-cost trade-off and resource leveling problem (DTCTRLP). A great majority of (97.28%) 10-activity problems are solved, successfully; nonetheless, the solution rate decreased as the problem's activity and mode numbers increase. In addition, the proposed procedure is compared with the current approach in the literature (i.e. consecutive implementation of TCT and RLP), to illustrate the contributions of the proposed approach.

Keywords: Time-cost-resource Trade-off Problem, Discrete Time-cost Trade-off Problem, Resource Leveling Problem, Exact Methods, Mixed Integer Programming.

ÖZ

ENTEGRE KESİKLİ ZAMAN-MALİYET ÖDÜNLEŞİM VE İŞGÜCÜ KAYNAKLARI DENGELEME PROBLEMİ İÇİN DOĞRUSAL TAMSAYILI PROGRAMLAMA YÖNTEMİ

Tatar, Ali Can

Yüksek Lisans, İnşaat Mühendisliği Bölümü

Tez Yöneticisi: Doç. Dr. Rifat Sönmez

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

Eylül 2016, 83 sayfa

İnşaat projeleri, bütün kapsam, kalite, pogram, ve bütçe hedeflerini eş zamanlı olarak yerine getirmelidir. Ancak bu hedefler, birbirinden bağımsız olarak ele alınamaz. Örneğin, direkt kaynaklardaki bir artış, genellikle daha düşük aktivite sürelerini beraberinde getirir. Daha düşük bir proje süresi, daha düşük endirekt maliyetlere yol açar; öte yandan ek kaynaklar, projenin direkt maliyetlerinde genellikle artışa yol açar. Bu olgu, zaman-maliyet ödünleşim problemi olarak tanımlanır (ZMÖP). Bununla birlikte, kaynak arzındaki artış, bazı durumlarda endirekt maliyetlerde de bir artışı beraberinde getirebilir. Bu sebeple, kaynakları maliyetle birlikte optimize etmek için, ZMÖP'nin en uygun işgücü kaynakları kullanım değerleri ile bütünleşmiş bir şekilde düşünüldüğü, kapsayıcı bir yaklaşıma olan ihtiyaç kaçınılmaz hale gelmektedir. Ancak, bu problem literatürde ZMÖP ve kaynak dengeleme problem (KDP) olmak üzere iki farklı alt-problem şeklinde ele alınmıştır.

Bu çalışma, ZMÖP ve KDP'nin eşzamanlı olarak değerlendirildiği, bütünleşik bir yöntem önermektedir. Bu bağlamda, kesikli zaman-maliyet ödünleşim problemi

(KZMÖP) ve KDP’nin eşzamanlı olarak çözümü için bir doğrusal tamsayılı programlama (DTP) modeli sunulmuştur. Önerilen problem için kıstas problemleri mevcut olmadığından, bütünleşik KZMÖP ve KDP (KZMÖPKDP) için 1215 adet kıstas problem üretilmiştir. 10 faaliyetli olan problemlerin çok büyük bir bölümü (%97.28) başarılı bir şekilde çözülmüştür, ancak problemlerin aktivite ve mod sayıları arttıkça çözüm oranı düşmüştür. Buna ek olarak, önerilen yöntemin katkılarının görülmesi için, önerilen yöntem ve literatürdeki mevcut yöntem (yani, ZMÖP ve KDP’nin art arda uygulanması) karşılaştırılmıştır.

Anahtar Kelimeler: Zaman-maliyet-kaynak Ödünleşim Problemi, Kesikli Zaman-maliyet Ödünleşim Problemi, Kaynak Dengeleme Problemi, Kesin Yöntemler, Doğrusal Tamsayılı Programlama.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor Assoc. Prof. Dr. Rifat Sönmez for his guidance, wisdom, and encouragement throughout this study.

Likewise, I wish to offer my thanks co-advisor Asst. Prof. Dr. S. Tankut Atan for his comments and suggestions.

Also, I would like to thank the examining committee members for their valuable feedbacks.

I wish to thank to my colleagues at Construction Management Division especially; Mert Bilir and Saman Aminbakhsh for their remarkable contributions, and to my roommates; Furkan Gökmen, Murat Altun, Ozan Uçal, Özgür Paşaoğlu and Yasin Özkılıç.

I specially thank to my friends Baran Yaman, Berkay Atıcı, Efe Orhan, Emre Konyar, Ferhat Keser, Osman Taştekin and Zafer Karakaş.

Finally, I am deeply grateful to my family; Fatma Tatar, İbrahim Tatar and Pınar Evren, Çınar Evren and Kayhan Deniz Evren for their constant support and motivation throughout my educational life.

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 ABBREVIATIONS

CPM	Critical Path Method
TCT	Time-cost Trade-off
TCTP	Time-cost Trade-off Problem
DTCTP	Discrete Time-cost Trade-off Problem
RLP	Resource Leveling Problem
MRD	Maximum Resource Demand
RCPSP	Resource-constraint Project Scheduling Problem
NP-hard	Non-polynomial Hard
MA	Memetic Algorithm
SFL	Shuffled Frog Leaping
PSO	Particle Swarm Optimization
ACO	Ant Colony Optimization
PMBOK	Project Management Body of Knowledge
GA	Genetic Algorithm
SAM	Siemens Approximation Method
DP	Dynamic Programming
ACT	Activity
RES	Resource
MIP	Mixed Integer Programming
MILP	Mixed Integer Linear Programming
MIQP	Mixed Integer Quadratic Programming (MIQP),
MIQCP	Mixed Integer Quadratically Constrained Programming
LP	Linear Programming
QP	Quadratic Programming
QCP	Quadratically Constrained Programming
BB	Branch-and-bound

DSS	Decision Support System
RCPSP	Resource Constraint Project Scheduling Problem
MRCPSP	Multi-mode Resource Constraint Project Scheduling Problem
MRCDTCTP	Multi-mode Resource Constraint Discrete Time-cost Trade-off Problem
SSGS	Serial Schedule Generation Scheme
IP	Integer Programming
BD	Benders Decomposition Algoritm
TCRO	Time-cost-resource Optimization
NPVP	Net Present Value Problem
DTCTRLP	Discrete Time-cost Trade-off and Resource Leveling Problem

CHAPTER 1

INTRODUCTION

Construction sector is classified under project based industries. A construction project is defined as a unique endeavour with the objective of creating a unique product, service or result by satisfying various constraints including the scope, quality, schedule, budget and resources (PMBOK guide, 2013). These constraints are interrelated to each other; hence, a change in one them affects the remaining constraints, too. For example, an increase in the resource supply will usually lead to a contraction in the schedule. A shorter project duration in general implies a drop in the indirect costs that depend to the project duration, on the other hand, additional resources will increase the project's direct costs. This phenomenon is defined as time-cost trade-off problem in the literature. However, increasing the resource utilization may trigger additional overhead costs (i.e. storage, transportation) in some cases, too. Therefore, a successful project management plan should include an adaptive analysis of total project cost and resource schedules throughout the project duration.

Delivery of a project by its deadline and providing predefined quality within the budget, is the most fundamental responsibility of a contractor. Thus, scheduling is an important component of project management. In construction industry, the most widely used project scheduling technique is the critical path method (CPM). It takes activity sequence relations and activity durations into consideration, and forms paths along successive activities. The longest path defines the total project duration, and activities through this path become critical activities. It means that, a change in the schedule of these activities leads to a change in whole network's schedule. Hence, modifications within the desired duration can be performed by means of shifting non-critical activities, only. In addition to this, the CPM is not capable of optimizing the budget and resources, that is, it cannot be used to effectively evaluate alternative construction methods with regards to their corresponding costs and resource demands. In order to overcome these drawbacks various techniques, such as time-cost trade-off analysis, resource leveling, and resource constraint project scheduling are suggested.

There is a trade-off between an activity's cost and duration. In the literature, this is defined as time-cost trade-off problem (TCTP). It may be either linear or discrete (DTCTP), depending upon the relation between different time-cost modes of a project's activities. Most of the resources used in construction industry are defined in terms of discrete units; hence, these problems are classified as DTCTP. In the literature, studies about DTCTP are categorized as exact methods, heuristics and meta-heuristics. De et al. (1997) define DTCTP as strongly non-polynomial hard (NP-hard). Hence, exact methods may require significant amount of computational time. Moussourakis and Haksever (2004) state that as the scale of networks enlarges, the solution time increases exponentially. Therefore, heuristic and meta-heuristic methods have been gaining attention for research, recently, to provide satisfying results within acceptable computational time. Nonetheless, criterion of a satisfying result cannot be defined unless the problem's optimal result is known; therefore, quality and performance of these methods cannot be evaluated. Exact methods, on the other hand, guarantee the optimality. Hence, they provide a benchmark to be compared with the results of heuristic and meta-heuristics.

Resources utilized to execute the activities of a project are among the main components of project scheduling process. Their quality, productivity, and availability affects a project's success substantially. Construction projects demand utilization of a great variety of resources simultaneously, by their nature. In some cases, though, there might be problems related to the availability of these resources. In the literature, this is defined as resource-constraint project scheduling problem (RCPSP), the objective of which is to find the optimum schedule conforming to precedence relations and given resource availability limits.

Resource scheduling, that is planning and management of resources throughout the project duration, is another significant component of project management. Construction projects in general are known as their requirements for simultaneous utilization of several resources. Hence, cumulative resource schedules are as important as each resource's schedule. A project's resource curve is obtained by considering the resource demand of each activity with regards to the project's CPM schedule. In practice, it is desired to have balanced resource usage profiles with lower peaks instead of fluctuant ones with high peaks, due to economic and physical restrictions. In the literature, efforts to achieve a smooth profile from an arbitrary resource schedule are defined as resource leveling problem (RLP). Resource leveling methods include minimizing daily resource demand deviations, minimizing the surplus amount from a desired quantity, minimizing the resource idle days, and minimizing the daily maximum resource demand (MRD). In this thesis, the overhead costs that are directly associated with the number of maximum daily resource demand (storage, hosting, transportation, security, and so on) are minimized; hence, MRD is taken into consideration. In the literature, there exist exact methods as well as heuristics and meta-heuristics to solve RLP. As it is stated previously in the TCTP part, exact methods are criticized for their longer computational time compared to heuristic and meta-heuristic methods. Thus, the most of the studies are based on heuristics and meta-heuristics. Nonetheless, these methods do not provide optimality guarantee; hence, exact methods become inevitable.

In spite of all the above-mentioned capabilities, current studies analyze TCTP and RLP as they are different sub problems of project planning and management literature. In other words, these efforts do not include an integrated approach that considers a simultaneous implementation of time-cost trade-off and resource leveling, rather than application of these methods one after another. This practice, however, leads to a reduction in the problem's search space resulting in exclusion of better solution alternatives. In this context, there is a need for a more comprehensive study that takes time, cost, and resource relations into consideration.

Thus, within the scope of this study, a multi-objective procedure that performs a simultaneous time-cost trade-off analysis (minimizing the direct cost per available modes) and resource leveling (minimizing the daily maximum resource costs together with cumulative daily maximum resource costs) is described.

In the light of all the information mentioned above, following studies are performed as part of this thesis:

- Generation of benchmark instances that reflect the problem's nature,
- Development of a mixed-integer programming (MIP) model in order to obtain the exact solutions of these instances,
- Execution of the model by means of the optimization solver, GUROBI 6.0.5.

Generated instances are designed considering different control parameters in terms of activity relations and resource requirements. Networks are obtained from ProGen/max problem generator, and modes of different time-cost-resource alternatives are created implementing various resource usage plans. Then, an MIP model is developed so as to minimize the total cost of the project together with maximum daily resource demand per each resource type, and cumulative maximum daily resource demand of all resource types. After that, modelled problems are solved using an optimizer program.

The outline of this thesis is constructed as follows. Chapter 2 presents a literature review in terms of the time-cost trade-off problem and resource leveling problem. Chapter 3 explains generation of benchmark problems. Chapter 4 describes the proposed MIP model and computational experiments conducted within the scope of the thesis. Lastly, Chapter 5 summarizes outcomes of the study, together with recommendations for further research efforts.

CHAPTER 2

LITERATURE REVIEW

Research efforts in TCTP and RLP consist of similar methods: exact procedures, heuristics and meta-heuristics. Exact methods are used in order to obtain the global optimum solution of a problem. Prevalent exact methods studied in the literature include mixed integer programming (MIP), branch-and-bound algorithm (BB), dynamic programming (DP), and Benders decomposition (BD) method. Heuristic and meta-heuristic algorithms, on the other hand, accept near-optimum results for a problem's solution too, in case of being unable to obtain the global optimum. Heuristics are mainly priority-based, problem-specific solution procedures, contrary to generic meta-heuristic algorithms. The most prominent meta-heuristic methods in the literature are genetic algorithm (GA), particle swarm optimization (PSO), and ant colony optimization (ACO).

Studies for both TCTP and RLP in the literature are concentrated on heuristics and meta-heuristics rather than exact methods, mainly. Exact methods usually require significant amount of computational time as the problems' size enlarges, due to NP-hard nature of both problem types. Hence, the implementation of exact methods to medium and large-sized instances might become impractical. Heuristic and meta-heuristic methods, on the other hand, gain attention for these types of problems, due to their capability of providing near optimal results within a reasonable amount of time.

2.1. Literature Review of Time-cost Trade-off Problem

A construction project's principal objective is to be realized within its planned duration and cost. Efforts to obtain a proper project plan with optimum duration and cost, make TCT analysis inevitable. Siemens (1971) explains the primary aim of TCT as to expedite activities by choosing the optimal crashing alternatives to reduce the total project cost. An activity's direct cost arisen from resources (labor, equipment and material costs, namely) and duration are inversely proportional. On the other hand, a project's indirect costs that cannot be assigned to any specific activity are denoted on a daily basis. Hence, a decrease in the project duration leads to a decrease in the indirect costs.

Vanhoucke and Debels (2007) state that the initial TCT researches in literature including Kelley (1961), Siemens (1971), and Goyal (1975) are concentrated on the linear version of the problem. In other words, they approach the problem as if it consisted of activities with a linear time and cost relation. Afterwards, it is decided to investigate the discrete version of the problem in order to represent its practical aspect. DTCTP assumes a pair of time and cost option for each activity.

In the literature, discrete version of this problem (DTCTP) is examined in three categories as: deadline problem, budget problem and Pareto front curve. Deadline problem aims to minimize the total cost with respect to a given project deadline. Budget problem, on the other hand, aims to minimize the project duration within a given budget. The Pareto front problem aims to construct complete and efficient time-cost profile over the set of feasible durations (Vanhoucke and Debels, 2007). All these types are researched in the literature by means of several heuristic, meta-heuristic and exact methods.

2.1.1. Heuristic and Meta-heuristic Methods

As a pioneer study within the context of heuristics, Siemens (1971) presented the Siemens approximation method (SAM) for the cost optimization problem, and implemented it on a problem with eight activities. In this method, it is accepted that the project's each individual activity has convex time-cost curves that are approximated as piecewise linear curves. Then, the project's critical activities are crashed starting from the one with the mildest slope to the activity with steepest slope, as long as the total crash cost is lower than the daily indirect cost. The proposed approach is simple enough to be applied by hand computation. Nevertheless, its implementation becomes difficult for the large networks including numerous critical activities. Goyal (1975) proposed a modified version of SAM and used the same example with 8 activities. In this method, an activity's effective cost slope is redefined, and de-shortening is applied on the suitable activities that were shortened previously. Hence, the amount of over-shortening (unnecessary crash of activities) is reduced, as compared to SAM.

Apart from heuristic methods, numerous meta-heuristics have been presented for the TCTP, too. Among them, GA that is invented by Holland (1975) is the most popular one. Basically, GAs are the search algorithms that mimic the pattern of natural selection phenomenon in genetics. A prominent study that uses GA for TCTP is introduced by Feng et al. (1997), for Pareto front problem. The GA based algorithm obtained the points on the time-cost curve with a 95% accuracy for an 18-activity sample network. Li and Love (1997) introduced improvements in mutation and crossover procedures to basic GA, in order to increase search efficiency. A sample problem consists of 10 activities is solved with the basic GA and the proposed GA, and it is observed that the latter one outperforms. Hegazy (1999) also developed a GA for the cost optimization problem. It is implemented as a macro to Microsoft Project 1995 project scheduling program, using the improvements suggested by of Li and Love (1997) to reduce the computational time. The algorithm's performance is tested on the instances consists of 18, 36, 108, and 360 activities. The algorithm achieved its objective in all the cases. However, its computational time increased with the size of the networks, as expected. Zheng et al. (2005) introduced a GA based approach for Pareto front problem. The model implements the niche formation techniques to diversify the population, and integrates the adaptive weight in order to balance the priority of each objective according to the performance of the previous generation. The algorithm is programmed as a macro in Microsoft Project 2000, and the sample problem with 18 activities given by Feng (1997) is solved. The proposed algorithm increased the results' robustness. Kandil and El-Rayes (2006) proposed a multi-objective GA with parallel processing. The model is tested on 183 experiments consist of 180, 360, and 720-activity networks with a cluster of 50 parallel processors. The suggested algorithm decreases the computational time by distributing the calculations over a cluster of parallel processors. The solution time of 720-activity network, though, obtained as 136.50 hours, which may be regarded as relatively long. Sonmez and Bettemir (2012) developed a hybrid strategy based on GA, simulated annealing (SA), and quantum simulated annealing techniques (QSA) for the cost optimization problem. SA aims to improve hill-climbing ability of GA, and QSA aims to improve local search capability. In order to assess the proposed algorithm's performance, its results are compared with the results of a basic GA, in terms of 10 benchmark instances with known exact results. The problems include 18 to 630 activities. Test outputs show that the hybrid algorithm improved convergence of GA, considerably. Zhang et al. (2015) proposed a GA for repetitive projects in which the DTCTP is combined with soft logic. The concept of soft logic allows for changes in the precedence relation of activities and their modes, in repetitive projects. The model is tested on an example problem consists of 5 activities. The model has potential to reduce project duration and cost. However, it is only applicable to projects that consist a single resource for execution of each activity.

