Development of Heuristics for Integrated Selection and Scheduling of Projects

Submitted in

fulfillment of the requirements for the degree of

Doctor of Philosophy

by

MANISH KUMAR

ID: 2014RME9540

Under the supervision of

Prof. M. L. Mittal & Dr. Gunjan Soni

DEPARTMENT OF MECHANICAL ENGINEERING

MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR

May 2019

Respectfully Dedicated to

My Parents, Wife, Siblings

&

Teachers

MALAVIYA NATIONAL INSTITUTE OFTECHNOLOGY
JAIPUR – 302017 (RAJASTHAN) INDIA

DECLARATION

I, **Manish Kumar**, declare that this thesis titled, **“Development of Heuristics for Integrated Selection and Scheduling of Projects”** and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this institute.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this institute or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself, jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date: May 2019

Manish Kumar
(2014RME9540)

MALAVIYA NATIONAL INSTITUTE OFTECHNOLOGY
JAIPUR – 302017 (RAJASTHAN) INDIA

CERTIFICATE

This is to certify that the thesis entitled **“Development of Heuristics for Integrated Selection and Scheduling of Projects”** being submitted by **Mr. Manish Kumar (ID No.: 2014RME9540)** is a bonafide research work carried out under my supervision and guidance in fulfilment of the requirement for the award of the degree of **Doctor of Philosophy** in the Department of Mechanical Engineering, Malaviya National Institute of Technology, Jaipur, India. The matter embodied in this thesis is original and has not been submitted to any other University or Institute for the award of any other degree.

Prof. M. L. Mittal
(Supervisor)
Professor
Department of
Mechanical Engineering,
MNIT, Jaipur
India

Dr. Gunjan Soni
(Co-Supervisor)
Assistant Professor
Department of
Mechanical Engineering,
MNIT, Jaipur
India

Place: Jaipur.
Date: May 2019

ACKNOWLEDGEMENT

It is a matter of great pleasure and privilege for me to present this thesis entitled **“Development of Heuristics for Integrated Selection and Scheduling of Projects”**.

First, I want to thank the God for all the blessings and for giving the opportunity to fulfil my dreams. Next, I take this opportunity to express my deep sense of gratitude and respect to my supervisors **Prof. M. L. Mittal** and **Dr. Gunjan Soni** for their patience, motivation, enthusiasm, and immense knowledge. Their excellent guidance, advice, feedback and encouragement have been fundamental to the completion of this project.

Besides my supervisors, I would like to thank my DREC members: Prof. A.P.S. Rathore and Prof. Rakesh Jain for their insightful comments, constructive inputs, and encouragement given during PhD progress presentations. I am also grateful to Prof. Dilip Sharma (Head, Mechanical Engineering Department) for his help and motivation during the study.

My sincere thanks also go to my friends Mr. Dheeraj Joshi, Mr. Vinod Yadav, Mr. Pal Manojkumar Ramchandra, Mr. Ramkaran Yadav and Mr. Gaurav Kumar Badhotiya for their continuous moral support and help through their knowledge in some dimensions of my work. I also extend sincere thanks to other fellow PhD scholars and all those people, whom I have not mentioned, but they had helped me directly or indirectly during the work.

Last, but by no means the least, I would like to thank my parents and my wife for their continuous support, encouragement and sacrifices for this endeavour that has taken such a large amount of my time. My PhD would not have been possible without their inseparable support, prayers and sacrifices. Especially, I am indebted to my wife who stood by me for the successful completion of this work. It would be an injustice to express her support, patience and sacrifices in few words.

Manish Kumar
2014RME9540

ABSTRACT

Most of the organizations deal with multiple projects running simultaneously sharing a common pool of resources. Such organizations need efficient decision making in selecting the right mix of the projects and scheduling them. Traditionally, these two activities are performed in a sequential manner. With this sequential approach, however, it may not be possible to schedule the selected projects within the stipulated time frame. This may require some adjustments in selection and scheduling of the projects leading to sub-optimality. This situation can be avoided by consideration of scheduling while selecting the projects. Usually, the projects are highly interdependent in nature. Also, the limited availability of the resources and their efficient utilization are big issues to be kept in mind while decision making. The integration of selection and scheduling processes makes the decision making more complex but enhances the quality of the decision by including feasible and better projects. In recent years, the integrated problem of project selection and scheduling has gained the attention of many researchers.

Literature review on project selection and scheduling reveals that very little work has been done towards integrated project selection and scheduling. Further, the existing models for integrated project selection and scheduling suffer from a number of limitations. One of the main concerns is the technical interdependencies: mutual exclusiveness and complementariness. Mutual exclusiveness has been considered by a few researchers but complementariness is largely ignored. Another major limitation of the existing research is non-consideration of benefit interdependencies which are quite important in multiproduct scenarios. It is also observed that the problem of reinvestment of benefits and the problem of efficient utilization of the limited resources have not adequately addressed in the literature of integrated project selection and scheduling of projects.

Motivated by the abovementioned facts, this thesis aims at developing the models for the integrated selection and scheduling of the projects which handle the interdependencies among the projects and optimize the utilization of resources also. The research work in this thesis is carried out in three parts. In the first part, a single-objective model for the problem is developed considering technical interdependencies. In the second part, a multi-objective model is developed using the reinvestment strategy. In the third part, the multi-objective model is extended to consider benefit interdependencies.

The thesis starts by introducing a single objective model for integrated selection and scheduling of the projects formulated as a zero-one integer linear program. The proposed model considers two types of technical interdependencies: mutual exclusiveness and complementariness. Three meta-heuristic algorithms: TLBO, TS and Hybrid TLBO-TS have been proposed to solve the model. Taguchi approach is used to tune the parameters of the algorithms. The proposed algorithms are tested on four different complexity level data sets generated in this research. The proposed algorithms are compared with each other and are also compared with the SFLA existing in the literature. The results show that the proposed Hybrid TLBO-TS algorithm outperforms all other algorithms. The scope of this model is limited to the selection and scheduling of projects to maximize the expected benefit of the portfolio.

Optimal use of limited resources is a big challenge. In real life situations, underutilization of cost-intensive resources such as experts, machines etc. affect the expected benefit from the project. From literature analysis, it is clear that minimization of the resource usage variation is the best way to handle the underutilization of cost-intensive resources. The proposed model is extended to minimize the variation in the usage of resources. A mixed integer linear programming model (MILP) is presented for the problem with two objectives: maximization of total expected benefit and minimization of resource usage variation. Benefit from a project is considered to be time sensitive and added to the budget for consideration of more projects. Two types of technical interdependencies: mutual exclusiveness and complementariness are considered. The problem has been formulated as a multi-objective integer linear program. A modified Non-dominated Sorting Teaching Learning Based Optimization (NSTLBO) algorithm has been proposed to solve the problem. The algorithm is hybridized with Tabu Search (TS) algorithm, and Hybrid NSTLBO is also proposed. A grey-based Taguchi method is used to optimize the parameters of the algorithms. Proposed algorithms are compared with three well-known meta-heuristics, NSGA II, MOSS and SFLA developed for the problem by solving 96 randomly generated problems of different size and complexity. From the results, it can be concluded that the proposed Hybrid NSTLBO outperforms other algorithms in terms of diversification and intensification for all type of instances.

This research is further extended to consider benefit interdependencies. The total expected benefit consists of the benefits from the individual projects and the synergic benefit/loss due to benefit interdependencies. Benefit from a project is considered to be time sensitive and added to the budget for consideration of more projects. Technical interdependencies: mutual exclusiveness and complementariness are also considered. Two meta-heuristic algorithms: improved NSTLBO (I-NSTLBO) and Hybrid I-NSTLBO are developed to solve the problem. A grey-based Taguchi method is used to optimize the parameters of the algorithms. Proposed algorithms are compared with three well-known meta-heuristics, NSGA II, MOSS and SFLA developed for the problem by solving 96 randomly generated problems of different size and complexity. From results, it can be concluded that proposed Hybrid I-NSTLBO outperforms all other algorithms in all the comparison criteria for all type of instances. The results for I-NSTLBO are also promising.

The scope of the current research work is limited to the consideration of technical & benefit interdependencies, reinvestment of benefits and consideration of the time-sensitive nature of the project benefit. The models and the solution techniques proposed in this research may be useful to project managers in the simultaneous selection of projects in the portfolio and scheduling when projects are interdependent.

Contents

List of Figures

List of Tables

Abbreviations

ABC	Artificial Bee Colony
ACO	Ant Colony Optimisation
B&B	Branch & Bound
BR	Budget Requirement
CA	Cellular Automation
CD	Crowding Distance
FBS	Filtered Beam Search
GA	Genetic Algorithm
GIPB	Grey Integer Programming Branch & Bound
GSFLA	Grey Shuffled Frog Leaping Algorithm
HIA	Hybrid Intelligent Algorithm
IGA	Iterated Greedy Algorithm
IP	Integer Programming
ILP	Integer Linear Programming
IRM	Integrated Resource Management
IT	Information Technology
I-NSTLBO	Improved NSTLBO
MILP	Mixed Integer Linear Programming
MIP	Mixed Integer Programming
MOSS	Multi-Objective Scatter Search
MPM	Multi-Project Management
NDS	Non-dominated Solutions
NP-hard	Non-deterministic Polynomial-time hardness
NPV	Net Present Value
NS	Non-dominated Sorting
NSGA II	Non-dominated Sorting Genetic Algorithm 2
NSTLBO	Non-dominated Sorting Teaching Learning Based Optimization
PMBOK	Project Management Body of Knowledge
PMI	Project Management Institute
PPM	Project Portfolio Management
PPSSP	Project Portfolio Selection and Scheduling Problem

PSO	Particle Swarm Optimisation
RA	Resource Availability
RCMPSP	Resource-Constrained Multi-Project Scheduling Problem
RCPSP	Resource-Constrained Project Scheduling Problem
RR	Renewable Resource
RRR	Renewable Resource Requirement
RUV	Resource Usage Variation
R&D	Research & Development
SA	Simulated Annealing
SFLA	Shuffled Frog Leaping Algorithm
SS	Scatter Search
SSC	Student-Student Crossover coefficient
SSM	Self-Study Mutation coefficient
S-VNS	Stochastic-Variable Neighborhood Search
TL	Tabu List
TLBO	Teaching Learning-Based Optimization
TS	Tabu Search
TSC	Teacher-Student Crossover coefficient

Chapter-1

Introduction

Chapter 1
Introduction

1.1 Background

In the current era of innovation and competitiveness, the project management has become a discipline of increasing interest. Project management is a ubiquitous concept and finds application in a wide variety of industries and service sectors such as research & development, construction, defence, pharmaceuticals, hospitals, chemicals, banking, information systems, accounting, advertising and governments.

A project consists of multiple activities. These activities are precedence related and require time & resources for their completion. A project is said to complete if its constituent activities are complete. Gray & Larson (2000) have defined a project as "*A project is a complex, non-routine, one-time effort limited by time, budget, resources, and performance specifications designed to meet customer needs*". According to the Project Management Institute a project is "*A temporary endeavour undertaken to create a unique product or service*" (PMI, 2004).

Most of the time, despite internal coordination, a project needs to be coordinated with other projects in the portfolio executed by the same organization. This task is very important and complex. This act of managing projects internally and externally in an effective manner is known as project management. According to PMBOK "*Project management is the planning, organizing, directing, and controlling of company resources for a relatively short-term objective that has been established to complete specific goals and objectives*".

