Multi-Project Multi-Mode Resource Constrained Scheduling Problem with Material Ordering

A Thesis

In

The Department

Of

Mechanical, Industrial and Aerospace Engineering (MIAE)

Presented in Partial Fulfilment of Requirements

For the Degree of Master of Applied Science at

Concordia University

Montreal, Quebec, Canada

May, 2018

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Yasser Ghamary

Entitled: Multi Project Multi-Mode Resource Constrained Scheduling F

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

Master of Applied Science

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

Signed by the final examining committee:

______________________ Chair
Dr. Wen-Fang Xie
______________________ Examiner
Dr. Onur Kuzgunkaya
______________________ Examiner
Dr. Zhi Chen
______________________ Thesis Supervisor(s)
Dr. Mingyuan Chen

____

Approved by ______________________
Chair of Department or Graduate Program Director

____
Dean,

Date ______________________

Abstract

In Multi–mode Project Scheduling with Resource Constrained (MPSRCP), activities are sequenced under resource limitation. In this thesis, an extension of the problem is considered. Multi-project multi-mode resource constrained scheduling problem with material ordering is studied. Bonus and penalty are taken into account in solving the considered problem as it is the case in many different industries. A literature review is presented and various solution methods for solving the considered and similar problems are studied. A new mathematical model is proposed considering a multi-project version of the problem. A new decomposition based heuristic to solve the problem is developed in this thesis. The approach is to use three separated mathematical models for each part of the problem. The developed heuristic is examined using various example problems with different features and randomly generated data. It can generate close-to-optimal solutions for all tested example problems with much reduced computational time when off-shelf optimization software was used. The developed math model and heuristic method are applied to a larger size case study based on a practical system in a manufacturing company in northern Ontario.

Acknowledgment

First of all, I want to thank God which I believe without his will nothing is possible**.** He was my comfort in all the hardships.

I want to show my gratitude to the following people who helped me and shared beautiful moments with me during my study.

I would like to thank Dr. Mingyuan Chen, my supervisor, his support and understanding were beyond my expectations and I never forget that. I could not ask for a better supervisor and working with him was a pleasure.

I also would like to thank my dearest friends Hamidreza Rezaei, Keivan K Maram and Raham Radmanesh for their help during my study and sharing their knowledge and experience with me.

I would love to express my gratitude to my dear lab partners Jaeir, Cesar, Omar, Armaghan, Niloofar, Bahman and Zenan which made my study a pleasant time that I will never forget.

After that, I would love to appreciate my lovely parents, which without their support I was not able to follow my dreams. I also cannot forget my brothers, Dr. Yassin Ghamary and Sepehr which supported me during all this time and shared their knowledge and experience with me.

Table of Contents

Abstract .... ***ii***

Acknowledgment .... ***iii***

CHAPTER 1: INTRODUCTION AND MOTIVATION ***7***

1.1 Resource Constrained Multi-Project Scheduling Problem (RCMPSP) 7

1.2 Challenges and Motivation 8

1.3 Contribution 9

1.4 Outline of the thesis 9

CHAPTER 2: LITERATURE REVIEW ***10***

2.1 Introduction 10

2.2 Single-Project Scheduling problem 10

2.2.1 Makespan minimization 11

2.2.2 Cost minimization 14

2.2.3 Other objectives 15

2.3 Multi-Project Scheduling Problem 18

2.3.1 Makespan minimization 18

2.3.2 Tardiness minimization 20

2.3.3 Other objectives 21

CHAPTER 3: PROBLEM DEFINITION ***26***

3.1 Introduction 26

3.2 Definitions 28

3.2.1 Resources 28

3.2.2 Holding cost 29

3.2.3 Ordering cost and Purchase cost 29

3.2.4 Inventory related 30

3.2.5 Bonus and penalty 30

3.3 Assumptions 31

3.4 Mathematical model 32

3.4.1 Decision variables 32

3.4.2 Objective function 33

3.4.3 Notations 33

3.4.4 Model 35

CHAPTER 4: SOLUTION METHODOLOGY ***38***

4.1 Challenges and Approach 38

4.2 Developed Heuristic 39

4.2.1 First phase 39

4.2.2 Second phase: 44

4.2.3 Third phase 46

4.3 Results 47

4.3.1 Data description 47

4.3.2 Worked example 48
4.5.2 Experimental results 56
4.5.3 Case study 64
4.6 Summary 67
CHAPTER 5: CONCLUSION AND FUTURE WORK 68
Appendix i 70
Appendix ii 75
Appendix iii 78
Appendix iv: 81
References 91

List of Tables and Figures

Figure 2.1- Renewable and non-renewable resources matrix (Van Peteghem and Vanhoucke, 2011) 14
Figure 3.1 Project networks and information 27
Figure 4.1 Graph of objective function 40
Figure 4.2 Project Networks considered in three-project example 42
Table 4.1 Resource Average usage for each resource in each project 42
Table 4.2 Maximum Resource usage for each resource in each project 43
Table 4.3 Maximum Resource usage in for new combination 43
Table 4.4 Maximum Resource usage for new combination 43
Table 4.5 Resource Distribution between projects 44
Figure 4.3 Network and Information for instance 15 49
Figure 4.4 Projects Network and Information in instance 15 for optimization 50
Table 4.6 Renewable resource distribution in the optimal solution for instance 15 50
Table 4.7 Start time and mode for each activity in optimal solution for instance 15 51
Table 4.8 Results for instance 15 in optimal solution 51
Table 4.9 Resource Average usage by each renewable resource in instance 15 52
Table 4.10 Renewable resources distribution for instance 15 in heuristic method 52
Figure 4.5 Network and information for the second phase of the heuristic in the absence of nonrenewable resources in instance 15 53
Table 4.11 Start time and mode for each activity in the heuristic method for instance 15 53
Table 4.12 Nonrenewable resources requirement for each time for instance 15 in the heuristic method 54
Table 4.13 Results for instance 15 in phase three of the heuristic method 55
Table 4.14 Costs for the first instance for the main mode and the heuristic 55

Figure 4.6 Resource 1 allocation for the first project 57
Figure 4.7 Resource 2 allocation for the first project 57
Figure 4.8 purchase cost for instances 18 to 22..... 58
Figure 4.9 Order cost 59
Figure 4.10 Holding cost..... 60
Figure 4.11 Bonus..... 60
Figure 4.12 Penalty..... 61
Figure 4.13 Total cost of each project..... 61
Table 4.15 Cost for each project..... 62
Table 4.16 Time spend to solve each instance 63
Figure 4.14 Project 1 network 64
Table 4.17 time spent to solve project 1 with 327 activities in the second phase of the heuristic 65
Table 4.18 time spent to solve project 2 with 278 activities in the second phase of the heuristic 65
Table 4.19 Cost for schedule with two modes 66
Table 4.20 Cost for schedule with doubled order cost..... 66

CHAPTER 1: INTRODUCTION AND MOTIVATION

In this chapter, Resource Constrained Multi-Project Scheduling Problem (RCMPSP) is described. The challenges of the problem are discussed. The approaches to tackle the problem and the main contributions of the thesis are provided. In the end, an overview of the thesis outline is given.

1.1 Resource Constrained Multi-Project Scheduling Problem (RCMPSP)

As projects are getting more common structures for organizing work in modern companies, issues involving simultaneous management of multiple projects (or a portfolio of projects) are attracting more attention. Based on Payne (1993), up to 90% of the value of all projects results from multi-project context. This fact shows the importance of this topic and an enormous benefit can be achieved by even a small improvement in their management. Managers of multiple projects with overly constrained resources face the challenge of allocating resources to projects in order to minimize the total duration of projects or the average delay per project, Browning and Yassine (2010). The basic of the RCMPSP is to prioritize activities in order to optimize a predefined objective function.

Sometimes, activities can operate in different modes. In each mode, the duration of each task is a function of the level and type of resources committed to it, Zapata et al. (2008). This arises the need for a new extension of the problem which is called multi-mode resource constrained multi-project scheduling problem.

1.2 Challenges and Motivation

Multi-project management is an important area of management in both manufacturing and business. Most of the approaches in the literature address a multi-project environment which projects can share their resources from a common pool. This policy makes it possible to create a general network for all projects combined together and treat them as a single large project, Besikci et al. (2012). In some cases, it is not possible to have a shared pool of resources for projects. For instance, in a case which projects are geographically far from each other, it is not possible to use this policy. In such cases, resource dedication (RD) policy is used. In this policy, the resources are dedicated to the individual project throughout the duration of the project. In these cases, multi-project environment becomes different and presents many new challenges.

Research conducted on multi-project scheduling, generally focus on minimizing the duration of projects. Another aspect that can affect the schedule is cost. Project scheduling problems are often extended in order to deal with more realistic cases. Bonus and penalty are two well-known terms which can be seen in some problems. A Penalty is paid when the project is finished after its due date and a bonus is allotted when the project is finished before the due date.

Additionally, nonrenewable resources can add another complexity to the problem. Traditionally, the nonrenewable resources, known as materials, need to be available at the beginning of the project which leads to higher holding cost. This can be solved by taking into account material ordering as a decision tool alongside the scheduling process in order to balance the holding and order cost related to nonrenewable resources. A schedule which considers all these costs can be closer to practice, Zoraghi et al. (2017).

1.3 Contribution

This research proposes a mathematical model for Multi-project Scheduling with material ordering which is an extended version of the model in Zoraghi et al. (2017). In our model, new constraints regarding inventory size and delivery date of each nonrenewable resource are considered and a new term considering the purchase cost is added to the objective function. The objective of the model is to minimize the total cost of all projects, including holding cost, order cost, purchase cost and penalty and. The model is tested and validated using numerical examples with data generated by RanGen1 generator.

A new heuristic is introduced to solve larger instances. The heuristic includes three phases. Each phase solves a small part of the problem and provides the data for the next phase. The approach is tested on 22 small instances generated by RanGen1 and the result are compared with optimal schedules generated using off-shelf optimization software. Also, a case study is used to test the method in a real case.

1.4 Outline of the thesis

In the next chapter, the Resource Constrained Scheduling Problem (RCSP) research literature is reviewed. In Chapter 3 a new mathematical model is presented to describe Multi-Project Resource Constrained Scheduling with Material Ordering (MPMRCSMO). A new heuristic is introduced to solve the problem, in Chapter 4, and the results are shown. Chapter 5 presents the summary and conclusion of the thesis.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

Various mathematical and optimization models have been developed to solve different types of Resource Constrained Project Scheduling Problem (RCPSP) since modeling and optimization are effective tools in solving RCPSP and similar problems. In practical applications, however, further model and solution method development are required to address different practical issues. As a result, RCPSP has been a starting phase for many researchers to develop more general project scheduling problems, Hartmann and Brickorn (2010). A large number of variants and extensions of RCPSP models have been studied as can be found in the literature with respect to model objective functions, constraints, characteristics etc. in this chapter, a literature review related to RCPSP modeling and solution methodologies is presented into two main sections. The first section covers single-project problems and different solution methods for solving the problem. The second section gives a review on approaches taken to model and solve multi-project scheduling problems.

2.2 Single-Project Scheduling problem

In this section, different versions of single-project scheduling model are discussed with emphasis on various model objective functions. The main objectives of single project scheduling are makespan minimization and cost minimization. Other objectives such as net present value, penalty minimization etc. are also considered.

2.2.1 Makespan minimization

Duration of a project, in some industries, can be a crucial factor for everyone involved in the project. Many researchers have tried to minimize the total duration of a project using various approaches. Mingozzi et al. (1998) is suggesting an exact algorithm to solve the classical RCPSP. A new and 0-1 linear programming formulation requiring an exponential number of variables is presented to describe the problem. The model is aiming to minimize makespan of project. The formulation is relaxed in different ways to drive new lower bounds for the value of longest path on precedence graph. Based on the formulation, a tree search algorithm is defined which uses the new lower bounds to reach to the optimal solution.

In practice, activities might have more than one execution mode. One of the important extensions of RCPSP is Multi-Mode Resource Constrained Project Scheduling Problem (MRCPS). When different modes are introduced to each activity in a way that in each mode, activities' duration and resource requirement are different, the problem is generalized to MRCPSP, Van Peteghem and Vanhoucke (2014).

Slowinski (1980) has proposed a model considering multi-mode activities and distinguished resources into three categories of renewable resources (e.g. labor, machines), nonrenewable resources (e.g. materials) and doubly constrained resources (e.g. cash-flow per time-unit). Preemption for activities is allowed and there is no penalty associated with it. The model assumes activities preemption is arbitrary and they can restart again later. As a solution, Slowinski has introduced two approaches, both using linear programming. First one is a one-stage approach that was mainly used to solve single-mode project scheduling. The second one is a two-stage approach which the results from the first LP model are the input data for the second stage to reach to the optimal schedule.

In Drexl and Gruenewald (1993), a mathematical formulation for multi-mode and non-preemptive activities is proposed. In this model, all the three resources (renewable, nonrenewable and doubly-constrained) are considered. The objective function is minimizing makespan of project. Furthermore, an extension of the problem is considered in which job-specific resource profiles varies with time. In this case, resource usage of activities are not constant during their running time. A mathematical formulation is given for this extension. As the solution methodology, a stochastic scheduling method is presented which solve problems to sub-optimality.

Boctor (1993) only discusses renewable resources. Preemption in activities is not allowed and the objective is to minimize makespan. 21 heuristics are developed and a comparison between them is made. All the heuristics are tested on 240 instances which are divided into two groups of 50 and 100 activities. Boctor in his later work, Boctor (1996), proposes a new simulated annealing algorithm to deal with single and multi-mode activities. He proposes a formulation which considers different objective functions such as minimization of project duration, minimization of project cost and maximization of project net-present-value.

Mika et al. (2008) introduce scheduling with schedule-dependent setup times. The schedule can affect setup times for each activity. Depending on the schedule, required resources for an activity could be in different places. In this situation, the time required to prepare an activity to start is longer. A model is presented considering only renewable resources and aims to minimize the duration of project. A number of resource locations are defined and each activity should start exactly in one location. Mika has used a tabu search as the solution methodology and reached a better solution than multi-start iterative improvement method and random sampling.

A time/resource trade-off is discussed by Ranjbar et al. (2009). Each activity has a work content which should be satisfied with a combination of resources. Multiple resource type of discrete time/resource trade-off problem (MDTRTP) is considered here. Duration of each activity differs according to the discrete non-increasing function of the number of renewable resources dedicated to it. A mathematical formulation with the objective of minimizing makespan is presented. A hybrid scatter search is used as the solution methodology of this problem.

The scarceness of resources is taken into account by Van Peteghem and Vanhoucke (2011). Scarceness in renewable and nonrenewable resources leads towards different difficulties is scheduling. While the shortage of renewable resources increases the variation from the critical path, scarceness of nonrenewable resources cause smaller feasible area in mode selection. Each problem has a different characteristic regarding resource scarcity, as a result, various behaviors. A resource scarceness matrix is introduced which divides the scarceness into four areas. The matrix is shown in Figure 2.1 and considers renewable and nonrenewable resources. The partitions one and four include the lowest and highest scarceness respectively. The paper proposes a scatter search algorithm which uses different improvement methods based on the scarceness value of the problem to find a feasible schedule in a way that the makespan of the project is minimized.

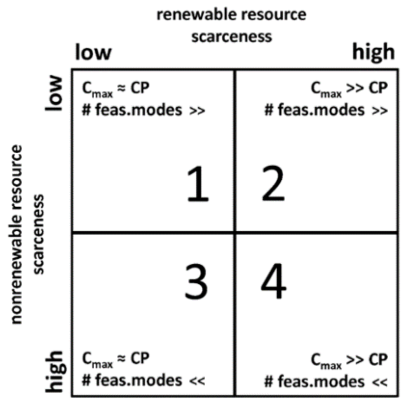


Figure 2.1- Renewable and non-renewable resources matrix (Van Peteghem and Vanhoucke, 2011)

There are three improvement methods specified to the characteristic of problems. Additionally, two local searches are used and the results are promising.

2.2.2 Cost minimization

Minimizing the cost of a project has always been an important issue for managers. Renewable resources are one of the main aspects of project scheduling and play a dominant rule in the cost of a project. Resource availability cost problem (RACP) is presented by Demeulemeester (1995). It is a single project scheduling problem and the objective is to minimize the cost of assigning resources to the project by deciding the resource availability levels. The basic of the problem is similar to RCPSP with some differences in objective function and constraints related to project duration.

Another extension of MRCPSP is discussed in Salewski et al. (1997). The non-preemptive variant of a resource constrained project scheduling problem with mode identity is considered. The problem referred to as Mode Identity Resource Constrained Project Scheduling Problem (MIRCPSP). The problem is defined by precedence constraints and renewable and nonrenewable resources. Set of all jobs is divided into disjoint subsets and all the jobs existing in a subset should be executed in the same mode. For each job, a release date and deadline are considered and the objective is to minimize overall costs. A tailored parallel randomized approach is presented as the solution methodology which is called RAMES and uses both static and dynamic priority rules.

2.2.3 Other objectives

Apart from makespan and cost, there are other objectives in scheduling a project. Another aspect of the problem is the financial part and it appears when, generally, a series of cash flows (positive and/or negative) occur during the time horizon of the project. In Mika et al. (2005) a problem called Multi-Mode Resource Constrained Project Scheduling Problem with Discounted Cash Flows (MRCPSPDCF) is proposed. Only positive cash flows are considered and assigned to each activity. A mathematical model is developed and the objective is to maximize net present value (NPV). The model is from contractor's point of view and four different payment methods are considered. These methods are lump-sum payment at the completion of the project (LSP), payments at activities' completion time (PAC), payments at equal time intervals (ETI) and progress payments (PP). Apart from any of the payment methods, the sum of all payments is the same and equal to the sum of the cash flows of all the project's activities. In the presented model, time value of money is considered. Money which is received today is more valuable than the money received in the future. For that, a discount rate α is considered. The model tries to find the best schedule and mode assignment by using simulated annealing and tabu search.

In Elloumi and Fortemps (2010), both renewable and nonrenewable resources are considered in a multi-mode project. Two new ideas are investigated. The model of single objective MRCPSP is modified to a bi-objective model which is capable of dealing with the potential violation of nonrenewable resource constraints. A penalty function is assigned to the nonrenewable resources violation and the penalty should be minimized. Allowing non-renewable resource violation expands the solution set and simplifies the evolutionary algorithm which is used to solve the problem.

Peteghem and Vanhoucke (2010) introduce the preemptive extension of the problem. Activities can split and restart another time. It is called Preemptive Multi-Mode Resource Constrained Project Scheduling Problem (PMRCPSP). A bi-population genetic algorithm is applied to solve the problem. The paper tests the impact of preemption on schedule's quality and compares the results with the results from the non-preemptive problem.