The other popular meta-heuristic methods are PSO, and ACO in TCTP literature. PSO, developed by Eberhart and Kennedy (1995), is a search algorithm that mimics the social behavior of a flock of migrating birds. Likewise, ACO, proposed by Colorni et al. (1991) aims to implement ants' food searching behavior in optimization process.

Early efforts to use PSO in TCTP literature go back to Elbeltagi et al. (2005). This study; in fact, compares the performance of 5 different algorithms (GA, PSO, ACO, memetic algorithm, and shuffled frog leaping algorithm). These algorithms are tested on two benchmark instances. Both problems are solved by 4 algorithms, except ACO. The success rate and solution quality of PSO is better compared to the others; however, it is ranked second in terms of computational time requirement. Yang (2007) proposed a PSO algorithm for time-cost curve problem. The model is tested on an 8-activity problem includes linear, piecewise linear, nonlinear, and discrete time-cost relations, as well as a 28-activity example. The algorithm's implementation is rather simple, and its results are satisfactory in terms of effectiveness, efficiency, and consistency. Zhang and Li (2010) developed a PSO introduces a new methodology for time-cost curve problem. The model combines sparse-degree and roulette-wheel selections to increase the diversity and convergence. Two sample problems with 7 and 18 activities are solved with the model. The proposed algorithm is compared with another GA based on Feng et al. (1997), and the former one provides better results in terms of convergence degree, diversity, and the speed of convergence. Ng and Zhang (2008) introduced an ACO based multi-objective algorithm for time-cost curve problem. In this method, two objectives (time and cost) are integrated into a single objective by applying a modified adaptive weight approach proposed by Zheng et al. (2004), in order to improve the algorithm's performance. The proposed model is compared with the ACO based algorithm introduced by Elbeltagi et al. (2005), solving an 18-activity problem. Both results are similar; however, the proposed algorithm executes with fewer ants and provides its results within a shorter computational time. Nonetheless, its performance is sensitive to selected parameters such as number of ants in each iteration, termination criteria, weighing factors, and so on. Afshar (2009) proposed a multi-colony ACO algorithm for Pareto front problem. It is tested on the 18-activity problem introduced by Feng et al. (1997), and the proposed algorithm performs better as the number of non-dominated solutions increases.

In the literature there exists other meta-heuristic methods other than GA, PSO, and ACO. A summary of leading meta-heuristic and heuristic researches in TCT literature is provided in Table 2.1.

2.1.2. Exact Methods

One of the most prominent exact solution methods in TCTP literature is the mixed integer programming. MIP procedures are introduced for the solution of optimization problems consist variables with a restriction to take integer values. The remaining variables and the objective function may be assumed as linear. Initial MIP studies include the model introduced by Crowston and Thompson (1967). It is an algorithm for decision CPM method, that is, the simultaneous consideration of the scheduling and planning phase of a project. The model is updated regularly throughout the project duration, using progress information of activities. Hence, it enables feasible solutions for changing conditions, too. The method is tested on a sample network with 8 activities. The proposed concept has a potential to provide satisfactory results. Nevertheless, it is only applicable to small networks. Reda and Carr (1989) suggested an MIP model for TCT analysis among related activities, stating that crashing of an activity should be considered together with its effects on the related activities. Liu et al. (1995) proposed a hybrid method as a combination of linear and integer programming. The method aims to provide optimal or near optimal results within reasonable solution times. Basically, linear programming is used to provide lower bounds for the cost curve, and integer programming is used to locate the exact solution. The model is programmed as a macro in Microsoft Excel 5.0, and a sample instance with 7 activities is solved. Moussourakis and Haksever (2004) introduced a flexible MIP model for TCTPs with linear, piecewise linear, or discrete activity time-cost modes. The model is proposed for deadline problems; however, it may also applied on budget problems, by means of small modifications. The model is capable of solving a 7-activity network successfully; however, its performance deteriorates as the network size enlarges. Chassiakos and Sakellaropoulos (2005) presented an MIP model for Pareto front problem. In this model, delay penalty and incentive calculations are included, too. It is tested on a 29-activity problem with different precedence relations including finish to start, and finish to start with lead and lag times. Szmerekovsky and Venkateshan (2012) introduced an MIP model and compared its results with 3 other MIP models. All models are tested on instances including 30 to 90 activities.

The proposed model is capable of solving 90-activity problems. In addition to this, it outperforms to the other models as the networks become denser as a result of its tightness of the LP relaxation, sparseness of the constraint matrix, and the reduced number of binary constraints. Bilir (2015) proposed an MIP model for practical size of deadline and Pareto front problems. The model solved the networks with 1000 activities and 20 modes for deadline problem, and 200 activities with 5 modes for time-cost curve problem, within reasonable computational times.

DP, BB, and BD algorithm are the other prominent exact methods. DP is a procedure that separates a complex problem into a group of simpler problems, solving them, and storing the solutions. BB, firstly introduced by Land and Doig (1960), is an algorithm that splits a problem's search space into smaller paths, provide corresponding lower and upper bounds, and iterate this process until obtaining the global optimum. Likewise, BD (Benders, 1962) is a method that divides large linear programming models into two, in order to provide a lower and an upper bound for the objective function. Early DP studies in TCT literature include Butcher (1967), Robinson (1975), and Panagiotakopoulos (1977). Butcher's (1967) study is only applicable to fully series or fully parallel networks, Robinson's (1975) model is based on network decomposition, and, Panagiotakopoulos's (1977) model is based on problem simplification and enumeration. Demeulemeester et al. (1993) introduced two DP models for DTCTP. One of them aims to find the minimum number of reductions in order to transform a general network to a series-parallel network, whereas the other one minimizes the estimated number of possibilities to be considered during the solution procedure. De et al. (1995) presented a DP model with modular decomposition/reduction technique. A pioneer BB study in DTCTP literature is performed by Demeulemeester et al. (1998). Branching is applied to group an activity's execution mode set into two subsets in order to derive improved convex piecewise linear underestimations of activity time-cost curves. The model is tested on the instances with 10, 20, 30, 40, and 50 activities up to 11 time-cost modes. Problems with 10 and 20 activities are solved rather easily; nevertheless, nearly 45% of the problems with 50 activities and six or more modes remained unsolved.

Vanhoucke (2005) suggested a BB algorithm for DTCTP with time-switch constraints that impose a specified start time on activities and force them to be inactive for specific time periods. Instances with 10, 20, and 30 activities up to 7 modes are solved with the model. One of the studies using BD method in the solution of TCT is proposed by Hazir et al. (2010). The model is a modified version of Benders algorithm, with improved decomposition approach and branch-and-cut procedure. The suggested model outperforms both basic Benders algorithm and IBM's CPLEX 9.1 optimizer. Hazir et al. (2010) introduced another Benders model for the robust optimization of DTCTP, using interval uncertainty for activity costs. The model assumes that an activity's start and finish times are certain once an activity's mode is fixed; hence, all the uncertainty is endured by the activity's cost. Problems with 85 to 136 activities are solved with the suggested algorithm.

A summary of major exact solution studies in TCT literature is provided in Table 2.2.

Table 2.1: Heuristic and Meta-heuristic Methods in Literature to Solve TCTP

Year	Authors	Problem Type	Method	No of Activity	CPU Time	Summary
1971	Siemens	Deadline	Heuristic	8	-	A simple and systematic model capable of solving problems with non-linear cost slopes is introduced for TCTP.
1997	Feng, Liu, and Burns	Pareto front	GA	18	-	A GA is proposed for Pareto front problem. The model is tested on an 18-activity sample network by evaluating 20000 possible schedules, and obtained the points on the time-cost curve with a 95% accuracy.
1997	Li and Love	Deadline	GA	10	-	A modified GA procedure is introduced stating that the current practice in mutation and crossover increases the computational time. Hence, the mutation and crossover procedures of Feng et al. (1997) model is improved.
1999	Hegazy	Deadline	GA	18	390 sec.	The improvements introduced by Li and Love (1997) are used in the proposed model. For example, a crossover between identical chromosomes is prevented. In addition to this, CPM and resource leveling features of Microsoft Project 1995 is used in order to schedule the problem considering available resource limits.
2005	Elbeltagi, Hegazy, and Grierson	Deadline	GA MA ACO PSO SFL	18 18 18 18 18	16 sec. 21 sec. 15 sec. 10 sec. 15 sec.	The performances of 5 different meta-heuristics are tested on two benchmark instances. The weakest one is ACO. The success rate and solution quality of PSO is better compared to the others; however, it is ranked second in terms of computational time requirement.
2006	Kandil and El-Rayes	Pareto front	GA	720	8220 min.	A multi-objective GA is proposed with parallel processing, to reduce the computational time.

Table 2.1: Heuristic and Meta-heuristic Methods in Literature to Solve TCTP (continued)

Year	Authors	Problem Type	Method	No of Activity	CPU Time	Summary
2007	Vanhoucke and Debels	DTCTP DTCTP (ts) DTCTP (wc) DTCTP (npv)	Meta-heuristic	50 50 50 50	1.605 sec. 3.135 sec. 2.548 sec. 9.136 sec.	A model is developed including tabu-search and truncated dynamic programming, for three extensions of the DTCTP, namely the DTCTP with time-switch constraint (ts), the DTCTP with work continuity constraints (wc), and the DTCTP with net present value maximization (npv).
2007	Yang	Deadline	PSO	8	48 sec.	A PSO algorithm is introduced. It is simple, easy to implement, and capable of evaluating linear, nonlinear, continuous, discrete, convex, concave, and piecewise discontinuous time-cost functions.
2010	Zhang and Li	Pareto front	PSO	18	205.08 sec.	A multi-objective PSO model is proposed, based on a combined scheme for determining the global best.
2011	Abdel-Raheem and Khalafallah	Deadline	Meta-heuristic	18	-	An evolutionary algorithm is introduced that simulates the behavior of electrons moving through electric circuit branches.
2012	Ashuri and Tavakolan	Pareto front	GA/PSO Hybrid	14	19 min.	A GA/PSO hybrid approach, using fuzzy set theory to include uncertainty about the input data, is proposed for a simultaneous optimization of total project cost, total project duration, and total deviations in resource utilization.
2012	Zhang and Ng	Pareto front	ACO	18	-	A modified adaptive weight approach is implemented in the proposed ACO algorithm, in order to combine the time and cost into a single objective function with relevant weights.

Table 2.2: Exact Methods in Literature to Solve TCTP

Year	Authors	Problem Type	Method	No of Activity	CPU Time	Summary
1967	Crowston and Thompson	Deadline	MIP	8	5 sec.	The Decision CPM problem is modelled mathematically, enabling to include path reductions.
1975	Robinson	Deadline	DP	-	-	A DP model is proposed based on network decomposition. Although complex networks cannot be fully decomposed, the model enables a reduction in the dimensions of these functions.
1995	De, Dunne, Ghosh, and Wells	Pareto front	DP	-	-	A new dynamic programming model based on network decomposition/reduction is introduced.
1995	Liu, Burns, and Feng	Deadline	LP/IP Hybrid	7	1800 sec.	A combination of linear and integer programming is presented. LP provides lower bounds and IP finds the exact solution.
1996	Demeulemeester, Herroelen, and Elmaghraby	Pareto front	DP	45	530.40 sec.	Two algorithms are described for solving DTCTP in deterministic activity on arc networks of the CPM type.
1997	De, Dunne, Ghosh, and Wells	Pareto front	DP	16	-	A correction is proposed for the flawed algorithm of Hindelang and Muth (1979). It is proved that certain structures are more difficult to solve.
1998	Demeulemeester, De Reyck, Foubert, Herroelen, and Vanhoucke	Pareto front	BB	50	175.91 sec. (On average)	A BB model is introduced that computes lower bounds by making convex piecewise linear underestimations of activity time-cost curves, and used them as an input for the adapted version of the Fulkerson labelling algorithm.
2004	Moussourakis and Haksever	Deadline	MIP	7	-	An MIP model capable of solving different TCTP types is introduced. The basic model is applicable to deadline problems; however, it may be modified to solve budget problems, too.

Table 2.2: Exact Methods in Literature to Solve TCTP (continued)

Year	Authors	Problem Type	Method	No of Activity	CPU Time	Summary
2005	Chassiakos and Sakellaropoulos	Pareto front	MIP	29	<1 sec.	An exact and an approximate methods are proposed, and their results are compared. Both methods solved a 29-activity problem (including incentive and penalty) less than a second.
2005	Vanhoucke	Pareto front	BB	30	11.506 sec.	A new BB algorithm is presented for the DTCTP with time-switch constraints.
2010	Hazır, Haourai, and Erel	Budget	BD	136	-	A modified BD algorithm is proposed to solve the DTCTP instances of realistic sizes. Including several algorithmic features to model prevents slow convergence.
2011	Hazır, Erel, and Günalay	Cost Uncertainty	BD	136	19139.61 sec.	Three models are suggested to formulate robust DTCTP. The models assume interval uncertainty for activity costs. Nevertheless, it assumes the activities with the same cost intervals as if they were equally uncertain, and all activities are likely to have cost values at the upper bounds.
2012	Sönmez and Bettemir	Deadline	MIP	63	-	An MIP model is developed in order to test the performance of proposed meta-heuristics.
2012	Szmerekovsky and Venkateshan	Irregular Cost	MIP	90	206 sec.	An MIP model is proposed for problems with irregular TCTs. Problems consist up to 90 activities are solved, successfully.
2015	Bilir	Deadline	MIP	1000	273.48 sec. (On average)	An MIP model is developed to solve medium and large size DTCTPs.

2.2. Literature Review of Resource Leveling Problem

Construction projects, in general, are composed of a large number of activities. Liberatore et al. (2001) indicates that a real-life construction project consists more than 300 activities. Hence, each project requires simultaneous utilization of numerous resources, in practice. Concurrent utilization of a large number of resources within short time periods may result in a cost overrun, as compared to a more uniform resource utilization schedule over longer time periods. Hence, resource leveling process aims to provide the optimum resource utilization for a project, with a smooth and leveled resource histogram that minimizes the peaks and deviations in resource demands.

The most prominent approach in project scheduling is the CPM. Briefly, it works on the basis of putting a network's activities in order according to their precedence relation, and it calculates the early start, early finish, late start, late finish, and total float of each activity. An activity with a positive total float implies that the activity might be rescheduled to an earlier or a later date, within its float value. Resource leveling is performed using this feature of the CPM. Rearranging the non-critical activities' start and finish times enables obtaining a project's optimum resource utilization schedule, without changing the total project duration. In addition to this, the extent of resource leveling may be enlarged to include critical activities, too. There are usually additional indirect costs associated with resources such as: storage and transportation costs. These costs are directly associated with the peak utilization values of resources. Hence, it may become mandatory to minimize these values, in order to obtain the optimum total project cost.

In the literature, different resource leveling metrics are studied. Among them, minimum moment method targets obtaining a rectangular resource histogram with a minimum moment value (Martinez and Ioannou, 1993). Sum of squares method aims to minimize the sum of daily resource usages' squares per each resource type. Overload metric minimizes the surplus amount of daily resource usages from a specific level. Likewise, absolute deviation approach minimizes the absolute deviations of daily resource utilization values (Yeniocak, 2013). Release and rehire method calculates the total amount of resources that to be released during low demand periods and rehired during high demand periods. Finally, resource idle days approach minimizes the total number of idle and nonproductive resource days as a result of undesirable resource fluctuations (El-Rayes and Jun, 2009).

2.2.1. Heuristic and Meta-heuristic Methods

A pioneer heuristic in RLP literature is proposed by Burgess and Killebrew (1962). It is a priority-based method that enables the implementation of different priority rules in activity sorting such as: ascending or descending activity numbers, ascending or descending total floats, and so on. Harris (1990) introduced a heuristic model, called as PACK, with the minimum moment approach. It follows a three-stage priority procedure for activity sorting: the descending resource demand, ascending total float, and descending activity number, respectively. Hiyassat (2000) also proposed a heuristic method for RLP, based on the minimum moment approach. In the model, a modified activity sorting procedure is used instead of the traditional approach (Harris, 1978), so that the number of iterations in each step is decreased, and the model's efficiency is increased. The proposed model gives as accurate results as the traditional approach within fewer calculations. Ballestin et al. (2007) suggested a heuristic model based on iterated greedy method (a meta-heuristic for stochastic local search), with the objective of minimizing daily resource utilization deviations from a desired level. The model is able to solve up to 1000-activity problems, and its performance is satisfying in terms of accuracy and solution time.

In addition to these, several meta-heuristic methods are introduced for the solution of RLP. Chan et al. (1996) proposed a GA with the objective of minimizing resource utilization deviations from a targeted level. The model is applicable both RLP and RCPSP, unlike the other models available at that time. Hegazy (1999) suggested a GA combined with an improved heuristic method. The model has a multi-objective approach considering both RLP and RCPSP simultaneously. In the model, heuristics are applied to the selected tasks with random priorities, and corresponding schedules are evaluated. Then, GA is used to search for better priorities in order to obtain shorter schedules with smoother resource histograms. Leu et al. (2000) introduced a GA model together with a decision support system (DSS) for daily deviation problem. Its performance is compared with sample heuristics and exact models. Neither the proposed model nor the heuristics in question are not capable of providing the global optimal results; however, near-optimal results of GA model may become valuable in performing a what-if analysis by means of its DSS. In addition to this, GA has the advantage to avoid combinatorial explosion that is a potential drawback in exact methods. Zheng et al. (2003) proposed a multi-objective GA together with adaptive weight concept, for multiple RLP. In the model, the selection criterion is decided as a combination of improvements in all resources utilization values; hence, the dominance of a single resource is prevented. El-Rayes and Jun (2009) suggested a GA for RLP by describing two new resource leveling metrics: release and rehire and resource idle days. The model is tested on a problem with 20 activities, and the results are promising. Ghoddousi et al. (2013) introduced a multi-objective GA that considers multi-mode RCPSP (MRCPSP), DTCTP, and RLP simultaneously. The model is tested on two sample networks with 6 and 33 activities, respectively, and is capable of enabling more practical solutions in terms of resource allocation and leveling, compared to another multi-mode resource constraint DTCTP (MRCDTCTP) model without resource leveling objective.