Project management primarily focuses on selection, scheduling and monitoring & control of the projects. Project selection problem deals with selection of the best subset of projects from the many competing proposals to attain the stated goals and objectives of the organization. Once the projects to be implemented are final, the selected projects are then scheduled. Project monitoring & control helps to know about the progress in the implementation of the projects and take the necessary action if deviated from the plan. In the case of multiple projects, the project management deals with project portfolio management (PPM) and multi-project management (MPM). The PPM is about the strategic decisions of project management such as project valuation, ranking and then selection to achieve the firm's objectives. Whereas the MPM is concerned with tactical and operational decisions such as project scheduling (determining the start and finish times of the activities), resource allocation and execution. Traditionally, these two processes are treated as independent and performed in a sequential manner. However, the two processes are interrelated and affect the final portfolio. The present work focusses on the integrated selection and scheduling of projects which has recently gained the attention of many researchers.

1.2 Project Selection

Choosing the most appropriate projects for implementation is a vital task in the project management to optimize the organization's objectives. Before selection of projects, it is ensured that the information about the set of candidate projects, resource requirements and resource availabilities are known a priori. But in actual practice, the number of candidate projects available are more than the budget and resources available for their implementation. The project selection problem targets to select a subset of projects from the set of available project alternatives to attain the organization's goals satisfying the budget and resources constraints. This set of selected project is called the project portfolio and the project selection process is widely studied as project portfolio selection problem in the literature.

In real life problems, often multiple objectives may be associated with the projects which are conflicting in nature. These objectives may be related to the profit, resources and the success of the portfolio. Projects which provide the best trade-off between multiple and conflicting objectives become the part of the portfolio. A list of the projects ranked according to their importance and contribution in achieving the stated objectives is determined which is treated as the output of the project selection process. This list is used as input to the project scheduling process.

1.3 Project Scheduling

Project scheduling is a necessary process required to make the implementation of a project successful. The classical project scheduling problem aims at ascertaining the start and finish time of the activities respecting the precedence relations, resource availability and other project characteristics. Precedence relations between the activities, their durations, resource requirement and availability, are known before scheduling. As the resources are limited, the problem is generally known as "resource-constrained project scheduling problem (RCPSP)".

An ample research has been carried out on single project RCPSP with different dimensions. However, very few projects are performed in a single project environment. Usually, firms deal with multiple projects running simultaneously. According to a study by Turner (1993), it is found that up to 90% of all projects, by value, are implemented in multi-project environment. In such situations, a number of projects run simultaneously sharing the same set of limited resources. Sharing of resources among the projects poses a problem of allocation of resources to the projects. This problem of determining the resource allocation and scheduling of multiple projects is called “resource-constrained multi-project scheduling problem (RCMPSP)”. The list of projects selected and ranked according to their importance during project selection process is input to the RCMPSP with other information such as precedence relations between the activities, their durations, resource requirement and availability etc. There are several issues added to multi-project scheduling problems which are not there in single project scheduling problems. These issues may be interdependencies among the projects, due dates, penalties for delay and other factors related to resource allocation and availability.

1.4 Integrated project selection and scheduling

In recent years the integrated problem of project selection and project scheduling has attracted the attention of many researchers. Organizations dealing with multiple projects often face the challenge of selection and scheduling of optimal mix of projects. The traditional project selection exercise seeks to select the portfolio of projects from a set of candidate projects satisfying the budget constraints. The criteria for project evaluation may be expected benefit and probability of success of the portfolio of selected projects. Once the projects to be implemented are final, an attempt is made to schedule them respecting the resource availability and time constraints. This sequential approach, however, may lead to difficulty in scheduling the selected portfolio. It may not be feasible to schedule all the projects selected in the portfolio during the given time frame. This may result in the adoption of some alternative approaches to avoid the conflict. Projects with shorter durations may be considered in place of the ones with larger durations. Resources may need to be increased to ensure completion of projects within the given time frame. If it is not possible to drop the projects which are not feasible to be scheduled in a given time frame, the schedule may be relaxed or economic goals may be compromised. These alternative approaches do not provide the best solution which was expected with the original selection criteria. Thus, the selection and scheduling of projects in a sequential manner result into sub-optimality (Coffin & Taylor, 1996a, 1996b). Hence the scheduling needs to be considered as an integral part of the project selection process. Project scheduling may be considered as one of the selection criteria in the project selection problem. This joint problem is termed as the project portfolio selection and scheduling problem (PPSSP) in literature. This problem not only determines the projects to be implemented but also provide the schedule of them with budget and resource requirement in each period. The PPSSP can be stated as the simultaneous problem of selection and scheduling of projects to optimize organization's stated objectives in specified time horizon without violating budget and resource constraints (Chen & Askin, 2009).

1.5 Issues in PPSSP

1.5.1 Time-dependent nature of the benefit

The benefit from a project is one of the most significant criteria in the project selection problem as well as in the integrated problem of project selection and scheduling. However, the benefit from a project may vary according to the completion time of a project which may affect the overall benefit from the portfolio. This is quite common for projects related to the introduction of new products and high tech products. The market share of these products largely depends on the time of their entry into the market. Early entrants have an advantage of larger market share and hence the benefit. According to a study by McKinsey & Company, a delay of six months in the introduction of high tech products may result into a loss of one-third of the profit over a duration of five years (Musselwhite, 1990). The literature from marketing science reveals that the effect of time-to-market on subsequent market share is substantially high. Consideration of this aspect attempts early scheduling of projects whose benefit is more sensitive to the completion time.

1.5.2 Interdependencies among the projects

The candidate projects for selection often exhibit a variety of interdependencies. Interdependencies are present if inclusion/exclusion of a project in the portfolio is affected by the selection of other candidate project(s). Consideration of interdependencies yields more profit as the total benefit/cost from interdependent projects is not same as the sum of the individual project's benefit/cost (Aaker & Tyebjee, 1978; Fox et al., 1984). In the literature three types of interdependencies have been considered viz. resource, benefit and technical.

The first type of interdependency is resource interdependency which arises due to common resources needed for implementation of a set of candidate projects. These resources may be machinery, equipment, experts or facilities. The amount of resources required would be high if such projects are implemented in an independent manner in contrast to their simultaneous implementation. For example in IT projects where programming experts, hardware and software can be shared among various projects. Another example can be found in R&D projects where costly lab equipment can be utilized for different research activities.

The second type of interdependency is benefit interdependency results when benefits expected from the projects are non-additive. In this sense, the projects may have positive or negative synergies. Positive synergy arises when the total benefit from the projects implemented together is greater than the total benefit when implemented individually. Such projects are recognized as benefit complementary projects. On the other hand, negative synergy arises when the total benefit from the projects implemented together is less than the total benefit when implemented individually. Such projects are recognized as competitive projects. For example, consider two projects each related to the introduction of a new or modified product for the customers. The projects are said to be benefit complementary if the simultaneous introduction of these products to the market can capture a larger market share and hence the sale than the sum of their individual sales. If their simultaneous introduction cannibalizes the market sale of each other resulting in reduced sales than the sum of individual sales, then the projects are said to be competitive. Good examples can be found in the automobile sector, IT gadgets etc.

The third type of interdependency is technical interdependency among the projects in which the selection of one project largely depends on the selection of other projects. This interdependency is further divided into two types: mutual exclusiveness and complementariness. Projects are said to be mutually exclusive if only one project from a subset of projects can be considered for the portfolio. For example, suppose a firm has to select only one project among the various available proposals for the construction of its new manufacturing plant. These different project proposals would be mutually exclusive. On the other hand, projects are said to be complementary if all projects from the subset are included or excluded from the portfolio. The complementariness can be observed in many real-life situations. Consider, e.g. electricity generation and transmission projects. In this situation construction of power plant and laying of transmission lines could be two projects which need to be selected or rejected together. Take another example where a municipal corporation is going for cleanliness drive in which one project is the development of a system for waste collection and the other is making the user aware through an awareness campaign. The real benefit from the drive could only be achieved when both of the projects are selected.

1.5.3 Reinvestment of benefit from the projects

During the project selection process a significant number of candidate projects qualify for implementation. A certain amount of resources is needed for the implementation of a project which can be started at any time when adequate resources are available. But all the projects may not be implemented due to financial limitations. However, the benefit from a completed project can be reinvested to add more projects for the higher expected benefit (Belenky, 2012). The benefit accrued for reinvestment is affected by selection of projects and their start times. This strategy provides an opportunity for the projects which otherwise do not get selected due to limited budget even when renewable resources are available. Besides increasing the benefit, this strategy also increases the utilization of renewable resources. Despite the fact that reinvestment strategy is commonly used in the parlance of financial portfolios, it is scantly considered in PPSSP.

1.6 Solution Approaches for Project Portfolio Selection and Scheduling Problem

In current work, the PPSSP involves the integration of project selection and project scheduling processes with consideration of interdependencies and reinvestment strategy. Multiple projects are selected from available candidate projects, scheduled and allocated resources which are available in limited quantity. Problems related to the scheduling of a single project (RCPSP) have already been identified as NP-hard in the literature (Demeulemeester & Herroelen, 2006). Thus, PPSSP becomes even more complex as it involves scheduling of multiple projects along with project selection. Further, the problem is complicated by interdependencies and reinvestment strategy also. From a mathematical viewpoint, this problem is very much complex as compared to pure scheduling problems hence determining an optimal solution within a reasonable time is very difficult. Moreover, the multiple conflicting objectives enhance the complexity of the problem as it is difficult to determine superiority of one solution over other, unlike single-objective optimization.

It is evident that as problem complexity increases, the computational requirement becomes too high for a real-life large-sized problem and it becomes almost impossible to solve the problem optimally using exact approaches. In such cases, it is important to find a near optimal solution. Heuristics may be employed to find acceptable solutions with less computation effort. Heuristics may provide a good feasible solution but do not guarantee its optimality. But, heuristics are subjected to the serious problem of getting trapped into local optima.

In recent years, meta-heuristics have been used to solve real life complex problems to reach acceptable solutions near to the optimal solution in reasonable computational time. Meta-heuristics are efficient, flexible and independent of the problem and model. These algorithms provide a good balance between the quality of solution and the computational exercise. Efficient working of a meta-heuristic is ensured by a good trade-off between its ability to explore (global search) and exploit (local search) the search space.

The meta-heuristics approaches can be classified into two sub-categories: (i) single solution based (ii) population-based. The former type of meta-heuristics continues with a single solution at each iteration of the algorithm. Simulated annealing (SA) and Tabu search (TS) are the two very common examples of this kind of meta-heuristic algorithms. These approaches advance iteratively and attempt to find a superior solution than present solution at each stage. However, the quality of the final solution attained greatly depends on the initial solution provided. These approaches have a good capability of exploiting the solution space locally but poor at exploration. The population-based meta-heuristics proceeds with a group of initial solutions and operators are applied to improve the population iteratively. Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Shuffled Frog Leaping Algorithm (SFLA) and Ant Colony Optimization are some well-known population-based meta-heuristics. Hybrid algorithms are also developed to tackle the problem in a fast and effective manner. Hybrid optimization techniques utilize the benefit characteristics of individual optimization algorithms. Meta-heuristics can be hybridized with classical solution approaches as well as other meta-heuristics.