Ghoddousi et al. (2013) try to model the problem to minimize three objective functions simultaneously. Objectives are project's total time, project's total cost, and resource leveling (by minimizing resource moment deviation on the project). A Multi-mode Resource Constrained Discrete Time-Cost-Resource Optimization (MRC-DTCTP) model is introduced by the paper. The model is a result of integrating previous ideas in the literature and aims to find the starting time and execution mode of each activity in a discrete time horizon. To minimize the total duration, the finish time of the last activity will be minimized. The project cost is calculated by summing direct and indirect costs. Direct cost is a cost coming from the mode assignment of each activity and indirect cost is assumed to be fixed in all time periods. But, for the whole project, the indirect cost varies based on the total makespan. A penalty function is considered, which is effective if the project takes longer than the contract. As the third objective, the model is using squared deviation (SD) to evaluate the deviation of the resource usage from a given profile in a resource histogram. Due to having different objectives with different characteristics and sometimes confliction, A Pareto based multi-objective genetic algorithm (NSGA-II) approach is used to determine Pareto front solutions. The solutions are considered by a decision maker and the best one is chosen based on the situation. The manager can give preferences to time, cost and resource fluctuation in order to have sorted non-dominated solutions.

In today world, for many companies, it is crucial to meet the compromised due dates. A minimum and a maximum time lag between activities in the MRCPSP is considered by Prez et al. (2014) and the objective function of the problem is to minimize project tardiness and temporal constraints infeasibilities. Each activity has a due date and needs to be finished before that. Minimum time lag means an activity cannot start before than start time of another activity and maximum time lag means an activity cannot start after than finish time of another activity. For implementing time lags a generalized precedence relationships (GPRs) is used. Start-start (SS), start-finish (SF), finish-start (FS) and finish-finish (FF) are the notations used to show the relations. The mathematical model is a multi-objective model aiming to minimize tardiness and GPRs infeasibilities. The GPR constraints belonging to strong components can be violated to improve the due dates. A multi-objective evolutionary algorithm is proposed to solve the problem. The algorithm is basically a genetic algorithm. Two local searches are used to improve each objective separately. In the algorithm, all the precedence relationships can be violated but at the end, it will be repaired. Finally, a Pareto front of solution is produced which is a trade of between tardiness and temporal infeasibilities. A decision maker chose the best solution among the Pareto optimal solutions.

An extensive survey on single project in which activities can be processed using a finite and infinite number of modes can be found in Weglarz et al. (2011). In this survey problems with single objective are considered.

2.3 Multi-Project Scheduling Problem

Another extension of resource constrained scheduling problem is multi-project scheduling. Based on Geiger (2017), this situation happens when a set of several projects independent of the others, with respect to their activities, should be scheduled together. The problem has the same constraints set as the classical RCPSP, but the difference is that at least one or two resources are shared between all projects. Pritsker et al. (1969) was one of the earliest research on multi-project scheduling. A mathematical model was suggested including most of the aspects of the problem. The main objectives of multi-project scheduling are minimization of makespan and tardiness over all projects. There are other objectives considered by authors. This section is divided into three subsections. These subsections are makespan minimization, tardiness minimization, and other objectives.

2.3.1 Makespan minimization

Similar to single-project scheduling, makespan minimization is a common objective in multi-project scheduling. The approach used in Sperenza and Vercellis (1993) to solve the problem is a two-stage approach. In their approach precedence relationship between projects can exist. Projects are defined as activities with multi-mode in the first stage. Each mode, in this step, is defined by solving a mathematical model for the project with budget limitation. The mathematical model chooses the finish time of projects. It is assumed that different budget estimation for projects can lead to different modes. The objective of the first stage is to maximize the net present value. The outcome of the first phase are start and finish time of each project and the total renewable and nonrenewable resource capacities that a project can use in a given period. The information coming from the first stage is used to schedule each individual project in the second stage with the objective of makespan minimization.

In the real world, when the same resources are shared between different projects or activities, transfer time can play a rule. To fill this gap, Kruger and Scholl (2009) study another resource management policy called resource sharing with sequence dependent transfer times. In this problem, when a resource is shared between projects or different activities of same project, a transfer time is considered. The objective of single project is minimizing the duration of the project and for the multi-project is minimizing the mean project duration.

Toffolo et al. (2016) propose a time-indexed model for the problem. The objective function of the model is to minimize the sum of completion times for projects and completion time of the last project. In the first step of the solution methodology, an initial feasible mode assignment is constructed by using an IP heuristic based on decomposition. Authors have used a hybrid algorithm with several IP-based components for MRCMPSP. These components are mode-selection IP model, IP constructive algorithm, forward-backward improvement (FBI) procedures, IP local search algorithm, and biased rebuild solution algorithm.

In some real-world applications, activities might not have a constant resource usage during their executing time. Resource-constrained project scheduling problem with flexible profiles (FRCPSP) is introduced by Naber and Kolisch (2014) to deal with such cases. Four different discrete-time model formulations are proposed, each one with the objective of makespan minimization. All the models are solved using CPLEX and compared to each other. A new classification of resources is presented in this paper and resources required by an activity are categorized into three general type: principal resource, dependent resource, and independent resource.

2.3.2 Tardiness minimization

Tardiness minimization is an objective which is more dominant in multi-project environments. Researchers have approached this objective in different ways. Kurtulus and Narula (1985) analyze only single mode activities with the objective of minimizing tardiness cost performance of projects. In their work, each project has a different weight. Project networks are characterized by factors such as maximum load factor and average utilization factor. Penalty is defined for projects with different functions based on total work content and critical path.

Resource pricing is discussed by Lawrance and Morton (1993). The pricing is based on priority heuristic rules for multi-project scheduling. In their works, like Kurtulus and Narula (1985), each project has a weight based on the relative importance of the project. The objective of the authors' work is to minimize the total weighted tardiness cost of projects. Different approaches are chosen to estimate resource price and some heuristics are developed based on the estimation.

Resource dedication policy is defined as assigning a set of limited resources to multiple projects in a way that each project cannot exceed the number of assigned resources. In Besiksi et al (2013), a multi-project environment with resource dedication policy is considered. A mathematical model with the objective of minimizing total weighted tardiness cost over all projects is presented. Two solution approaches are suggested by authors. First approach is a genetic algorithm with a new local improvement heuristic called combinatorial auction and the Second one is Lagrangian relaxation. The problem is solved in two phases. In the first phase, the number of resources dedicated to each project is decided and after that, the schedule is constructed in the second phase. A new extension of RCPSP which is Combinatorial Multi-mode Resource Constrained Multi-Project Scheduling Problem (CMRCMPSP) is presented by Pinha et al (2016). This model is more general and capable of dealing with RCPSP, MRCPSP, RCMPSP, and MRCMPSP. The idea behind this problem is that the multiple modes for each activity should fulfill the required work content, whether there are the same resources or not. Meaning that a mode is no longer a set of resources only, but a set of combinatorial subsets of required resources capable of conducting a given task. The model which is proposed for the CMRCMPSP in nonlinear due to nonlinear constraints. The objective function is to minimize the total tardiness of all projects and total cost. This model is designed to deal with the real-world problems, as a result, it makes a fewer assumption to get closer to the reality. Simulation is used for this study and project manager is an integral part of the resource allocation process. The project manager uses a software tool named STREAM in order to analyze data and provide the appropriate input for the model. A ship repair and maintenance company is used as a case study.

Besikci et al. (2015) include budget limitation into a multi-project environment with resource dedication (RD) policy. Each project has a due date and the objective function is to minimize the total tardiness of all projects. The first step is to determine the amount of budget assigned to each specific resource. In the next step, the number of resources dedicated to each project is decided. The final step is to schedule activities. Two solution approaches both based on genetic algorithm are proposed.

2.3.3 Other objectives

In multi-project scheduling, shared pool resources is a policy in which, all projects use the same pool of resources. Browning and Yassine (2010) consider a project portfolio with shared pool resources. The problem contains single-mode activities. A mathematical model is proposed and five different objectives is suggested: total delay, average delay, average percentage delay, total portfolio delay and portfolio percentage delay. All the objectives are trying to minimize the delay in different shapes. Three of them aim to minimize the delay for each project separately while the other focus on the whole portfolio. 20 different priority rules are used to establish a schedule which is acceptable. Four characteristics of RCMPSP (objective function, network complexity, resource distribution and resource contention) are considered to create various types of problems and test each PR's functionality. The study has used 616 test problems with 20 replications and using 20 PRs on every one of them resulting in 12320 problems. The result shows the best PR varies from problem to problem and is dependent on the objective function, whether it is from the perspective of the project manager or the portfolio manager.

There are a lot of organizations dealing with multiple projects running simultaneously. These kind of organizations are usually capacity driven. An important aspect here is capacity planning. Gadmann and Schutten (2005) propose a mathematical formulation in which aspects such as capacity flexibility, precedence relations between work packages, and maximum work content per period can be taken into account. The approach is using two planning levels for scheduling which is proposed by De Boer (1998). The first level in known as Rough-Cut Capacity Planning (RCCP) and the second level is Resource Constrained Project Scheduling Problem (RCPSP). RCCP concerns medium-term capacity planning problem. It is assumed that the time horizon is divided into weekly portions instead of being continues. Jobs have work content for several weeks. In the second planning level which is RCPSP, jobs are subdivided into several activities to be scheduled. In each portion of time, a fraction of jobs is completed. It is assumed that all resources spend an equal fraction on jobs in a specific time portion. To complete each job their work content should be satisfied. There is a regular capacity of hours available for each resource and non-regular capacity (sub-contracting, working overtime, hiring etc.) is allowed too. Preemption is allowed and the mathematical model is presented with the objective function of minimizing total cost of required non-regular capacity. For maintaining the feasibility of the schedule an Allowed to Work (ATW) window is introduced. For example, an ATW (S_j, C_j) for job J_j means this job cannot be worked on before week S_j or after week C_j. Three heuristics based on linear programming is proposed to solve the problem.

In some industries, decisions are made in the presence of uncertainty. As a common approach, to simplify, these problems are formulated using deterministic MILP. Aside from all the simplifications still there are limitations for solving large problems with exact algorithms. Zapate et al. (2008) propose three different mathematical formulations for resolving limitations posing from indexing of the task execution mode, the indexing of time periods and discrete character of the resources. The objective is to schedule all jobs in each time horizon in a way that the total non-discounted profit of all projects is maximized. For each project, there is an expected return associated with it. There are lower and upper bounds for combination multiples that can be allocated to each task based on the values used to represent the resource makeup ratios. Two of the formulations which are developed are using continuous time representations, continuously divisible resources, and short-term horizon. The other one is using a discrete time representation. The results show the new formulations are not able to solve larger problems.

Wei-Xin et al. (2014) present a multi-objective model for multi-project scheduling on critical chain. All the activities can only be executed in a single mode. There are four objectives: total duration, cost, quality, and robustness. The objective is to maximize the utilization of all the objectives. The model is capable of producing different schedules based on the magnitude of the objective functions. For solving this problem, a cloud genetic algorithm is used. The algorithm is using the randomness and stability of normal cloud model. Although, the shortcoming of the model is the process of determining a weight for each factor. This process can be influenced by subjective factors.

In real-world situations, there are a lot of industries which does not have all the resources at the beginning of the project due to some considerations such as inventory cost. They order the materials when they are needed, to reduce the cost of the project. Material ordering is considered in Zoraghi et al. (2017). In their work, a new version of the problem is considered and formulated. The problem is called Multi-mode Resource Constrained Project Scheduling with Material Ordering (MRCPSMO). Bonus and penalty policies are included to make the problem more realistic. The intention of the model is to schedule all the activities in all the project and to decide the time and quantity of the materials to order. The objective function is to minimize the total cost over all projects. The material holding cost, the material ordering cost, the bonus paid by the client and the cost of delay in project completion are four elements which are considered in the objective function. Three meta-heuristic algorithms are used as the solution methodology.

Hartmann and Brikorn (2010) present a survey of variants and extensions of the resource constrained project scheduling problem. For a survey on different heuristics used for solving resource-constrained project scheduling, Kolisch and Hartmann (2006) is a valuable source. Kolisch and Padman (2001) is a survey of deterministic project scheduling and discuss various mathematical formulations and their different objective functions. A classification of different aspects of project scheduling is presented in Horroelen et al. (1998). A comparison between exact algorithms can be seen in Hartmann and Drexl (1997).

In this chapter, the literature related to resource-constrained scheduling problem was presented. Different approaches to model the problem were studied and various versions and extensions of the problem were discussed. Solution methodologies used to solve the problem were mentioned. Different versions and extensions of the problem were divided by their objective functions. In the next chapter, a new mathematical model is presented to describe a new extension of the problem.

CHAPTER 3: PROBLEM DEFINITION

3.1 Introduction

In this chapter, Multi-Project Multi-Mode Resource Constrained Scheduling Problem with Material Ordering (MPMRCSMO) is studied. General definitions and constraints are described and a mathematical model is developed. The developed model is similar to that in Zoraghi et al. (2017) with significant extensions considering additional problem features. Zoraghi has studied a single project environment with material ordering. In our study, the multi-project version of the model is presented considering resource dedication (RD) policy. The model considers a portfolio of projects all available at time zero. All projects are presented by activity on node network. There are two dummy activities showing start and finish of each project with the duration and resource usage of zero. Activities use both renewable and nonrenewable resources. Each activity can operate in various modes with different resource usages and durations. We assume all the information related to resource usage and duration of each mode is given. The number of renewable resources dedicated to each project, the order time and the amount of order of all nonrenewable resources, mode assignment for each activity, and finally the start time of each activity are the decisions that need to be made. Figure 3.1 shows an example of project networks and their resource usage and duration. The information of each activity is shown in a specific format. The format is (*R1, R2, N1, N2, D).* Notations *R1* and *R2* show the renewable resource usage of each activity in each mode. Notations *N1* and *N2* indicate the nonrenewable resource type 1 and 2 usages in each mode. In the end, the duration of each activity in each mode is presented by *D*.

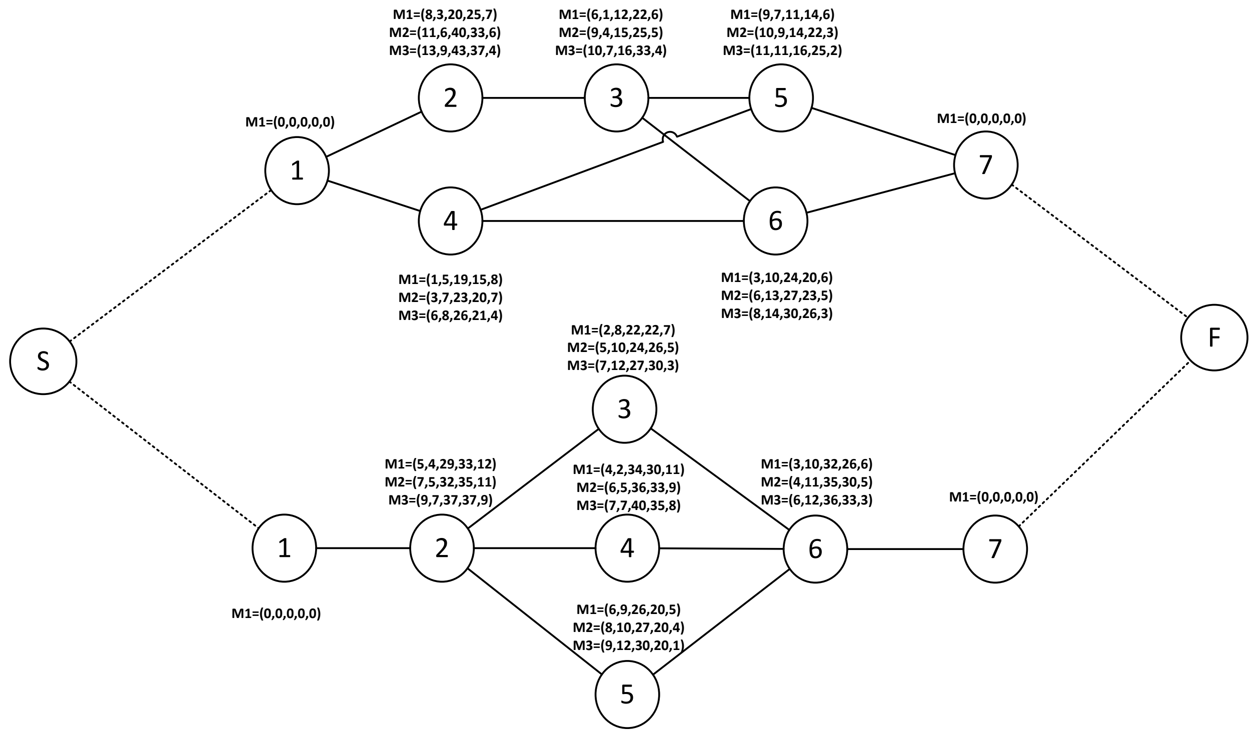


Figure 3.1 Project networks and information

In this figure, two projects which belong to a portfolio are shown. Both projects are available at time zero. The portfolio is finished when both projects are finished. The nodes S and F are used to show the relation between the projects. Aside dummy activities which have resource usage and duration of zero, all activities can operate in three modes with the information of each mode given for each activity. As an example, activity 3 of project 1 in mode 1, uses 6 and 1 renewable resources 1 and 2 respectively and 12 and 22 nonrenewable resources 1 and 2 in order to finish in 6 time unit. The same activity in the second mode requires 9 renewable resources 1, 4 renewable resource 2 and 15 and 25 nonrenewable resources 1 and 2 respectively and takes 5 unit of time to complete. The limited number of renewable resources are allotted to projects and stay constant until the end of the projects. All activities use all type of resources.

3.2 Definitions

The aim of this model is to be close to the real world situation. As a result, the assumptions have been considered in a way which can be applied in practice. The definition of each term is based on its use in our study.

3.2.1 Resources

As it was mentioned, both renewable and nonrenewable resources are considered in our study. A renewable resource is a resource which can be used repeatedly. Examples of renewable resources in industry are human labor, machines etc. A nonrenewable resource is a resource which cannot be used more than once. When a nonrenewable resource is used by an activity, it is consumed and cannot be used for another activity. An example of nonrenewable resources is raw materials. In this study, the amount of renewable resources is known at the beginning of the time horizon. It is possible to have R different kind of renewable resources. When a renewable resource is occupied by an activity, it stays busy until the activity is finished. All nonrenewable resources have an initial inventory of zero. The amount of nonrenewable resources varies during the time scope of the project and depends on the consumption of the resources and the order amount of them. In our study N different kinds of nonrenewable resources are considered.