PSO and ACO algorithms are also proposed for solving the RLP, Pang et al. (2008) introduced an improved PSO model for PSO. In this approach, the basic PSO model is modified using a constriction factor, in order to prevent early convergence in complicated problems. As a result, the local searching capability of the algorithm is increased. The model is not capable of solving multi-resource problems. Xiong and Kuang (2006) proposed a hybrid approach as a combination of serial schedule generation scheme (SSGS) and ACO for RLP. The model uses SSGS to generate a feasible schedule and ACO to search the global optimal solution.

In order to test the model, the sample problem proposed by Son and Skibniewski (1999) is solved, and the model is capable of providing the global optimal solution by searching only a small fraction of the total search space.

A summary of prominent heuristic and meta-heuristic researches in RLP literature is provided in Table 2.3.

2.2.2. Exact Methods

Wagner et al. (1964) introduced five MIP models with different resource leveling objectives. These are: minimizing the sum of absolute changes in resource utilization, minimizing the sum of the increases in total weekly resource usage, minimizing the sum of absolute deviations from the average weekly resource utilization, minimizing the weekly peak resource usage, and minimizing the maximum change in weekly resource utilization. All the models are tested on two small scale problem set and no single model is obtained as a best suited approach for RLP. However, the fourth model provided the least satisfactory results in every case. Easa (1989) suggested an MIP model for RLP of single resource, aiming to minimize the total absolute deviation between the actual and the desired resource utilization amounts. Nonetheless, the algorithm's application is limited to small size problems. Karshenas and Haber (1990) suggested an MIP model to minimize the total project cost, and the resource cost is considered as a decision variable within this context. Mattila and Abraham (1998) proposed an MIP model to achieve a daily desired resource usage in projects planned by linear scheduling method. The proposed resource leveling procedure is independent of network analysis and the CPM. Additionally, it enables a reduction in the number of integer variables used in formulations, in order to increase its efficiency. For example, a problem with 138 integers is described in terms of 24 integers with the proposed model, by reviewing some constraints. However, the implementation becomes harder as the number of integer variables increases. Son and Mattila (2004) suggested an MIP model for RLP, allowing activity splitting. The model is tested in three different cases: activity splitting is prohibited, activity splitting is allowed only for the selected activities, activity splitting is allowed for all the activities. The proposed model's results are compared with the results of Primavera P3 and Primavera SureTrak. They are tested on a problem with 11 activities, and the proposed model outperformed the others. Rieck et al. (2012) introduced an MIP model for daily deviation and overload problems. The procedure includes pre-processing techniques to reduce the problem's search space. Problems including 50 activities with tight project deadlines are solved optimally for the first time, using CPLEX 12.1.

Petrovic (1968) proposed a DP model for multi-resource daily deviation problem, evaluating it as a multistage decision problem. Bandelloni et al. (1994) suggested a non-serial DP approach to minimize the deviation between the actual and the desired resource utilization amounts. Its performance is compared with Burgess and Killebrew's (1962) heuristic method. The proposed algorithm is capable of providing results for small size problems, by requiring modest computing facilities. Mason and Moodie (1971) introduced a BB algorithm to minimize the combined cost of deviations in resource utilization and delay of project completion. The model included cost bounding procedures by means of dominance relationships, in order to reduce to computational time. It is tested on a 10-activity problem, and the parameters affecting the model's performance are evaluated. Accordingly, the number of activities in a network, their durations and resource requirements affect the model's performance more, compared to the network's structure. On the contrary, changes in the ratio of the project delay cost to the resource deviation cost have a relatively lower impact on the results. Neumann and Zimmermann (2000) proposed a BB algorithm for RLP and net present value problem (NPVP). The model is applicable to three different resource leveling metrics: minimization of fluctuation costs, minimization of deviation from a targeted level, and minimization of daily deviations. It enables solving RLPs with up to 20 activities and 5 resources, for the first time in the literature. Gather et al. (2011) suggested a BB model combined with a tree-based enumeration scheme for RLP subject to general temporal constraints. Mutlu (2010) introduced a BB model for RLP with single and multi-resource networks. The algorithm uses several lower bound calculation techniques in order to reduce the computational time.

It is applicable to four different resource leveling metrics: sum of squares, daily deviation, resource idle days, and maximum resource demand together with resource idle days. Results show that the model is capable of solving 20-activity problems. Yeniocak (2013) proposed a BB algorithm for RLP that uses an adaptive BB heuristic for upper bound calculations, and a dual calculation procedure for lower bound calculations. It is applicable to four different resource leveling metrics: sum of squares, daily deviation, overload, and resource idle days together with daily peaks. The suggested BB model shows provides the best computational times for problems up to 20 activities in terms of all metrics. Additionally, problems up to 30 activities with the objective of minimize resource idle days are solved for the first time in the literature.

A summary of primary exact solution studies in RLP literature is provided in Table 2.4.

In conclusion, all the above-mentioned studies focus on RLP and TCTP, separately. Hence, current practice in project management literature involves the successive implementation of them, for the solution of discrete time-cost trade-off and resource leveling problems. That is, a project's resource utilization values are leveled within its schedule obtained from the project's TCT analysis. Therefore, resource leveling process is performed within a smaller search space, in which better solutions might be ignored. This study, on the other hand, introduces an integrated approach for both of these problems. As a result of this, the model investigates a larger search space in order to obtain better results as compared to the existing approach.

Table 2.3: Heuristic and Meta-heuristic Methods in Literature to Solve RLP

Year	Authors	Problem Type	Method	No of Activity	CPU Time	Summary
1962	Burgess and Killebrew	RLP	Heuristic	11	2.41 min.	A simple, priority based heuristic method is introduced. It is regarded as a leading heuristic study in RLP literature.
1990	Harris	RLP	Heuristic	11	-	PACK, a priority based minimum resource moment method is introduced for RLP.
1996	Chan, Chua, and Kannan	RLP and RCPSP	GA	51	5.84 min.	A model is that is applicable to both resource leveling and limited resource allocation problems, is introduced.
1999	Hegazy	RLP and RCPSP	GA	20	120 min.	An improved heuristic is suggested for resource allocation, using random activity priorities. A modification is proposed for resource leveling using a double-moment method. A GA-based procedure for simultaneous resource allocation and resource leveling is introduced.
1999	Son and Skibniewski	RLP	Heuristic	13	-	A multi-heuristic procedure and a hybrid model (a combination of multi-heuristic with simulated annealing) is proposed.
2000	Hiyassat	RLP	Heuristic	12	-	A modified activity selection procedure is proposed to minimum moment method suggested by Harris (1978)
2000	Leu, Yang, and Huang	RLP	GA	13	-	A model is developed to minimize resource utilization variation, and a decision support system is developed to perform what-if analyses.
2003	Zheng, Ng, and Kumaraswamy	RLP	GA	6	-	A GA for multi-resource leveling problem is introduced with adaptive resource weight concept.

Table 2.3: Heuristic and Meta-heuristic Methods in Literature to Solve RLP (continued)

Year	**Authors**	**Problem Type**	**Method**	**No of Activity**	**CPU Time**	**Summary**
2004	Senouci and Eldin	RLP and RCPSP	GA	11	-	An augmented Lagrangian GA model for simultaneous resource leveling and RCPSP is suggested.
2006	Xiong and Kuang	RLP	ACO	13	-	A hybrid model using serial schedule generation scheme (to generate a feasible schedule) together with ACO (to search the optimal schedule) is proposed.
2007	Ballestin, Schwindt, and Zimmermann	RLP	Heuristic	1000	459.70 sec.	A procedure is introduced for population-based resource leveling of the iterated greedy type.
2008	Pang, Shi, and You	RLP	PSO	9	-	An improved PSO, including a constriction factor to enhance local search ability of the model, is proposed.
2008	Roca, Pugnaghi and Libert	RLP and RCPSP	GA	13	2.64 sec.	A modified GA is developed with multiple objectives as: minimization of the project duration and leveling of each resource within given limits.
2009	Guo, Li, and Ye	RLP	PSO	-	-	A model to for multiple resource leveling in multiple project scheduling is developed.
2010	Doulabi, Seifi, and Shariat	RLP	GA	5000	14502 sec.	A hybrid GA is proposed for multiple resource leveling, by allowing activity splitting.
2010	Geng, Weng, and Li	RLP	ACO	9	-	A directional ACO is introduced for non-linear resource leveling problems.
2011	Jun and El-Rayes	RLP and RCPSP	GA	20	-	A multi-objective model is developed to minimize both daily resource fluctuations and project duration.

Table 2.4: Exact Methods in Literature to Solve RLP

Year	Authors	Problem Type	Method	No of Activity	CPU Time	Summary
1968	Petrovic	RLP	DP	-	-	Project execution is considered as a discrete multistage decision process, and optimization of this stage is evaluated in terms of dynamic programming method. In order to reduce the solution time, 3 approaches are proposed.
1971	Mason and Moodie	RLP and RCPSP	BB	10	-	An algorithm is developed to minimize cost of fluctuations in resource demand and delay of project completion. Initial feasible schedules are generated using heuristics.
1984	Patterson	RCPSP	Bounded Enumeration BB Implicit Enumeration	7-50 7-50 7-50	14.02 sec. 0.82 sec. 14.98 sec. (On average)	Three enumerative-based procedure are examined. Specific classes of problems suitable for each procedure are identified. 110 instances tested in this study are used as benchmark problems, in the following researches.
1989	Easa	RLP	MIP	7	-	An MIP model is introduced to minimize daily resource deviations for single resource type and continuous activities.
1990	Karshenas and Haber	RLP	MIP	5	-	An MIP model is developed to minimize total project cost, including resource cost and cost of time.
1994	Bandelloni, Tucci, and Rinaldi	RLP	DP	21	-	A non-serial DP model is proposed for minimization of the deviation between resource requirements and stated desirable resource usage levels.
1998	Mattila and Abraham	RLP	MIP	9	-	An MIP method is presented for resource leveling in linear project scheduling.

Table 2.4: Exact Methods in Literature to Solve RLP (continued)

Year	Authors	Problem Type	Method	No of Activity	CPU Time	Summary
2000	Neumann and Zimmermann	RLP and NPVP	BB	20	94.4 sec. to 97.8 sec.	Different heuristic and exact procedures are proposed for resource leveling and net present value problems. Branch-and-bound and truncated branch-and-bound methods are introduced for resource leveling with and without resource constraints.
2004	Son and Mattila	RLP	MIP	11	-	An MIP method is proposed for RLP in which activity splitting (remaining an activity as unfinished and resuming it later) is allowed.
2010	Mutlu	RLP	BB	20	2 sec. to 335 sec.	A BB algorithm is developed for RLPs consist of single and multi-resources. Networks consist of up to 20 activities are solved optimally. Among them, resource idle day (RID) metric is implemented with an exact method for the first time in the literature.
2011	Gather, Zimmermann, and Bartels	RLP	BB	20	-	A BB procedure with is introduced with a new tree-based enumeration scheme. It is tested on the problem set of Kolisch et al. (1999), and even 20-activity problems are solved to optimal within a time limit of 36000 sec.
2012	Rieck, Zimmermann, and Gather	RLP	MIP	50	288.89 sec. to 926.27 sec.	A MIP model is proposed for daily deviation and overload metrics. The model is improved with problem-specific preprocessing techniques in order to increase its efficiency. Instances with 50 activities and small deadlines are solved to optimal for the first time in the literature.
2013	Yeniocak	RLP	BB	30	1500.02 sec. (On average)	A flexible BB procedure that supports the implementation of several acceleration techniques is proposed for RLP. Networks including 30 activities are solved optimally in terms of maximum resource demand and resource idle days metric.

CHAPTER 3

GENERATION OF BENCHMARK PROBLEMS

As discrete time-cost trade-off and resource leveling problems studied separately in the literature, there are no instance sets for the integrated discrete time-cost trade-off and resource leveling problem (DTCTRLP). Hence, problem instance are generated within the context of this thesis for the DTCTRLP. This chapter clarifies the employed procedure and defines used parameters in the problem set generation stage.

3.1. Problem Network Generation

Firstly, problem networks are produced using ProGen/max (Schwindt, 1995). In fact, ProGen/max is designed to generate resource-constrained project scheduling problems, it does not produce any activity time-cost-resource modes. Hence, ProGen/max is used in order to generate networks with predefined parameters, only. These parameters include both generic items such as: network complexity (the intensity of predecessor-successor relation), resource demand, activity duration, fraction of the resources used per activity, and problem-specific items like Thesen restrictiveness coefficient, resource number, resource factor, as well as the number of activities . The value of each parameter used in the network generation is given in Table 3.1 and Table 3.2.

Table 3.1: Generic Parameter Input Values for Network Generation in ProGen/max

Parameter	Value
Minimal duration of activity	8
Maximal duration of activity	15
Slack of project duration	0
Minimal cost coefficient	1
Maximal cost coefficient	1
Threshold factor	1
Minimal number of initial activities	1
Maximal number of initial activities	6
Minimal number of terminal activities	1
Maximal number of terminal activities	6
Maximal number of predecessor activities	6
Maximal number of successor activities	6
Degree of redundancy	0.1
Minimal demand of resources per activity	2
Maximal demand of resources per activity	8
Minimal resource strength	1
Maximal resource strength	1

Table 3.2: Problem-specific Parameter Input Values for Network Generation in ProGen/max

Thesen Restrictiveness Coefficient	**Activity Number**			**Resource Number**	**Resource Factor**		
0.25	10	15	20		0.20	0.5	0.8
0.50	10	15	20	1 RES	0.20	0.5	0.8
0.75	10	15	20		0.20	0.5	0.8
Thesen Restrictiveness Coefficient	**Activity Number**			**Resource Number**	**Resource Factor**		
0.25	10	15	20		0.20	0.5	0.8
0.50	10	15	20	2 RES	0.20	0.5	0.8
0.75	10	15	20		0.20	0.5	0.8
Thesen Restrictiveness Coefficient	**Activity Number**			**Resource Number**	**Resource Factor**		
0.25	10	15	20		0.20	0.5	0.8
0.50	10	15	20	4 RES	0.20	0.5	0.8
0.75	10	15	20		0.20	0.5	0.8

Firstly, it is assumed that successive activities have finish to start relationship without any lag. In other words, an activity can start as soon as its predecessor activity finishes. The initial activities succeeding the dummy start is set as six, the final activities preceding the dummy finish is also set as six. Likewise, remaining activities might have at most 6 predecessor and successor activities, respectively. The durations of activities are set between the range of 8 and 15 days. The resource demand for each activity is defined in the range of 2 and 8 per each resource type (except for the dummy start and the dummy finish activities, they do not require any resource). Resource strength is taken as 1 due to the fact that generated instances do not have any resource constraint. In other words, each resource type is considered as available when required. Finally, values of the slack of project duration, minimal and maximal cost coefficients, threshold factor and the degree of redundancy are determined according to the study by Rieck et al. (2012).

In addition to these, problem specific parameters are defined in order to take the level of network hardness, number of activities, number of resources, and average fraction of the resources used per activity into consideration. “Thesen restrictiveness coefficient” defines the complexity of a network generated in ProGen/max. As it increases, the number of predecessor and successor of an activity increases, within identified limits. In this study, networks with a Thesen restrictiveness coefficient of 0.25, 0.50, and 0.75 are generated. “Resource number” represents the number of different resource types used in problems. One third of the instances in total have only 1 resource type, and the remaining ones have up to 2 and 4 resource types, respectively. “Resource factor” denotes the average value of resource demand for each type, proportionally. Within the context of this study, 0.2, 0.5, and 0.8 are selected in order to represent sparse, medium and dense resource fraction intervals, respectively. Finally, “Activity number” states the number of activities composing the network. Generated instances include 10, 15 and 20 activities, equally. Actually, each problem involves two additional activities as the dummy start and the dummy finish; however, they do not have any duration and resource demand. They are generated as a result of ProGen/max format.