A variety of meta-heuristic approaches such as genetic algorithm (GA), particle swarm optimization (PSO), ant colony optimization (ACO), simulated annealing (SA), tabu search (TS) and artificial bee colony (ABC) have been developed for solving a large number of engineering and management problems. However, the proper tuning of common control and algorithm specific parameters is required for the efficient working of a meta-heuristic algorithm. For example, in GA optimal tuning of the selection operator, crossover probability and mutation probability is required. Similarly, ACO, PSO, artificial bee colony algorithm (ABC) and other meta-heuristics require algorithm specific and common control parameters to be set optimally to avoid the local optima.

Teaching learning-based optimization (TLBO) is one of the recently developed meta-heuristic which has been successfully applied to a variety of complex optimization problems (Yu, Wang, & Wang, 2016; Dokeroglu, 2015; Xu et al. 2015; Tuncel, & Aydin, 2014; Baykasoğlu, Hamzadayi, & Köse, 2014; Keesari, & Rao, 2014; Rao, & Patel, 2012) with the benefit of very less number of parameters to be tuned compared to other meta-heuristics. This algorithm derives its philosophy from the conventional classroom teaching-learning process. TLBO is developed for the proposed versions of PPSSP in this study.

1.7 Research motivation

Selection of projects from a set of available projects is an important decision every organization faces as this may affect the long-term profitability of the organization. These projects need to be allocated resources and scheduled for their timely completion. The sequential process of selection and scheduling, however, results in sub-optimality. Hence the scheduling needs to be considered as an integral part of the project selection process. In recent years the integrated problem of project selection and project scheduling has attracted the attention of many researchers, but still, little research is available in the PPSSP. There are a number of factors which motivate us to continue the research in this area.

a) The interdependencies among the projects have been considered in the project selection process but have been considered in a limited way in PPSSP. Out of the two technical interdependencies (as mentioned in section 1.5.2), the mutual exclusiveness has been considered as project interdependencies in the PPSSP, but complementariness is yet ignored in the existing literature.
b) There is a lack of research in consideration of benefit interdependencies which renders an increase in the overall benefit of the portfolio.
c) The reinvestment strategy in PPSSP has been addressed scantly in the literature.
d) Very few researchers have considered time-dependent nature of the return in PPSSP.

e) Last motivation comes from the need for an efficient solution approach for such a complex problem.

1.8 Research objectives and scope of the thesis

The following are the main research objectives outlined for the current study:

1. To develop a mathematical model for project portfolio selection and scheduling problem.
2. To develop efficient meta-heuristics for single objective project portfolio selection and scheduling problem.
3. To develop efficient meta-heuristics for multi-objective project portfolio selection and scheduling problem.

The scope of the current study covers

1. This study considers technical interdependencies: mutual exclusiveness and complementariness among the projects.
2. This study considers the positive and negative synergies existing among the projects due to benefit interdependencies to optimize the total expected benefit.
3. The benefit accrued from the completed projects is reinvested in the portfolio to add more projects to the portfolio which increases the total benefit and utilization of renewable resources.
4. The time-sensitive nature of the project benefit has been taken into consideration in the current study to ensure the early completion (scheduling) of high benefit projects and hence the incresed overall benefit from the portfolio.

1.9 Organization of the thesis

In this section, the organization of the thesis has been presented. The thesis is organized into six chapters. Figure 1.1 shows the chapter-wise layout of contents of the thesis.

A brief description of the contents of each chapter is given below.

Chapter 1 provides an introduction to the thesis. It includes basics of project selection, project scheduling and integrated project selection and scheduling problems. It then provides an overview of the solution approaches to the integrated project selection and scheduling problems. After this, the objectives of this research are presented. Lastly, the chapter-wise organisation of the contents of the thesis is given.

Chapter 2 deals with the state-of-the-art research on integrated project selection and scheduling. This chapter first discusses the nature of the problem and its importance and then presents various modelling approaches existing in the literature. It subsequently provides an overview of the solution approaches used in the literature for the considered problem. Finally, it presents an examination of the gaps in the existing body of knowledge of PPSSP.

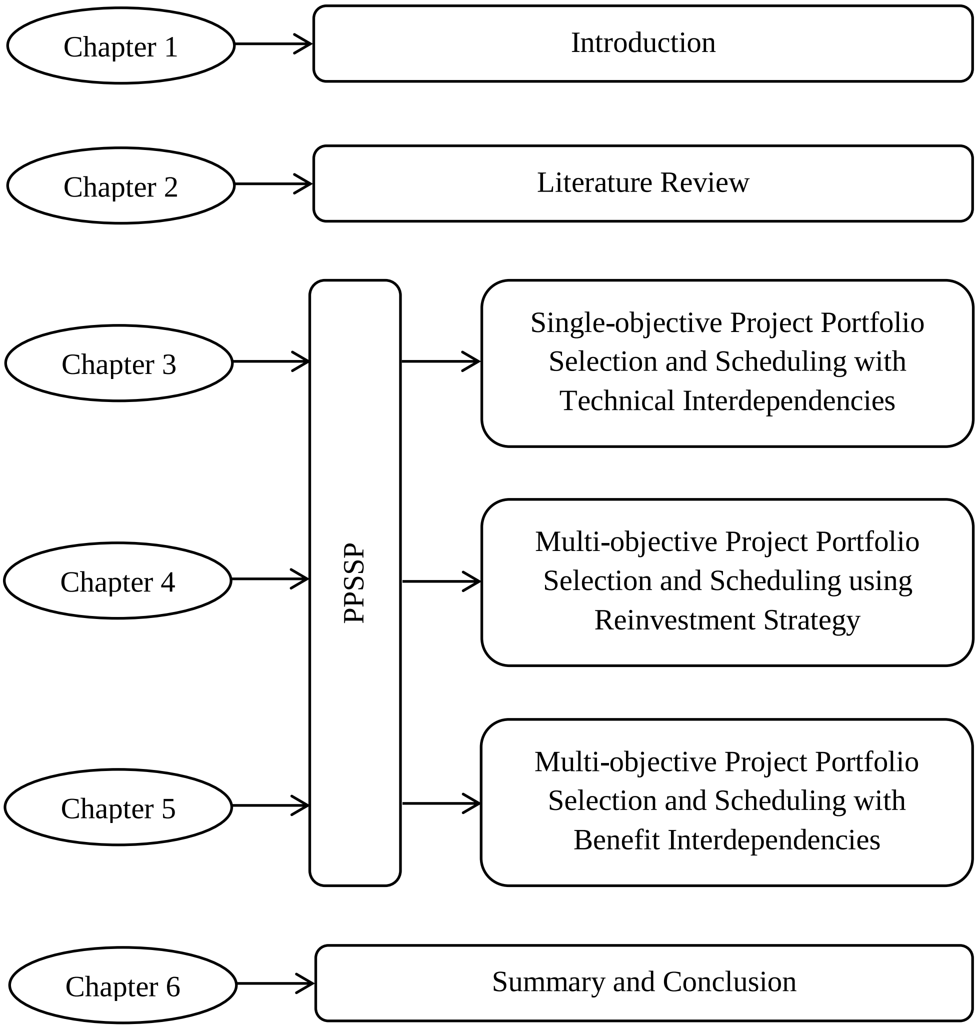

Chapter 1
Introduction
Chapter 2
Literature Review
Chapter 3
Chapter 4
Chapter 5
PPSSP
Single-objective Project Portfolio Selection and Scheduling with Technical Interdependencies
Multi-objective Project Portfolio Selection and Scheduling using Reinvestment Strategy
Multi-objective Project Portfolio Selection and Scheduling with Benefit Interdependencies
Chapter 6
Summary and Conclusion

Figure 1.1: Chapter wise layout of the thesis

Chapter 3 presents problem definition, mathematical formulation and proposed meta-heuristic algorithms for the solution of the single-objective PPSSP with technical interdependencies. The detailed procedure of the proposed algorithms, parameter settings, instance generation scheme to examine the behaviour of the algorithm and results are discussed in this chapter. In this way, this chapter aims to fulfil the second research objective.

Chapter 4 provides the problem description and mathematical formulation and proposed solution approaches for the multi-objective PPSSP using reinvestment strategy. The detailed procedure of the proposed algorithms, parameter settings, instance generation scheme to examine the behaviour of the algorithm and results are discussed in this chapter. In this way, this chapter aims to fulfil the third research objective.

Chapter 5 considers the extended version of the multi-objective PPSSP presented in chapter 4 considering benefit interdependencies also and describes the proposed solution algorithms. The mathematical model is modified to accommodate the benefit interdependencies. The detailed procedure of the proposed algorithms, parameter settings, instance generation scheme to examine the behaviour of the algorithm and results are discussed in this chapter. In this way, this chapter aims to fulfil the third research objective.

Chapter 6 concludes the thesis. This chapter focuses on the contribution of the present work to the body of knowledge. In addition, the practical implications of the work with some recommendations for future research are also summarized.

Chapter-2

Literature Review

Chapter 2
Literature Review

In this chapter, a state-of-the-art review of the literature available in the domain of project portfolio selection and scheduling problem (PPSSP) has been provided. First, the review methodology and the important aspects of the problem considered in the modelling are described. Next, a discussion on commonly used objective functions is provided followed by a classification of the solution approaches used to solve the PPSSP. Based on this literature review, some future research directions on integrated selection and scheduling of projects have been identified.

2.1 Review methodology

In this section, a methodological framework for analysing the current literature on PPSSP has been introduced. This literature review process is carried out in four different steps as shown in Figure 2.1. A total of 33 articles have been collected and analysed according to four criteria namely important aspects focussed, number of objectives considered, modelling approach and solution approach. This analysis has been done primarily to get an insight into the present state of work on PPSSP and to find critical issues, gaps and hence the potential areas for further research.

2.1.1 Material collection

This literature review focusses on papers available from reputed journal publications and conferences. The papers which are included are easily accessible through online databases such as Elsevier, Emerald Insight, Springer, Taylor & Francis, Wiley Online Library, Inderscience, Sage, IEEE Xplore, Hindawi, ASME and INFORMS. Duration of the study spans from the beginning (the point the term integration of project selection and scheduling was introduced) to March 2018. It also covers the accepted papers also which are available online. Papers selected from Google Scholar are cross verified with the online databases of the individual publications to ensure the authenticity of the paper selection process. The articles are searched using “project selection and scheduling”, “project portfolio selection”, “multi-project scheduling” and “multi-project environments” as keywords. The articles relevant to PPSSP are acquired after rectification and cross-referencing. Table 2.1 lists the journal publications and the number of papers available in that publication. Table 2.2 provides the journal-wise list of articles considered for review.