Activities can operate in multiple modes. The modes are predefined. Each mode has a specific resource consumption (renewable and nonrenewable). The duration of each activity varies based on its mode. The more the resource usage is, the less the duration. Each activity can only be executed in one mode. To start an activity all the required resources should be available. There is no preemption in activities. All the precedence relations are finish-to-start, meaning that an activity cannot start unless all the precedence activities are finished.

3.2.2 Holding cost

A large number of industries are dealing with holding cost which is an important factor in overall cost. Holding cost is the cost paid to keep an item of nonrenewable resources in inventory at each unit of time. Each resource has a specific holding cost. The holding cost of a resource is applicable from the time the resource arrives in the inventory to the time it leaves the inventory. To be more realistic, holding cost is considered in our model.

3.2.3 Ordering cost and Purchase cost

Ordering cost is the cost of placing an order. This cost is independent of the quantity of the resources ordered. In our proposed mathematical model, each kind of nonrenewable resources has an ordering cost. If an order of a resource is placed, apart from the quantity of the order, a fixed order cost of that resource should be paid. Each nonrenewable resource has a price which is for a unit of that resource. This price is different from ordering cost. The total price paid for resource k is dependent on the quantity of that resource and is calculated by the formulation $OR_k\ N_k^N$. In this formulation, N_k^N is the price of nonrenewable resource k and OR_k is the number of resource k ordered. It is possible to place orders at any time.

3.2.4 Inventory related

Safety stock, inventory space, and space requirement are some of the features related to inventory which are considered in the model. In our model, each nonrenewable resource has a safety stock. It means the amount of that resource in inventory cannot be less than its safety stock. The safety stock is used to deal with unpredicted events during a project, such as a delay in delivery. The inventory has a limited space. This fact is considered here as inventory space. We can store resources in inventory as long as we have space. The inventory is only used for nonrenewable resources. Each type of nonrenewable resources stored in inventory has a specific size, as a result, they occupy a different space in the inventory. Space requirement defines the area needed to fit each unit of an item into the inventory.

3.2.5 Bonus and penalty

To be more realistic and close to practice, we have included bonus and penalty into the mathematical model. The bonus is the money which client pays to the company if the project is finished before the due date. Each project has its own bonus. It is assumed that the total bonus paid for a project is the multiplication of the days that the project is finished before the due date and bonus amount for one day. The penalty is the amount of money paid to the client because of delay in the project. Due to a lot of reasons a project might face a delay. In that case, the company should pay a penalty cost for each day the project is delayed.

3.3 Assumptions

In the model presented here, as it was mentioned, to be closer to the real-world situation, penalty cost and bonus are considered. The company should pay a corresponding penalty per each day delay in each project. If a project is finished before the due date (DD_p), a bonus per day is received. Due date is the date that a project is expected to be delivered. Bonus for each day for project p is shown by BN_p in the model. Notation PN_p is used for showing penalty which is paid if project p is delayed for each day. The number of renewable resources is given, but nonrenewable resources should be purchased and kept in the inventory. Holding cost is the price paid for keeping a unit of a nonrenewable resource in inventory for a unit of time. The holding cost of resource k for each unit of time is shown by H_k in the presented model. There is a limited capacity of the inventory which we call it inventory size. In the model IS is used to define the inventory size. There is a cost of A_k for placing an order of resource k at any time. Price of each unit of nonrenewable resource k is shown by N_k^N in the model. The space requirement of nonrenewable resource k is defined by RS_k in our study. To deal with unexpected situations, for each nonrenewable resource, there is a safety stock. The inventory level of each item should not be less than its safety stock in anytime. To show the safety stock of resource k notation SS_k is used. Each resource k ordered takes DL_k unit of time to arrive at the inventory which is called delivery time of that resource. There are multiple projects which all are available to schedule at the beginning of time horizon. Each project includes several activities which can be run in different modes. Each mode has a specific usage of renewable and nonrenewable resources and its duration varies correspondingly. Notations U_{pjrm}^R and U_{pjkm}^N indicate renewable resource r usage for activity j of project p operating on mode m and nonrenewable resource k usage for activity j of project p operating on mode m respectively. Notation d_{pjm} indicates the duration of activity j of project p operating on mode m.

3.4 Mathematical model

The mathematical model is presented in this section. Decision variables and objective function are described and constraints are explained in detail.

3.4.1 Decision variables

In this model, aside from start time of activities, the time of placing an order for each nonrenewable and the amount of the order are considered as variables too. Two binary variables S_{pjmt} and γ_{kt} are used to indicate the start time of activity j of project p operating on mode m and order time of resource k, respectively. The number of non-renewable resource k ordered at time t is shown by OR_{kt} in the model.

Resource dedication policy requires renewable resources for each project to be predefined and stay constant in the time horizon of the project. In this policy, each project only uses the renewable resources which are assigned to it. Another variable is needed to decide how many renewable resources should be assigned to each project. We use D_{pr}^{R} to present the number of renewable resource r dedicated to project p.

3.4.2 Objective function

Zoraghi considered four parts in his objective function. The material holding cost, the material ordering cost, the bonus paid by the client and the penalty cost paid by the company are the elements of the objective function in Zoraghi model. In our study, another part is added to the objective function which is the purchase cost of nonrenewable resources. While order cost is independent of quantity, purchase cost varies by the number of nonrenewable resources ordered.

3.4.3 Notations

Sets, parameters, and variables used in the model are defined in this section. To consider all the aspects which were mentioned, the model uses 6 different sets of variables.

Sets:

T -Set of time periods, t= 1…T

R -Set of all renewable resources, r= 1…R

K -Set of all nonrenewable resources, k= 1…K

J_p *-Set of all activities for project p, j= 1…*J_p

A_p *-Set of all precedence relationships of project p*

M_{jp} *-Set of modes for activity j of project p, m= 1…*M_{jp}

P -Set of projects, p= 1…P

Parameters:

E_{pj} *-Earliest start time of activity j of project p*

L_{pj} *-Latest start time of activity j of project p*

d_{pjm} *-Duration of activity j of project p operating in mode m*

U^R_{pjrm} *-Renewable resource r usage in activity j of project p operating in mode m*

U^n_{pjkm} *-Nonrenewable resource k usage in activity j of project p operating in mode m*

DD_p *-Due date of project p*

N^N_k *-Price of nonrenewable resource k*

M *-A big number*

A_k *-Order cost of material k*

H_k *-Holding cost of each nonrenewable resource k per unit time*

BN_p *-Bonus for each day finishing project p before deadline*

PN_p *-Penalty for each day delay in project p*

DL_k *-Delivery time of nonrenewable resource k*

SS_k *-Safety stock of nonrenewable k*

IS *-Inventory size*

RS_k *-Space requirement of material k*

Decision variables:

$$S_{pjmt} \begin{cases} 1 \; if\ activity\ j\ of\ project\ p\ operating\ at\ mode\ m\ start\ at\ time\ t \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad otherwise \end{cases}$$

D^R_{pr} *-Number of renewable resource r dedicated to project p*

$$\gamma_{kt} \begin{cases} 1 \qquad if\ material\ k\ is\ ordered\ at\ time\ t \\ 0 \qquad\qquad\qquad\qquad otherwise \end{cases}$$

I_{kt} -$Invetory\ level\ of\ nonrenewable\ resource\ k\ at\ time\ t$

AR_{kt} -$The\ number\ of\ nonrenewable\ resource\ k\ arrived\ at\ time\ t$

OR_{kt} -$Ordered\ amount\ of\ nonrenewable\ resource\ k\ at\ time\ t$

3.4.4 Model

The mathematical model is presented as follows:

$$Min(\sum_{k=1}^{K}\sum_{t=0}^{T} A_k\,\gamma_{kt} + \sum_{k=1}^{K}\sum_{t=0}^{T} OR_{kt}\,N_k^N + \sum_{k=1}^{K}\sum_{t=0}^{T} H_k\,I_{kt} - \sum_{p=1}^{P}\sum_{t=E_{pJ_p}}^{DD_p-1} BN_p\big(DD_p - t\big)\,S_{pJ_pmt} +$$

$$\sum_{p=1}^{P}\sum_{t=DD_p+1}^{T} PN_p(\,t - DD_p)S_{pJ_pmt}) \qquad (1)$$

S.t:

$$\sum_{m=1}^{M_{ap}}\sum_{t=E_{ap}}^{L_{ap}} S_{pamt}(t+\,d_{pam}) \;\le\; \sum_{m=1}^{M_{bp}}\sum_{t=E_{bp}}^{L_{bp}} S_{pbmt} * t \qquad \forall\,(a,b)\in A\,, p\in P \qquad (2)$$

$$\sum_{m=1}^{M_{jp}}\sum_{t=E_{pj}}^{L_{pj}} S_{pjmt} = 1 \qquad \forall\, j\in J_p\,, p\,\in P \qquad (3)$$

$$\sum_{m=1}^{M_{jp}}\sum_{j=1}^{J_p}\sum_{w=\max(t-d_{pjm},E_{jp})}^{\min(t\,,L_{jp})} U_{pjrm}^{R} * S_{pjmw} \;\le\; D_{pr}^{R} \qquad \forall\, p\in P, r\in R\,, t\,\in T \qquad (4)$$

$$\sum_{p=1}^{P} D_{pr}^{R} \;\le CAP_r \qquad \forall\, r\in R \qquad (5)$$

$$I_{kt} =\; I_{k(t-1)} +\; AR_{kt} - \sum_{p=1}^{P}\sum_{m=1}^{M_{jp}}\sum_{j=1}^{J_p} U_{pjkm}^{N} *\; S_{pjmt} \qquad \forall\, k\in K\,, t\,\in T \qquad (6)$$

$$AR_{kt} = OR_{k(t-DL_k)} \qquad \forall\, k\in K\,, t\,\in T \qquad (7)$$

$$OR_{kt} \le \gamma_{kt} * M \qquad \forall\, k\in K\,, t\,\in T \qquad (8)$$

$$\sum_{p=1}^{P}\sum_{m=1}^{M_{jp}}\sum_{j=1}^{J_p} U_{pjkm}^{N} * S_{pjmt} \;\le\; I_{k(t-1)} \qquad \forall\, k\in K\,, t\,\in T \qquad (9)$$

$$I_{kt} \ge\; SS_k \qquad \forall\, k\in K\,, t\,\in T \qquad (10)$$

$$\sum_{k=1}^{K} I_{kt} * RS_k \le\; IS \qquad \forall\;\, t\,\in T \qquad (11)$$

$$AR_{kt} \in Z^{+} \qquad \forall\, k\in K, t\in T \qquad (12)$$

$$OR_{kt} \in Z^{+} \qquad \forall\, k\in K, t\,\in T \qquad (13)$$

$$S_{pjmt}, \gamma_{kt} \;= \{0\,,1\} \qquad \forall\, j\in J_p\,, t\in T\,, m\in M_{jp}, p\in P \qquad (14)$$

The objective function of the problem modeled in Eq(1) is the minimization of cost over all the projects. The objective function includes five parts. The material ordering cost, the material holding cost, the material purchase cost, the bonus, and penalty are the five parts.

Constraint in Eq(2) ensures the precedence relations between activities of each project. An activity can be started if all the precedence activities are finished. Constraint in Eq(3) enforces that each activity can only be executed on one mode and at one time. Inequality in Eq(4) shows that at each time unit the use of renewable resources for each project should not be more than the resources dedicated to that project. In constraint in Eq(5) it is guaranteed that resources dedicated to all projects are not more than the capacity of renewable resources available. The inventory level of each nonrenewable resource at each time is calculated in Eq(6). In constraint shown in Eq(7) the arrival time of each order is reflected. Inequality in Eq(8) emphasizes that when there is an order in a specific period, the binary variable γ_{kt} at that time is one, or zero otherwise. In other words, if we place an order, the order cost should be paid. Constraint in Eq(9) shows that nonrenewable resource use at each day cannot be more than the level of inventory for the end of the previous day. Inequality in Eq(10) indicates that inventory level of each nonrenewable resource should not be less than its safety stock. In Eq(11), the constraint related to inventory size can be seen. Base on this constraint, the sum of size of all the resources in the inventory should not exceed the inventory size. The constraints in Eqs(12) – (14) show the domain of decision variables.

In this chapter, the mathematical model is presented. The terms used in the mathematical model are defined. Sets, parameters, and variables are defined and all the constrained are explained. Zoraghi model is used as the base of this study and it is extended into a multi-project scheduling model with material ordering with more features.

In the next chapter, we first use 22 smaller size instances to validate the developed model and its performances in solving problems of different features. As we notice, solving the developed model directly using off-shelf optimization software requires extensive computational time. We developed a straightforward decomposition based heuristic solution approach aiming at solving larger size practical problems. Details of the heuristic method along with the testing results are also presented in Chapter 4. In the end, a practical case is used and the approach is tested on it.

CHAPTER 4: SOLUTION METHODOLOGY

Ulman (1975) has proved that scheduling is an NP-Complete problem. As a result, the problem that is considered here is NP-Complete and more complex than a classic scheduling problem. Trying to solve this problem with off-shelf optimization software is computationally expensive as a small instance can take hours to solve.

A new heuristic to solve the problem is presented. The approach consists of three stages. Different instances are used to validate the model and test the heuristic. A case study is used to experiment the approach in practice. CPLEX version 12.7.1.0 is used as the software to solve the problem.

4.1 Challenges and Approach

The multi-mode resource constrained project scheduling with material ordering for a single project environment was presented by Zoraghi et al. (2017) and it is an NP-hard problem. In this study, a mathematical model is developed to deal with multi-project problems, also, more features are added to the model. As it was mentioned, the problem with one project is NP-hard so it is clear that the multi-project version of the problem is NP-hard as well due to the increased complexity. Trying to solve the problem with current programs available (without using heuristic and meta-heuristic methods) is time-consuming and a small example can take a long time to run. There is a need for a method to be able to solve a real case with a large number of data in a reasonable time. This is why we have developed a new heuristic to tackle the problem in the real world. In the next section, the developed heuristic is explained in details.

4.2 Developed Heuristic

The heuristic developed for solving the problem is a three-phase method. As we noticed, using off-shelf optimization to solve the model for a case with large data is computationally expensive. As the solution methodology, the problem is broken down into three sub-problems. Instead of solving a large model with many variables, three smaller models with fewer variables are solved. The first phase is to solve a nonlinear model. In the second and the third phases two linear models are solved. Each phase deals with a different aspect of the problem. The results of the first stage are the input for the second one and the results of the second stage are the data for the third stage.

4.2.1 First phase

Resource dedication policy is considered here. The first stage decides about the number of renewable resources dedicated to each project. A model with nonlinear objective function is presented. Once the number of resources is specified, projects stick to that amount of resources until the end. It means projects cannot swap resources with each other. For example, if there are 10 renewable resource *A* available and it is distributed between two projects 1 and 2 by the amount of 7 and 3 respectively, at all time the resource availability for projects 1 and 2 is 7 and 3 respectively. The objective function consists of two terms. The function of the terms used in the objective function is hyperbolic. All the constraints in the model are linear. Figure 4.1 shows the graph of the objective function.

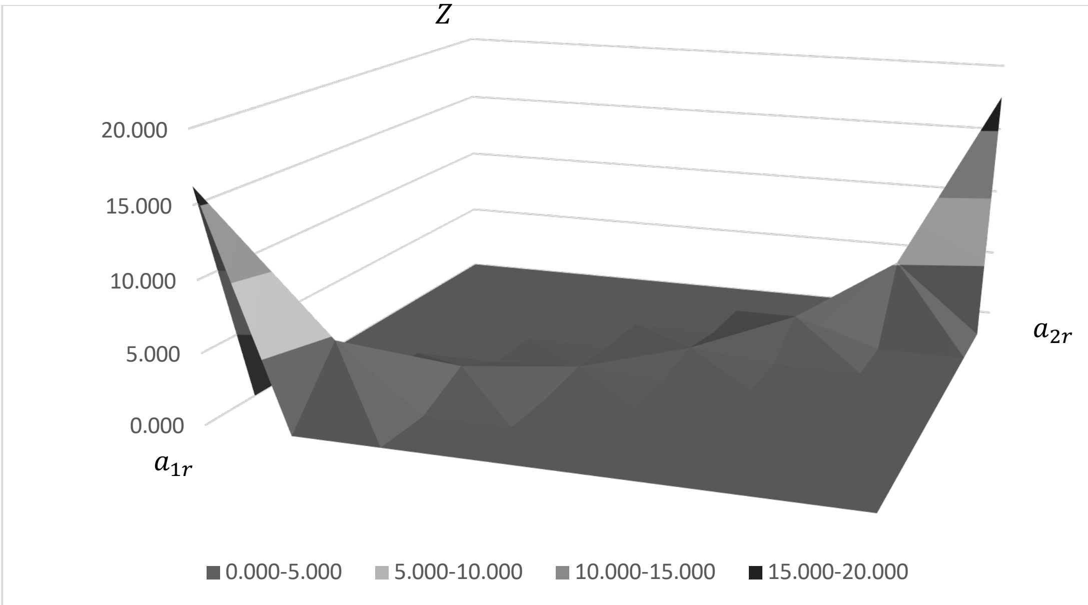


Figure 4.1 Graph of objective function

In the first stage, a nonlinear mathematical model is solved. The objective function is the only nonlinear part of the model, aside from that, all the constraints are linear. The model is presented as below.

Minimizing $Z = \sum_{r=1}^{R} \left| \frac{\bar{r}_{1r}}{a_{1r}} - \frac{\bar{r}_{2r}}{a_{2r}} \right|$ (15)

Subject to:

$a_{pr} \geq MaxU_{pr}$ $\forall\, p \in P\, , r \in R$ (16)

$\sum_{p=1}^{2} a_{pr} = CAP_r$ $\forall\; r \in R$ (17)

$a_{pr} \in Z^+$ $\forall\, p \in P, r \in R$ (18)

In this model, the objective function is to minimize the sum of difference between Resource Constraint (RC) of each resource in projects. Resource Constraint (RC) was introduced by Patterson (1976) and used by Demeulemeester et al. (2003) in RanGen1 to generate project network. RC is defined as $\frac{\bar{r}_r}{a_r}$ which in this equation, a_r is the total availability of renewable resource type *r* and $\bar{r}_r$ denotes the average quantity of resource type *r* demanded when required by an activity. The formulation for calculating $\bar{r}_r$ is given as $\frac{\sum_{i=1}^{n} r_{ir}}{\sum_{i=1}^{n} X_{ir}}$ where r_{ik} is the number of resource *r* required for activity *i* and X_{ik} is a binary which is 1 if $r_{ir} > 0$, and 0 otherwise.