3.2. Mode Generation

ProGen/max does not generate activity time-cost-resource modes. Hence, modes are created, based on the data provided in ProGen/max output. Results of the network generated by the program are considered as “Mode 1”. Using these values, “Mode 2”, “Mode 3” and “Mode 4” alternatives are created, correspondingly, by taking into consideration the following criteria:

- In order to calculate the direct cost of an activity, firstly, an hourly unit resource cost is assigned per each resource type. Then, daily working hours and daily overtime hours are defined. Using them, a daily cost is found. Finally, the direct cost of each mode is estimated multiplying daily cost by resource quantities, per each resource type.
- It is assumed that “Mode 2”, “Mode 3” and “Mode 4” are all requiring daily overtime hours, on top of daily working hours defined in “Mode 1”.
- Resource quantities used in performing activities are increased, for “Mode 3” and “Mode 4”. For example, the cumulative resource quantity in “Mode 4” is higher as compared to “Mode 3”. As a result of this, activity durations in “Mode 3” become shorter than activity durations in “Mode 4”, as it should be. The same rule applies to the relation between “Mode 3” and “Mode 2”, too. However, resource quantities in “Mode 1” and “Mode 2” are the same. In “Mode 2” overtime is used to decrease the durations.
- Activity durations in “Mode 2”, “Mode 3” and “Mode 4” are calculated with reference to the total work executed in “Mode 1”. Firstly, resource quantities are multiplied by daily working hours per each resource type, in order to find total man-hour required to complete an activity. Then, the estimated value is divided by the cumulative resource quantity in the corresponding mode, to obtain the activity duration in that mode.
- The direct cost of “Mode 2”, “Mode 3” and “Mode 4” are estimated multiplying the daily cost by duration and cumulative resource quantity in the corresponding mode. Resource quantities for “Mode 3” and “Mode 4” increase

as the mode number increases, in order to crash the activity duration. However, an increase in the cumulative resource quantity may not result in the same amount of decrease in the activity duration, all the time, due to practical conditions. In other words, a decline in the productivity of resources may occur. The effect of this phenomenon is reflected to the costs, in this study. The cost of “Mode 2”, “Mode 3” and “Mode 4” is multiplied by different coefficients, as shown in Table 3.3.

After the formation of time-cost-resource mode values corresponding to each activity in a network, a problem’s indirect costs are determined to be minimized in the model’s objective function. These costs are: “unit cost for the peak value of daily resource usage per each resource type”, “unit cost for the peak value of cumulative daily resource usage including all resource types”, “daily indirect cost”, and “daily delay penalty”. Among them, the first cost is determined as 250 USD per resource for each resource type, whereas the second one is decided as 500 USD per resource for each resource within the cumulative amount. Likewise, daily direct cost is determined as 500 USD per each day throughout the project duration. Daily indirect penalty, with a cost of 1,000 USD per day, is applied to the project if its duration exceeds the deadline, that is, the rounded average value of the maximum and the minimum probable project durations. The total cost of maximum daily resource usage per each resource type in a problem is obtained, by multiplying “unit cost for the peak value of daily resource usage per each resource type” and the peak value of daily resource usage per each resource type. Likewise, the total cost of maximum cumulative daily resource usage for all resource types, is calculated by multiplying “unit cost for the peak value of cumulative daily resource usage including all resource types” and the peak value of daily cumulative resource utilization value.

Basic principles explained above and selected values of each coefficient, are summarized in Table 3.3 and Table 3.4, respectively. While deciding the corresponding numerical values of all the cost items included in these tables, it is aimed to represent real-life conditions in this study. Thus, costs of an actual construction project performed by a Turkish company in Jordan are obtained, and their up to date values are used as a basis for the cost values used in this research.

The distribution of instances with respect to different control parameters is represented in Table 3.5. Accordingly, a total of 1215 problems are generated as part of the study. Format of a problem file is explained in Figure 3.1.

```
12   2

1    0    0    0    5    3    4    2    6    5
2    9    5    0    1    11
3    15   5    0    4    8    7    10   9
4    12   0    3    4    8    11   10   9
5    10   8    0    1    8
6    14   0    5    2    7    10
7    12   0    7    1    12
8    8    5    0    1    12
9    14   0    6    1    12
10   8    0    2    1    12
11   13   7    0    1    12
12   0    0    0    0
1 0
2 3 9 5 0 1800 7 5 0 2407 5 7 0 3282
3 3 15 5 0 3000 11 5 0 3782 8 7 0 5250
4 3 12 0 3 1440 9 0 3 1857 7 0 4 2625
5 3 10 8 0 3200 8 8 0 4400 6 11 0 6188
6 3 14 0 5 2800 11 0 5 3782 8 0 7 5250
7 3 12 0 7 3360 9 0 7 4332 7 0 10 6563
8 3 8 5 0 1600 6 5 0 2063 5 7 0 3282
9 3 14 0 6 3360 11 0 6 4538 8 0 8 6000
10 3 8 0 2 640 6 0 2 825 4 0 3 1125
11 3 13 7 0 3640 10 7 0 4813 7 10 0 6563
12 0
```

Figure 3.1: Format of a Problem File

The sample problem file consists of 10 activities, each of which has 3 different time-cost-resource modes, as well as the dummy start and dummy finish activities. Furthermore, all these 10 activities require up to 2 resource types. In the figure, this is stated in the first row as “12” (the number of activities including the dummy start and dummy finish) and “2” (the number of resource types). The following rows, located between “1” and “12” can be classified as the first group. Each character through a row in this group denotes the activity ID, resource usage quantity per each type, total number of successors and the activity ID of each successor, respectively.

On the other hand, the rows in the last group include different time-cost-resource usage alternatives. The first character of each row shows the activity ID and the second character denotes the number of modes, available for corresponding activity. Remaining characters represent duration of the activity, resource usage quantity per each type and cost of the related mode, in a repetitive manner.

Table 3.3: Coefficients Used in Mode Generation

	Mode 1	**Mode 2**	**Mode 3**	**Mode 4**
Daily working hours	*8 hours*	**8 hours**	**8 hours**	**8 hours**
Daily overtime hours	-	**3 hours**	**3 hours**	**3 hours**
Unit resource cost during working hours per each resource type	*5 USD*	**5 USD**	**5 USD**	**5 USD**
Unit resource cost during overtime hours per each resource type	-	**7.5 USD**	**7.5 USD**	**7.5 USD**
Resource quantity per each resource type	*ProGen/max output value*	**ProGen/max output value**	**1.3 x ProGen/max output value**	**2 x ProGen/max output value**
Duration of the mode	*ProGen/max output value*	$\frac{\text{Total work in Mode 1}}{\text{Total resource in Mode 2}}$	$\frac{\text{Total work in Mode 1}}{\text{Total resource in Mode 3}}$	$\frac{\text{Total work in Mode 1}}{\text{Total resource in Mode 4}}$
Cost of the mode	*Total resource cost*	**1.1 x Total resource cost**	**1.5 x Total resource cost**	**2.25 x Total resource cost**

Table 3.4: Selected Cost Values Corresponding to Each Problem

Activity Number	Resource Number			Mode Number			Thesen Restrictiveness Coefficient								
							0.25 T			0.50 T			0.75 T		
							Resource Factor			Resource Factor			Resource Factor		
							0.20 RF	0.50 RF	0.80 RF	0.20 RF	0.50 RF	0.80 RF	0.20 RF	0.50 RF	0.80 RF
							IC [1] (USD) / DP [2] (USD) UC [3] (USD) / TUC [4] (USD)			IC (USD) / DP (USD) UC (USD) / TUC (USD)			IC (USD) / DP (USD) UC (USD) / TUC (USD)		
10	1	2	4	2	3	4	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500
							500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500
							500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500
15	1	2	4	2	3	4	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500
							500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500
							500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500
20	1	2	4	2	3	4	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500
							500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500
							500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500	500/1000 250/500

1 (IC): Daily indirect cost, **2 (DP):** Daily delay penalty, **3 (UC):** Unit cost for the peak value of daily resource usage per each resource type
4 (TUC): Unit cost for the peak value of cumulative daily resource usage including all resource types

Table 3.5: The Number of Instances with Respect to Problem-specific Control Parameters

							Thesen Restrictiveness Coefficient								
							0.25 T			0.50 T			0.75 T		
Activity Number	**Resource Number**			**Mode Number**			**Resource Factor**			**Resource Factor**			**Resource Factor**		
							0.20 RF	0.50 RF	0.80 RF	0.20 RF	0.50 RF	0.80 RF	0.20 RF	0.50 RF	0.80 RF
							Number of Problems	**Number of Problems**	**Number of Problems**	**Number of Problems**	**Number of Problems**	**Number of Problems**	**Number of Problems**	**Number of Problems**	**Number of Problems**
10	1	2	4	2	3	4	5	5	5	5	5	5	5	5	5
							5	5	5	5	5	5	5	5	5
							5	5	5	5	5	5	5	5	5
15	1	2	4	2	3	4	5	5	5	5	5	5	5	5	5
							5	5	5	5	5	5	5	5	5
							5	5	5	5	5	5	5	5	5
20	1	2	4	2	3	4	5	5	5	5	5	5	5	5	5
							5	5	5	5	5	5	5	5	5
							5	5	5	5	5	5	5	5	5

CHAPTER 4

MIXED INTEGER PROGRAMMING MODEL AND COMPUTATIONAL EXPERIMENTS

In order to find the exact solutions of the generated benchmark problems, a Mixed Integer Programming model is developed. In accordance with this model, instances are solved using GUROBI optimizer. In this chapter, firstly, the suggested MIP model is presented. Then, the process of solving the problems with the GUROBI optimizer is described. Finally, results of the computational experiments are presented.

4.1. Mixed Integer Programming Model

The proposed model aims to achieve the following objectives:

- Minimizing a project's total cost including direct, indirect and delay penalty (in case the deadline is exceeded)
- Minimizing maximum daily manpower resource demand, per each resource type,
- Minimizing cumulative maximum daily manpower resource demand, for all resource types.

Sets, parameters and decision variables used in the model are explained below.

4.1.1. Sets

$I =$ Project activities, $i = 1,...,N$
(where; 1 is the dummy start activity, N is the dummy finish activity)
$P_i =$ Activities which directly precede activity i
$J =$ Activity modes, $j = 1,...,M$
$K =$ Number of resource type, $k = 1,...,H$
$T =$ Days in the project, $t = 0,...,D_{max}$

4.1.2. Parameters

$IC =$ Daily indirect cost
$UC =$ Unit cost for the peak value of daily manpower resource usage per each resource type
$TUC =$ Unit cost for the peak value of cumulative daily manpower resource usage, including all manpower resource types
$DP =$ Daily delay cost
$D_{max} =$ Maximum possible project duration in terms of the longest activity modes
$D_{\min} =$ Minimum possible project duration in terms of the shortest activity modes
$D_{avg} =$ Rounded average project duration ($D_{avg} = \dfrac{D_{max} + D_{\min}}{2}$)
$c_{i,j} =$ Cost of activity i under mode j
$d_{i,j} =$ Duration of activity i under mode j
$u_{i,j} =$ Amount of manpower resource used daily by activity i under mode j

4.1.3. Decision Variables

$TC =$ Total cost
$DUR =$ Project duration
$L_k =$ Peak value of daily usage of manpower resource type k
$LL =$ Peak value of cumulative daily manpower resource usage, including all manpower resources
$D_{dly} =$ Project delay
$x_{i,j} =$ 1 if activity i is conducted under mode j; 0 otherwise
$ys_{i,t} =$ 1 if activity i is started on day t; 0 otherwise
$z_{i,j,t} =$ 1 if activity i is started on day t under mode j; 0 otherwise
$f_i =$ Finish day of activity i
$r_{kt} =$ Amount of manpower resource k used on day t

4.1.4. Model

$$\min TC = \sum_{i=1}^{N}\sum_{j=1}^{M} c_{i,j} \cdot x_{i,j} + IC \cdot DUR + DP \cdot D_{dly} + \sum_{k=1}^{H} UC \cdot L_k + TUC \cdot LL \tag{4.1}$$

Subject to:

$$\sum_{j=1}^{M} x_{i,j} = 1 \quad \forall i \in I \tag{4.2}$$

$$f_i \geq f_p + \sum_{j=1}^{M} d_{i,j} \cdot x_{i,j} \quad \forall i \in I,\ \forall p \in P_i \tag{4.3}$$

$$f_i = \sum_{t'=1}^{Dmax} t' \cdot ys_{i,t'} + \sum_{j=1}^{M} d_{i,j} \cdot x_{i,j} \quad \forall i \in I,\ \forall t' \in T \tag{4.4}$$

$$\sum_{t=1}^{Dmax} ys_{i,t} = 1 \quad \forall i \in I \tag{4.5}$$

$$r_{kt} = \sum_{i}^{N} \sum_{j}^{M} \sum_{t'=1}^{Dmax} u_{i,j} \cdot z_{i,j,t'} \quad \forall k \in K,\ \forall t' \in T \tag{4.6}$$

$$r_{kt} \le L_k \quad \forall k \in K,\ \forall t \in T \tag{4.7}$$

$$\sum_{k=1}^{H} r_{kt} \le LL \quad \forall t \in T \tag{4.8}$$

$$z_{i,j,t} \le x_{i,j} \quad \forall i \in I,\ \forall j \in J,\ \forall t \in T \tag{4.9}$$

$$z_{i,j,t} \le ys_{i,t} \quad \forall i \in I,\ \forall j \in J,\ \forall t \in T \tag{4.10}$$

$$z_{i,j,t} \ge x_{i,j} + ys_{i,t} - 1 \quad \forall i \in I,\ \forall j \in J,\ \forall t \in T \tag{4.11}$$

$$f_N \le DUR \tag{4.12}$$

$$DUR \le D_{max} \tag{4.13}$$

$$D_{dly} \le DUR - D_{avg} \tag{4.14}$$

$$f_1 = 0 \tag{4.15}$$

$$x_{i,j} \in \{0,1\} \quad \forall i \in I,\ \forall j \in J \tag{4.16}$$

$$ys_{i,t} \in \{0,1\} \quad \forall i \in I,\ \forall t \in T \tag{4.17}$$

$$z_{i,j,t} \in \{0,1\} \quad \forall i \in I,\ \forall j \in J,\ \forall t \in T \tag{4.18}$$

$$r_{kt} \ge 0 \quad \forall k \in K,\ \forall t \in T \tag{4.19}$$

$$L_k \geq 0 \quad \forall k \in K \tag{4.20}$$

$$f_i \geq 0 \quad \forall i \in I \tag{4.21}$$

$$DUR,\ LL,\ D_{dly} \geq 0 \tag{4.22}$$

The objective function (4.1) aims to minimize the total cost of the project consisting of direct cost, indirect cost, delay penalty (in case the deadline is exceeded), peak manpower resource usage cost per each resource type, and cumulative peak manpower resource usage cost of all resource types. In the objective function – and throughout the whole study- the term “manpower resource” is emphasized to indicate the considered resources’ type. According to this, all the resources used in the study are defined as manpower resource. For example, one of the resources might be a welder, and another might be a scaffolder. Therefore, summation of these resources’ peak values does not lead to ambiguity. To illustrate, the peak values of welders and scaffolders can be summed, and the resulting cumulative value can be used to calculate the total accommodation cost (i.e. the camp size) of both resources. This is also explained in Figure 4.1, Figure 4.2, and Figure 4.3, respectively. Figure 4.1 shows daily resource histogram of a manpower resource requirement, named as Resource 1. Resource 1 has its peak value (7) on the second day throughout the project. Likewise, Figure 4.2 represents that Resource 2 has its peak value (8) on the fourth day. Figure 4.3, that includes the cumulative resource histogram of Resource 1 and Resource 2, indicates that the cumulative peak value (14) on the fourth day.