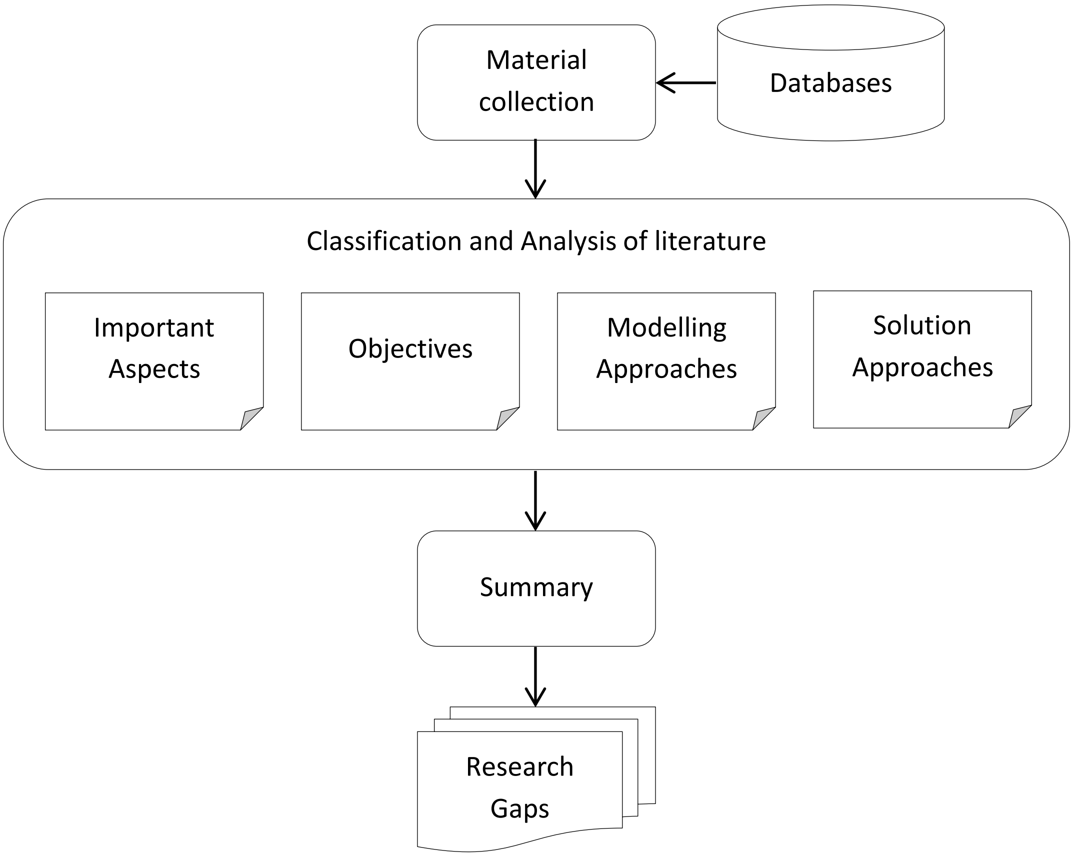


Figure 2.1: The literature review process

Table 2.1: Sources of publication

Publication	No. of Papers
Elsevier	16
Springer	7
Taylor and Francis	2
Emerald Insight	1
Inderscience	2
IEEE Xplore	1
INFORMS	1
Hindawi	1
Others	2
Total	33

Table 2.2: Journal-wise list of the papers reviewed

Journal Name	No. of Papers
European Journal of Operational Research	3
Computers & Industrial Engineering	3
Computers & Operations Research	2
Management Science	1
Applied Mathematics and Computation	1
IIE Transactions	1
Journal of the Operational Research Society	1
Decision Support Systems	1
Automation in Construction	1
Advances in Engineering Software	1
Knowledge-Based Systems	1
Technovation	1
Technological Forecasting & Social Change	1
Central European Journal of Operations Research	1
Kybernetes	1
International Journal of Productivity and Quality Management	1
Mathematical Problems in Engineering	1
International Journal of Industrial and Systems Engineering	1
Journal of Zhejiang University-SCIENCE C (Computers & Electronics)	1
Journal of Industrial and Production Engineering	1

Figure 2.2 shows the number of papers published or accepted in each year throughout the span of the study. The increasing trend on the graph shows that in recent times the more attention is given to the integration of scheduling with project selection. Figure 2.2 also indicates that research in this field is in its beginning stage and has gained more attention from 2009.

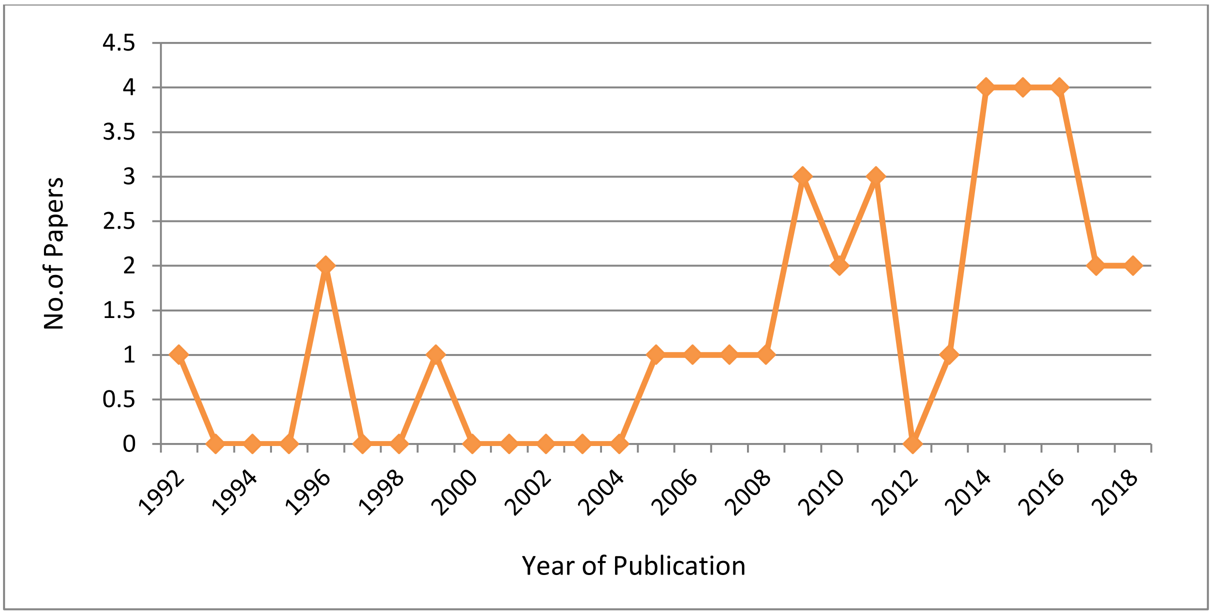


Figure 2.2: Publishing trend in the area of PPSSP

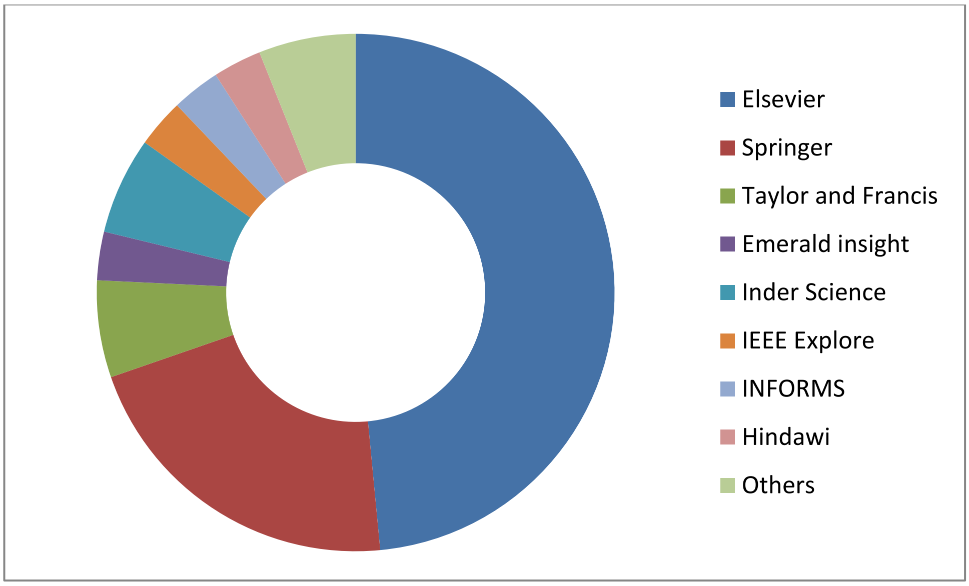


Figure 2.3: Publisher wise contribution to study of PPSSP

Figure 2.3 shows the contribution of each publisher to the research in PPSSP. It is clear that major contribution is provided by the Elsevier and Springer. Elsevier alone contributes almost 50% of the total papers.

2.2 Project portfolio selection and scheduling problem (PPSSP)

The first attempt to integrate project selection and scheduling was made by Gupta et al. (1992) through dynamic programming. However, authors assumed that a project can be scheduled after the completion of the previous one. This means only one project can be pursued at a time and the projects are merely sequenced one after other. After that various authors have considered this problem with different modelling aspects and approaches. A summary of the research articles on PPSSP is given in Table 2.3. The detailed literature on important aspects of the problem, different objectives considered, various modelling approaches and solution techniques is provided next in this chapter.

2.3 Important aspects of the PPSSP

The PPSSP has been studied with respect to five important aspects described below.

2.3.1 Time-dependent return

Generally, the project benefit is taken as time-dependent (Chen & Askin, 2009). The early completion of high return projects tends to increase the overall benefit. Consider, for example; IT projects where new gadgets are introduced in the market almost every day. Delayed introduction of a gadget may lead to reduced market share and hence profitability. This aspect is one of the most important aspects considered in PPSSP as the scheduling of the projects affects the overall benefit to the organization. This time-sensitive aspect of benefit in PPSSP is first considered by Chen & Askin (2009) and followed by Ghorbani & Rabbani (2009), Tofighian & Naderi (2015), Ganji et al. (2016), Shariatmadari et al. (2017) and Amirian & Sahraeian (2017; 2018).

2.3.2 Interdependencies

Interdependency among the projects is quite common. Interdependency comes into play when the decision on choosing a project depends on the decision of other project(s). Consideration of interdependencies though makes project selection difficult, yields more profit as the total benefit/cost from interdependent projects is not the same as the sum of the individual project's benefit/cost. In the literature three types of interdependencies have been considered viz. benefit, resource and technical. Aaker & Tyebjee (1978) and Fox et al. (1984) were the pioneers who considered project interdependencies during project selection. Later on, project interdependencies in project selection have been considered, e.g. by Li et al. (2016); Abbassi, Ashrafi, & Tashnizi, (2014); Bhattacharyya, Kumar, & Kar, (2011); Liesiö, Mild, & Salo, (2008); Lee, & Kim, (2000); Santhanam, & Kyparisis, (1996).