In the objective function, a_{pr} is the only variable and decides about the amount of renewable resource type *r* dedicated to each project. The resources are decided in a way that more resources are assigned to a project with more $\bar{r}_r$ to make the two RC values as close as possible.

Constraint in Eq(16) ensures that the number of resource *r* assigned to project *p* is more and equal than the maximum number of that resource used by an activity in project *p* operating in mode one. In other words, there should be the minimum requirement for all activities in a project. Each activity in its mode one uses the least number of resources. For all activities operating at mode one, the maximum use of resource *r* in project *p* among all activities is shown by $MaxU_{pr}$ in the first phase. In constraint shown in Eq(17), it is guaranteed that all available renewable resources are assigned to projects. The last constraint indicates that the number of resources dedicated to a project should be positive and integer. Excel solver is used as the tool to solve this model.

To use this method in a case with more than two projects, at each time, the resource assignment for one project is decided until all projects are assigned. In this case, the $\bar{r}_r$ for the project that we want to decide about its resources is considered alone and the rest of projects are considered together as a single larger project and the $\bar{r}_r$ values of them are added to calculate the new $\bar{r}_r$ for the new combined project. The new value for $MaxU_{pr}$ is calculated by summing $MaxU_{pr}$ for all projects which are considered in the large project. When the resource allocation for one project is finished, the resources which are assigned to that project are subtracted from CAP_r to calculate the new CAP_r. The same process is repeated until the resource allocation for all projects is decided. Let consider three projects with the following characteristics.

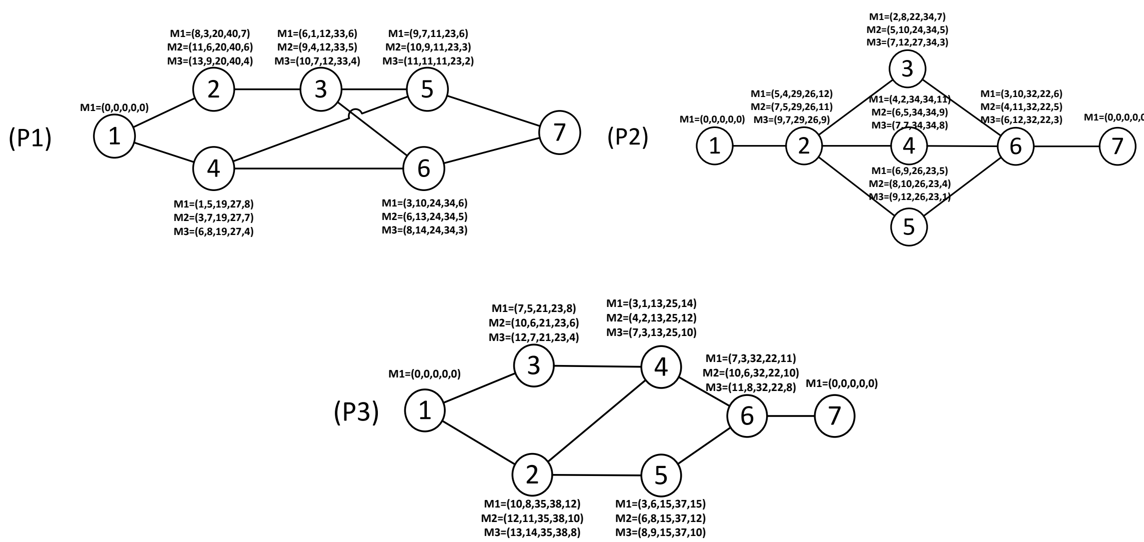


Figure 4.2 Project Networks considered in three-project example

The value of $\bar{r}_r$ and $MaxU_{pr}$ for each resource in each project are given in Table 4.1 and Table 4.2. There are 45 unit of each renewable resource available.

Table 4.1 Resource Average usage for each resource in each project

	R1	**R2**
P1	5.4	5.2
P2	4	6.6
P3	6	4.6

Table 4.2 Maximum Resource usage for each resource in each project

	R1	R2
P1	9	10
P2	6	10
P3	10	8

First, to distribute the resources between these three projects, we consider project one alone and projects 2 and 3 as a single larger project. The value of $\bar{r}_r$ and $MaxU_{pr}$ for the new combined project is the sum of the same values in the other two projects. The new values for $\bar{r}_r$ and $MaxU_{pr}$ are shown in Tables 4.3 and Table 4.4.

Table 4.3 Maximum Resource usage in for new combination

	R1	R2
P1	5.4	5.2
New P	10	11.2

Table 4.4 Maximum Resource usage for new combination

	R1	R2
P1	9	10
New P	16	18

Now we treat them as two projects and solve the first model. The result shows that 16 and 14 of resources one and two are assigned to project 1 and the rest are assigned to the combination of projects 2 and 3. Now that we have the resource assignment for project 1 we can exclude it from the portfolio and decide about the other two projects. The new resource availability for resources 1 and 2 now is 29 and 31 respectively, instead of 45. We repeat the same process to find the resource distribution for projects 2 and 3. The final result is reflected in Table 4.5.

Table 4.5 Resource Distribution between projects

	R1	R2
P1	16	14
P2	12	18
P3	17	13

Due to the long time which it takes CPLEX to solve the optimization model for a problem with more than two projects, we have only tested problems with two projects.

4.2.2 Second phase:

Having the number of renewable resources allotted to each project, in the second phase, each project is considered separately and is solved to reach an optimal schedule without considering nonrenewable resources availability. The schedule minimizes the makespan of each project in order to minimize the penalty and maximize the bonus. A linear mathematical model is solved in this phase. Solving this sub-problem is still NP-hard as it is a classic scheduling problem, but the advantage is the decreased complexity which makes it possible to apply it on larger size problems.

For the second phase, two objective functions are proposed. In this stage there is no connection between projects and each project is considered alone.

$$Max \sum_{t=E_{pJ}}^{DD_p-1} \sum_{m=1}^{M_{Jp}} BN_p \times (DD_p - t) \times S_{pJmt} - \sum_{t=DD_p+1}^{T} \sum_{m=1}^{M_{Jp}} PN_p \times (t - DD_p) \times S_{pJmt} \qquad (19)$$

$$\text{Min } \sum_{t=1}^{T} \sum_{m=1}^{M_{Jp}} (t \times S_{pJmt}) \qquad (20)$$

Subject to: Eqs. (2), (3), (4) and (14)

Objective functions in Eqs(19) and (20), both have the same meaning and both are to minimize the makespan of the project. The difference is that the first objective calculates the penalty and bonus at the same time while the second one is just minimization of the makespan. It is important to say that the schedule resulting from each objective function is identical. In objective function in Eq(19) the first term calculates the bonus and the second term calculates the penalty. To maximize the objective function, the project should end as soon as possible. Maximizing the Bonus and minimizing the penalty produce the best schedule. In the objective function in Eq(20), as it was mentioned, the makespan of the project is minimized. The model finds the earliest start time for the last activity of each project in order to minimize the duration.

Constraint in Eq(2) indicates the precedence relationship between activities. To decrease the time required for processing, the start time of each activity is considered within a time window related to that activity. The time window of each activity starts from the earliest possible start time of each activity and ends at the latest possible start time of that activity. Equality in Eq(3) guarantees that each activity is scheduled once and only once. Inequality in Eq(4) shows the fact that renewable resources used at each time slut cannot exceed the number of renewable resources dedicated to that project. The number of renewable resources dedicated to each project is already decided in the first phase. The last constraint shows that the variable for deciding the start time of each activity is a binary variable.

This stage generates the optimal schedule regarding renewable resources allotted to each project. The schedule from this phase ignores the availability of nonrenewable resources. The results from the second stage are the input for the final stage.

4.2.3 Third phase

The results from the previous phase provide a complete schedule for each project. Having the schedules gives us the information for nonrenewable resources requirement at each time. The third phase is to solve a linear model for all nonrenewable resources for all projects together. The best time to place an order and the number of order is decided with the objective of minimizing the total cost. This is the final stage of the heuristic. The nonrenewable resources usage for all projects is considered here and a general plan for ordering resources is produced. The model is flexible to consider a large number of resources at the same time. The model is presented as below.

$$\text{Min } \sum_{k=1}^{K} \sum_{t=0}^{T} A_k \, \gamma_{kt} + \sum_{k=1}^{K} \sum_{t=0}^{T} OR_{kt} \, N_k^N + \sum_{k=1}^{K} \sum_{t=0}^{T} H_k \, I_{kt} \tag{21}$$

Subject to: Eqs. (6), (7), (8), (9), (10), (11), (12), and (13)

The objective function consists of three terms all related to inventory and order cost. The first term is to minimize the cost of placing orders. There is a binary variable γ_{kt} which indicates whether resource *k* is ordered at time *t* or not. If the order is placed for resource *k* at time *t,* the value for γ_{kt} is 1, otherwise, it is 0. The second term shows the cost spent on buying nonrenewable resources. Variable OR_{kt} indicates the number of resource *k* ordered at time *t*. the parameter N_k^N shows the price of each renewable resource *k*. The last term calculates the inventory cost of all the resources during the time horizon of portfolio. In this term H_k and I_{kt} represent holding cost of each resource *k* at each unit of time and inventory level of resource *k* at time *t*, respectively. In this phase S_{pjmw} which was variable in the second phase, is a parameter now. The number of non-renewable resources required in each time is calculated by the schedule from the second phase.

The objective function of the model is to minimize the sum of the inventory cost, order cost, and purchase cost. The constraints related to inventory and non-renewable resources are considered here. In our examples, to make the analysis easier, the delivery time and safety stock are zero. It means that as soon as the order is placed the item will be in the inventory and the inventory can also be empty. Additionally, the size of the inventory is considered too large and the constraints related to inventory size are ineffective in our instances. Another assumption is that all nonrenewable resources for each activity are consumed at the beginning of each activity. In other words, to start an activity all required resources should be available in the inventory.

4.3 Results

In this section, the data used in the model and the results using optimization and the developed heuristic are shown and a comparison is made.

4.3.1 Data description

The data used for this study are generated by popular generator RanGen1 which was introduced by Demeulemeester et al. (2003). For generating networks, RanGen1 uses OS as complexity measure. OS is defined as the number of precedence relations which includes only the transitive ones divided by the theoretical maximum number of precedence relations. Base on the definition, OS is calculated by equation $n(n-1)/2$, which n is the number of non-dummy activities in the network. The more the OS number is the more complex the network.

The data generated by RanGen1 includes the project network, resource availability, duration of each activity, and resource usage for each activity. The time window for each activity is calculated based on project network and the projects which are considered in a portfolio. The time window indicates early start (ES) and late start (LS) of each activity. RanGen1 does not generate multi-mode instances. We considered the data generated from RanGen1 as the information for the first mode. To generate the information for mode 2 and 3, random distribution is used. It is assumed that the higher the mode is, the more the resources usage and the shorter the duration.

The instances generated are projects with 7 activities, including dummy activities. Each activity has three modes and uses two renewable and two nonrenewable resources. All activities use all type of resources. In each mode, the usage of renewable and nonrenewable resources vary. The number of renewable resources is fixed and known. The initial inventory of all nonrenewable resources is 0. As it was mentioned before, to monitor and analyze the results easier, safety stock and delivery time are zero. The inventory size is too large as well. It means these three constraints are practically infective.

4.3.2 Worked example

In this section, one of the tested example instances is elaborated in details. Instance 15 which includes projects 3 and 7 is considered here. The approach and steps taken in each method are studied and the results are discussed in details. The networks for projects 3 and 7 are shown in Figure 4.3.

As it can be seen, each project consists of 7 activities (two dummy activities). Each non-dummy activity has 3 modes. Dummy start and dummy finish have only one mode in which resource usage and duration are zero. There are two renewable and two nonrenewable resources for each activity. By increasing the mode of activities the renewable resource usage of activities increases and the duration drops. In this example, the nonrenewable resource usage for activities is constant and does not increase by mode increment.

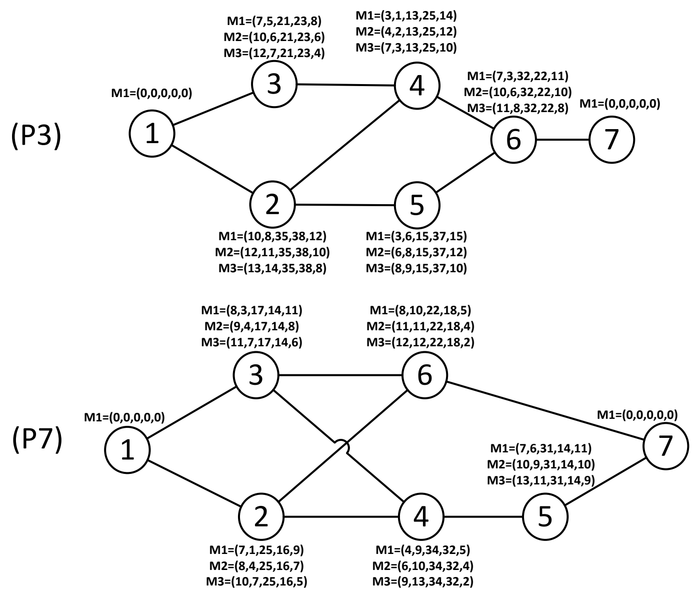


Figure 4.3 Network and Information for instance 15

Using optimization to solve the model, both projects are considered and renewable resources are distributed between them by looking at the whole scope of the portfolio. The model looks at two projects as a single larger project as Figure 4.4 shows. Nodes S and F are added to show the relationship between the projects.

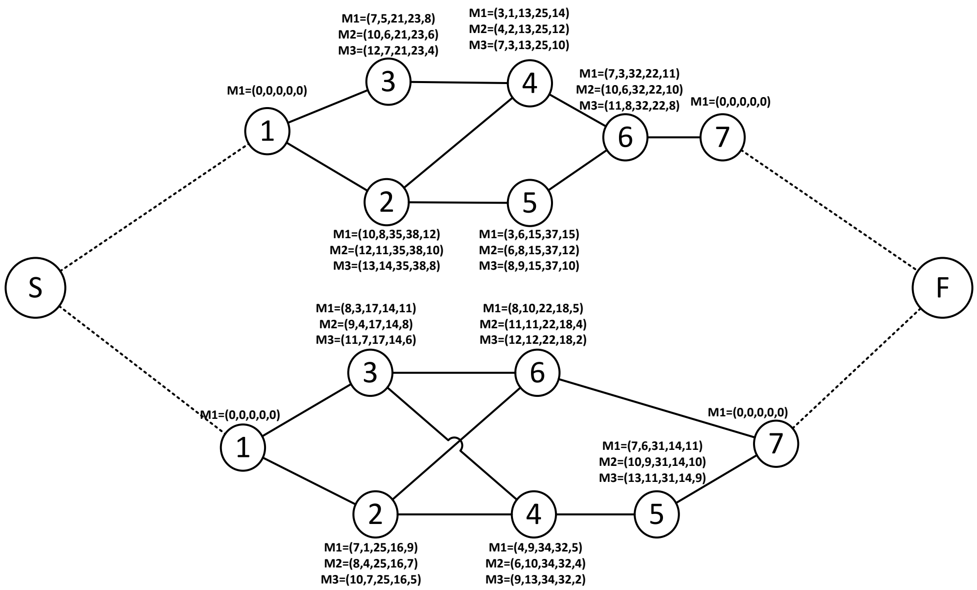


Figure 4.4 Projects Network and Information in instance 15 for optimization

We have 30 unit of each renewable resource available. Since the resource dedication policy is used here, we do not have a shared pool of resources. Renewable resources should be allocated to each project. Table 4.6 shows the optimal distribution of renewable resources.

Table 4.6 Renewable resource distribution in the optimal solution for instance 15

Projects	Resources	Optimal
P3	R1	14
	R2	14
P7	R1	16
	R2	16

To start an activity, all the required resources (renewable and nonrenewable) should be available at the beginning of the activity. The model should decide about the number of nonrenewable resources to order and also the timing of the order. The model is solved aiming minimizing the cost. For this instance, CPLEX reaches to the optimal solution after 10 minutes and 25 seconds. The optimal solution gives us the start time and the mode of all the activities and a plan to order nonrenewable resources. The optimal results for the example are shown in Table 4.7 and Table

4.8. Table 4.7 shows the start time and the mode selection for each activity and Table 4.8 shows the inventory level, the time to order and the number of each resource to order at each time. In the order time columns, the value of 1 means that we place the order, and 0 means we do not order the resource at that time. The rows in which all the values are zero are not shown.

Table 4.7 Start time and mode for each activity in optimal solution for instance 15

	Project3		Project7	
Activities	**Start time**	**Mode**	**Start time**	**Mode**
1	1	1	1	1
2	1	3	1	1
3	10	3	1	2
4	15	2	10	3
5	18	3	15	1
6	29	3	15	1
7	37	1	26	1

Table 4.8 Results for instance 15 in optimal solution

	Order Time		Order Amount		Inventory Level	
Time	**R1**	**R2**	**R1**	**R2**	**R1**	**R2**
0	1	1	77	68	77	68
9	1	1	55	55	55	55
14	1	1	66	57	66	57
17	1	1	15	37	15	37
28	1	1	32	22	32	22

The optimal solution suggests operating activities 2, 3, 5, and 6 in their third mode for the first project. While running the activities in the given mode for project 2 can finish the project in 23 time unit, you can see that project 7 is finished at time 26. Based on the schedule no activity is running from time 12 to time 15 in project 2 and all the renewable resources are idle. Although we have enough renewable resources to schedule activities, and we can order nonrenewable resources required for activities, the holding cost or order cost will exceed the bonus received for that 3 days.

That is why it is better to delay the project for 3 days instead of paying the holding cost of non-renewable resources or place another order to finish the project faster.

While optimization considers all the projects together to decide on all variables, the heuristic considers the projects together only in the first stage to allocate the renewable resources. By looking at the information from each project, $\overline{r}$ can be calculated for each renewable resource in each project. Resource average usage for renewable resources 1 and 2 in projects 3 and 7 is shown in Table 4.9.

Table 4.9 Resource Average usage by each renewable resource in instance 15

		R1	**R2**
RA	**P3**	6	4.6
	P7	6.8	5.8

The results from the first phase which is solved in Excel Solver are shown in Table 4.10. This information indicates the number of each renewable resource which is allotted to each project.