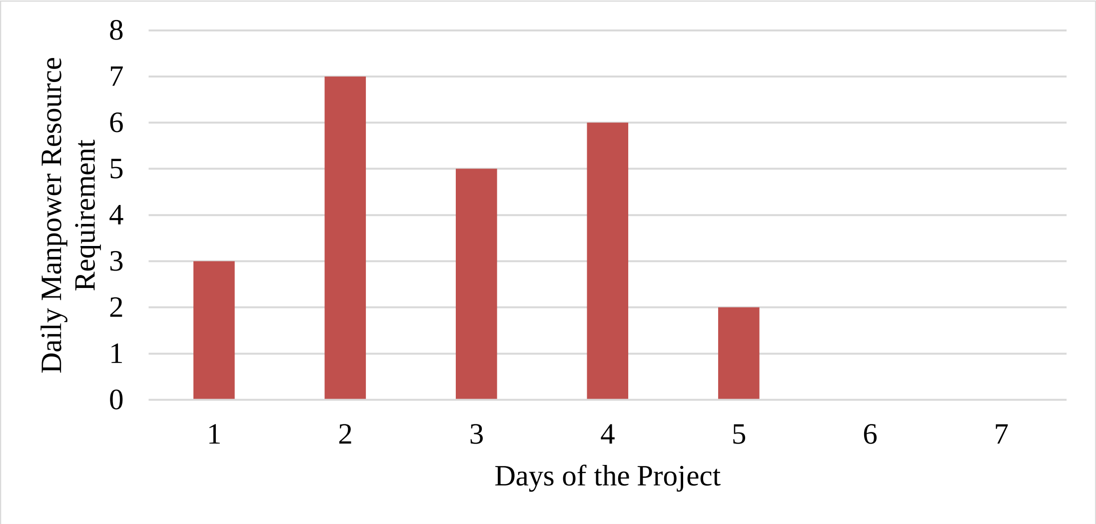

Figure 4.1: Manpower Resource Histogram of Resource 1

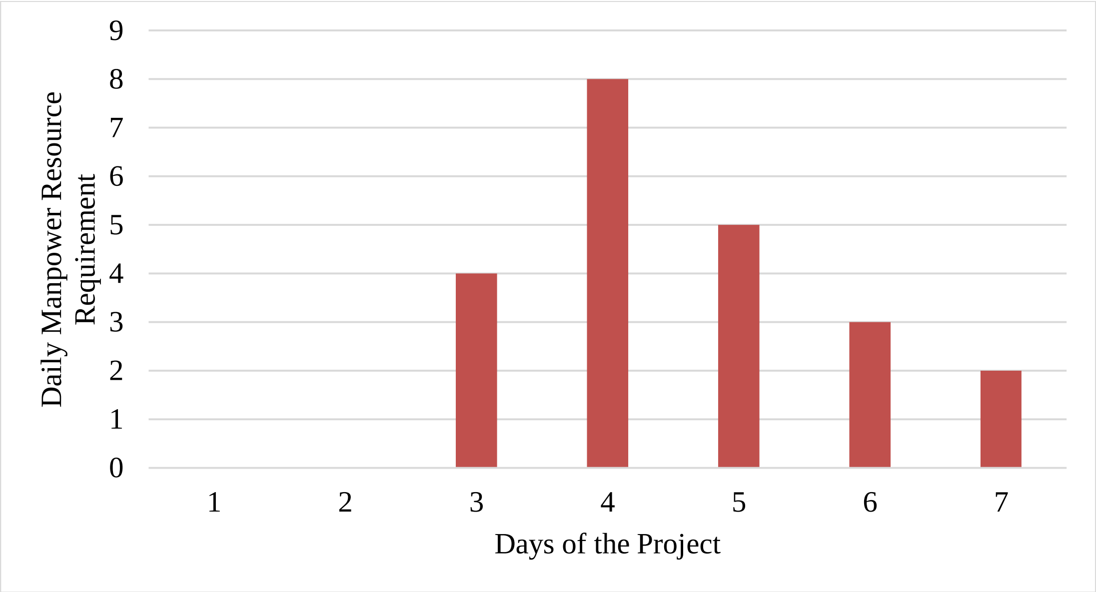

Figure 4.2: Manpower Resource Histogram of Resource 2

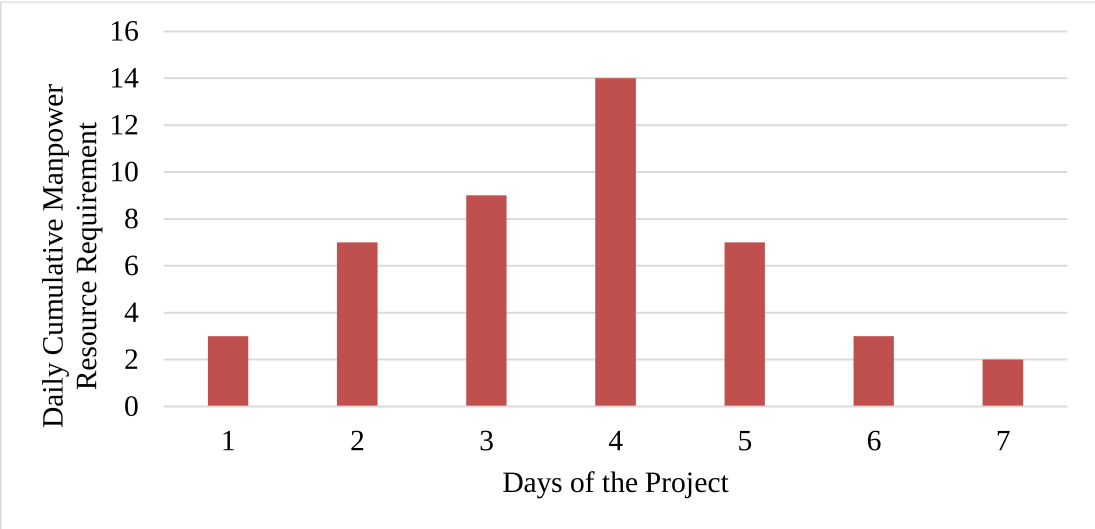

Figure 4.3: Cumulative Manpower Resource Histogram of Resource 1 & Resource 2

The objective function performs these peak minimizations considering a project's direct costs, daily indirect cost, and delay penalty. Direct costs are taken into consideration in the activity level. Different cost alternatives are obtained for each activity with respect to different manpower resource utilization options, as mentioned in Chapter 3. The objective function evaluates these alternatives together with the peak resource utilization values, daily indirect cost and delay penalty, then it decides the optimum execution mode for each activity. Indirect costs cannot be assigned to any specific activity; hence, they are included in the objective function on a daily basis, throughout the whole project duration. For instance, manpower resource's insurance expenditure is a type of indirect cost. The objective function incorporates delay penalty in order to minimize the costs arisen from exceeding a project's deadline (rounded average of the maximum and the minimum probable project durations, calculated using the longest and the shortest activity modes). This part is not taken into consideration if the project duration is smaller than or equal to the project's deadline. Constraint (4.2) indicates that only one execution mode of all alternatives should be selected for each activity. (4.3) ensures that the finish date of an activity cannot be earlier than the date calculated by the summation of the activity's duration of the selected mode and the finish date of its predecessors.

Constraint (4.4) correlates the start date of an activity to its finish date. (4.5) provides a specific start date for each activity in the network. Constraint (4.6) calculates daily resource usage per each resource type with respect to corresponding execution modes of each activity, throughout the project duration. (4.7) finds the peak value of daily resource usage per each resource type. Constraint (4.8) estimates the peak value of cumulative daily resource usage including all resource types. (4.9), (4.10) and (4.11) are defined to prevent x, y and z variables to have values that are logically conflicting. . Constraint (4.12) expresses that the project cannot finish earlier than the final dummy activity's end date. (4.13) identifies the relation between the project duration and the maximum possible project duration. (4.14) states the amount of delay. Constraint (4.15) ensures that the dummy start activity has no duration; in other words, it starts and finishes on "day 0". (4.16), (4.17) and (4.18) explain that $x_{i,j}$, $ys_{i,t}$ and $z_{i,j,t}$ are binary variables, respectively. Constraint (4.19) represents that daily resource usage per each resource type must be a positive value. As a result of this, the peak value of daily resource usage per each resource type is also defined as a positive value in (4.20). Likewise, constraint (4.21) indicates that each activity must have a positive finish date, correspondingly. Finally, (4.22) defines that the project duration, the amount of delay and the peak value of cumulative daily resource usage of all resource types are positive values, altogether.

This study proposes that the described integrated model (considering both time-cost trade-off and resource leveling in the same objective function) is capable of providing better results compared to individual implementation of time-cost trade-off and resource leveling (the successive implementation of time-cost trade-off and resource leveling, respectively). In order to represent the differences between these approaches, DTCTP model is given in Equation 4.23 to Equation 4.33, and resource leveling model in terms of peak minimizations is given in Equation 4.34 to Equation 4.47.

DTCTP model:

$$\min TC = \sum_{i=1}^{N}\sum_{j=1}^{M} c_{i,j} \cdot x_{i,j} + IC \cdot DUR + DP \cdot D_{dly} \tag{4.23}$$

Subject to:

$$\sum_{j=1}^{M} x_{i,j} = 1 \quad \forall i \in I \tag{4.24}$$

$$f_i \geq f_p + \sum_{j=1}^{M} d_{i,j} \cdot x_{i,j} \quad \forall i \in I,\ \forall p \in P_i \tag{4.25}$$

$$f_i = \sum_{j=1}^{M} d_{i,j} \cdot x_{i,j} \quad \forall i \in I \tag{4.26}$$

$$f_N \leq DUR \tag{4.27}$$

$$DUR \leq D_{max} \tag{4.28}$$

$$D_{dly} \leq DUR - D_{avg} \tag{4.29}$$

$$f_1 = 0 \tag{4.30}$$

$$x_{i,j} \in \{0,1\} \quad \forall i \in I,\ \forall j \in J \tag{4.31}$$

$$f_i \geq 0 \quad \forall i \in I \tag{4.32}$$

$$DUR,\ D_{dly} \geq 0 \tag{4.33}$$

Resource leveling model in terms of peak minimizations (performed using each activity's optimum time-cost alternative obtained from DTCTP analysis):

$$\min TC = \sum_{k=1}^{H} UC \cdot L_k + TUC \cdot LL \qquad (4.34)$$

Subject to:

$$f_i \geq f_p + d_i \quad \forall i \in I, \ \forall p \in P_i \qquad (4.35)$$

$$f_i = \sum_{t'=1}^{DUR} t' \cdot ys_{i,t'} + d_i \quad \forall i \in I \qquad (4.36)$$

$$\sum_{t=1}^{DUR} ys_{i,t} = 1 \quad \forall i \in I \qquad (4.37)$$

$$r_{kt} = \sum_{i}^{N} \sum_{t'=1}^{DUR} u_{i,j} \cdot ys_{i,t'} \quad \forall k \in K \qquad (4.38)$$

$$r_{kt} \leq L_k \quad \forall k \in K, \ \forall t \in T \qquad (4.39)$$

$$\sum_{k=1}^{H} r_{kt} \leq LL \quad \forall t \in T \qquad (4.40)$$

$$f_N = DUR \qquad (4.41)$$

$$f_1 = 0 \qquad (4.42)$$

$$ys_{i,t} \in \{0,1\} \quad \forall i \in I, \ \forall t \in T \qquad (4.43)$$

$$r_{kt} \geq 0 \quad \forall k \in K, \ \forall t \in T \qquad (4.44)$$

$$L_k \geq 0 \quad \forall k \in K \qquad (4.45)$$

$$f_i \geq 0 \quad \forall i \in I \qquad (4.46)$$

$$LL \geq 0 \qquad (4.47)$$

4.2. Modelling Process of the Generated Instances

In order to obtain exact solutions of the generated problems, each instance is modelled using MIP model. Then, modelled problems are solved using a mathematical optimization software. Available solvers on the market include CPLEX, GUROBI, XPRESS, and so on. Among them, GUROBI is selected, due to its superior performance (GUROBI 6.5 Performance Benchmarks, 2015) as well as its free academic license option. In addition to this, it supports a great variety of problem types as: Mixed Integer Programming (MIP) in general (including Mixed Integer Linear Programming (MILP), Mixed Integer Quadratic Programming (MIQP), and Mixed Integer Quadratically Constrained Programming (MIQCP)), Linear Programming (LP), Quadratic Programming (QP) and Quadratically Constrained Programming (QCP). Furthermore, GUROBI is compatible with numerous programming languages including C, C++, Java, .NET (C#), Python, MATLAB, and R (GUROBI Optimizer Reference Manual, 2016).

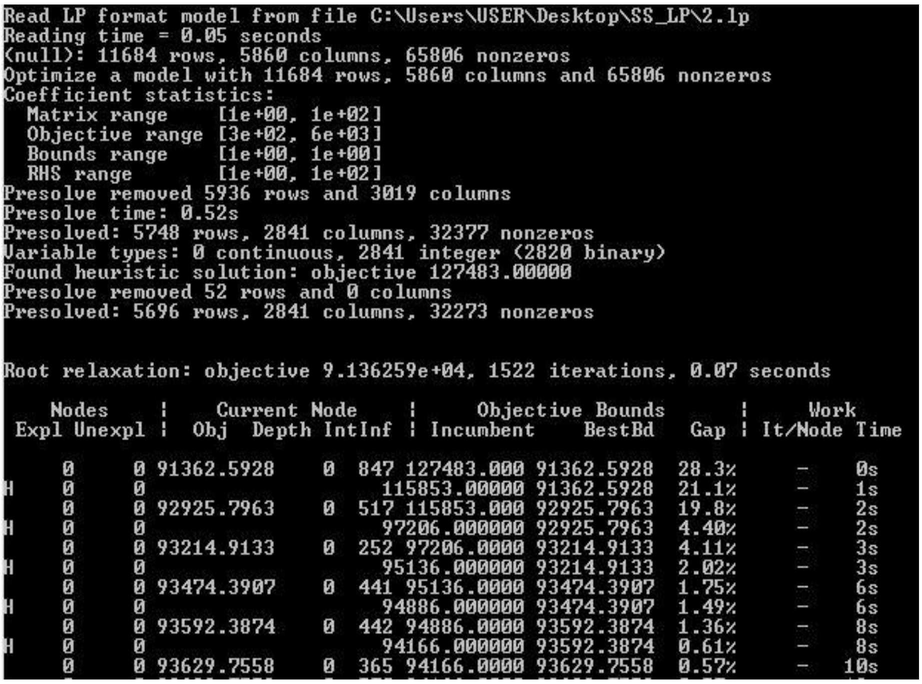

Figure 4.4: A View from GUROBI's Execution Process

4.2.1. GUROBI Input Format

The GUROBI optimizer is compatible with different file formats. In this study, input files are prepared according to LP format. One of the main advantages of LP file format is, it is human-readable; hence, it is easy to produce LP files. In addition to this, it does not preserve the order of variables; thus, different solution paths may be obtained if the model is run at different times. As a result, LP format appears as the most convenient one for model debugging. (GUROBI Optimizer Reference Manual, 2016).

An LP file is composed of 4 section headings. Firstly, the objective function is defined. This part declares the characteristics of the problem either to be minimized or maximized. Then, constraints that must be satisfied by each variable are given under "Subject To" heading. Thirdly, variable types are identified in the following section. Finally, completion of the model is denoted by "End" phrase in the last section. Figure 4.5 illustrates the LP format with a simplified problem.

```
Maximize
A + 2 B + 3 C
Subject To
A + B <= 4
B + 2 C <= 10
Generals
A B C
End
```

Figure 4.5: Format of a Sample LP File

Within the context of this study, a C# code is developed in Visual Studio 2013 environment, in order to convert problem sets into LP files. A total of 1215 LP files are composed. Then, they are read and solved using GUROBI, by means of another C# code developed in Visual Studio 2013 environment. The code sets a time limit of 600 seconds per each problem. If GUROBI still continues execution at the end of this time limit, the problem is accepted as unsolved. In addition to this, solution processes of all problems are recorded by means of created log files.

Finally, results of the solved instances are saved in terms of text files that include the corresponding value of each variable in order to obtain the optimum solution. Sample sections from LP files, log files and result files with respect to scope of the study are provided in Figure 4.6, Figure 4.7, and Figure 4.8, respectively.

```
Minimize

1080 X21 + 1444 X22 + 4200 X31 + 5294 X32 + 2000 X41 + 2750 X42 +
2240 X51 + 2888 X52 + 1120 X61 + 1513 X62 + 2600 X71 + 3438 X72 +
3200 X81 + 4400 X82 + 1800 X91 + 2407 X92 + 3120 X101 + 4125 X102
+ 2880 X111 + 3713 X112 + 250 L1 + 500 DUR + 1000 DDLY

Subject To

X21 + X22 = 1
X31 + X32 = 1
X41 + X42 = 1
X51 + X52 = 1
X61 + X62 = 1
X71 + X72 = 1
X81 + X82 = 1
X91 + X92 = 1
X101 + X102 = 1
X111 + X112 = 1
F2 - F1 - 9 X21 - 7 X22 >= 0
F3 - F1 - 15 X31 - 11 X32 >= 0
F4 - F1 - 10 X41 - 8 X42 >= 0
F5 - F1 - 8 X51 - 6 X52 >= 0
F6 - F1 - 14 X61 - 11 X62 >= 0
F7 - F5 - 13 X71 - 10 X72 >= 0
F8 - F4 - 10 X81 - 8 X82 >= 0
F8 - F7 - 10 X81 - 8 X82 >= 0
F9 - F3 - 9 X91 - 7 X92 >= 0
F10 - F2 - 13 X101 - 10 X102 >= 0
F10 - F6 - 13 X101 - 10 X102 >= 0
F11 - F2 - 12 X111 - 9 X112 >= 0
F11 - F8 - 12 X111 - 9 X112 >= 0
F12 - F9 >= 0
F12 - F10 >= 0
F12 - F11 >= 0
DUR - F12 >= 0
DDLY >= 0
DUR - DDLY <= 38
DUR <= 43

YS201 + YS202 + YS203 + YS204 + YS205 + YS206 + YS207 + YS208 +
YS209 + YS2010 + YS2011 + YS2012 + YS2013 + YS2014 + YS2015 +
YS2016 + YS2017 + YS2018 + YS2019 + YS2020 + YS2021 + YS2022 +
YS2023 + YS2024 + YS2025 + YS2026 + YS2027 + YS2028 + YS2029 +
YS2030 + YS2031 + YS2032 + YS2033 + YS2034 + YS2035 + YS2036 +
YS2037 + YS2038 + YS2039 + YS2040 + YS2041 + YS2042 + YS2043 = 1
```

Figure 4.6: A Sample Section of an LP File

```
 Expl Unexpl |   Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0      0 45000.4954    0  155 63722.0000 45000.4954  29.4%     -     0s
H    0      0                    58471.000000 45000.4954  23.0%     -     0s
     0      0 47759.3075    0   84 58471.0000 47759.3075  18.3%     -     0s
H    0      0                    49309.000000 47759.3075  3.14%     -     0s
     0      0 48040.5590    0   51 49309.0000 48040.5590  2.57%     -     0s
H    0      0                    48809.000000 48040.5590  1.57%     -     0s
     0      0 48081.4289    0  108 48809.0000 48081.4289  1.49%     -     0s
     0      0 48081.4289    0  104 48809.0000 48081.4289  1.49%     -     0s
     0      0 48081.4289    0   48 48809.0000 48081.4289  1.49%     -     0s
     0      0 48113.7300    0   87 48809.0000 48113.7300  1.42%     -     0s
     0      0 48154.3620    0   77 48809.0000 48154.3620  1.34%     -     0s
     0      0 48211.5138    0   75 48809.0000 48211.5138  1.22%     -     0s
     0      0 48241.0938    0   75 48809.0000 48241.0938  1.16%     -     0s
     0      0 48265.9904    0   88 48809.0000 48265.9904  1.11%     -     0s
     0      0 48278.4574    0   86 48809.0000 48278.4574  1.09%     -     0s
     0      0 48279.9351    0   93 48809.0000 48279.9351  1.08%     -     1s
     0      0 48283.0928    0   94 48809.0000 48283.0928  1.08%     -     1s
     0      0 48284.4148    0   98 48809.0000 48284.4148  1.07%     -     1s
     0      0 48285.7621    0   97 48809.0000 48285.7621  1.07%     -     1s
H    0      0                    48661.000000 48285.7621  0.77%     -     1s
     0      0 48289.7410    0   96 48661.0000 48289.7410  0.76%     -     1s
     0      0 48289.7410    0   29 48661.0000 48289.7410  0.76%     -     1s
     0      0 48289.7410    0   60 48661.0000 48289.7410  0.76%     -     1s
     0      0 48289.7410    0   59 48661.0000 48289.7410  0.76%     -     1s
     0      0 48292.7698    0   66 48661.0000 48292.7698  0.76%     -     1s
     0      0 48294.7958    0   60 48661.0000 48294.7958  0.75%     -     1s
     0      0 48295.3307    0   63 48661.0000 48295.3307  0.75%     -     1s
     0      0 48295.7029    0   58 48661.0000 48295.7029  0.75%     -     1s
     0      0 48295.7029    0   58 48661.0000 48295.7029  0.75%     -     1s
     0      0 48295.7029    0   26 48661.0000 48295.7029  0.75%     -     1s
     0      0 48295.7029    0   51 48661.0000 48295.7029  0.75%     -     1s
     0      0 48299.4800    0   44 48661.0000 48299.4800  0.74%     -     1s
     0      0 48306.7235    0   48 48661.0000 48306.7235  0.73%     -     1s
     0      0 48310.0027    0   51 48661.0000 48310.0027  0.72%     -     1s
     0      0 48310.0027    0   24 48661.0000 48310.0027  0.72%     -     1s
     0      0 48310.0027    0   49 48661.0000 48310.0027  0.72%     -     1s
     0      0 48310.0027    0   48 48661.0000 48310.0027  0.72%     -     1s
     0      0 48310.0027    0   49 48661.0000 48310.0027  0.72%     -     2s
     0      0 48310.0027    0   26 48661.0000 48310.0027  0.72%     -     2s
     0      0 48310.0027    0   51 48661.0000 48310.0027  0.72%     -     2s
     0      0 48310.0027    0   54 48661.0000 48310.0027  0.72%     -     2s
     0      0 48310.0027    0   44 48661.0000 48310.0027  0.72%     -     2s
```