Table 2.3: Summary of the research articles on PPSSP

Authors	Characteristic						Modelling Approach	Solution Methodology
	Time dependent Return	Inter-dependencies	Re-investment Strategy	Uncertainty	Activity Scheduling	Multi-objective		
Gupta et al. (1992)							Dynamic programming	Dynamic programming
Coffin & Taylor (1996a)				✓		✓	0-1 integer linear programming, Fuzzy Logic	Fuzzy logic with a beam search approach
Coffin & Taylor (1996b)						✓	0-1 integer linear programming	Filtered beam search (FBS)
Ghazemzadeh et al. (1999)		✓					0-1 integer linear programming	Branch & bound (B&B) using LINGO & CPLEX
Sun and Ma (2005)							0-1 integer linear programming	Packing-multiple-boxes heuristic
Kolisch and Meyer (2006)							0-1 integer linear programming	Sequential and concurrent scheduling heuristics
Gutjahr et al. (2007)				✓		✓	stochastic combinatorial optimization	Stochastic variable neighborhood search (S-VNS)
Gutjahr et al. (2008)						✓	Non-linear mixed-integer programming	Ant colony optimization (ACO) and genetic algorithm (GA)
Ghorbani et al. (2009)	✓	✓				✓	Mixed integer linear	Multi-objective scatter

							programming	search (MOSS)
Chen and Askin (2009)	✓				✓		0-1 integer linear programming	Implicit enumeration algorithm
Shou and Huang (2010)					✓		0-1 integer programming	An iterative multi-unit combinatorial auction algorithm
Carazo et al. (2010)		✓				✓	Non-linear binary programming	Scatter Search (SS-PSS)
Shi et al. (2011)							0-1 integer programming	Genetic algorithm (GA)
Liu & Wang (2011a)		✓					Constraint programming	Constraint programming approach
Liu & Wang (2011b)		✓					Constraint programming	Constraint programming approach
Zhao and Huang (2013)				✓			Chance-constrained programming	Implicit enumeration algorithm
Garcia (2014)							Integer linear programming	A greedy heuristic & a meta-heuristic (Meta-RaPS)
Shou et al. (2014)					✓		0-1 integer programming	A multi-agent evolutionary algorithm
Huang and Zhao (2014)		✓		✓			Uncertain programming	Genetic algorithm (GA)
Hosseininasab et al. (2015)							Integer linear programming, Mixed integer linear programming	First Phase of two-phase simplex method, Frank-Wolfe algorithm, and GA

Tofighian et al. (2015)	✓	✓				✓	Mixed integer linear programming	Ant colony optimization (ACO)
Jafarzadeh et al. (2015)			✓				Integer linear programming	BONMIN solver of GAMS software
Ghahremani and Naderi (2015)					✓		Mixed integer linear programming	Genetic algorithm (GA) & iterated greedy algorithm (IGA)
Ganji et al. (2016)	✓				✓		Mixed integer linear programming	GAMS software
Wang and Song (2016)		✓	✓				Integer linear programming	MATLAB and LINGO solvers
Arratia M. et al. (2016)				✓			Mixed integer linear programming	B&B technique using CPLEX
Huang et al. (2016)		✓		✓			Uncertain programming	Hybrid intelligent algorithm (HIA)
Shariatmadari et al. (2017)	✓				✓		Mixed integer linear programming	Gravitational search algorithm (GSA)
Pérez et al. (2018)				✓			Fuzzy Logic	
Amirian and Sahraeian (2017)	✓			✓	✓	✓	Grey integer programming	Modified grey shuffled frog leaping algorithm (GSFLA)
Amirian and Sahraeian (2018)	✓	✓		✓		✓	Grey integer programming	Grey integer programming branch & bound (GIPB)

Out of the three, technical interdependencies are the most common (Aaker & Tyebjee, 1978; Fox et al., 1984) which are further divided into two types: mutual exclusiveness and complementariness. Mutual exclusiveness has been considered the most in the PPSSP literature. Projects are said to be mutually exclusive if only one project from the set can be included in the portfolio. Ghazemzadeh et al. (1999) were probably the first to consider mutual exclusiveness of projects and developed a 0-1 integer linear programming model for PPSSP. Ghorbani & Rabbani (2009) offered a multi-objective meta-heuristic to obtain the diverse non-dominated solutions for the joint problem of selection and scheduling considering mutually exclusive projects. Tofighian & Naderi (2015) consider the same problem and developed a multi-objective ant colony optimization for PPSSP. Liu & Wang (2011a, 2011b) have considered the mutual exclusiveness among the projects with time-dependent resource constraints. Wang & Song (2016) considered mutual exclusiveness of projects with reinvestment strategy. Huang & Zhao (2014), Huang et al. (2016) and Amirian & Sahraeian (2018) have considered mutual exclusiveness with uncertain project parameters. The second technical interdependency: complementariness still has not been considered in PPSSP. Carazo et al. (2010) have included resource interdependencies in PPSSP model and also estimated their contribution to the overall benefit of the portfolio. Benefit interdependencies have been frequently used in project selection but still need attention in PPSSP.

2.3.3 Reinvestment strategy

The return from the completed projects, if reinvested in the portfolio, can provide an adequate budget for selection of more projects. Thus the order of implementation of the projects matters as more profitable projects would make more money available. Belenky (2012) is the first to consider the reinvestment strategy in project selection. The author assumed that profit can only be generated after project completion. The author formulated the basic project selection problem as a Boolean program and presented some generalizations.

Jafarzadeh et al. (2015) used reinvestment strategy in PPSSP for the first time. Authors considered the time horizon to be flexible. Authors corrected the model proposed by Belenky (2012) and presented an integer programming based model for the PPSSP. Authors assumed that the profit generated by a project is yielded just one period after the investment on that project ends.

Wang & Song (2016) applied reinvestment strategy while considering the annual budget to be time-dependent. The authors have also considered the time value of the capital with the objective to maximize the total profit and presented a mathematical model based on integer programming. To show the effectiveness of the reinvestment strategy, authors have compared scenarios with and without reinvestment strategy.

2.3.4 Uncertainty

In life situations, projects are always subjected to some degree of uncertainty. This uncertainty may arise due to either the scarce or no past information. An example can be seen in R & D projects where there is no exact information available about the total cost of the research carried out and the revenue collected by the resulting product. Only rough estimates are provided by the experts, which are not very reliable. In PPSSP, parameters such as profits and costs of projects, project durations, budget, cash inflows and resources may exhibit uncertainty. In PPSSP literature, such variables are treated as interval, stochastic, fuzzy, or grey variables.

Coffin & Taylor (1996a) were probably the first to consider uncertainty in PPSSP who considered the return from R&D projects as probabilistic. A risk is always associated with the return from R&D projects. Authors have linked a probability of success to each project so that the probability of success of the final portfolio can be maximized. Gutjahr et al. (2007) modelled PPSSP with staff assignment considering the wok times required by each work package to be a stochastic variable. The problem is an solved by an S-VNS algorithm based on Variable Neighborhood Search (VNS) meta-heuristic. Zhao & Huang (2013) have considered the uncertainty in the cash inflows, cash outflows and the initial cost of the projects which is managed by chance-constrained programming.

Huang & Zhao (2014) have considered some uncertain project parameters in R & D projects such as net income and investment cost. In R &D projects, initial the investment is calculated by rough estimates obtained by the experts' evaluations. The actual investment, however, generally exceeds the estimated investment, causing a risk of cost overrun. Authors have measured the exceeding amount of the average investment over budget using a cost overrun risk. Arratia M. et al. (2016) also included budget uncertainty in PPSSP. Huang et al. (2016) have also considered the cash inflows and initial outlays as uncertain variables. Authors have developed two models for the problem under uncertainty namely uncertain mean variance and uncertain mean semi-variance.

According to Pérez et al. (2018), it is very tough to gauge the exact amount of resources needed at the time of implementation of an activity. This may happen due to information ambiguity. Authors have treated the parameters associated with renewable resources as uncertain parameters and handled them as fuzzy numbers. Amirian & Sahraeian (2017) have assumed the project duration, cost and budget availability per period as grey parameters. They have first modelled the problem as deterministic then the problem is converted into grey equivalent by assuming the above three parameters as grey numbers. Amirian & Sahraeian (2018) have considered the project profit, cost of resource usage variation, and amount of resources (required and available) as grey parameters while modelling the PPSSP.

2.3.5 Activity scheduling

In literature, the PPSSP has been modelled in two ways (Amirian & Sahraeian, 2017). In first, a project is considered as a single unit assuming the project schedule to be fixed and predetermined. In this only start and finish time of the projects are determined. The start and finish times of the activities are established based on the project's start time. In second, activity scheduling is considered in which the start and finish times of the activities are also established with the selection. Chen & Askin (2009) are the first to model the PPSSP at activity level. Few other authors have also included project scheduling at activity level in PPSSP (Shariatmadari et al., 2017; Amirian & Sahraeian, 2017; Ganji et al., 2016; Ghahremani & Naderi, 2015; Shou et al., 2014; Shou & Huang, 2010). In PPSSP, considering the project scheduling at activity level makes the decision making more complex but also enhances the resource utilization. Further, the real advantage of resource interdependencies can only be realized when project scheduling is considered at activity level. Shou & Huang (2010) state that by considering scheduling at activity level the limited resources can be utilized in a better way. Sometimes an extra project may be included in the portfolio just by adjusting the resource allocation without affecting other projects. Shariatmadari et al. (2017) have considered the issue of integrated resource management which is possible only when detailed scheduling is included in PPSSP.

2.4 Objectives for PPSSP

In literature, PPSSP has been modelled with objectives such as maximizing the overall benefit, minimizing the make-span, maximizing the success probability and minimizing the resource usage variation etc. But maximizing the overall benefit has always been considered as the key objective. Table 2.4 lists the various objectives used in PPSSP.

Table 2.4: Objectives considered in PPSSP

S. No.	Objectives used in the formulation of PPSSP	No. of Papers
1	Expected benefit or NPV	27
2	Expected strategic benefit	2
3	Recourse usage variation	3
4	Probability of success of the portfolio	2
5	Make-span	2
6	Total Cost	1
7	Expected value of the total tardiness	1

From Table 2.4 it can be seen that the expected benefit or NPV of the portfolio has been the most popular and primary objective in PPSSP. Some authors, however, have taken it as the sum of the return from each selected project in the portfolio (Gutjahr et al., 2007; Gutjahr et al., 2008; Liu & Wang, 2011; Garcia, 2014) while some have considered this objective in form of NPV (Gupta at al., 1992; Ghazemzadeh et al., 1999; Shou & Huang, 2010; Shi et al., 2011; Zhao & Huang, 2013; Huang & Zhao, 2014; Huang et al., 2016; Ghahremani & Naderi, 2015).

Some authors have maximized the expected return from the project portfolio. This form of the economic objective is undertaken when the return from a project is not fixed. Either the return from a project is associated with the success probability of the project or varies according to the point of implementation or completion of a given project. In the first situation, the return is associated with the success probability of the project and can be assessed as a product of the return and success probability (Coffin & Taylor, 1996). In the second situation, the outcome benefit from a project is supposed to be time-sensitive. The amount of return decreases with the delay in the commencement or completion time of the project (Chen & Askin, 2009). In PPSSP, this aspect of economic benefit has been considered by Chen & Askin (2009), Ghorbani & Rabbani (2009), Shou et al. (2014), Tofighian & Naderi (2015), Amirian & Sahraeian (2017), Shariatmadari et al. (2017), and Amirian & Sahraeian (2018). Ganji et al. (2016) have used the maximization of the NPV of the total expected benefit of the portfolio.

One more form of benefit has been addressed as the objective in the literature called strategic benefit. This is not a direct benefit from the project. Gutjahr et al. (2007) and Gutjahr et al. (2008) have considered the strategic benefit which is realized from the increment in the work efficiency of staff over the planning horizon.

In some of the cases, minimization of the total cost of implementation of the projects may be an objective. This objective has been considered by Amirian & Sahraeian (2017) for optimization.

In real life situations, underutilization of cost-intensive resources such as experts, machines etc. affect the expected benefit from the project. This aspect has been considered in the literature by minimization of total required resources/maximum amount of a resource required in each time-period. Minimization of total required resources and minimization of the maximum amount of resource may lead to hiring/firing cost and exclusion of some of the projects respectively. To avoid this situation Ghorbani & Rabbani (2009) suggested the use of minimization of resource usage variation which was also considered by Tofighian & Naderi (2015). Amirian & Sahraeian (2018) have considered the cost associated with the resource usage variation as their second objective. However, Amirian & Sahraeian (2017) have considered the minimization of total unused resources to utilize the limited resources better.