Table 4.10 Renewable resources distribution for instance 15 in heuristic method

Projects	**Resources**	**Heuristic**
P3	R1	14
	R2	13
P7	R1	16
	R2	17

After the first stage, in the second phase, the projects are considered alone as a single project in scheduling phase. The renewable resource availability for each project is known and the schedule is generated in the absence of nonrenewable resources. The networks information for this phase have the format of (R1,R2,D) and there is no nonrenewable resource limitation that is considered here. The network information is modified as it can be seen in Figure 4.5.

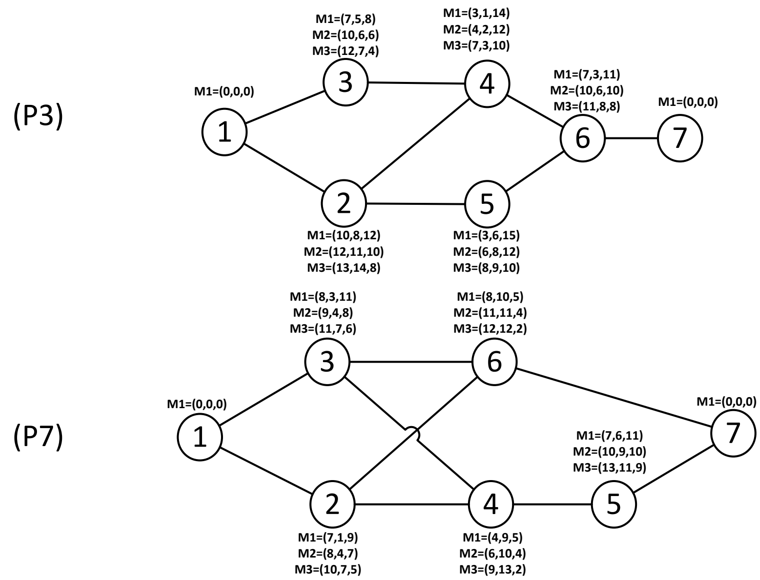


Figure 4.5 Network and information for the second phase of the heuristic in the absence of nonrenewable resources in instance 15

In the second stage, the schedule is generated for projects and the start time and the mode in which each activity operates is decided. The information of this stage is reflected in Table 4.11.

Table 4.11 Start time and mode for each activity in the heuristic method for instance 15

	Project 3		**Project 7**	
	Start time	**Mode**	**Start time**	**Mode**
1	1	1	1	1
2	1	2	1	1
3	12	3	1	2
4	17	2	10	3
5	18	3	13	1
6	30	3	14	1
7	38	1	24	1

As results show, it is possible to operate activities 3, 5, and 6 in project 3 in their fastest mode but, resource limitation for activities 2 and 4 makes it impossible to run all activities in mode 3. Take activity 2 of project 3 as an example. This activity is operated in mode 2. In this mode, the activity needs 12 renewable resource 1 and 11 renewable resource 2 and takes 10 unit of time to finish.

Although in mode 3 it only takes 8 unit of time to finish the activity, the resource requirement for this mode for resources 1 and 2 is 13 and 14 respectively. While there is 30 renewable resource 2, only 13 of them are assigned to project 3. As a result, there is not enough resource 2 for activity 2 of project 3 to operate in mode 3.

After the schedules for both projects are constructed, the nonrenewable resources requirement for each time can be calculated. The nonrenewable resources requirement for both projects are considered together as a whole. The nonrenewable resource requirement for each unit of time in this example based on the generated schedule in phase two is shown in Table 4.12. The table only shows the times in which there is a request for nonrenewable resources.

Table 4.12 Nonrenewable resources requirement for each time for instance 15 in the heuristic method

Time	N1	N2
1	77	68
10	34	32
12	21	23
13	31	14
14	22	18
17	13	25
18	15	37
30	32	22

In the last phase of the heuristic, a plan for ordering the required nonrenewable resources is produced. The objective function of this phase is to minimize the holding cost, purchase cost and order cost. It takes 38 time unit to finish both projects (project 3 end in time 38 and project 7 ends in time 24). As a result, the time horizon which is considered by phase 3 is 38 unit of time. Table 4.13 presents the results from the last phase. It is clear that the model is trying to minimize the time that the resources are in the inventory in order to minimize the holding cost. As a result, the order is placed as late as possible to avoid inventory. For example, at time 30 activities require 32 and 22 nonrenewable resources 1 and 2 respectively. To be able to be on track, the inventory coming to time 30 should fulfill this need. It means the inventory at the end of time 29 should be at least 32 for the first resource and 22 for the second resource. As it is clear from the table, an order is placed at time 29 at the amount required at time 30. This keeps the inventory level for the end of time 29 at the level of resources which are required at the beginning of time 30. Because there is a holding cost, no more resources are ordered at time 29 and the order is placed as late as possible.

Table 4.13 Results for instance 15 in phase three of the heuristic method

	Order Time		**Order Amount**		**Inventory Level**	
Time	**R1**	**R2**	**R1**	**R2**	**R1**	**R2**
0	1	1	77	68	77	68
9	1	1	34	32	34	32
11	1	1	21	23	21	23
12	1	1	31	14	31	14
13	1	1	22	18	22	18
16	1	1	13	25	13	25
17	1	1	15	37	15	37
29	1	1	32	22	32	22

The results for this instance are shown in Table 4.14. The table shows the results for both optimal and the heuristic method.

Table 4.14 Costs for the first instance for the main mode and the heuristic

COSTS	**Main Model**	**Heuristic Method**
Purchase Cost	8724	8724
Order Cost	1235	1976
Holding Cost	4601	4601
Bonus	0	0
Penalty	3016	3016
Total	17576	18317

For this instance, the results show 4% gap between the developed heuristic and optimal solutions. Although, the time spend to reach to a schedule in the heuristic method is minimized. It takes 10 minutes and 25 seconds to solve the model with ILOG CPLEX to optimality, while, only 8 seconds is required to solve the problem with the heuristic and reach a suboptimal solution.

4.5.2 Experimental results

To validate the model, 22 portfolios each including 2 projects are generated. The developed heuristic is tested on the instances and the results are compared with optimal solution. Aside from in the first phase of the heuristic, which is solved by excel solver, all the models are coded and solved by ILOG CPLEX. Due to the complexity of the model solving instances with more than 7 activities using off-shelf optimization would take a long time.

The objective function of the presented model is minimizing the total cost. Penalty, Bonus, purchase cost, order cost, and holding cost are the elements that play a rule in the total cost. All these factors are discussed separately and the results are shown. The results from the first stage are illustrated in Figures 4.6 and 4.7. The first figure shows the way that renewable resource number 1 is assigned to the first project in the optimal solution and the heuristic method. The second figure shows the same information for resource 2. The figures show the percentage of each resource which is dedicated to project 1. Each table corresponds to the resource allocation for project 2 as well. For example, in case 4 in Figure 4.6, in the optimal solution 50% of resource 1 is allocated to project 1 and in the heuristic method, 40% is assigned. It can be concluded that in the optimal solution 50% of resources 1 is assigned to project 2 and also in the heuristic method 60% of resource 1 is allocated to project 2.

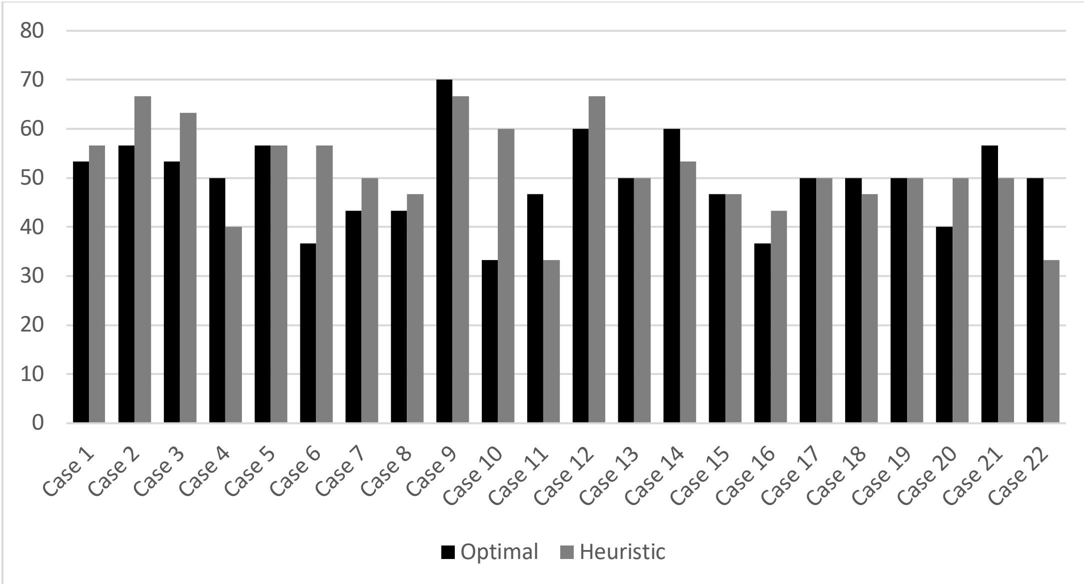


Figure 4.6 Resource 1 allocation for the first project

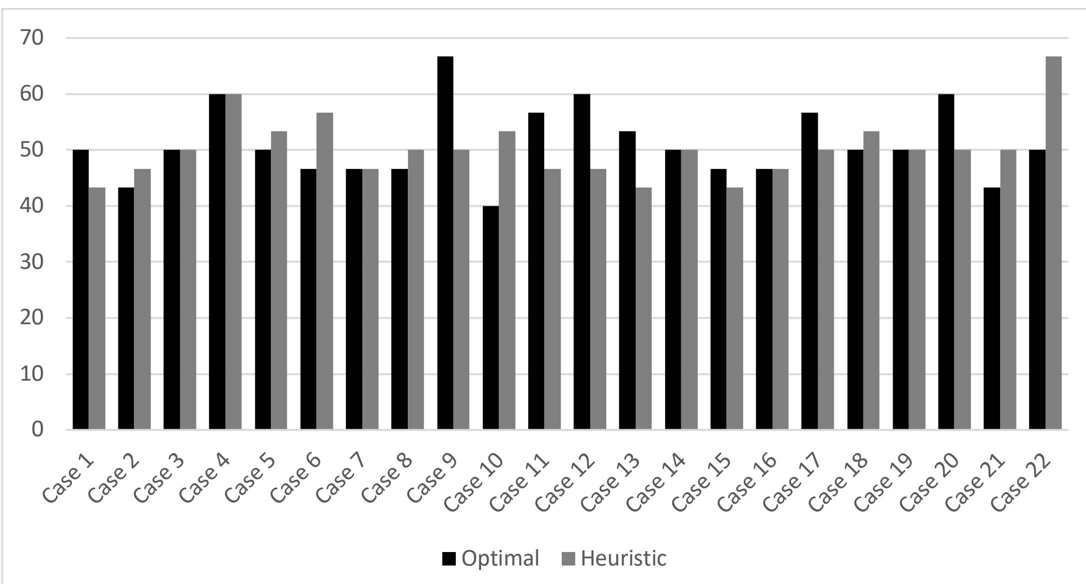


Figure 4.7 Resource 2 allocation for the first project

The average deviation of resource allocation in the heuristic method from the optimal solution is 2.2 for the first resource and 1.8 for the second resource.

Figure 4.8 shows the results for purchase cost for both approaches in instances 18 to 22. In instances 1 to 17, the nonrenewable resource usage in all modes is constant. In other words, nonrenewable resources in the first 17 instances have only one mode. As a result, the purchase cost in the optimal and the heuristic method is identical. In instances 18 to 22, similar to renewable resources, the need for nonrenewable resources varies in each mode. It leads to different purchase cost in different scenarios.

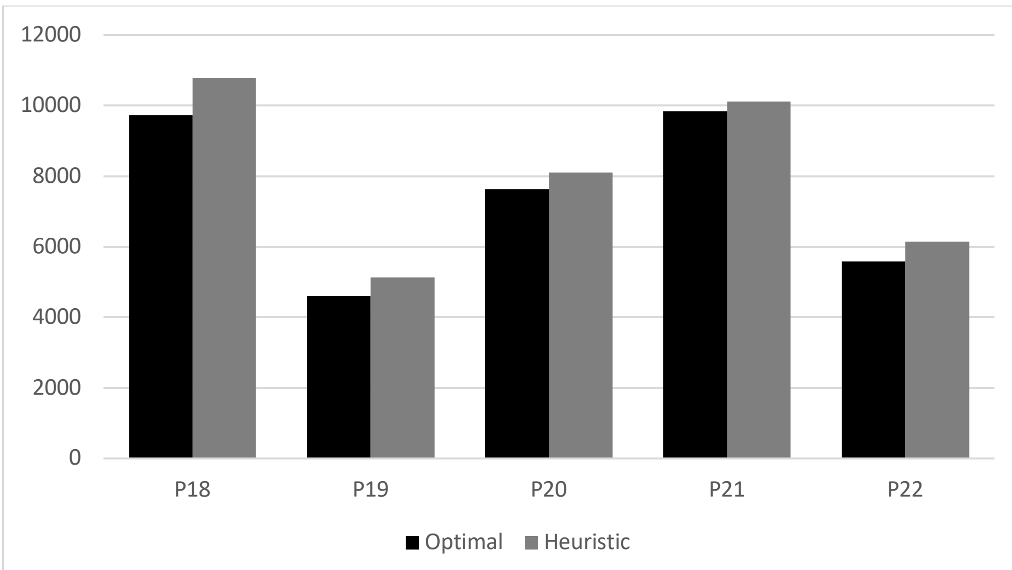


Figure 4.8 purchase cost for instances 18 to 22

Over these 5 cases, the purchase cost shows 10% increase in heuristic method compared to optimal solution. Over all 22 instances this amount is 2.4%.

The results for order cost is presented in Figure 4.9. Order cost defined as the cost which is paid when an order is placed. This cost is independent of the volume of the order.

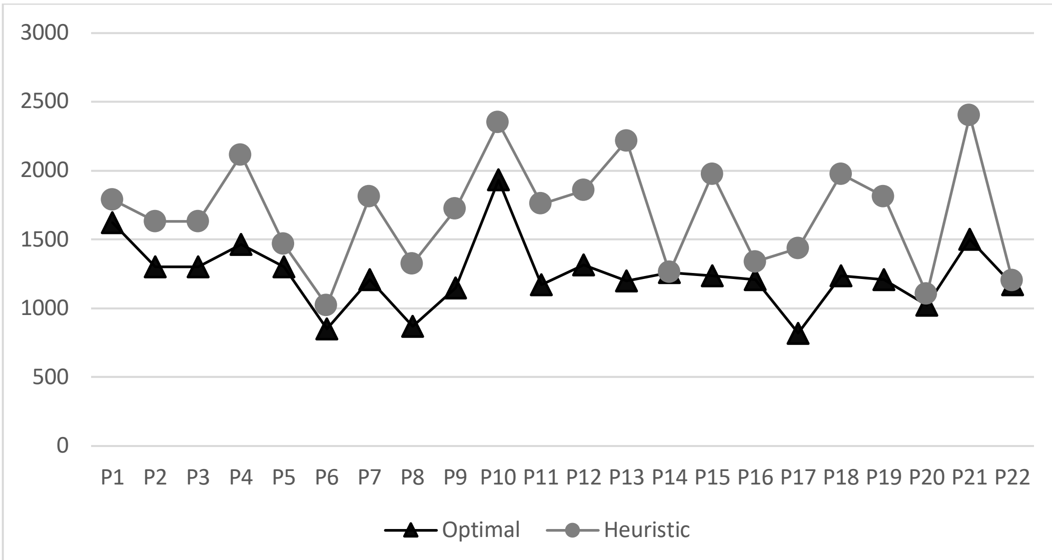


Figure 4.9 Order cost

The presented heuristic, on average, shows about 38% gap in order cost. As a result, the approach is not successful in dealing with order cost. The approach shows different results in dealing with holding cost. While the outcomes for order cost is too far from the optimal schedule, it is not the case in holding cost. On average, the results show 7% increase in holding cost in the first 17 instances with one mode for their nonrenewable resources. The average of all 22 instances increases the gap from 7% to 9%. The holding cost for each project is given in Figure 4.10.

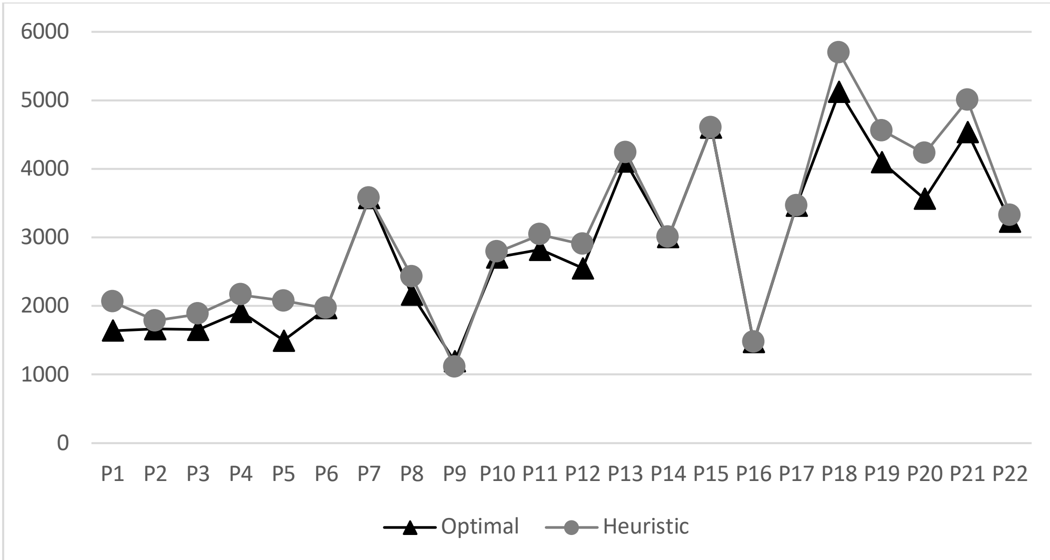


Figure 4.10 Holding cost

Bonus and Penalty are the last factors in the total cost of a project. The heuristic approach has shown a good result in Bonus and Penalty. In average, both bonus and penalty show 3% gap from the optimal solution. Figures 4.11 and 4.12 reflect results for Bonus and Penalty, respectively.