Figure 4.7: A Sample Section of a Log File

```
Optimal objective: 48661
Solution 0 has objective: 48661
Solution 1 has objective: 48809
Solution 2 has objective: 49309
Solution 3 has objective: 58471
Solution 4 has objective: 63722
X21 1
X22 0
X31 1
X32 0
X41 1
X42 0
X51 1
X52 0
X61 1
X62 0
X71 0
X72 1
X81 1
X82 0
X91 1
X92 0
X101 1
X102 0
X111 0
X112 1
L1 17
DUR 37
DDLY 0
F2 9
F1 0
F3 24
F4 10
F5 8
F6 24
F7 18
F8 28
F9 37
F10 37
F11 37
F12 37
YS201 1
YS202 0
YS203 0
YS204 0
YS205 0
YS206 0
```

Figure 4.8: A Sample Section of a Result File

4.3. Computational Experiments

In order to assess suggested MIP model's performance, generated problems are solved in GUROBI, with a time limit of 600 seconds per each problem. Parameters used in each problem set are given in Table 3.5, explicitly. In order to observe the effects of these parameters to the results, average computational times of the solved instances, and their solution percentages are taken into consideration. Table 4.1 represents the instances' average solution times, together with corresponding solution rates, with respect to number of activities, number of modes, number of resource types, resource factor, and restrictiveness of Thesen, respectively. Table 4.2 shows the problems' solution percentages with respect to different number of modes and resource types. According to these results, the most explicit parameters affecting the model's performance are given in Figure 4.9, Figure 4.10, and Figure 4.11, respectively.

Table 4.1: Average Solution Times and Corresponding Solution Percentages with Respect to Different Parameters

		Average Solution Time (seconds)	Solution Percentage (%)
Number of Activity	**10**	49.70	97.28
	15	114.73	58.77
	20	147.90	35.06
Number of Mode	**2**	68.72	78.27
	3	97.18	58.52
	4	104.87	54.32
Number of Resource Type	**1**	51.83	94.81
	2	115.74	53.33
	4	132.11	42.96
Restrictiveness of Thesen	**0.25**	62.34	63.21
	0.50	101.52	63.95
	0.75	98.99	63.95
Resource Factor	**0.20**	93.62	69.63
	0.50	83.11	62.72
	0.80	85.62	58.77

Table 4.2: The Problems' Solution Percentages with Respect to Different Modes & Resource Types

Problem Set	The Number of Problems			Solution Percentage (%)
	Total	Solved	Unsolved	
10 ACT & 1 RES & 2 MODE	45	45	0	100.00
10 ACT & 1 RES & 3 MODE	45	45	0	100.00
10 ACT & 1 RES & 4 MODE	45	45	0	100.00
10 ACT & 2 RES & 2 MODE	45	45	0	100.00
10 ACT & 2 RES & 3 MODE	45	45	0	100.00
10 ACT & 2 RES & 4 MODE	45	45	0	100.00
10 ACT & 4 RES & 2 MODE	45	45	0	100.00
10 ACT & 4 RES & 3 MODE	45	40	5	88.89
10 ACT & 4 RES & 4 MODE	45	39	6	86.67
15 ACT & 1 RES & 2 MODE	45	45	0	100.00
15 ACT & 1 RES & 3 MODE	45	45	0	100.00
15 ACT & 1 RES & 4 MODE	45	45	0	100.00
15 ACT & 2 RES & 2 MODE	45	39	6	86.67
15 ACT & 2 RES & 3 MODE	45	20	25	44.44
15 ACT & 2 RES & 4 MODE	45	5	40	11.11
15 ACT & 4 RES & 2 MODE	45	29	16	64.44
15 ACT & 4 RES & 3 MODE	45	8	37	17.78
15 ACT & 4 RES & 4 MODE	45	2	43	4.44
20 ACT & 1 RES & 2 MODE	45	43	2	95.56
20 ACT & 1 RES & 3 MODE	45	32	13	71.11
20 ACT & 1 RES & 4 MODE	45	39	6	86.67
20 ACT & 2 RES & 2 MODE	45	16	29	35.56
20 ACT & 2 RES & 3 MODE	45	1	44	2.22
20 ACT & 2 RES & 4 MODE	45	0	45	0.00
20 ACT & 4 RES & 2 MODE	45	10	35	22.22
20 ACT & 4 RES & 3 MODE	45	1	44	2.22
20 ACT & 4 RES & 4 MODE	45	0	45	0.00

It is obvious that the activity number has a substantial impact on the results. 97.28% of all the problems consisting of 10 activities are solved successfully, within around 50 sec. However, the solution rate of 15-activity problems is obtained as 58.77%, together with an average computational time of 114.73 sec. These values become 35.06% and 147.90 sec. for problems comprised of 20 activities. That is, the average computational time of the solved instances is almost tripled, compared to the problems with 10 activities.

In fact, this is an expected result, because the number of activities in a problem directly indicates its size. As a problem's size expands, the number of constraints to be satisfied by each variable increases, too. Hence, the problems become difficult as the number of activities increase. Figure 4.9 illustrates this effect, comparing the number of activities with their average solution times and corresponding solution rates.

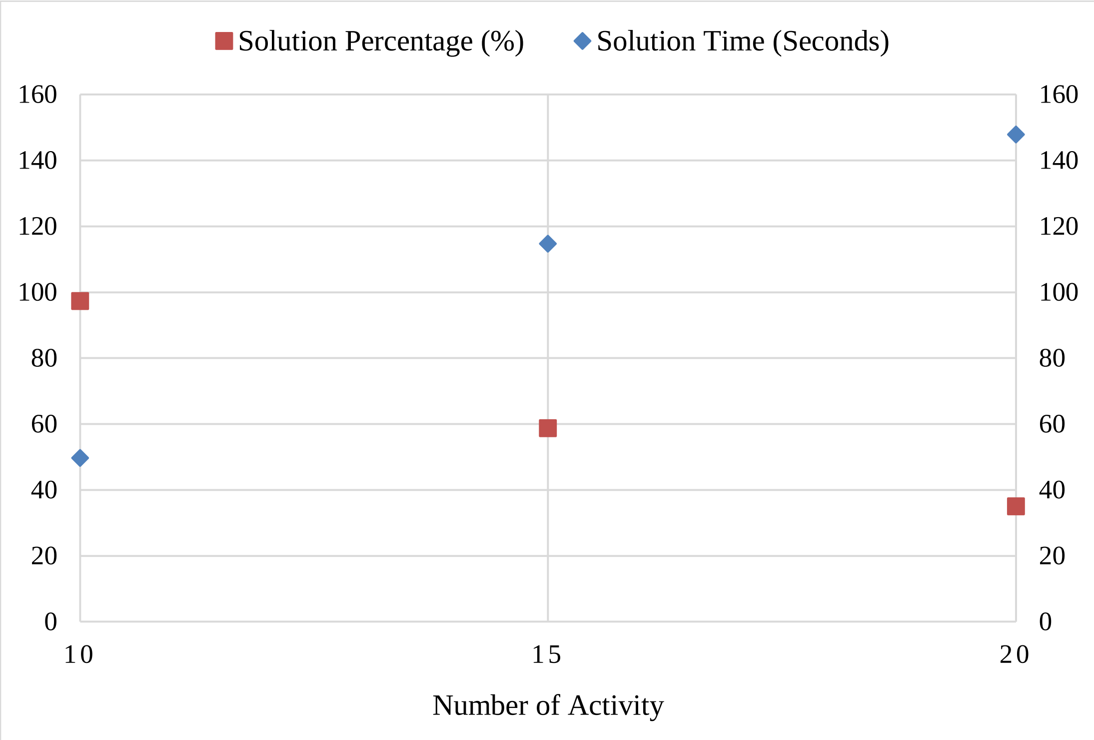

Figure 4.9: Average Solution Time & Solution Percentages vs. Number of Activities

The other important parameter is the number of different time-cost-resource alternatives per activity, in a problem. In the experiments, 78.27% of the problems with 2 modes are solved within 69 sec., nearly. This rate becomes 58.52% and 54.32%, associated with solution times of 97.18 sec. and 104.87 sec., for the 3 and 4 mode-problems, respectively. For example, 10-activity problems include 11 unsolved instances out of 405, and 6 of them have 4 modes. In other words, more than half (54.55%) of the total unsolved instances with 10 activities arise from 4-mode problems.

This rate becomes 49.70% and 36.50%, for 15 and 20-activity problems, respectively. On the other hand, all the 2-mode problems with 10 activities are solved. 13.17% and 25.10% of all the unsolved problems with 15 and 20 activities, respectively, is composed of 2-mode instances. In fact, comparative differences of these results denote that 2-mode problems are much easier than the others, due to their reduced search space. Figure 4.10 indicates the proposed model's performance for different mode numbers, in terms of their average solution times and corresponding solution rates.

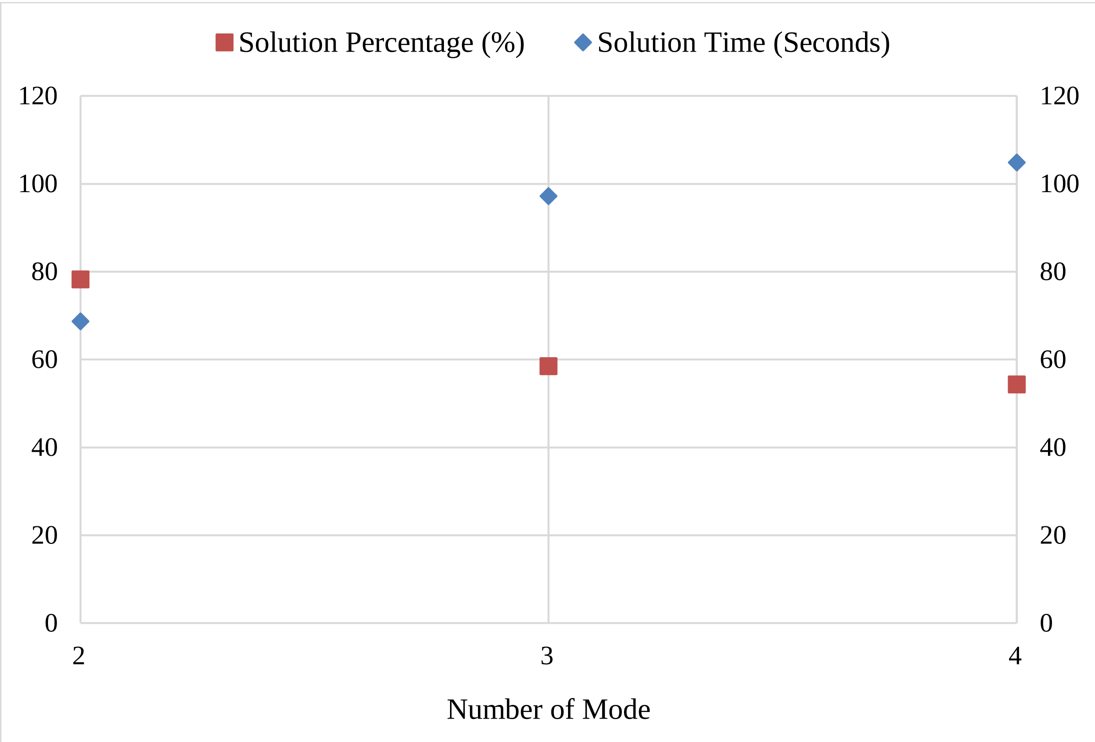

Figure 4.10: Average Solution Time & Solution Percentages vs. Number of Modes

Another significant criterion affecting the model's performance is the number of resource types. Solution rate of the problems with one resource is 94.81%, together with a fair amount of average computational time (51.83 sec.). The model is capable of solving most of the 20-activity problems in case of a single resource requirement, despite the average solution rate of 20-activity problems in general. To illustrate, 142 problems out of 405 samples of 20-activity problems in total are solved. Among these, 114 of them are the ones with a single resource. In other words, 80.28% of all the solved 20-activity problems consist of one resource. Nevertheless, the average solution ratio of the problems including 2 and 4 different resource types, decreases to 53.33% and 42.96%, respectively. In fact, this results from the model's nature, optimization of the cumulative maximum daily resource usage of all resources. As the resources diversify, the number of constraints increases too, resulting in a decline in the model's success rate. Figure 4.11 proves this, comparing the effect of number of different resource types in terms of their average solution times and corresponding solution rates.

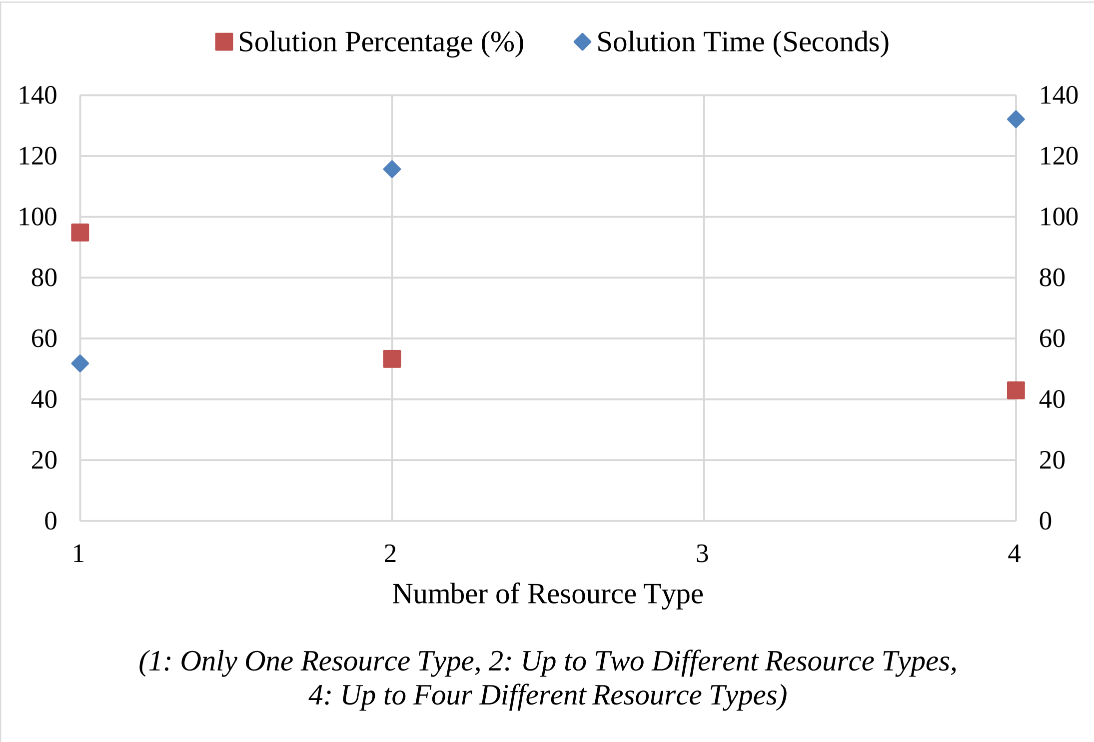

Figure 4.11: Average Solution Time & Solution Percentages vs. Number of Different Resource Types

Remaining parameters are restrictiveness of Thesen, and resource factor. Actually, both of these parameters are investigated with the anticipation of affecting the model's performance inversely (i.e. an increase in these parameters leads to a decrease in the solution percentages, and an increase in the solution times, correspondingly). Nonetheless, experiment results deviate from this pattern. For example, average computational times of the problems with Thesen coefficients of 0.25, 0.50, and 0.75 are obtained as 62.34 sec., 101.52 sec., and 98.99 sec., respectively. Likewise, problems with resource factors of 0.20, 0.50, and 0.75 are solved within 93.62 sec., 83.11 sec., and 85.62 sec., respectively. This irregular pattern arises from the other parameter's strong influence on the instances. To illustrate, the problems with 10 activities and 0.75 Thesen coefficient have a solution rate of 93.33%; on the other hand, this rate becomes 31.11% for the ones with 20 activities and 0.25. In other words, although the former ones consist of three times more complex networks than the latter ones, their solution rate is three times higher, due to their lower activity number. In addition to this, unsolved problems are not included in average computational time calculations, as mentioned above. Considering these facts, obtained experiment results in terms of different Thesen coefficients and resource factors become clearer. Nevertheless, these factors may not have a significant effect on the computation time and a more comprehensive research is needed to make more elaborate inferences about the effect of these parameters.