The success of R&D projects is usually not sure, and there is always a risk associated with them. In such cases, an objective related to the probability of success is included so that the only those projects find a place in the portfolio which have the best success probability (Coffin & Taylor, 1996). However, it is difficult to gauge the risk associated with the success of R&D projects accurately. Coffin & Taylor (1996) have considered the success probability of the portfolio as the mean value of the success probability of each project selected in the portfolio.

In PPSSP, minimizing the make-span can also be considered as an important objective. This is a scheduling goal for the problem when there is no fixed planning horizon or due dates for the projects. Coffin & Taylor (1996) have considered minimizing the make-span as scheduling objective for the problem. In the case of due dates of the selected projects are known, the scheduling objective could be minimizing the total expected tardiness (Gutjahr et al., 2007). The tardiness can be calculated with respect to the due dates of the selected projects.

Most of the authors have considered single objective and benefit maximization has been the most popular. There are, however, some authors who have considered multiple objectives. Figure 2.4 shows the number of papers using single or multiple objectives in PPSSP.

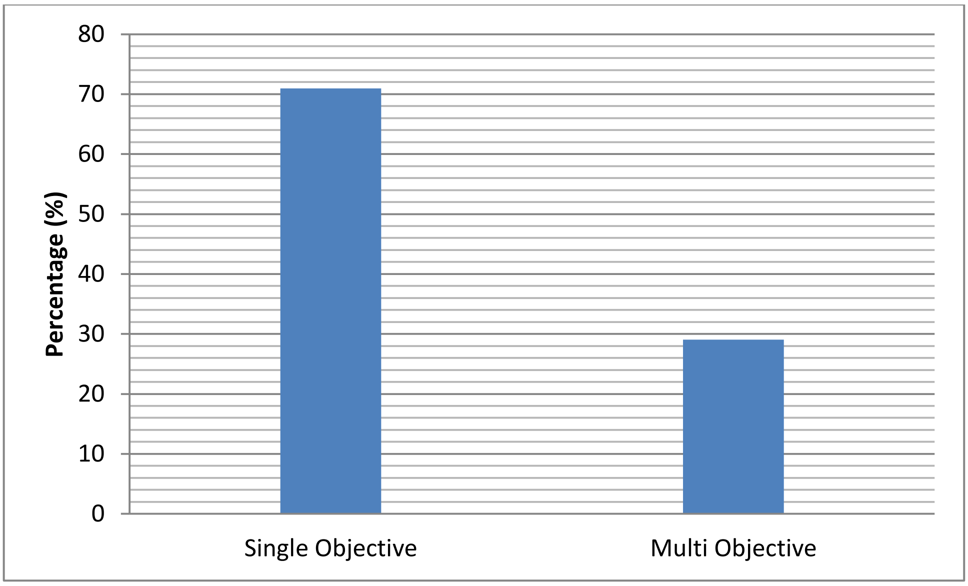


Figure 2.4: Single and multi-objective papers on PPSSP

Coffin & Taylor (1996a) have considered three objectives for optimization: (i) expected profit, (ii) average success probability, and (iii) make-span of the portfolio of R&D projects. In the above mentioned objectives, the first two objectives are to be maximized, and the third is to be minimized. By using fuzzy logic, a fuzzy set is generated for each of the three objectives and grades of membership for these objectives have been summed up to give a single objective function called "overall objective function".

Coffin & Taylor (1996b) considered the same objectives and have used the probability of success as the first filtering objective to screen the projects. The set of screened projects is then subjected to the second filter based on the make-span. Finally, the projects are evaluated and selected on the basis of their expected profit.

Gutjahr et al. (2007) have modelled the problem of PPSSP with three objectives: maximization of economic benefit, expected strategic benefit and minimization of total tardiness. The economic benefit is the return from each selected project while expected strategic benefit arises from the competence increment of the staff. The weighted aggregate of the three objectives is taken to give a single objective function. Gutjahr et al. (2008) have also considered maximization of economic benefit and expected strategic benefit.

Carazo et al. (2010) have formulated the problem to optimize the weighted aggregate of different attributes at different periods. The objective function to optimize different attribute consists of two parts. The first part is the sum of profits from the individual contribution of projects while the second part is the total of the benefit or loss resulting from the synergies between the projects.

Amirian & Sahraeian (2018) have proposed the model for PPSSP in grey environment with two objectives: maximizing total benefit from the portfolio and minimizing the total cost resulting from resource usage variations. Multiple objectives are handled using a simple additive weighting method.

In above-discussed cases of multiple objective optimization of PPSSP, either the objectives are combined into single one or objectives are used in a stage-wise manner to make the decisions. All these techniques are known as priori approaches and avoid the complexities and facilitate the easy decision making. However, this is not true multi-objective optimization. According to Deb et al. (2002), the multi-objective optimization is not a simple extension of the single-objective optimization.

True multi-objective optimization leads to multiple solutions depending upon the trade-offs between the objectives. Decision makers choose the final solution in accordance with the importance of objectives for their requirement. This approach of decision making in the multi-objective environment is known as posteriori approach. Ghorbani & Rabbani (2009) have considered maximizing the total expected benefit and minimizing the total resource usage variation as the two objectives for the problem. Authors have used the posteriori approach of decision making. The problem taken by Ghorbani & Rabbani (2009) has been extended by the Tofighian & Naderi (2015) with the same set of objectives. Amirian & Sahraeian (2017) have modelled the problem with three conflicting objectives: maximizing the total benefit, minimizing the total cost and the total unused resources.

2.5 Modelling approaches

The PPSSP has been formulated using various modelling approaches. The available approaches in literature have been divided into eight different categories: (1) linear and MIP, (2) non-linear programming, (3) grey programming, (4) constraint programming, (5) uncertain programming, (6) fuzzy logic, (7) stochastic, (8) dynamic programming, and (9) chance-constrained programming. Figure 2.5 shows the relative use of various modelling approaches. From the figure it is clear that most of the authors have used linear or MIP approach of modelling for their problems.

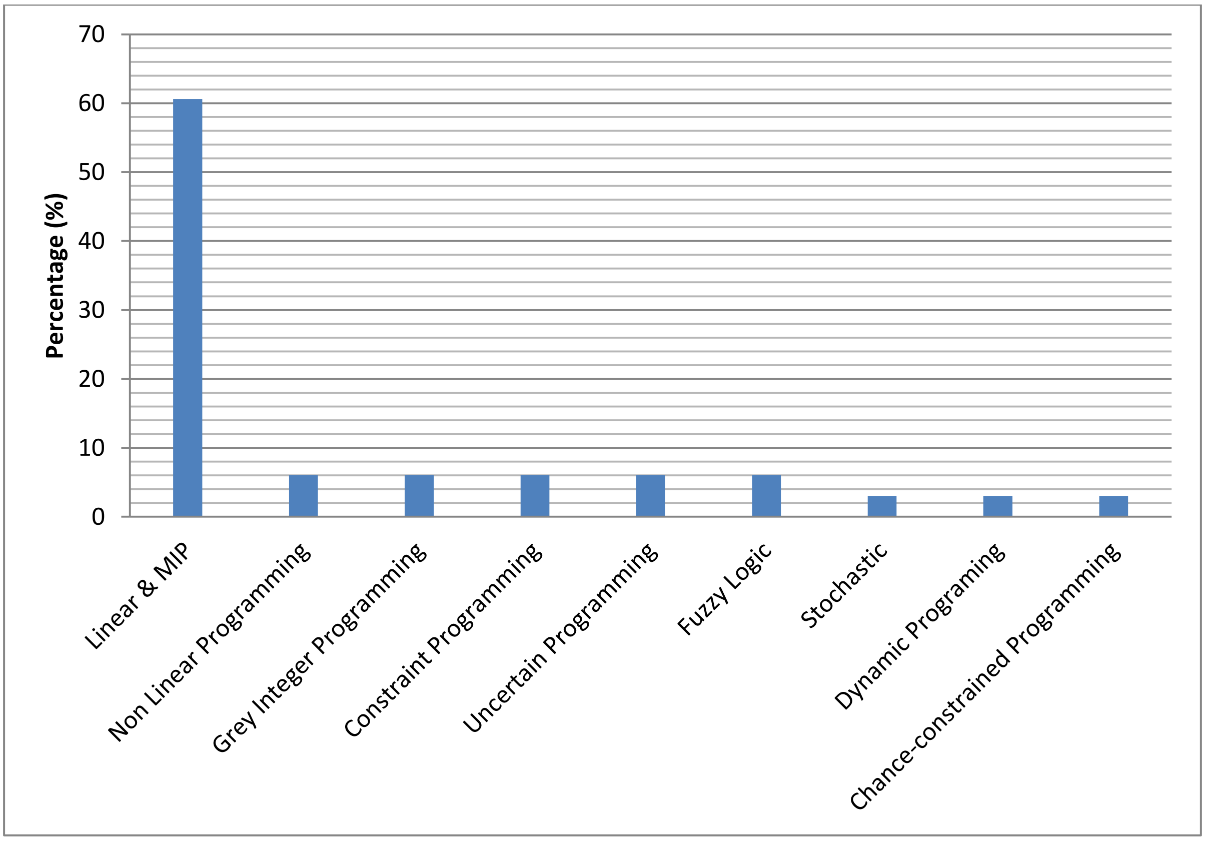


Figure 2.5: Various modelling approaches for PPSSP

The first attempt to model the PPSSP was made by Gupta et al. (1992) through dynamic programming. However, this model suffers from a very unrealistic assumption of implementing only one project at a time. This means the projects are just sequenced one after the other. Later the problem was modelled using 0-1 integer-linear programming (ILP) and mixed-integer linear programming (MILP) models by several authors. Coffin & Taylor (1996b) have proposed a 0-1 integer programming model to solve the problem. The authors have considered multiple objectives in the model. Ghazemzadeh et al. (1999) have also presented a 0-1 ILP model which considers the issues of mutual exclusiveness and non-uniformity of resource availability and consumption over the planning horizon of the portfolio. Sun & Ma (2005) presented a 0-1 ILP model for selection and scheduling of R&D projects where authors have considered only financial constraints.

Chen & Askin (2009) proposed a 0-1 ILP model which addresses a more complex version of PPSSP. The model includes the activity scheduling and time-dependent returns also. This model is further used by Shou and Huang (2010) and Shou et al. (2014) with minor modifications. Kolisch and Meyer (2006) and Shi et al. (2011) have also used the 0-1 ILP modelling approach to model their problem.

Ghorbani & Rabbani (2009) proposed a MILP model considering the resource usage variation and mutual exclusiveness among the projects. Tofighian & Naderi (2015) have modified the problem proposed by Ghorbani & Rabbani (2009) and modelled in the same way. Garcia (2014) developed an integer programming (IP) model considering limited inventory buffer and with hard due windows for the completion of the projects. Hosseininasab et al. (2015) presented three different models based on integer and mixed integer programming for urban road construction projects. The models also include network traffic assignment problems which increase the problem complexity.

Jafarzadeh et al. (2015) considered the reinvestment strategy taking the time horizon as flexible and formulated the problem as an integer program. Wang & Song (2016) also consider reinvestment strategy in PPSSP and presented the model in integer programming form. The authors also consider the time-dependent budget and time value of the capital. However, the planning horizon is kept fixed in the above mentioned case. Shariatmadari et al. (2017) developed an MILP model for the PPSSP with integrated resource management (IRM) approach. Ghahremani & Naderi (2015), Ganji et al. (2016) and Arratia M. et al. (2016) have used the same approach to model the PPSSP.