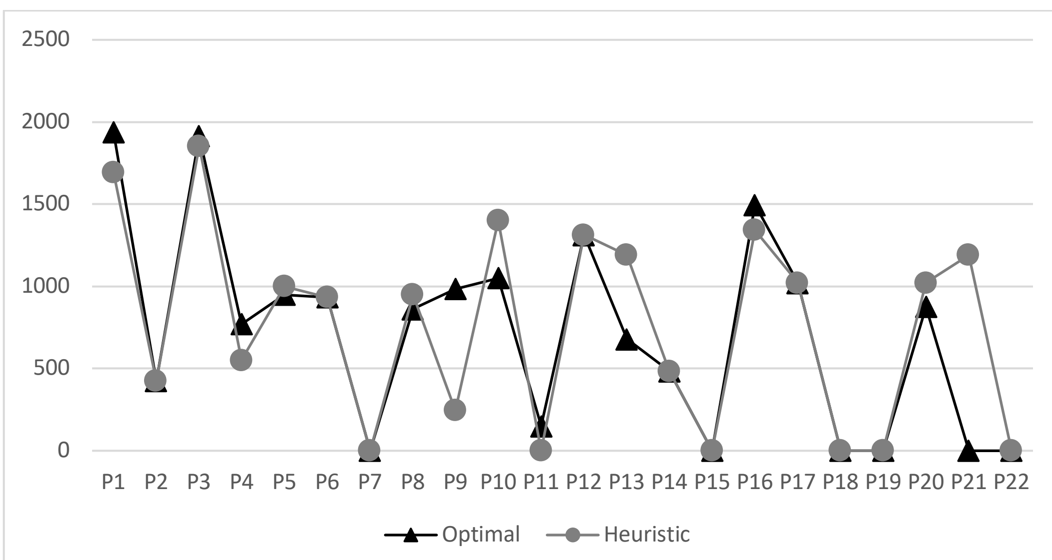


Figure 4.11 Bonus

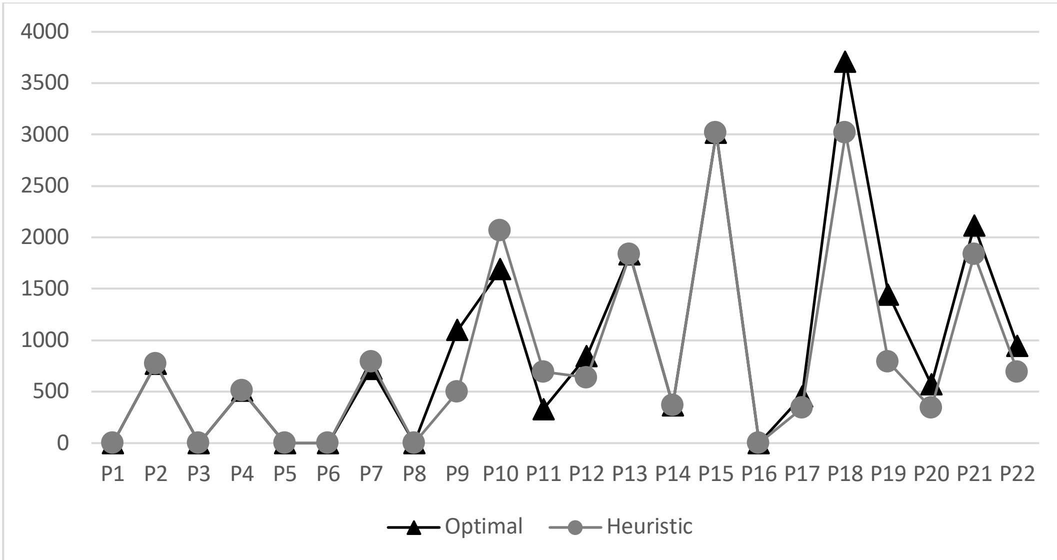


Figure 4.12 Penalty

In average, the results for 22 instances show 7% gap between the heuristic method and the optimal solutions. Figure 4.13 and Table 4.15 illustrate the total cost in the optimal case and the heuristic method in all instances.

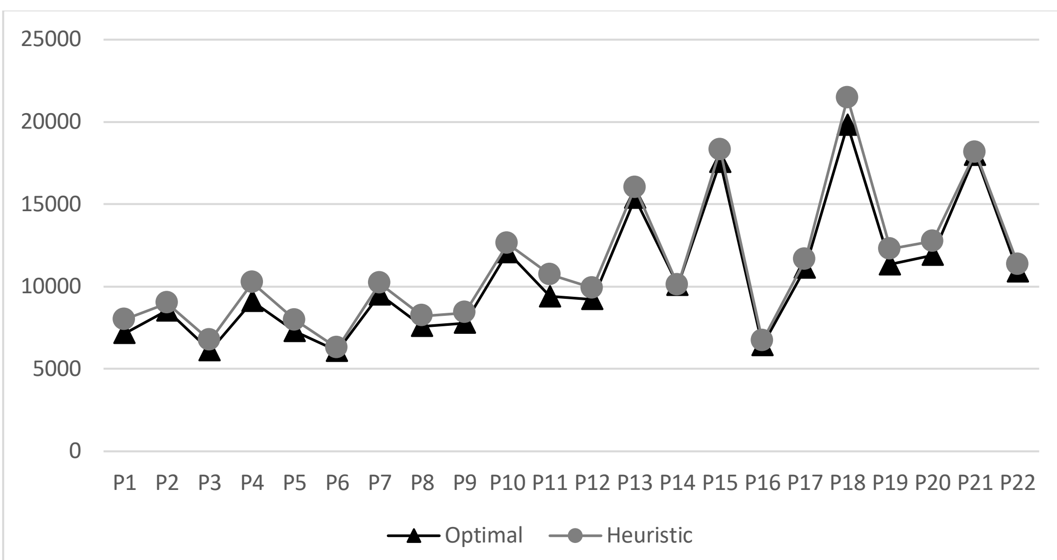


Figure 4.13 Total cost of each project

Table 4.15 Cost for each project

	Pu Cost		O Cost		H Cost		B Cost		Penalty		total	
Proj	**O**	**H**	**O**	**H**	**O**	**H**	**O**	**H**	**O**	**H**	**O**	**H**
P1	5842	5842	1625	1790	1642	2062	1940	1695	0	0	7169	7999
P2	5234	5234	1300	1630	1666	1786	425	425	770	770	8545	8995
P3	5078	5078	1300	1630	1654	1884	1915	1855	0	0	6117	6737
P4	5998	5998	1465	2115	1916	2164	770	550	510	510	9119	10237
P5	5432	5432	1300	1465	1496	2074	950	1000	0	0	7278	7971
P6	4212	4212	850	1020	1968	1968	935	935	0	0	6095	6265
P7	4023	4023	1208	1812	3576	3576	0	0	724	788	9531	10199
P8	5414	5414	868	1325	2169	2429	862	949	0	0	7589	8219
P9	5329	5329	1148	1722	1192	1110	984	246	1100	500	7785	8415
P10	6804	6804	1932	2350	2710	2788	1050	1400	1692	2068	12088	12610
P11	5240	5240	1172	1758	2820	3041	150	0	330	693	9412	10732
P12	5827	5827	1315	1857	2553	2901	1314	1314	848	636	9229	9907
P13	8912	8912	1200	2216	4104	4239	680	1190	1833	1833	15369	16010
P14	5929	5929	1262	1262	3004	3004	485	485	368	368	10078	10078
P15	8724	8724	1235	1976	4601	4601	0	0	3016	3016	17576	18317
P16	5234	5234	1211	1338	1475	1476	1497	1342	0	0	6423	6706
P17	7425	7425	820	1435	3465	3465	1022	1022	456	342	11144	11645
P18	9736	10784	1235	1976	5129	5691	0	0	3712	3016	19812	21467
P19	4608	5130	1208	1812	4096	4560	0	0	1444	788	11356	12290
P20	7635	9060	1025	1435	3563	4228	876	1022	570	342	11917	12755
P21	9848	10103	1500	2400	4536	5003	0	1190	2115	1833	17999	18149
P22	5576	6140	1172	1200	3233	3324	0	0	946	693	10927	11357
Ave	**6275**	**6405**	**1243**	**1691**	**2844**	**3062**	**721**	**755**	**929**	**827**	**10571**	**11230**

Table 4.16 shows the time spent to solve each instance and the total cost. It is clear from the table that the time is minimized compared to optimization. In average, each instance including 7 activities takes 48 minutes to solve while in the proposed heuristic only 8 seconds is spent on each example.

Table 4.16 Time spend to solve each instance

	Optimal		Heuristic	
Instances	**Time**	**Total cost**	**Time**	**Total cost**
Case 1	2h	7169	8s	7999
Case 2	17m	8545	10s	8995
Case 3	1h 45m	6117	10s	6737
Case 4	7m	9119	7s	10237
Case 5	23m	7278	9s	7971
Case 6	3m	6095	7s	6265
Case 7	2m	9531	10s	10199
Case 8	30m	7589	6s	8219
Case 9	1h 3m	7785	7s	8415
Case 10	1h	12088	9s	12610
Case 11	9m	9412	8s	10732
Case 12	34m	9229	8s	9907
Case 13	1m	15369	8s	16010
Case 14	30m	10078	7s	10078
Case 15	10m	17576	8s	18317
Case 16	6h 11m	6423	8s	6706
Case 17	16m	11144	7s	11645
Case 18	1m	19812	9s	21467
Case 19	56m	11356	10s	12290
Case 20	1m	11917	7s	12755
Case 21	1h 13m	17999	7s	18149
Case 22	21m	10927	8s	11357
Average	**48m**	**10571**	**8s**	**11230**

As it was mentioned, from 22 instances, in 5 cases both renewable and nonrenewable resources usage for an activity vary in different modes while in other examples only renewable resources have different usage in different modes and nonrenewable resources are constant during all modes. While the results for all instances show 7% gap, results considering only instances with multi-mode nonrenewable resources show 9% gap. We observe the developed heuristic is more practical in cases which the total cost is more relied on bonus, penalty, or holding cost.

4.5.3 Case study

The developed heuristic is applied in a case study. The project networks used here are from a company in northern Ontario. Two main projects in the company are considered and studied. The first project includes 327 activities in 10 stations, and the second project has 278 activities in 9 stations. The results for each project is shown here and the time spent is shown as well.

There are 7 main renewable resources in the company which should be assigned to projects in the beginning. In the first phase of the heuristic these 7 resources are assigned to each project. For this stage, it takes about 18 seconds to run the model and reach to a solution.

In the second phase, which is the scheduling of activities based on the assigned renewable resources, each station is considered separately. Aside from the precedence relations among the activities of a station, there is a precedence relation between stations as well. In this case, the stations are numbered based on their priority. Figure 4.14 shows the network for project 1.

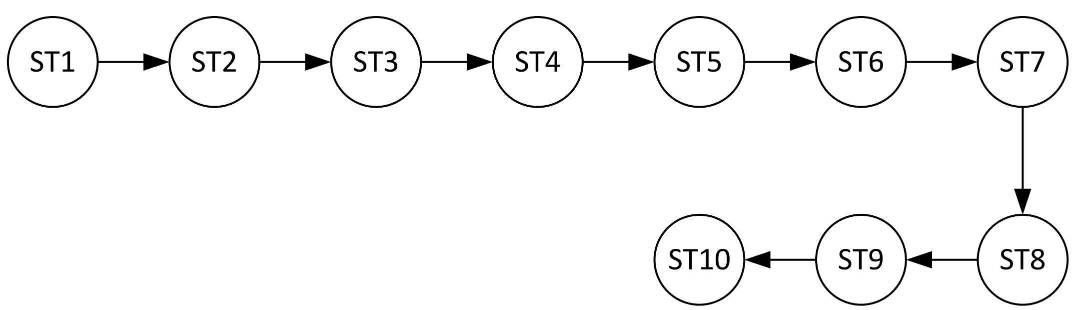


Figure 4.14 Project 1 network

The number of activities in each station and the time which it takes to schedule them is given in Table 4.17 and Table 4.18 for projects 1 and 2, respectively. Except stations 1 and 4 in project 1 and stations 1 and 3 in project 2 which have only one mode for their activities, the activities of other stations can be operated in two modes. Nonrenewable resources usage in all modes is constant for all stations.

Table 4.17 time spent to solve project 1 with 327 activities in the second phase of the heuristic

Project 1	**Number of activities**	**Time**
Station 1	47	3m 48s
Station 2	52	24s
Station 3	33	2m 49s
Station 4	79	32s
Station 5	6	6s
Station 6	21	12s
Station 7	37	1m 34s
Station 8	13	11s
Station 9	30	14s
Station 10	9	9s
	327	**10m**

Table 4.18 time spent to solve project 2 with 278 activities in the second phase of the heuristic

Project 1	**Number of activities**	**Time**
Station 1	63	2m 7s
Station 2	27	29m 20s
Station 3	52	7m
Station 4	20	40s
Station 5	18	46s
Station 6	17	20s
Station 7	9	6s
Station 8	21	45s
Station 9	51	1m 27s
	278	**42.5 m**

In phase 3, the heuristic is able to generate a schedule for ordering 500 nonrenewable resources in less than 2 minutes.

In total, a complete schedule can be generated within an hour for two projects of the company. In this schedule project 1 takes 190 days and project 2 takes 264 days to finish. The data related to cost are modified from the original data of the company. The results are shown in Table 4.19.

Table 4.19 Cost for schedule with two modes

order cost	115490
purchase cost	89367
holding cost	38162
Penalty	0
Bonus	58400
Total	184619

To monitor the effect of the order cost on total cost, we doubled the cost for ordering and reran the last phase. The results are reflected in Table 4.20.

Table 4.20 Cost for schedule with doubled order cost

order cost	212160
purchase cost	89367
holding cost	51996
Penalty	0
Bonus	58400
Total	295123

As it can be interpreted from the results, both order cost and holding cost are increased. While the total order cost is almost twice than before, the holding cost is increased by 36%.

The problem is solved by considering only the first mode for all activities as well. The results show, as it was expected, the projects take longer to finish. Project 1 ends after 250 days and it takes 345 days to end project 2. The computational time in the second phase for the first projects drops from 10 minutes to 5 minutes and from 42.5 minutes to 11.65 minutes for the second project.

4.6 Summary

In this chapter a new heuristic to solve MPMRCSMO has been proposed. The heuristic includes three phases. The first stage is to solve a nonlinear model which is solved by excel solver. This phase decides about the number of renewable resources assigned to each project. In stage two a scheduling model is solved for each project in the absence of nonrenewable resources. Finally, the last phase uses the schedules generated by the second phase to calculate the number of nonrenewable resources needed for each time period. A plan to order nonrenewable resources based on the generated schedules is produced in this phase.

The introduced heuristic takes minimum computational time to solve the problem. On average, the results from the heuristic show that there is 7% gap between heuristic and optimal solutions. The results for instances with multi-mode nonrenewable resource usage shows 9% gap in the heuristic solution. The developed heuristic shows a slightly better performance in cases which nonrenewable resource usage is constant in all modes. In cases which order cost has a dominant effect on total cost, the developed heuristic with a large gap in order cost is not the most efficient approach to take.

The results from applying the heuristic on a case study indicate that the model is able to solve larger size instances. Although, the results consider the ideal situation and also a lot of factors are not taken into account. However, the fact that an acceptable solution was obtained for a real-word case in less than an hour for such a large instance is promising.

CHAPTER 5: CONCLUSION AND FUTURE WORK

In this study, a new extension of project scheduling problem was investigated. In this extension, Multi-Project Resource Constrained Scheduling Problem with Material Ordering (MPMRCSMO) was introduced in Chapter 3. Some new constraints were added to the model to make it more practical and realistic. Resource dedication policy was considered and implemented. In Chapter 4 a new heuristic to solve the problem was developed and tested.

Using off-shelf optimization for solving the model takes a long time for even small instances and it cannot be applied on large and practical cases. The heuristic uses three phases to reach a suboptimal solution with minimum computational time. The approach uses three mathematical models in three phases. The first phase uses a nonlinear model to calculate the number of renewable resources dedicated to each project. The model uses the concept of resource constraint to decide the number of renewable resources. In the second phase, a linear model using the data from the first phase, generates the schedule for each project individually. In the last phase, another linear model is used to produce a plan for ordering nonrenewable resources based on the schedule generated in the second phase.

To validate the proposed model and compare it with the developed heuristic, a large number of example instances were tested and the results are compared. The first phase of the heuristic is solved by excel solver. Aside from that, the main model and the second and the third phases of the heuristic are solved with ILOG CPLEX. In the end, the proposed heuristic is applied to a real case problem based on a large scale facility in a manufacturing company in northern Ontario.

As future work, we plan to further improve the developed heuristic method for better performance in solving large size practical problems. Additionally, although the developed heuristic requires much less computing effort than optimization, the sub-problem solved in the second phase of the solution procedure is still a classic scheduling problem which is NP-Hard. A new method to tackle this problem can be studied in the future.