The model is capable of performing a simultaneous time-cost trade-off analysis with resource leveling, unlike current practice in the literature, that is, the implementation of resource leveling based on selected mode options of time-cost trade-off. This study claims that the suggested model enables a better representation for the problem; thus, it provides better results than the current approach. In order to prove that, two sample instances are selected, and each one is solved by both methodologies. Then, the results are compared.

The first instance consists of 10 activities. Each activity requires up to 4 resource types and 4 time-cost-resource modes. The network's coefficient of Thesen restrictiveness is 0.25, and resource factor is 0.80. Precedence relation and resource information of the problem is given as follows:

```
12   4

1    0    0    0    0    0    5    3    2    6    5    4
2    9    4    3    8    5    1    10
3    14   4    6    6    6    2    8    11
4    13   2    0    0    5    1    8
5    8    5    0    0    6    1    7
6    15   5    8    5    7    1    8
7    12   0    4    7    2    3    11   9    8
8    14   4    2    7    4    1    12
9    12   0    0    0    3    1    12
10   14   4    3    5    4    1    12
11   9    6    6    2    8    1    12
12   0    0    0    0    0    0
1 0
2 4 9 4 3 8 5 7200 7 4 3 8 5 9625 5 6 4 11 7 13125 4 8 6 16 10
22500
3 4 14 4 6 6 6 12320 11 4 6 6 6 16638 8 6 8 8 8 22500 6 8 12 12
12 37125
4 4 13 2 0 0 5 3640 10 2 0 0 5 4813 7 3 0 0 7 6563 5 4 0 0 10
9844
5 4 8 5 0 0 6 3520 6 5 0 0 6 4538 5 7 0 0 8 7032 3 10 0 0 12 9282
6 4 15 5 8 5 7 15000 11 5 8 5 7 18907 8 7 11 7 10 26250 6 10 16
10 14 42188
7 4 12 0 4 7 2 6240 9 0 4 7 2 8044 6 0 6 10 3 10688 5 0 8 14 4
18282
8 4 14 4 2 7 4 9520 11 4 2 7 4 12857 7 6 3 10 6 16407 6 8 4 14 8
28688
9 4 12 0 0 0 3 1440 9 0 0 0 3 1857 7 0 0 0 4 2625 5 0 0 0 6 4219
10 4 14 4 3 5 4 8960 11 4 3 5 4 12100 8 6 4 7 6 17250 6 8 6 10 8
27000
11 4 9 6 6 2 8 7920 7 6 6 2 8 10588 5 8 8 3 11 14063 4 12 12 4 16
24750
12 0
```

Figure 4.12: Network and Resource Properties of the First Problem

At first, the problem experienced the implementation of time-cost trade-off and resource leveling, in a sequential manner. A delay penalty is applied in both cases, at a cost of 1,000 USD per day. In time-cost trade-off phase, the delay penalty is started to be applied if the project's duration exceeds the deadline (the average of maximum and minimum project durations). In this case, resource leveling phase's delay penalty calculation is started from TCT analysis's project finish date, instead of the project's deadline, in order to prevent a duplicate delay penalty implementation. In the light of these, following results are obtained:

Table 4.3: The First Instance's TCT Results

Activity Number	Selected Mode	Start Date	Finish Date
1 (Dummy Start)			
2	1	1	9
3	1	1	14
4	1	1	13
5	2	1	6
6	1	1	15
7	2	7	15
8	2	16	26
9	2	18	26
10	1	16	26
11	1	18	26
12 (Dummy Finish)			

Table 4.4: The First Instance's Resource Leveling (MRD) Results

Activity Number	Start Date	Finish Date
1 (Dummy Start)		
2	1	9
3	1	14
4	3	15
5	1	6
6	1	15
7	7	15
8	16	26
9	16	24
10	13	23
11	16	24
12 (Dummy Finish)		

Table 4.3 shows the selected mode of each activity, and corresponding project schedule as a result of the first instance's time-cost trade-off analysis. Table 4.4 represents the modified schedule in order to minimize the daily peak resource demand of each resource type and cumulative daily resource demand of all resource types. Accordingly, peak demands are calculated as 20, 21, 26, and 29 for each resource,

respectively. Cumulative daily peak value, on the other hand, is obtained as 87. Peak costs are determined as 250 and 500 USD/resource for each and cumulative resources, respectively. Therefore, total cost of this part is 67,500 USD. In addition to this, total cost of the TCT part is obtained as 97,336 USD. As a result of this, total cumulative cost (including time-cost trade-off and resource leveling) of instance becomes 164,836 USD. Then, the same instance is solved by the proposed model. Results of this integrated approach is provided in Table 4.5

Table 4.5: The First Instance's Proposed Model Outputs

Activity Number	Selected Mode	Start Date	Finish Date
1 (Dummy Start)			
2	1	1	9
3	1	10	23
4	1	10	22
5	1	1	8
6	2	1	11
7	1	12	23
8	2	24	34
9	2	26	34
10	1	12	25
11	1	26	34
12 (Dummy Finish)			

Daily peak value of each resource is obtained as 14, 14, 18, and 18, respectively. Cumulative daily peak resource demand is calculated as 58. Total project cost becomes 155,421 USD. According to these results, the proposed model improves not only the resource demands but also the total cost. For example, the daily peak value of Resource 1 is lowered by 30%. Likewise, the amount of reduction becomes 33.33%, 30.77%, 37.93%, and 33.33% for daily peak values of Resource 2, Resource 3, and Resource 4, respectively. Additively, the daily peak value of the cumulative resource demand is reduced by 33.33%. Finally, the total cost is decreased by 5.71%. In spite of these improvements, the proposed integrated approach requires much longer computational times, as compared to the sequential application of TCT and resource leveling.

For instance, solution times for the former and the latter approaches are obtained as 151.35 seconds and 0.12 seconds, respectively. The main reason of this major gap between solution times is arisen from the proposed model's nature. That is, evaluating the problem with an integrated manner leads to a more complicated problem. Therefore, the model's computational time increases substantially.

The second instance consists of 15 activities. Each activity requires up to 4 resource types and 2 time-cost-resource modes. The network's coefficient of Thesen restrictiveness is 0.75, and resource factor is 0.50. Precedence relation and resource information of the problem is given as follows:

```
17    4

1     0     0     0     0     0     5     3     2     4     6     5
2     8     0     0     2     0     4     8     11    15    7
3     15    2     6     7     6     4     8     11    12    7
4     8     8     5     7     7     2     10    12
5     10    5     6     4     0     4     10    12    7     16
6     13    3     4     3     6     2     13    9
7     11    3     4     2     2     2     14    15
8     11    3     2     0     4     4     13    15    14    16
9     10    0     6     6     4     5     11    12    7     15    16
10    14    7     2     3     8     4     11    13    15    14
11    12    2     0     0     0     1     17
12    13    5     7     8     3     1     17
13    10    8     7     0     3     1     17
14    11    0     3     0     4     1     17
15    13    8     4     3     3     1     17
16    15    5     6     5     5     1     17
17    0     0     0     0     0     0
1 0
2 2 8 0 0 2 0 640 6 0 0 2 0 825
3 2 15 2 6 7 6 12600 11 2 6 7 6 15882
4 2 8 8 5 7 7 8640 6 8 5 7 7 11138
5 2 10 5 6 4 0 6000 8 5 6 4 0 8250
6 2 13 3 4 3 6 8320 10 3 4 3 6 11000
7 2 11 3 4 2 2 4840 8 3 4 2 2 6050
8 2 11 3 2 0 4 3960 8 3 2 0 4 4950
9 2 10 0 6 6 4 6400 8 0 6 6 4 8800
10 2 14 7 2 3 8 11200 11 7 2 3 8 15125
11 2 12 2 0 0 0 960 9 2 0 0 0 1238
12 2 13 5 7 8 3 11960 10 5 7 8 3 15813
13 2 10 8 7 0 3 7200 8 8 7 0 3 9900
14 2 11 0 3 0 4 3080 8 0 3 0 4 3850
15 2 13 8 4 3 3 9360 10 8 4 3 3 12375
16 2 15 5 6 5 5 12600 11 5 6 5 5 15882
17 0
```

Figure 4.13: Network and Resource Properties of the Second Problem

The same procedure is applied for the second instance, too. Results of TCT and resource leveling parts are given in Table 4.6, and Table 4.7, respectively. The proposed model's results are given in Table 4.8. Both approaches' comparative results in terms of peak resource demands and total project cost, are given in Table 4.9.

Table 4.6: The Second Instance's TCT Results

Activity Number	Selected Mode	Start Date	Finish Date
1 (Dummy Start)			
2	1	1	8
3	1	1	15
4	1	1	8
5	1	1	10
6	2	1	10
7	2	21	28
8	1	16	26
9	1	11	20
10	1	11	24
11	1	30	41
12	1	29	41
13	1	32	41
14	1	31	41
15	1	29	41
16	1	27	41
17 (Dummy Finish)			

Table 4.7: The Second Instance's Resource Leveling Results

Activity Number	Start Date	Finish Date
1 (Dummy Start)		
2	1	8
3	1	15
4	7	14
5	1	10
6	1	10
7	21	28
8	16	26
9	11	20
10	15	28
11	30	41
12	22	34
13	31	40
14	30	40
15	29	41
16	27	41
17 (Dummy Finish)		

Table 4.8: The Second Instance's Proposed Model Outputs

Activity Number	Selected Mode	Start Date	Finish Date
1 (Dummy Start)			
2	1	1	8
3	1	9	23
4	1	1	8
5	1	1	10
6	1	1	13
7	1	24	34
8	2	24	31
9	1	14	23
10	1	14	27
11	2	29	37
12	1	24	36
13	1	38	47
14	1	37	47
15	1	35	47
16	1	33	47
17 (Dummy Finish)			

Table 4.9: Comparative Results of Both Approaches

	Time-cost Trade-off & Resource Leveling	The Proposed Model	Acquired Improvement (%)
Daily Peak Value of Resource 1 (R1)	28	21	25
Daily Peak Value of Resource 2 (R2)	27	20	25.93
Daily Peak Value of Resource 3 (R3)	23	16	30.43
Daily Peak Value of Resource 4 (R4)	19	18	5.26
Daily Peak Value of Cumulative Resource Demand	89	64	28.09
Total Cost (USD)	200,900	188,278	6.28
Solution Time (Seconds)	0.03 (TCT and RLP)	130.85	-

In the first approach (leveling of resources according to TCT part's results), the problem's TCT analysis provided the optimum cost as 132,150 USD, together with a project duration of 41 days. Then, resource leveling is performed with MRD metric, and daily peak value of each resource is obtained as 28, 27, 23, and 19, respectively. Cumulative daily peak resource demand is calculated as 89. According to these values, resource leveling part's total cost is 68,750 USD. As a result of this, the cumulative cost of TCT and resource leveling parts is obtained as 200,900 USD. The integrated approach, on the other hand, provided the optimum total cost as 188,278 USD. Furthermore, it reduced the daily peak value of each resource to 21, 20, 16, and 18, respectively. Finally daily cumulative peak value is reduced to 64. Improvements achieved by the proposed approach in terms of percentage change, are given in Table 4.9.

After solving these problems, scope of the study is extended to solve all the generated problems by means of successive implementation of TCT and RLP, in order to make a more detailed comparison between the current application in the literature and the proposed model. That is, a total of 1215 instances are solved in both means, and each instance's result (in terms of the total cost) together with its solution time is compared. In both practices, a time limit of 600 seconds is defined per each problem, and the problems that cannot be solved within this limit are defined as unsolved. In the successive implementation stage, this time limit is set separately, for time-cost trade-off and result leveling analyses. Similarly, in this stage, average solution times of time-cost trade-off and resource leveling parts are summed, and their cumulative value is considered as the problem's solution time. Results of both the successive implementation of TCT and RLP and the integrated approach are provided in Table 4.10, with respect to problem sets with different activities, manpower resources, and time-cost-resource modes. According to the results of the problems solved within time limit, the proposed integrated model either provides the same result with the current approach (successive implementation of TCT and RLP), or it finds a better result. Table 4.10 denotes the number of problems with improved results by means of the integrated model's application, as well as the average, maximum, and minimum rates of the acquired improvement. However, the number of the solved problems with the integrated approach is lower than the number of the solved problems with the current successive approach. Furthermore, average solution times of the integrated approach is considerably higher, as compared to the current approach. Table 4.11 includes information about the number of solved and unsolved problems in both means, together with the average solution times of the problems solved within the time limit, in both practices.

Table 4.10: The Extended Comparison of Both Approaches

Problem Type (ACT_RES_MODE)	The Number of Problems (Total / Solved / Unsolved)		The Number of Instances with Better Results	Acquired Improvement Rate of Better Results (%) (Avg. / Max. / Min.)
	Successive Implementation of TCT and RLP	The Integrated Model		
10_1_2	45 / 45 / 0	45 / 45 / 0	19	0.89 / 2.35 / 0.14
10_1_3	45 / 45 / 0	45 / 45 / 0	16	1.06 / 3.21 / 0.21
10_1_4	45 / 45 / 0	45 / 45 / 0	14	0.73 / 2.53 / 0.10
10_2_2	45 / 45 / 0	45 / 45 / 0	37	3.49 / 11.60 / 0.19
10_2_3	45 / 45 / 0	45 / 45 / 0	35	3.22 / 10.71 / 0.09
10_2_4	45 / 45 / 0	45 / 45 / 0	35	2.85 / 9.93 / 0.18
10_4_2	45 / 45 / 0	45 / 45 / 0	33	3.88 / 10.18 / 0.04
10_4_3	45 / 45 / 0	45 / 40 / 5	31	3.32 / 12.45 / 0.11
10_4_4	45 / 45 / 0	45 / 39 / 6	26	2.96 / 7.17 / 0.26
15_1_2	45 / 45 / 0	45 / 45 / 0	14	0.64 / 2.38 / 0.01
15_1_3	45 / 45 / 0	45 / 45 / 0	24	0.58 / 1.43 / 0.17
15_1_4	45 / 45 / 0	45 / 45 / 0	18	0.61 / 1.27 / 0.09
15_2_2	45 / 45 / 0	45 / 39 / 6	29	1.31 / 5.12 / 0.08
15_2_3	45 / 45 / 0	45 / 20 / 25	18	2.33 / 6.56 / 0.66
15_2_4	45 / 45 / 0	45 / 5 / 40	3	1.62 / 2.28 / 1.11
15_4_2	45 / 45 / 0	45 / 29 / 16	21	1.87 / 6.58 / 0.03
15_4_3	45 / 45 / 0	45 / 8 / 37	7	1.68 / 3.90 / 0.18
15_4_4	45 / 45 / 0	45 / 2 / 43	1	8.79 / 8.79 / 8.79
20_1_2	45 / 45 / 0	45 / 43 / 2	9	0.40 / 1.35 / 0.01
20_1_3	45 / 45 / 0	45 / 32 / 13	4	0.63 / 1.62 / 0.15
20_1_4	45 / 45 / 0	45 / 39 / 6	21	0.29 / 2.82 / 0.12
20_2_2	45 / 44 / 1	45 / 16 / 29	7	1.07 / 2.82 / 0.12
20_2_3	45 / 45 / 0	45 / 1 / 44	1	0.67 / 0.67 / 0.67
20_2_4	45 / 45 / 0	45 / 0 / 45	0	-
20_4_2	45 / 43 / 2	45 / 10 / 35	6	2.20 / 4.82 / 0.69
20_4_3	45 / 45 / 0	45 / 1 / 44	0	-
20_4_4	45 / 45 / 0	45 / 0 / 45	0	-

Table 4.11: The Extended Comparison of Both Approaches-II

Problem Type (ACT_RES_MODE)	The Number of Problems (Total / Solved / Unsolved)		Average Solution Time (seconds)	
	Successive Implementation of TCT and RLP	The Integrated Model	Successive Implementation of TCT and RLP	The Integrated Model
10_1_2	45 / 45 / 0	45 / 45 / 0	0.01	1.21
10_1_3	45 / 45 / 0	45 / 45 / 0	0.01	4.96
10_1_4	45 / 45 / 0	45 / 45 / 0	0.01	6.82
10_2_2	45 / 45 / 0	45 / 45 / 0	0.05	11.62
10_2_3	45 / 45 / 0	45 / 45 / 0	0.02	60.94
10_2_4	45 / 45 / 0	45 / 45 / 0	0.02	110.75
10_4_2	45 / 45 / 0	45 / 45 / 0	0.06	21.50
10_4_3	45 / 45 / 0	45 / 40 / 5	0.04	93.32
10_4_4	45 / 45 / 0	45 / 39 / 6	0.06	155.10
15_1_2	45 / 45 / 0	45 / 45 / 0	0.03	11.08
15_1_3	45 / 45 / 0	45 / 45 / 0	0.02	40.05
15_1_4	45 / 45 / 0	45 / 45 / 0	0.02	48.28
15_2_2	45 / 45 / 0	45 / 39 / 6	0.48	153.95
15_2_3	45 / 45 / 0	45 / 20 / 25	0.29	303.20
15_2_4	45 / 45 / 0	45 / 5 / 40	0.14	361.39
15_4_2	45 / 45 / 0	45 / 29 / 16	1.86	178.03
15_4_3	45 / 45 / 0	45 / 8 / 37	0.94	374.09
15_4_4	45 / 45 / 0	45 / 2 / 43	2.31	400.63
20_1_2	45 / 45 / 0	45 / 43 / 2	0.08	73.65
20_1_3	45 / 45 / 0	45 / 32 / 13	0.11	147.65
20_1_4	45 / 45 / 0	45 / 39 / 6	0.05	178.25
20_2_2	45 / 44 / 1	45 / 16 / 29	5.79	153.30
20_2_3	45 / 45 / 0	45 / 1 / 44	0.80	423.73
20_2_4	45 / 45 / 0	45 / 0 / 45	1.01	-
20_4_2	45 / 43 / 2	45 / 10 / 35	2.77	295.57
20_4_3	45 / 45 / 0	45 / 1 / 44	6.09	325.67
20_4_4	45 / 45 / 0	45 / 0 / 45	14.53	-

CHAPTER 5

CONCLUSION

This study introduces an improved approach for two major project scheduling problems, namely DTCTP and RLP. The existing studies investigating these problems classify DTCTP and RLP as two different subclasses of project scheduling literature; hence, concentrate on these problems separately. Nevertheless, this study claims that an integrated approach for these problems will provide better results. Thus, this thesis describes a multi-objective MIP model in this context. In order to evaluate the model's performance, benchmark instances are generated for the discrete time-cost trade-off and resource leveling problem.