Liu & Wang (2011a) and Liu & Wang (2011b) have addressed various practical issues such interdependencies, periodical budget limitations and time-dependent resource constraints in modelling the PPSSP. The model is formulated using constraint programming. This model selects and schedules the projects in such a way that both renewable and non-renewable resource requirements are satisfied within the limited annual budget.

Gutjahr et al. (2008) presented a model for PPSSP with staff assignment. The authors formulated the problem using non-linear mixed-integer programming. The model pays special attention to competence development. Carazo et al. (2010) used a non-linear binary programming approach for the PPSSP. The model allows the transfer of the unconsumed resources from one period to the next. Project interdependencies and other resource constraints have also been taken into account.

Various authors have considered uncertainty in different project parameters of the problem and modelled them using suitable modelling approach. Coffin & Taylor (1996a) have represented the objectives of the problem as fuzzy sets. First, the problem is modelled as a 0-1 ILP model, and then fuzzy sets are created for each objective. The final objective function is obtained by taking the sum of grade membership of all the objectives. Gutjahr et al. (2007) have considered the human competencies as resources on which the project work time depends. Work times are treated as uncertain variable and problem is formulated using stochastic combinatorial optimization.

Zhao and Huang (2013) proposed a chance-constrained programming model to handle the uncertainty in the problem. The model treats cash inflows and cash outflows as stochastic variables. Huang and Zhao (2014) consider net income and investment cost as uncertain project parameters. An uncertain programming model is proposed for the problem with a new cost overrun risk. Uncertain programming is also employed by Huang et al. (2016) to formulate two models for the PPSSP. Net cash inflows and Initial outlays are the uncertain project parameters treated as variables. The two models are based on the method of measuring the risk by mean-variance and a mean-semi variance.

Pérez et al. (2017) offered a model with fuzzy renewable resource constraints. This uncertainty is handled by using fuzzy triangular numbers. The uncertain project parameters can be modelled using grey programming also. Amirian & Sahraeian (2017) have considered the uncertainty associated with project cost, budget limit and project duration. Authors have used the grey number to convert them to their grey equivalent, and a grey integer programming model is proposed. Amirian & Sahraeian (2018) have also used the grey integer programming to model their problem considering some uncertain parameters. These uncertain parameters are project profit, cost of variation in resource usage, and available and required resources.

2.6 Solution Approaches

The resource-constrained project scheduling problems (RCPSP) have already been identified as NP-hard in the literature (Demeulemeester & Herroelen, 2006). Thus, integration of selection with consideration of project interdependencies and scheduling makes it even more complex hence determining an optimal solution within a reasonable time is very difficult. In literature, a large number of solution methods exist for the PPSSP. These solution methodologies have been classified into four main categories: (1) exact, (2) heuristics, (3) meta-heuristics, and (4) others. The exact approaches and general exact solvers (e.g. GAMS, LINGO, and CPLEX etc.) have considered into one category. Figure 2.6 illustrates the frequencies of applying different solution approaches.

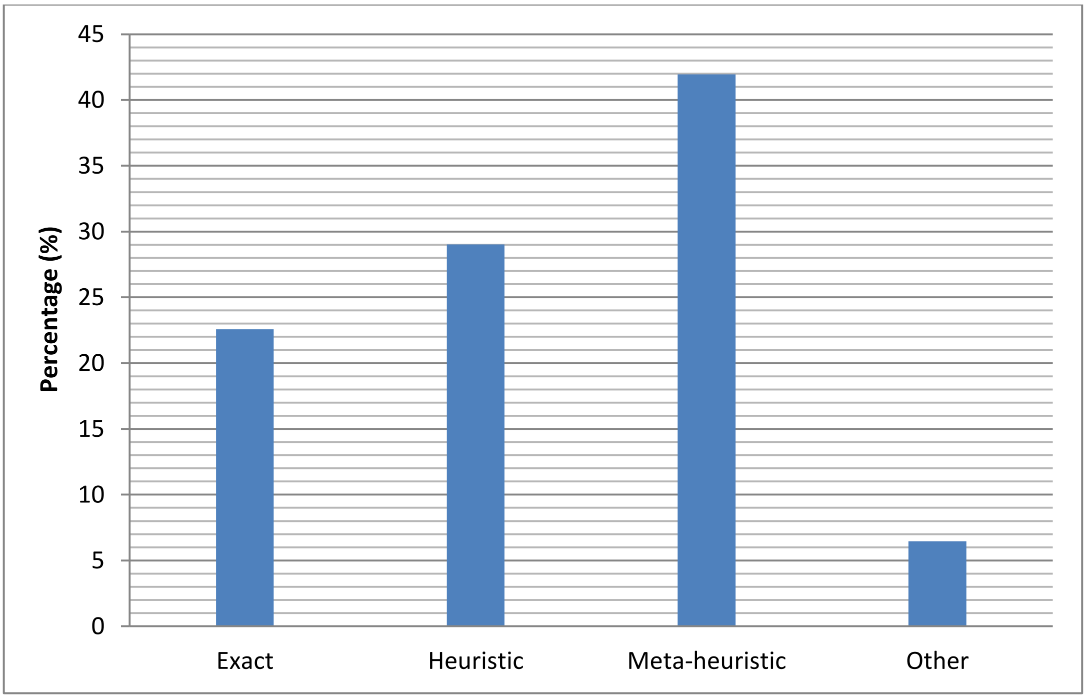


Figure 2.6: Solution methodologies for PPSSP

Some authors have used exact approaches to solve the PPSSP. Gupta et al. (1992) used dynamic programming for the PPSSP. But the problems considered have a very unrealistic condition of scheduling only one project at a time. Ghazemzadeh et al. (1999) used a branch & bound (B&B) method and taken a case example to validate the model and solved it using LINGO and CPLEX solvers. Jafarzadeh et al. (2015) have solved the integer programming model using BONMIN solver of GAMS software. Ganji et al. (2016) have also used GAMS software to solve small sized problems. Wang and Song (2016) solved the proposed integer programming model using LINGO and MATLAB solvers. Arratia M. et al. (2016) applied the B&B technique using CPLEX software to measure the computational effort required. Amirian & Sahraeian (2018) combined the two methods grey programming and B&B, and the resultant solution technique is named as grey integer programming branch & bound (GIPB). This technique provides Pareto interval solutions where the greyness in the input parameters determines the lengths of intervals. All these exact approaches are limited to small sized problems. It is almost impractical to solve the large-sized real-life problems with exact approaches.

Since exact methods can not handle large problems, several authors have developed heuristics to solve the PPSSP. Coffin & Taylor (1996b) proposed a filtered beam search (FBS) approach for multi-objective PPSSP model. The proposed FBS method uses two filters to restrict the poor solutions to proceed further in the search process. The first filter is based on the lowest important objective (probability of success) and the second filter is based on the second important objective (make-span). The final solution is selected on the basis of the most important objective (expected profit). However, this method may restrict the solutions at lower level search based on the poor performance for least important objective while the solution may have high performance for the most important objective.

Coffin & Taylor (1996a) presented a heuristic based on beam search and fuzzy logic approach for multi-objective PPSSP. Fuzzy logic has been used to obtain a single objective function from multiple objectives so that standard beam search can be applied. This technique is also associated with a trade-off between the computational effort and solution quality. Sun & Ma (2005) developed a heuristic approach for the problem named as packing-multiple-boxes. This heuristics approach is an extension of the packing-single-box method. Kolisch & Meyer (2006) outlined two heuristics for the PPSSP dealing with pharmaceutical research projects. The first heuristic is named as sequential scheduling which starts with empty portfolio and projects are selected one by one for scheduling. The second heuristic is named as concurrent method which starts with all projects selected initially and projects which are not feasible to scheduled are removed one by one.

Chen & Askin (2009) developed an implicit enumeration algorithm to select the best possible portfolio. The proposed heuristic uses the depth-first approach for sequencing of the projects. Zhao & Huang (2013) have also used the implicit enumeration algorithm for solving the proposed chance constrained programming model. But this algorithm has been applied in a different manner. First the probable groups (portfolios) of the projects are identified based on the logical relationship, and then each group is evaluated on the basis of NPV to find the best group. Liu & Wang (2011a; 2011b) proposed constraint programming approach to solve the PPSSP with time-dependent resource constraints. The consistency of the variables is checked using a backtracking method. This approach avoids unnecessary branching during the search process and hence improves the search efficiency. Garcia (2014) has developed a greedy heuristic for solving the IP model.

Meta-heuristic approaches have also been developed for PPSSP. Meta-heuristics provide reasonable trade-off between the computational effort and solution quality. Meta-heuristics are efficient, flexible and independent of the problem and model. Gutjahr et al. (2007) developed a stochastic variable neighbourhood search (S-VNS) meta-heuristic for the problem considering uncertainty in the work times. Gutjahr et al. (2008) developed two meta-heuristics: ACO and GA for the project selection part while a greedy heuristic has been employed for scheduling & staff assignment. Ghorbani & Rabbani (2009) developed a multi-objective scatter search (MOSS) algorithm to solve the multi-objective project selection and scheduling problem and compared their results with the NSGA II algorithm. Carazo et al. (2010) developed a Scatter Search (SS) for the multi-objective PPSSP which uses a tabu search (TS) for initialization of population. The solutions are then improved iteratively using the SS algorithm. Shi et al. (2011) presented a GA for solving the proposed 0-1 integer linear programming model. Huang and Zhao (2014) also developed GA to solve the PPSSP with uncertain project parameters.

Garcia (2014) developed a meta-heuristic approach by adding a randomized priority search to the greedy priority-based heuristic. This randomized priority search is used in a controlled manner to avoid local optima. Hosseininasab et al. (2015) have combined three approaches: the first phase of the two-phase simplex method, along with two meta-heuristics GA and

Frank-Wolfe algorithm to solve the problem of urban road construction projects. The first phase of two-phase simplex method is used for checking the feasibility of solutions generated by GA at the upper level. The feasible solutions identified are then carried over by the frank-wolfe algorithm at a lower level to get the final solution. Tofighian & Naderi (2015) proposed an ACO algorithm to maximize the total expected benefit of the selected projects and to minimize resource usage variation. The features of the ACO are devised to enhance the performance of the algorithm and suit the problem at hand. Ghahremani and Naderi (2015) have developed two meta-heuristics: GA and iterated greedy algorithm (IGA) for solving the PPSSP.

Huang et al. (2016) developed a hybrid intelligent algorithm (HIA) for solving the mean-semi-variance and mean-variance model with uncertain project parameters. Cellular automation (CA) and GA are combined to give proposed HIA. CA is employed to diversify the solutions throughout the search process. Since the problem contains uncertain projects parameters, the inverse uncertainty distributions are used to calculate the value of objective and constraints. Shariatmadari et al. (2017) proposed an integrated resource management approach and developed a gravitational search algorithm (GSA) for the problem. A heuristic is used to generate the initial set of solutions which are further updated by GSA update mechanism. Positions of agents are changed iteratively in GSA update process till the termination criterion is met. Amirian & Sahraeian (2017) considered grey parameters and proposed a modified grey shuffled frog leaping algorithm (GSFLA). Monte Carlo simulation is used to accommodate the greyness in the problem with embedded GSFLA. The censorship process is modified to suit the multi-objective problem. Four frogs are added to find the extreme point in the search space, and no frog is eliminated.