Appendix i

Main model Coded in ILOG CPLEX:

```
/*********************************************
 * OPL 12.6.3.0 Model
 * Author: umroot
 * Creation Date: Mar 10, 2018 at 3:29:35 PM
 *********************************************/

int NR = ...; //number of different kind of renewable resources//
int NK = ...; //number of different kind of nonrenewable resources//
int P = ...; //number of projects
int T = ...; // set of time sluts
int M = ...; //number of modes//
int N= ...;  //number of activity in projects

range Time= 1..T;  //range of time sluts
range Renewable= 1..NR;  //range of renewable resources
range Nonrenewable = 1..NK;  //range of nonrenewable resources
range Project = 1..P;  //range of projects
range Mode= 1..M;  //range of modes
range Activity = 1..N;

int DA[Activity][Mode] = ...;//Durations for activities of project 1
int DB[Activity][Mode] = ...;//Durations for activities of project 2//int DC[Activity][Mode] = ...;//Durations for activities of project 3

int D[Project][Activity][Mode]; // duration of activity j of project p operating in mode m

int UseR[Project][Activity][Mode][Renewable];//the usage of renewable resource r for the activity j of project p operating in mode m
int UseN[Project][Activity][Nonrenewable];//the usage of nonrenewable resource k for the activity j of project p
int W[Project]=...;//the weight of project p
int DD[Project]=...; // due date of project p

// data

int Use11[Activity][Mode] = ...;
int Use12[Activity][Mode] = ...;
int Use21[Activity][Mode] = ...;
int Use22[Activity][Mode] = ...;


int NUse1[Activity][Nonrenewable] = ...;
int NUse2[Activity][Nonrenewable] = ...;

int A[Nonrenewable]= ...; //order cost of material k
int H[Nonrenewable]= ...; // holding cost of each item k per unit of time
int BN[Project]= ...; //bunos for each day finishing project p before deadline
int PN[Project]=...; //penalty for each day delay in project p
int CAP[Renewable]=...; // capacity of renewable resource r


int CN[Nonrenewable] = ...; //cost of buying each brand new nonrenewable resource//
int ET[Project][Activity] = ...; //Early start time of activity j of project p//
```

```
int LT[Project][Activity] = ...; //Late start time of activity j of project pp//

dvar boolean S[Project][Activity][Mode][Time]; //1 if activity j operating at mode m starting at time t, 0 otherwise//


dvar int+ OR[Nonrenewable][Time]; //ordered amount of k at time t FIX THIS ONE
dvar int+ I[Nonrenewable][Time]; // inventory level of material k at time t    FIX THIS ONE
dvar boolean L[Nonrenewable][Time];// if material k is ordered at time t
dvar int+ DR[Project][Renewable]; // amount of renewabler resource r dedicated to project p     FIX THIS ONE


tuple Precedence {
int before;
int after;
}
{Precedence} AB[Project];   //precedence relationship for all projects
{Precedence} PreA = ...;
{Precedence} PreB = ...;


execute {                    //use of renewable resource r for activity j of project p operating in mode m
        for (var p in Project) {
                        for (var j in Activity) {
                                for (var m in Mode)                     {
                                        for (var r in Renewable)    {

                                        if (p == 1 && r==1)
                                        UseR[p][j][m][r]= Use11[j][m] ;

                                        if (p == 1 && r==2)
                                        UseR[p][j][m][r]= Use12[j][m] ;

                                        if (p == 2 && r==1)
                                        UseR[p][j][m][r]= Use21[j][m];

                                        if (p == 2 && r==2)
                                        UseR[p][j][m][r]= Use22[j][m] ;

                                }
                        }
                }
        }
}



execute {                  // use of nonrenewable resource k for activity j of project p operating in mode m
        for (var p in Project) {
                for (var j in Activity) {

                                for (var k in Nonrenewable) {

                                        if (p == 1)
                                        UseN[p][j][k]= NUse1[j][k] ;

                                        if (p == 2)
```

```
                                        UseN[p][j][k]= NUse2[j][k] ;

                              }
                    }
          }
}



execute {
          for (var p in Project) {
                    for (var j in Activity) {
                              for (var m in Mode) {

                              if (p == 1)
                              D[p][j][m]= DA[j][m];

                              if (p == 2)
                              D[p][j][m]= DB[j][m];

                              }
                    }
          }
}


execute {
          for (var p in Project) {

          if (p == 1)
          AB[p]= PreA;

          if (p == 2)
          AB[p]= PreB;

          }
}

dexpr int OrderCost =  sum ( k in Nonrenewable, t in Time) A[k]* L[k][t] ;
dexpr int PurchaceCost= sum(k in Nonrenewable, t in Time) OR[k][t]* CN[k];
dexpr int HoldingCost=  sum(k in Nonrenewable, t in Time) H[k]* I[k][t];
dexpr int Penalty=   sum(p in Project,m in Mode, u in Time: (u >= (DD[p]+1) && u<=T)) PN[p]*(u-
DD[p])*S[p][7][m][u]* W[p];
dexpr int Bunus=  sum(p in Project, m in Mode,q in Time:(q >= ET[p][7] && q<= (DD[p]-1))) BN[p]* (DD[p]-
q)*S[p][7][m][q]* W[p] ; //minimizing the total weighted tardiness of all projects



//The Model
minimize
 OrderCost + PurchaceCost + HoldingCost + Penalty - Bunus;

subject to {


 C1:  forall (p in Project , <i,j> in AB[p])
```

```
    sum (m in Mode, t in Time : t >= ET[p][i] && t<= LT[p][i])
  S[p][i][m][t] * (t + D[p][i][m]) <= sum (m in Mode, t in Time : (t >= ET[p][j] && t<= LT[p][j])) S[p][j][m][t] * t;

 C2: forall (j in Activity, p in Project)
   sum (m in Mode, t in Time : t >= ET[p][j] && t<= LT[p][j]) S[p][j][m][t] == 1; //each activity should start exac


 C3:   forall(r in Renewable)
  sum(p in Project) DR[p][r] == CAP[r];


 C4:    forall (p in Project, t in Time, r in Renewable )
     sum(m in Mode, j in Activity, q in Time:(q >= maxl (ET[p][j] , t- D[p][j][m]) && q<= minl(LT[p][j] , t)))
S[p][j][m][q] * UseR[p][j][m][r] <= DR[p][r] ; // capacity of renewable resources//

 C5:    forall (k in Nonrenewable, t in Time)
  OR[k][t] <= L[k][t]* 10000; // if we decide to order at time t we can have OR more than 0



 C8:    forall (k in Nonrenewable, t in Time: t==1)
    I[k][t] == OR[k][t] - sum(p in Project, m in Mode, j in Activity) UseN[p][j][k] * S[p][j][m][t]; //inventory level at
time period 1


 C9:    forall (k in Nonrenewable, t in Time: t>=2)
    I[k][t] == I[k][t-1]+ OR[k][t] - sum(p in Project, m in Mode, j in Activity) UseN[p][j][k] * S[p][j][m][t];
//inventory level at each time period



 C10:   forall (k in Nonrenewable, t in Time)

            OR[k][t] >=0; // Ordered amount of k can't be negative

 C11:   forall (k in Nonrenewable, t in Time: t>=2)
                         sum(p in Project, m in Mode, j in Activity) UseN[p][j][k] * S[p][j][m][t] <= I[k][t-1];//
nonrenewable resources used at each time should'nt be more than inventory'


 C12:   forall (k in Nonrenewable, t in Time: t==1)
            sum(p in Project, m in Mode, j in Activity) UseN[p][j][k] * S[p][j][m][t] <= I[k][1];// nonrenewable
resources used at each time should'nt be more than inventory'

 }

execute{
 for(var p in Project)
  for (var i in Activity)
   for (var m in Mode)
   for (var t in Time)
  if(S[p][i][m][t] == 1 )
   writeln("Project "+(p)+" of Activity "+(i)+ " from Mode "+(m)+" at time "+(t)) ;

}
```

```
execute{
 for (var k in Nonrenewable)
            for (var q in Time)
            if (I[k][q] >= 1)
            writeln("I: "+ (k) + " at time "+ (q)+ " inventory is " + I[k][q]);
}

execute{
 for (var k in Nonrenewable)
            for (var q in Time)
            if (OR[k][q] >> 0)
            writeln("OR "+ (k) + " at time "+ (q)+ " ordered amount of " + OR[k][q]);
}

 execute{
 for (var k in Nonrenewable)
            for (var q in Time)
            if (L[k][q] == 1)
            writeln("Nonrenewable "+ (k) + " at time "+ (q)+ " is ordered" );

 //         writeln ("the best solution is " + (obj));

}
execute{

 writeln ("Order Cost= " + (OrderCost));
 writeln ("Purchace Cost= " + (PurchaceCost));
 writeln ("Holding Cost= " + (HoldingCost));
 writeln ("Penalty= " + (Penalty));
 writeln ("Bunus= " + (Bunus));
}
```

Appendix ii

Second phase of the heuristic:

```
* OPL 12.7.1.0 Model
 * Author: umroot
 * Creation Date: Mar 18, 2018 at 12:26:20 PM
 *********************************************/

int NR = ...; //number of different kind of renewable resources//
int P = ...; //number of projects
int T = ...; // set of time sluts
int M = ...; //number of modes//
int Nj= ...;  //number of activity in project p

range Time= 1..T;  //range of time sluts
range Renewable= 1..NR;  //range of renewable resources
range Project = 1..P;  //range of projects
range Mode= 1..M;  //range of modes
range Activity = 1..Nj;

int DA[Activity][Mode] = ...;//Durations for activities of project 1
int DB[Activity][Mode] = ...;

int D[Project][Activity][Mode]; // duration of activity j of project p operating in mode m

int UseR[Project][Activity][Mode][Renewable];//the usage of renewable resource r for the activity j of project p
operating in mode m

int Use11[Activity][Mode] = ...;
int Use12[Activity][Mode] = ...;
int Use21[Activity][Mode] = ...;
int Use22[Activity][Mode] = ...;

int ET[Project][Activity] = ...; //Early start time of activity j of project p//
int LT[Project][Activity] = ...; //Late start time of activity j of project pp//
int DR[Project][Renewable]=...;; // amount of renewabler resource r dedicated to project p
int W[Project]=...;//the weight of project p
int DD[Project]=...; // due date of project p

int BN[Project]= ...; //bunos for each day finishing project p before deadline
int PN[Project]=...; //penalty for each day delay in project p

dvar boolean S[Project][Activity][Mode][Time]; //1 if activity j operating at mode m starting at time t, 0 otherwise//

tuple Precedence {
int before;
int after;
}
{Precedence} AB[Project];   //precedence relationship for all projects
{Precedence} PreA = ...;
{Precedence} PreB = ...;

execute {                    //use of renewable resource r for activity j of project p operating in mode m
        for (var p in Project) {
                        for (var j in Activity) {
```

```
                    for (var m in Mode)            {
                        for (var r in Renewable)    {

                        if (p == 1 && r==1)
                        UseR[p][j][m][r]= Use11[j][m] ;

                        if (p == 1 && r==2)
                        UseR[p][j][m][r]= Use12[j][m] ;

                        if (p == 2 && r==1)
                        UseR[p][j][m][r]= Use21[j][m] ;

                        if (p == 2 && r==2)
                        UseR[p][j][m][r]= Use22[j][m] ;


                    }
                }
            }
        }
}


execute {
        for (var p in Project) {
            for (var j in Activity) {
                for (var m in Mode) {

                if (p == 1)
                D[p][j][m]= DA[j][m];
                if (p == 2)
                D[p][j][m]= DB[j][m];


                }
            }
        }
}


execute {
        for (var p in Project) {

        if (p == 1)
        AB[p]= PreA;

        if (p == 2)
        AB[p]= PreB;


        }
}

dexpr int Bonus = sum(p in Project,m in Mode, q in Time:(q >= ET[p][7] && q<= (DD[p]-1))) BN[p]* (DD[p]-
q)*S[p][7][m][q]* W[p];
```

```
dexpr int Penalty = sum(p in Project,m in Mode ,u in Time: (u >= (DD[p]+1) && u<=T)) PN[p]*(u-
DD[p])*S[p][7][m][u]* W[p];

maximize
Bonus - Penalty ;




 subject to{

 C1:  forall (p in Project , <i,j> in AB[p])
     sum (m in Mode, t in Time : t >= ET[p][i] && t<= LT[p][i])
  S[p][i][m][t] * (t + D[p][i][m]) <= sum (m in Mode, t in Time : (t >= ET[p][j] && t<= LT[p][j])) S[p][j][m][t] * t;

 C2: forall (j in Activity, p in Project)
    sum (m in Mode, t in Time : t >= ET[p][j] && t<= LT[p][j]) S[p][j][m][t] == 1; //each activity should start exac

 C4:    forall (p in Project, t in Time, r in Renewable )
      sum(m in Mode, j in Activity, q in Time:(q >= maxl (ET[p][j] , t- D[p][j][m]) && q<= minl(LT[p][j] , t)))
S[p][j][m][q] * UseR[p][j][m][r] <= DR[p][r] ; // capacity of renewable resources//



 }

 execute{

 for (var p in Project)
            for (var i in Activity)
                        for (var m in Mode)
                                    for (var t in Time)

                                    if (S[p][i][m][t]==1)

            writeln ("Project "+ (p)+" Activity "+(i)+ " Mode "+(m)+ "at time "+(t));

            writeln ("Penalty is "+(Penalty));
            writeln ("Bonus is "+(Bonus));

}
```

Appendix iii

Third phase of the heuristic:

```
* OPL 12.7.1.0 Model
 * Author: umroot
 * Creation Date: Mar 18, 2018 at 3:36:42 PM
 *********************************************/

int NK = ...; //number of different kind of nonrenewable resources//
int P = ...; //number of projects
int T = ...; // set of time sluts
int Nj= ...;  //number of activity in project p

range Time= 0..T;  //range of time sluts
range Nonrenewable = 1..NK;  //range of nonrenewable resources
range Project = 1..P;  //range of projects
range Activity = 1..Nj;

int UseN[Project][Activity][Nonrenewable];//the usage of nonrenewable resource k for the activity j of project p
operating in mode m


int NUse11[Activity] = ...;
int NUse12[Activity] = ...;
int NUse21[Activity] = ...;
int NUse22[Activity] = ...;


int A[Nonrenewable]= ...; //order cost of material k
int H[Nonrenewable]= ...; // holding cost of each item k per unit of time
int CN[Nonrenewable] = ...; //cost of buying each brand new nonrenewable resource//

int S[Project][Activity][Time];
int S11[Activity][Time] = ...;
int S21[Activity][Time] = ...;


execute {                // start time and mode of each activity in each project
        for (var p in Project) {
                for (var j in Activity) {
                        for (var t in Time) {

                                        if (p == 1 )
                                        S[p][j][t]= S11[j][t] ;

                                        if (p == 2 )
                                        S[p][j][t]= S21[j][t] ;

                        }
                }
        }
}


dvar int+ OR[Nonrenewable][Time]; //ordered amount of k at time t FIX THIS ONE
```

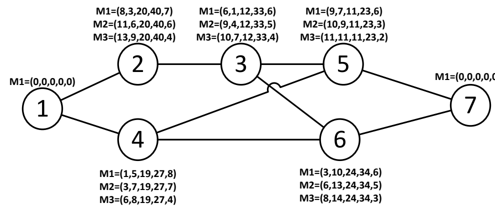
```
dvar int+ I[Nonrenewable][Time]; // inventory level of material k at time t    FIX THIS ONE
dvar boolean L[Nonrenewable][Time];// if material k is ordered at time t

execute {                  // use of nonrenewable resource k for activity j of project p operating in mode m
        for (var p in Project) {
                for (var j in Activity) {
                        for (var k in Nonrenewable) {

                                        if (p == 1 && k==1)
                                        UseN[p][j][k]= NUse11[j] ;

                                        if (p == 1 && k==2)
                                        UseN[p][j][k]= NUse12[j] ;

                                        if (p == 2 && k==1)
                                        UseN[p][j][k]= NUse21[j] ;

                                        if (p == 2 && k==2)
                                        UseN[p][j][k]= NUse22[j] ;

                        }
                }
        }
}

dexpr int OrderCost= sum ( k in Nonrenewable, t in Time) A[k]* L[k][t];
dexpr int PurchaseCost= sum(k in Nonrenewable, t in Time) OR[k][t]* CN[k];
dexpr int HoldingCost= sum(k in Nonrenewable, t in Time) H[k]* I[k][t];

//The Model
minimize
OrderCost + PurchaseCost+ HoldingCost;


 subject to
 {
 C5:    forall (k in Nonrenewable, t in Time)
 OR[k][t] <= L[k][t]* 100000; // if we decide to order at time t we can have OR more than 0


 C8:    forall (k in Nonrenewable, t in Time: t==1)
    I[k][t] == OR[k][t] - sum(p in Project, j in Activity) UseN[p][j][k] * S[p][j][t]; //inventory level at time period 1


 C9:    forall (k in Nonrenewable, t in Time: t>=2)
    I[k][t] == I[k][t-1]+ OR[k][t] - sum(p in Project, j in Activity) UseN[p][j][k] * S[p][j][t]; //inventory level at each
time period


 C11:   forall (k in Nonrenewable, t in Time: t>=2)
                    sum(p in Project, j in Activity) UseN[p][j][k] * S[p][j][t] <= I[k][t-1];// nonrenewable resources
used at each time should'nt be more than inventory'


 C12:   forall (k in Nonrenewable, t in Time: t==1)
```

```
            sum(p in Project, j in Activity) UseN[p][j][k] * S[p][j][t] <= I[k][t-1];// nonrenewable resources used at
each time should'nt be more than inventory'

    }

 execute {

                                    writeln ("Order Cost is " + OrderCost);
                                    writeln ("Purchase Cost is " + PurchaseCost);
                                    writeln ("Holding Cost is " + HoldingCost);
 }
```

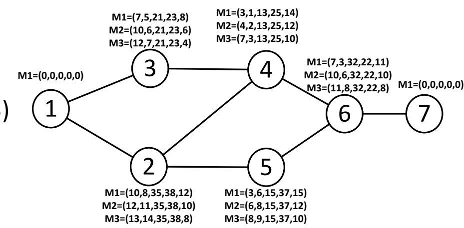
Appendix iv:

Data used for the model

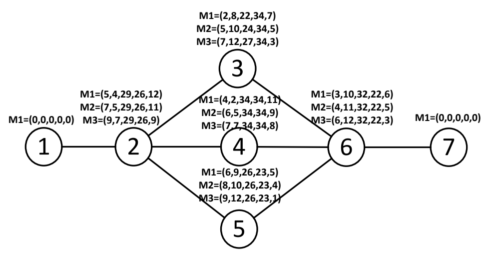
(P1)



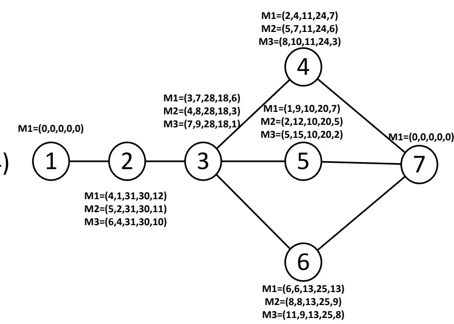
(P3)



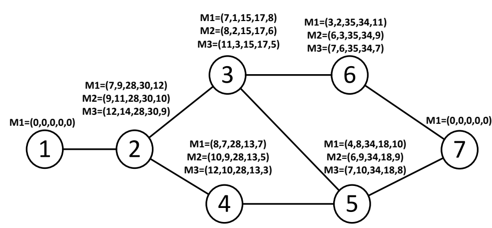
(P2)



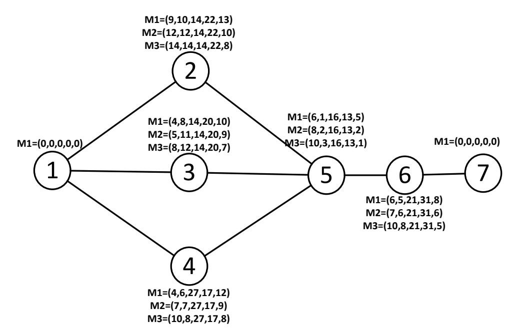
(P4)



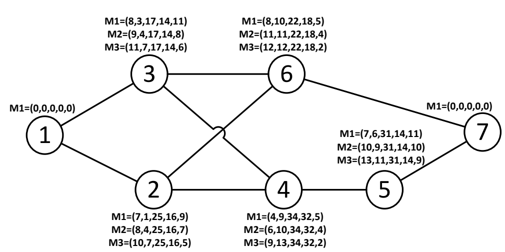
(P5)