Problem generation process includes different stages. Firstly, networks with various activity precedence relationships, durations, and resource utilization values are obtained using ProGen/max generator. Generated networks comprise of 3 different hardness coefficients (0.25, 0.50, and 0.75), 3 different resources (1 resource, up to 2 resources, and up to 4 resources), 3 different resource factors (0.20, 0.50, and 0.80), 3 different activity numbers (10, 15, and 20), and 3 alternative modes to execute each activity (2, 3, and 4). 5 sample instances are generated for each and every combination of these parameters. Then, a unit direct cost is assigned per each resource utilized to execute an activity, and the cumulative cost of these resources is calculated as the activity's direct cost. Finally, alternative modes are generated with different time-cost-resource options, using the direct cost of each activity and ProGen/max outputs in terms of activity durations and resource utilizations. As a result, 1215 instances are generated in total, which may also be used as benchmark instances in the future studies.

A multi-objective MIP model is proposed with the objective of minimizing the total cost of the project including the direct costs of each activity with respect to its selected execution mode, maximum daily resource utilization cost per each resource type, and cumulative maximum daily resource utilization cost of all resource types. The model also takes delay penalty into consideration in case a project's duration exceeds its deadline (i.e. the average of the maximum and minimum probable project durations). Using this model, all the generated problems are solved in GUROBI 6.0.5 optimization program. In order to be able to solve these instances, a C# code is developed in Visual Studio 2013 environment by setting a time limit of 600 seconds per each problem.

The results reveal the proposed model is very successful in solving 10-activity problems. However, its solution rate decreases as the number of activities increases. In fact, the majority of 20-activity problems remains unsolved within the time limit. Likewise, mode number is another important problem affecting the model's performance. The solution rate decreases as the number of modes increases. Network hardness, and resource factor are the remaining parameters; nonetheless, their impact on the solution rate is not as significant as the above-mentioned ones. Thus, further studies require an elaborate investigation in terms of the effect of these parameters on the model's performance. In addition to this, the cost coefficient used in problem generation stage may have an impact on the results. Hence, future research may include a sensitivity analysis for the effect of these cost values on the model's performance.

To fill the gap in the literature the integrated discrete time-cost trade-off and resource leveling problem are presented, benchmark instances are generated and solved with the proposed MIP model for the first time. The results provided by the proposed MIP model cannot be compared with any other benchmark result. This might lead to concerns about the results' accuracy. For this reason, a small group the generated instances are selected randomly, and solved with the proposed MIP model in GAMS 24.2.3 optimizer. Both GUROBI 6.0.5 and GAMS 24.2.3 provided the same results for the selected problem set.

The proposed model is capable of solving instances including up to 20 activities, and it is very limited for solving practical size problems. This is mainly related with the difficulty of the problem studied. However, future exact studies may focus on different MIP formulations or different optimization techniques such as, branch and bound method to improve the performance of the proposed model.

An important conclusion of this thesis is that the proposed integrated approach for the discrete time-cost trade-off and resource leveling problem provides better results than the existing approach. Hence, future heuristic and meta-heuristic research focusing on integrated discrete time-cost trade-off and resource leveling problem will enable better options within much reasonable computational times for resource scheduling of the construction projects. The optimal solutions of the benchmark instances presented in this study provide a benchmark for evaluation of the performance of the heuristic and meta-heuristic for the DTCTRLP. Large size benchmark instances for the DTCTRLP can also be created using the proposed procedure.

REFERENCES

Abdel-Raheem, M., & Khalafallah, A. (2011). Using Electimize to Solve the Time-Cost-Tradeoff Problem in Construction Engineering. In *Proceedings of 2011 ASCE International Workshop on Computing in Civil Engineering, June* (pp. 19-22).

Afshar, A., Ziaraty, A. K., Kaveh, A., & Sharifi, F. (2009). Nondominated archiving multicolony ant algorithm in time–cost trade-off optimization. *Journal of Construction Engineering and Management*, *135*(7), 668-674.

Ashuri, B., & Tavakolan, M. (2011). Fuzzy enabled hybrid genetic algorithm–particle swarm optimization approach to solve TCRO problems in construction project planning. *Journal of Construction Engineering and Management*, *138*(9), 1065-1074.

Ashuri, B., & Tavakolan, M. (2015). Shuffled Frog-Leaping Model for Solving Time-Cost-Resource Optimization Problems in Construction Project Planning. *Journal of Computing in Civil Engineering*, *29*(1).

Ballestín, F., Schwindt, C., & Zimmermann, J. (2007). Resource leveling in make-to-order production: modeling and heuristic solution method. *International Journal of Operations Research*, *4*(1), 50-62.

Bandelloni, M., Tucci, M., & Rinaldi, R. (1994). Optimal resource leveling using non-serial dyanamic programming. *European Journal of Operational Research*, *78*(2), 162-177.

Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, *4*(1), 238-252.

Bilir, M. (2015). *A Mixed integer programming method for Pareto front optimization of discrete time cost trade-off problem* (Master's thesis). Middle East Technical University, Ankara.

Burgess, A. R., & Killebrew, J. B. (1962). Variation in activity level on a cyclical arrow diagram. *Journal of Industrial Engineering*, *13*(2), 76-83.

Butcher, W. S. (1967). Dynamic programming for project cost-time curves. *Journal of the Construction Division, Proceedings of ASCE*, *93*(1), 59-74.

Chan, W. T., Chua, D. K., & Kannan, G. (1996). Construction resource scheduling with genetic algorithms. *Journal of Construction Engineering and Management*, *122*(2), 125-132.

Chassiakos, A. P., & Sakellaropoulos, S. P. (2005). Time-cost optimization of construction projects with generalized activity constraints. *Journal of Construction Engineering and Management*, *131*(10), 1115-1124.

Chassiakos, A. P., & Sakellaropoulos, S. P. (2005). Time-cost optimization of construction projects with generalized activity constraints. *Journal of Construction Engineering and Management*, *131*(10), 1115-1124.

Colorni, A., Dorigo, M., & Maniezzo, V. (1991). Distributed optimization by ant colonies. In *Proceedings of the first European conference on artificial life* (Vol. 142, pp. 134-142).

Crowston, W., & Thompson, G. L. (1967). Decision CPM: A method for simultaneous planning, scheduling, and control of projects. *Operations Research*, *15*(3), 407-426.

De, P., Ghosh, J. B., & Wells, C. E. (1992). Heuristic Estimation of the Efficient Frontier for a Bi-Criteria Scheduling Problem. *Decision Sciences, 23*(3), 596-609.

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

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

Demeulemeester, E., Herroelen, W., & Elmaghraby, S. E. (1993). An optimal procedure for the discrete time/cost trade-off scheduling problem in project networks. In *Working Paper# 9337*. Department of Applied Economics, Katholieke Universiteit Leuven.

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

Demeulemeester, E., De Reyck, B., Foubert, B., Herroelen, W., & Vanhoucke, A. M. (1998). New computational results on the discrete time/cost trade-off problem in project networks. *Journal of the Operational Research Society*, 1153-1163.

Easa, S. M. (1989). Resource leveling in construction by optimization. *Journal of Construction Engineering and Management*, *115*(2), 302-316.

Eberhart, R. C., & Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Proceedings of the sixth international symposium on micro machine and human science* (Vol. 1, pp. 39-43).

Elbeltagi, E., Hegazy, T., & Grierson, D. (2005). Comparison among five evolutionary-based optimization algorithms. *Advanced Engineering Informatics*, *19*(1), 43–53.

El-Rayes, K., & Jun, D. H. (2009). Optimizing resource leveling in construction projects. *Journal of Construction Engineering and Management*, *135*(11), 1172-1180.

Feng, C.W., Liu, L., & Burns, S. A. (1997). Using Genetic Algorithms to Solve Construction Time-Cost Trade-Off Problems. *Journal of Computing in Civil Engineering*, *11*(3), 184–189.

Gather, T., Zimmermann, J., & Bartels, J. H. (2011). Exact methods for the resource leveling problem. *Journal of Scheduling*, *14*(6), 557-569.

Geng, J. Q., Weng, L. P., & Liu, S. H. (2011). An improved ant colony optimization algorithm for nonlinear resource-leveling problems. *Computers & Mathematics with Applications*, *61*(8), 2300-2305.

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

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

Guo, Y., Li, N., & Ye, T. (2009). Multiple resources leveling in multiple projects scheduling problem using particle swarm optimization. *2009 Fifth International Conference on Natural Computation*, Tianjin, 2009, (pp. 260-264).

GUROBI Optimizer Reference Manual (2016). Retrieved from www.gurobi.com/documentation/6.5/refman.pdf, last accessed on June 2016.

GUROBI 6.5 Performance Benchmarks (2015). Retrieved from www.gurobi.com/pdfs/benchmarks.pdf, last accessed on June 2016.

Harris, R. (1978). *Precedence and arrow networks for construction*. New York: John Wiley and Sons.

Harris, R. B. (1990). Packing method for resource leveling (PACK). *Journal of Construction Engineering and Management*, *116*(2), 331-350.

Hazır, Ö., Haouari, M., & Erel, E. (2010). Discrete time/cost trade-off problem: A decomposition-based solution algorithm for the budget version. *Computers & Operations Research*, *37*(4), 649-655.

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

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

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

Heon Jun, D., & El-Rayes, K. (2011). Multiobjective optimization of resource leveling and allocation during construction scheduling. *Journal of Construction Engineering and Management*, *137*(12), 1080-1088.

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

Hiyassat, M. A. S. (2000). Modification of minimum moment approach in resource leveling. *Journal of Construction Engineering and Management*, *126*(4), 278-284.

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.

Hossein Hashemi Doulabi, S., Seifi, A., & Shariat, S. Y. (2010). Efficient hybrid genetic algorithm for resource leveling via activity splitting. *Journal of Construction Engineering and Management*, *137*(2), 137-146.

Kandil, A., & 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.

Karshenas, S., & Haber, D. (1990). Economic optimization of construction project scheduling. *Construction Management and Economics*, *8*(2), 135-146.

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

Kolisch, R., Schwindt, C., & Sprecher, A. (1999). Benchmark instances for project scheduling problems. In *Project Scheduling* (pp. 197-212). Springer US.

Land, A. H., & Doig, A. G. (1960). An automatic method of solving discrete programming problems. *Econometrica: Journal of the Econometric Society*, 497-520.

Leu, S. S., Yang, C. H., & Huang, J. C. (2000). Resource leveling in construction by genetic algorithm-based optimization and its decision support system application. *Automation in construction*, *10*(1), 27-41.

Li, H., & Love, P. (1997). Using improved genetic algorithms to facilitate time-cost optimization. *Journal of Construction Engineering and management*, *123*(3), 233-237.

Liberatore, M. J., Pollack-Johnson, B., & Smith, C. A. (2001). Project management in construction: Software use and research directions. *Journal of Construction Engineering and Management*, *127*(2), 101-107.

Liu, L., Burns, S. A., & Feng, C. W. (1995). Construction time-cost trade-off analysis using LP/IP hybrid method. *Journal of Construction Engineering and Management*, *121*(4), 446-454.

Martinez, J., & Ioannou, P. (1993). Resource leveling based on the modified minimum moment heuristic. In *Computing in Civil and Building Engineering (1993)* (pp. 287-294). ASCE.

Mason, A. T., & Moodie, C. L. (1971). A branch and bound algorithm for minimizing cost in project scheduling. *Management Science*, *18*(4-part-i), B-158.

Mattila, K. G., & Abraham, D. M. (1998). Resource leveling of linear schedules using integer linear programming. *Journal of Construction Engineering and Management*, *124*(3), 232-244.

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

Mutlu, M. Ç. (2010). *A branch and bound algorithm for resource leveling problem* (Master's thesis). Middle East Technical University, Ankara.

Neumann, K., & Zimmermann, J. (2000). Procedures for resource leveling and net present value problems in project scheduling with general temporal and resource constraints. *European Journal of Operational Research*, *127*(2), 425-443.

Ng, S. T., & Zhang, Y. (2008). Optimizing Construction Time and Cost Using Ant Colony Optimization Approach, *134*(September), 721–728.

Panagiotakopoulos, D. (1977). A CPM time-cost computational algorithm for arbitrary activity cost functions. *INFOR: Information Systems and Operational Research*, *15*(2), 183-195.

Pang, N., Shi, Y., & You, Y. (2008). Resource leveling optimization of network schedule based on particle swarm optimization with constriction factor. In *Advanced Computer Theory and Engineering, 2008. ICACTE'08. International Conference on* (pp. 652-656). IEEE.

Patterson, J. H. (1984). A comparison of exact approaches for solving the multiple constrained resource, project scheduling problem. *Management Science*, *30*(7), 854-867.

Petrović, R. (1968). Optimization of resource allocation in project planning. *Operations Research*, *16*(3), 559-568.

Project Management Institute. (2013). *A guide to the project management body of knowledge: (PMBOK guide)*. Newtown Square, Pa: Project Management Institute.

Reda, R., & Carr, R. I. (1989). Time-cost trade-off among related activities. *Journal of Construction Engineering and Management*, *115*(3), 475-486.

Rieck, J., Zimmermann, J., & Gather, T. (2012). Mixed-integer linear programming for resource leveling problems. *European Journal of Operational Research*, *221*(1), 27-37.

Robinson, D. R. (1975). A dynamic programming solution to cost-time tradeoff for CPM. *Management Science*, *22*(2), 158-166.

Roca, J., Pugnaghi, E., & Libert, G. (2008). Solving an extended resource leveling problem with multiobjective evolutionary algorithms. *International Journal of Computational Intelligence*, *4*(4), 289-300.

Schwindt, C. (1995). ProGen/max: A new problem generator for different resource-constrained project scheduling problems with minimal and maximal time lags.

Senouci, A. B., & Eldin, N. N. (2004). Use of genetic algorithms in resource scheduling of construction projects. *Journal of Construction Engineering and Management*, *130*(6), 869-877.

Siemens, N. (1971). A Simple CPM Time-Cost Tradeoff Algorithm. *Management Science*, *17*(6), B–354–B–363.

Son, J., & Mattila, K. G. (2004). Binary resource leveling model: Activity splitting allowed. *Journal of Construction Engineering and Management*, *130*(6), 887-894.

Son, J., & Skibniewski, M. J. (1999). Multiheuristic approach for resource leveling problem in construction engineering: Hybrid approach. *Journal of Construction Engineering and Management*, *125*(1), 23-31.

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

Szmerekovsky, J. G., & Venkateshan, P. (2012). An integer programming formulation for the project scheduling problem with irregular time–cost tradeoffs. *Computers & Operations Research*, *39*(7), 1402-1410.

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

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

Wagner, H. M., Giglio, R. J., & Glaser, R. G. (1964). Preventive maintenance scheduling by mathematical programming. *Management Science*, *10*(2), 316-334.

Xiong, Y., & Kuang, Y. (2006). Ant colony optimization algorithm for resource leveling problem of construction project. *Advancement of Construction Management and Real Estate*, *1*, 212-226.

Yang, I. T. (2007). Performing complex project crashing analysis with aid of particle swarm optimization algorithm. *International Journal of Project Management*, *25*(6), 637-646.

Yeniocak, H. (2013). *An efficient branch and bound algorithm for the resource leveling problem* (Master's thesis). Middle East Technical University, Ankara.

Zhang, H., & Li, H. (2010). Multi-objective particle swarm optimization for construction time-cost tradeoff problems. *Construction Management and Economics*, *28*(1), 75-88.

Zhang, Y., & Thomas Ng, S. (2012). An ant colony system based decision support system for construction time-cost optimization. *Journal of Civil Engineering and Management*, *18*(4), 580-589.

Zhang, L., Zou, X., & Qi, J. (2015). A trade-off between time and cost in scheduling repetitive construction projects. *Journal of Industrial and Management Optimization*, *11*(4), 1423-1434.

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

Zheng, D. X., Ng, S. T., & Kumaraswamy, M. M. (2003, March). GA-based multi-objective technique for multi-resource leveling. In *Construction Research Congress: Wind of Change: Integration and Innovation* (pp. 1-8). ASCE.

Zheng, D. X., Ng, S. T., & Kumaraswamy, M. M. (2004). Applying a genetic algorithm-based multiobjective approach for time-cost optimization. *Journal of Construction Engineering and Management*, *130*(2), 168-176.