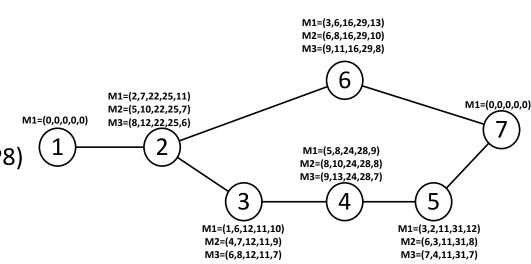
(P6)



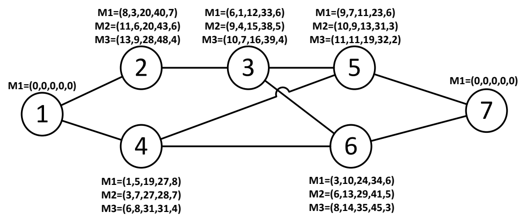
(P7)



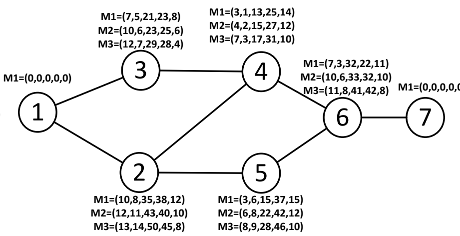
(P8)



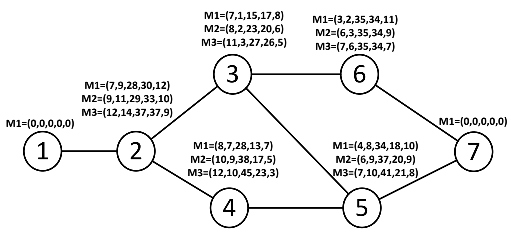
(P9)



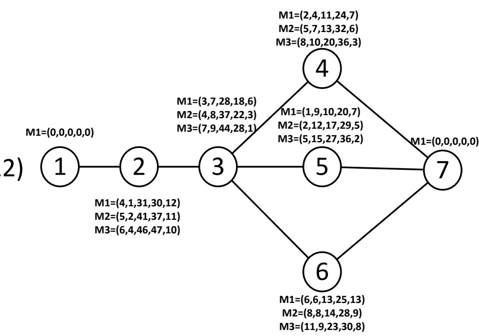
(P10)



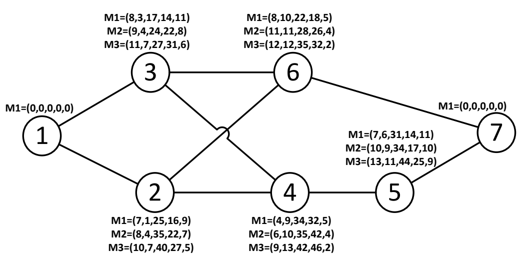
(P11)



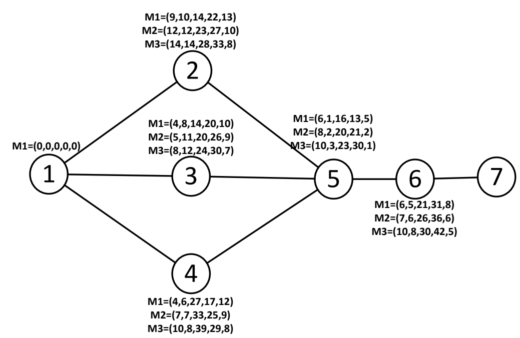
(P12)



(P13)



(P14)



These 14 projects are used in different combination of two to create a portfolio. 22 portfolios used in this study and the information related to them is given here. The first number in each row belongs to the first project and the second number belongs to the second project. In the network information section, the projects which are used in the example are mentioned. For instance, in example 1, projects 1 and 2 are considered.

Example	1
Due Date	30,31
Penalty	110,85
Bonus	110,85
Network Information	P1 & P2
order cost	160,165
holding cost	2,4
Cap of resources	30,30
price of nonrenewable	10,12

Example	2
Due Date	30,31
Penalty	110,85
Bonus	110,85
Network Information	P3 & P4
order cost	160,165
holding cost	2,4
Cap of resources	30,30
price of nonrenewable	10,12

Example	3
Due Date	30,31
Penalty	110,85
Bonus	110,85
Network Information	P1 & P4
order cost	160,165
holding cost	2,4
Cap of resources	30,30
price of nonrenewable	10,12

Example	4
Due Date	30,31
Penalty	110,85
Bonus	110,85
Network Information	P2 & P3
order cost	160,165
holding cost	2,4
Cap of resources	30,30
price of nonrenewable	10,12

Example	5
Due Date	30,31
Penalty	110,85
Bonus	110,85
Network Information	P2 & P4
order cost	160,165
holding cost	2,4
Cap of resources	30,30
price of nonrenewable	10,12

Example	6
Due Date	30,30
Penalty	85,85
Bonus	85,85
Network Information	P2 & P4
order cost	70,100
holding cost	4,4
Cap of resources	30,30
price of nonrenewable	7,10

Example	7
Due Date	22,25
Penalty	66,196
Bonus	134,179
Network Information	P5 & P6
order cost	135,167
holding cost	8,8
Cap of resources	30,30
price of nonrenewable	9,9

Example	8
Due Date	34,32
Penalty	202,236
Bonus	85,87
Network Information	P5 & P7
order cost	80,137
holding cost	5,4
Cap of resources	30,30
price of nonrenewable	14,8

Example	9
Due Date	27,28
Penalty	160,100
Bonus	246,204
Network Information	P6 & P8
order cost	174,113
holding cost	3,2
Cap of resources	30,30
price of nonrenewable	16,11

Example	10
Due Date	32,21
Penalty	113,188
Bonus	175,223
Network Information	P2 & P8
order cost	218,212
holding cost	1,8
Cap of resources	30,30
price of nonrenewable	16,12

Example	11
Due Date	26,20
Penalty	176,55
Bonus	150,176
Network Information	P4 & P7
order cost	135,158
holding cost	7,6
Cap of resources	30,30
price of nonrenewable	16,8

Example	12
Due Date	26,21
Penalty	219,53
Bonus	146,95
Network Information	P1 & P8
order cost	93,170
holding cost	10,3
Cap of resources	30,30
price of nonrenewable	16,11

Example	13
Due Date	24,34
Penalty	141,246
Bonus	113,170
Network Information	P3 & P6
order cost	184,116
holding cost	9,9
Cap of resources	30,30
price of nonrenewable	19,20

Example	14
Due Date	29,30
Penalty	94,184
Bonus	97,175
Network Information	P4 & P8
order cost	154,82
holding cost	3,10
Cap of resources	30,30
price of nonrenewable	13,15

Example	15
Due Date	26,22
Penalty	232,116
Bonus	146,115
Network Information	P3 & P7
order cost	133,114
holding cost	10,9
Cap of resources	30,30
price of nonrenewable	20,16

Example	16
Due Date	25,26
Penalty	244,50
Bonus	172,155
Network Information	1,7
order cost	192,127
holding cost	P1 & P4
Cap of resources	30,30
price of nonrenewable	8,14

Example	17
Due Date	25,22
Penalty	159,57
Bonus	146,82
Network Information	P1 & P5
order cost	111,94
holding cost	7,7
Cap of resources	30,30
price of nonrenewable	15,15

Example	18
Due Date	26,22
Penalty	232,116
Bonus	146,115
Weight	1,1
Network Information	P9 & P13
order cost	133,114
holding cost	10,9
Cap of resources	30,30
price of nonrenewable	20,16

Example	19
Due Date	22,25
Penalty	66,196
Bonus	134,179
Network Information	P11 & P14
order cost	135,167
holding cost	8,8
Cap of resources	30,30
price of nonrenewable	9,9

Example	20
Due Date	25,22
Penalty	159,57
Bonus	146,82
Network Information	P9 & P11
order cost	111,94
holding cost	7,7
Cap of resources	30,30
price of nonrenewable	15,15

Example	21
Due Date	24,34
Penalty	141,246
Bonus	113,170
Network Information	P10 & P14
order cost	184,116
holding cost	9,9
Cap of resources	30,30
price of nonrenewable	19,20

Example	22
Due Date	26,20
Penalty	176,55
Bonus	150,176
Network Information	P12 & P13
order cost	135,158
holding cost	7,6
Cap of resources	30,30
price of nonrenewable	16,8

References

[1] U. Beşikci, Ü. Bilge and G. Ulusoy, "Multi-mode resource constrained multi-project scheduling and resource portfolio problem," *Eur. J. Oper. Res.,* vol. 240, *(1),* pp. 22-31, 2015.

[2] U. Beşikci, Ü. Bilge and G. Ulusoy, "Resource dedication problem in a multi-project environment," *Flexible Services and Manufacturing Journal,* vol. 25, *(1-2),* pp. 206-229, 2013.

[3] F. F. Boctor, "Resource-constrained project scheduling by simulated annealing," *Int J Prod Res,* vol. 34, *(8),* pp. 2335-2351, 1996.

[4] F. F. Boctor, "Heuristics for scheduling projects with resource restrictions and several resource-duration modes," *Int. J. of Pro. Res.,* vol. 31, *(11),* pp. 2547-2558, 1993.

[5] T. R. Browning and A. A. Yassine, "Resource-constrained multi-project scheduling: Priority rule performance revisited," *Int J Prod Econ,* vol. 126, *(2),* pp. 212-228, 2010.

[6] J. Coelho and M. Vanhoucke, "Multi-mode resource-constrained project scheduling using RCPSP and SAT solvers," *Eur. J. Oper. Res.,* vol. 213, *(1),* pp. 73-82, 2011.

[7] R. De Boer, *Resource-Constrained Multi-Project Management,* 1998.

[8] E. Demeulemeester *et al*, "New computational results on the discrete time," *J. Oper. Res. Soc.,* vol. 49, *(11),* pp. 1153-1163, 1998.

[9] E. Demeulemeester, "Minimizing resource availability costs in time-limited project networks," *Management Science,* vol. 41, *(10),* pp. 1590-1598, 1995.

[10] E. Demeulemeester, "Minimizing resource availability costs in time-limited project networks," *Management Science,* vol. 41, *(10),* pp. 1590-1598, 1995.

[11] E. L. Demeulemeester, W. S. Herroelen and S. E. Elmaghraby, "Optimal procedures for the discrete time/cost trade-off problem in project networks," *Eur. J. Oper. Res.,* vol. 88, *(1),* pp. 50-68, 1996.

[12] E. Demeulemeester, B. D. REYCK and W. Herroelen, "The discrete time/resource trade-off problem in project networks: a branch-and-bound approach," *IIE Transactions,* vol. 32, *(11),* pp. 1059-1069, 2000.

[13] E. Demeulemeester, M. Vanhoucke and W. Herroelen, "RanGen: A random network generator for activity-on-the-node networks," *Journal of Scheduling,* vol. 6, *(1),* pp. 17-38, 2003.

[14] A. Drexl and J. Gruenewald, "Nonpreemptive multi-mode resource-constrained project scheduling," *IIE Transactions,* vol. 25, *(5),* pp. 74-81, 1993.

[15] S. Elloumi and P. Fortemps, "A hybrid rank-based evolutionary algorithm applied to multi-mode resource-constrained project scheduling problem," *Eur. J. Oper. Res.,* vol. 205, *(1),* pp. 31-41, 2010.

[16] N. Gademann and M. Schutten, "Linear-programming-based heuristics for project capacity planning," *Iie Transactions,* vol. 37, *(2),* pp. 153-165, 2005.

[17] M. J. Geiger, "A multi-threaded local search algorithm and computer implementation for the multi-mode, resource-constrained multi-project scheduling problem," *Eur. J. Oper. Res.,* vol. 256, *(3),* pp. 729-741, 2017.

[18] P. Ghoddousi *et al*, "Multi-mode resource-constrained discrete time–cost-resource optimization in project scheduling using non-dominated sorting genetic algorithm," *Autom. Constr.,* vol. 30, pp. 216-227, 2013.

[19] S. Hartmann and D. Briskorn, "A survey of variants and extensions of the resource-constrained project scheduling problem," *Eur. J. Oper. Res.,* vol. 207, *(1),* pp. 1-14, 2010.

[20] S. Hartmann and A. Drexl, *Project Scheduling with Multiple Modes: A Comparison of Exact Algorithms,* 1997.

[21] W. Herroelen, E. Demeulemeester and B. De Reyck, "A classification scheme for project scheduling problems," 1998.

[22] R. Kolisch and S. Hartmann, "Experimental investigation of heuristics for resource-constrained project scheduling: An update," *Eur. J. Oper. Res.,* vol. 174, *(1),* pp. 23-37, 2006.

[23] R. Kolisch and R. Padman, "An integrated survey of deterministic project scheduling," *Omega,* vol. 29, *(3),* pp. 249-272, 2001.

[24] D. Krüger and A. Scholl, "A heuristic solution framework for the resource constrained (multi-) project scheduling problem with sequence-dependent transfer times," *Eur. J. Oper. Res.,* vol. 197, *(2),* pp. 492-508, 2009.

[25] I. S. Kurtulus and S. C. Narula, "Multi-project scheduling: Analysis of project performance," *IIE Transactions,* vol. 17, *(1),* pp. 58-66, 1985.

[26] S. R. Lawrence and T. E. Morton, "Resource-constrained multi-project scheduling with tardy costs: Comparing myopic, bottleneck, and resource pricing heuristics," *Eur. J. Oper. Res.,* vol. 64, *(2),* pp. 168-187, 1993.

[27] A. Lova, C. Maroto and P. Tormos, "A multicriteria heuristic method to improve resource allocation in multiproject scheduling," *Eur. J. Oper. Res.,* vol. 127, *(2),* pp. 408-424, 2000.

[28] M. Mika, G. Waligora and J. Węglarz, "Tabu search for multi-mode resource-constrained project scheduling with schedule-dependent setup times," *Eur. J. Oper. Res.,* vol. 187, *(3),* pp. 1238-1250, 2008.

[29] M. Mika, G. Waligóra and J. Węglarz, "Simulated annealing and tabu search for multi-mode resource-constrained project scheduling with positive discounted cash flows and different payment models," *Eur. J. Oper. Res.,* vol. 164, *(3),* pp. 639-668, 2005.

[30] A. Mingozzi *et al*, "An exact algorithm for the resource-constrained project scheduling problem based on a new mathematical formulation," *Management Science,* vol. 44, *(5),* pp. 714-729, 1998.

[31] M. Mori and C. C. Tseng, "A genetic algorithm for multi-mode resource constrained project scheduling problem," *Eur. J. Oper. Res.,* vol. 100, *(1),* pp. 134-141, 1997.

[32] A. Naber and R. Kolisch, "MIP models for resource-constrained project scheduling with flexible resource profiles," *Eur. J. Oper. Res.,* vol. 239, *(2),* pp. 335-348, 2014.

[33] J. H. Patterson, "Project scheduling: The effects of problem structure on heuristic performance," *Naval Research Logistics (NRL),* vol. 23, *(1),* pp. 95-123, 1976.

[34] J. H. Payne, "Management of multiple simultaneous projects: a state-of-the-art review," *Int. J. Project Manage.,* vol. 13, *(3),* pp. 163-168, 1995.

[35] J. H. Payne, "Introducing formal project management into a traditional, functionally structured organization," *Int. J. Project Manage.,* vol. 11, *(4),* pp. 239-243, 1993.

[36] Á. Pérez *et al*, "A multi-objective approach for a project scheduling problem with due dates and temporal constraints infeasibilities," *Int. J. Prod Res,* vol. 52, *(13),* pp. 3950-3965, 2014.

[37] D. Pinha, R. Ahluwalia and P. Senna, "The Combinatorial Multi-Mode Resource Constrained Multi-Project Scheduling Problem," *Int. J. Supply and Operations Management*, vol. 3, *(3),* pp. 1391, 2016.

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

[39] M. Ranjbar, B. De Reyck and F. Kianfar, "A hybrid scatter search for the discrete time/resource trade-off problem in project scheduling," *Eur. J. Oper. Res.,* vol. 193, *(1),* pp. 35-48, 2009.

[40] F. Salewski, A. Schirmer and A. Drexl, "Project scheduling under resource and mode identity constraints: Model, complexity, methods, and application," *Eur. J. Oper. Res.,* vol. 102, *(1),* pp. 88-110, 1997.

[41] R. Słowiński, "Two approaches to problems of resource allocation among project activities—a comparative study," *J. Oper. Res. Soc.,* vol. 31, *(8),* pp. 711-723, 1980.

[42] D. E. Smith-Daniels and N. J. Aquilano, "Constrained resource project scheduling subject to material constraints," *J. Oper. Manage.,* vol. 4, *(4),* pp. 369-387, 1984.

[43] M. G. Speranza and C. Vercellis, "Hierarchical models for multi-project planning and scheduling," *Eur. J. Oper. Res.,* vol. 64, *(2),* pp. 312-325, 1993.

[44] T. A. Toffolo *et al*, "An integer programming approach to the multimode resource-constrained multiproject scheduling problem," *J. Sche,* vol. 19, *(3),* pp. 295-307, 2016.

[45] V. Van Peteghem and M. Vanhoucke, "An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances," *Eur. J. Oper. Res.,* vol. 235, *(1),* pp. 62-72, 2014.

[46] V. Van Peteghem and M. Vanhoucke, "Using resource scarceness characteristics to solve the multi-mode resource-constrained project scheduling problem," *J. Heuristics,* vol. 17, *(6),* pp. 705-728, 2011.

[47] V. Van Peteghem and M. Vanhoucke, "A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem," *Eur. J. Oper. Res.,* vol. 201, *(2),* pp. 409-418, 2010.

[48] W. Wang *et al*, "Multi-objective optimization model for multi-project scheduling on critical chain," *Adv. Eng. Software,* vol. 68, pp. 33-39, 2014.

[49] J. Węglarz *et al*, "Project scheduling with finite or infinite number of activity processing modes–A survey," *Eur. J. Oper. Res.,* vol. 208, *(3),* pp. 177-205, 2011.

[50] J. C. Zapata, B. M. Hodge and G. V. Reklaitis, "The multimode resource constrained multiproject scheduling problem: Alternative formulations," *Aiche j.,* vol. 54, *(8),* pp. 2101-2119, 2008.

[51] N. Zoraghi *et al*, "Multi-mode resource-constrained project scheduling problem with material ordering under bonus–penalty policies," *Top,* vol. 25, *(1),* pp. 49-79, 2017.