Some authors have used other techniques to solve the PPSSP. Shou and Huang (2010) proposed an iterative multi-unit combinatorial auction algorithm. A distributed bidding mechanism is used with two price update schemes in the combinatorial auction process. Shou et al. (2014) proposed a multi-agent evolutionary algorithm for solving the PPSSP. In order to enhance the evolution of agents, the algorithm is integrated with two new operators: neighbourhood competition and self-learning. The self-learning operator is designed with the adoption of GA.

2.7 Discussion

The results of the literature review in the domain of PPSSP are discussed in this section. Some potential research ideas can be derived from the analysis of the results.

A good research opportunity can be found in investigating the interdependencies among the projects. There exist three types of interdependencies among the projects namely benefit, resource and technical (Fox et al., 1984; Aaker & Tyebjee, 1978). These interdependencies have been considered frequently in project selection problems. But in PPSSP, project interdependencies have been considered in a limited way. From section 2.3.2 it is clear that out of the two technical interdependencies mutual exclusiveness is the only one which has got the attention of many researchers. Other important technical interdependency complementariness has been ignored in the literature. Further, resource interdependencies have been considered by Carazo et al. (2010) only. Consideration of resource interdependencies may lead to a great reduction in the overall cost of the portfolio due to resource sharing. The sharing of cost-intensive resources among two or more projects increases the utilization and hence a great saving in the cost. With consideration of resource interdependencies, some of the projects may get delayed which would have been implemented earlier otherwise. However, it requires scheduling of projects at activity level which results into increased problem complexity. Moreover, nobody has incorporated benefit interdependencies in their PPSSP model. In case of new product development projects benefit interdependencies play an important role.

Analysis of literature reveals that there are good opportunities for research in PPSSP with uncertainties. Authors have considered profits and costs of projects, project durations, budget, cash inflows and resource availabilities as the uncertain parameters. However, there could be other parameters also which may show uncertainty. These parameters include project success, skilled manpower availability and the rate of return of the project etc. Other factors also may influence the outcome of the PPSSP such as environmental, social, political and organizational risks.

Some vital future research direction can be extracted from the analysis of the literature on the basis of objectives considered for PPSSP. As mentioned in Section 2.4, most of the PPSSP problems have been modelled with a single objective. The project benefit maximization being the most popular. Further, in recent studies, the objective of benefit maximization has been shifted to expected benefit maximization where project benefit has been assumed to be time-sensitive. In practice, the single objective problems hardly persist. There are other objectives also which can be considered for the problem along with benefit maximization as illustrated in Table 2.4. Though, few multi-objective models exist in the literature for PPSSP still it is required to pay attention to the multi-objective problems in future. On the other hand, multi-objective problems are handled using different ways out of which obtaining the Pareto optimal solutions is suits the most. Opportunities are available for developing a more robust and reliable solution approaches to deal with multi-objective problems.

An important research direction revealed from the literature analysis is reinvestment strategy. Reinvestment of accrued benefits helps to add more projects to the portfolio and hence the overall benefit of the portfolio. A few papers have considered the reinvestment of benefits in the portfolio as illustrated in Section 2.3.3. The assumptions made by previous works can be simplified to address more realistic problems. The reinvestment of benefits can be seen with other perspectives also. For instance, the project benefits can be invested towards the payment of the loans. Earlier works are limited to reinvesting the benefits to the portfolio by adding it to the budget which in turn gets consumed for increasing availability of consumable resources. One more way to reinvest the benefits is to use the part of the generated benefit to increase the levels of renewable resources too instead of simply adding to the budget.

New and interesting research tracks can be found by analysis of the literature from the perspective of consideration of the problem in static and dynamic environments. Literature analysis reveals that all problems have been considered in a static environment. The portfolios are selected and scheduled as closed portfolios, and new incoming projects are considered as a candidate for new portfolio. Integration of a new candidate project to the existing portfolio could be an interesting research direction for further research. A new project if integrated into the existing portfolio influences the decision on other projects in the portfolio partially implemented or waiting to be implemented. A good amount of research is already available on the scheduling of multiple projects in a dynamic environment. However, it makes the decision making complex and repetitive. Development of suitable solution approaches would be essential then for such problems.

The PPSSP belongs to NP-hard category of the problems. Consideration of practical situations makes the problem more complicated and almost impossible to be handled by exact approaches. The literature review reveals that exact approaches are either used for very simplified problems or very small problems. Heuristics, on the other hand, suffer from their inherent greedy nature. In comparison to exact and heuristic approaches, the meta-heuristics are found to more promising and used widely in solving complex PPSSP’s. The meta-heuristics are independent of the problem type and nature. The meta-heuristics can be utilized for solving the problems with uncertainties also (Gutjahr et al., 2007; Amirian & Sahraeian, 2017; Huang et al., 2016). The meta-heuristics can be integrated with other solution approaches easily (Hosseininasab et al., 2015; Huang et al., 2016; Shariatmadari et al., 2017; Amirian & Sahraeian, 2017; Carazo et al., 2010). However, it is difficult to decide which meta-heuristic approach suits best to a specific problem. One more issue to be managed is the trade-off between computational effort and solution quality. Since, no meta-heuristic guarantees the optimality of the solution obtained, so it is important to choose the meta-heuristics wisely.

The literature review also reveals that the problem formulation, decision variables and constraints play a vital role in choosing a solution approach. For example, most of the problems which contain some uncertain parameters are solved using heuristics or meta-heuristics.

2.8 Research Gaps

The research gaps identified after analysing the results of the review can be listed as below:

1. Out of the two technical interdependencies, mutual exclusiveness has been considered by many authors in the PPSSP. There is, however, another important interdependency – complementariness - yet, ignored in the existing literature.
2. Some projects may alter the outcome of each other when selected simultaneously in the portfolio. This happens due to the existence of benefit interdependencies among them. So, it is important to address the benefit interdependencies to estimate the total expected benefit more accurately.
3. The resource interdependencies among the projects affect the aggregate resource requirement of the portfolio and may lead to reduced resource requirements. In case of projects demanding same cost intensive resources (e.g. machine, equipment or a lab), the projects need to be scheduled in such a manner that the same resource can be used for all such projects. The actual benefit of resources interdependencies can be realized with the detailed scheduling of activities. Activity scheduling also helps to determine the start and

finish time of each project precisely. However, in PPSSP, scheduling of projects at activity level increases the complexity of the problem.

4. In the absence of proper information or historical data, some parameters may be associated with the uncertainties. For example, in the case of R&D, new product development, and innovation projects, the exact information about the expected benefit, cost, duration and cash flows is not available.
5. In most of the PPSSP formulations, project benefit has been considered as the only objective, with less attention being paid to the multi-objective problems. There are other objectives also which should be considered with project benefit. These objectives may be conflicting in nature. For example, resource usage variation and strategic benefit realized form the consideration of benefit interdependent projects.
6. Handling of the multiple and conflicting objectives is another critical task. Posteriori approaches are best suited to handle such problems. Efficient meta-heuristics need to be developed to generate Pareto solutions for the problem. The efforts should also be made towards the development of hybrid algorithms to solve complex PPSSP.
7. The return from a project after completion can be reinvested in the portfolio. More profitable and early completed projects provide an adequate budget to select more projects. The issue of reinvestment strategy in integrated project selection and scheduling problem has been scantly addressed. The problem becomes more difficult with consideration of reinvestment of benefits.
8. In PPSSP, the projects are selected scheduled in a static environment. The problem can be formulated in the dynamic environment also. The best portfolio is selected initially based on the objectives of the problem. The existing portfolio is kept open for the new projects to be part of the portfolio. New projects can be included in the portfolio based on the project features. The dynamic selection and scheduling of projects has not been considered yet.

Chapter-3

Single-objective Project Portfolio Selection and Scheduling with Technical Interdependencies

Chapter 3
Single-objective Project Portfolio Selection and Scheduling with Technical Interdependencies

3.1 Introduction

The chapter considers the single objective PPSSP with two types of technical interdependencies: mutual exclusiveness and complementariness. The problem is formulated as 0-1 ILP model and a modified TLBO algorithm has been proposed for the problem. In order to improve the performance of the algorithm, the hybridization of the TLBO with well-known tabu search algorithm is proposed. The proposed algorithms are tested on four different complexity level data sets generated in this research. Performance of the proposed algorithms has been compared with the existing algorithms available in the literature for the problem.

The chapter has been organized in the following manner. Section 3.2 proposes an improved mathematical model for the single objective PPSSP. In Section 3.3 the methodology for the proposed algorithms has been described. The scheme for test problem generation, parameter settings for the proposed algorithms, results obtained and performance of the algorithms is discussed in Section 3.4. The chapter is concluded in Section 3.5.

3.2 Problem definition and mathematical formulation

This section presents a zero-one integer linear programming model for the PPSSP and an illustrative example for the same. Let, there be a set of *N* projects out of which a subset of projects is to be selected optimally respecting resource availability constraints and interdependencies. The projects may have two types of technical interdependencies among them viz. mutual exclusiveness (technical) and complementariness (technical). *K* types of renewable resources are needed to carry out the portfolio of the selected projects in a planning horizon spanning *T* time periods. The resources are available in limited quantity during each period of the planning horizon. The objective of the problem is to maximize the total expected benefit from a selected portfolio of projects. The expected benefit from a project is considered to be time-dependent. This means a delay in the implementation of a project will lead to a decrease in expected benefit.

3.2.1 Mathematical model:

A formal mathematical model, its decision variables and coefficients are as below:

Decision variable:

X_{it} = 1; if project *i* is selected and starts in period *t*,

0; otherwise.

Technological coefficients and parameters:

N = number of candidate projects; (*i = 1,2, ……. , N*)

K = number of resource types; (*k = 1,2,……., K*)

T = time periods; (*t = 1,2,……. , T*)

P_{it} = expected profit if project *i* starts in period *t*.

d_i = duration of the project *i*.

r_{ik} = requirement of resource type *k* for project *i* in each time period.

R_{kt} = resource availability of type *k* in period *t*.

h = project which is complementary (technical) to project *i*.

H_i = set of projects which are complementary (technical) to *i*.

e = project mutually exclusive (technical) to project *i*.

E_i = set of projects mutually exclusive (technical) to project *i*.

Formulation:

Objective function:

Max $$\sum_{i=1}^{N} \sum_{t=1}^{T-d_i+1} P_{it} * X_{it} \tag{3.1}$$

Constraints:

$$\sum_{t=1}^{T-d_i+1} X_{it} \leq 1 \qquad \forall\, i \qquad (3.2)$$

$$\sum_{i=1}^{N} \left(\sum_{j=\max\{1,t-d_i+1\}}^{\min\{t,T-d_i+1\}} X_{ij}\right) * r_{ik} \ \leq R_{kt} \qquad \forall\, k, t \qquad (3.3)$$

$$\sum_{t=1}^{T-d_i+1} X_{it} + \sum_{t=1}^{T-d_e+1} X_{et} \leq 1 \qquad \forall\, i, e \in E_i \qquad (3.4)$$

$$\sum_{t=1}^{T-d_i+1} X_{it} = \sum_{t=1}^{T-d_h+1} X_{ht} \qquad \forall\, i, h \in H_i \qquad (3.5)$$

$$X_{it} = [0,1] \qquad \forall\, i, t \leq T - d_i + 1 \qquad (3.6)$$

The objective function (3.1) is for maximization of the total expected benefit from the selected portfolio of the projects. Constraint (3.2) determines the starting time of each of the selected project(s) so that it is completed within the planning horizon. Constraint (3.3) ensures that the resource availability limitations are not violated. Constraints (3.4) and (3.5) enforce the mutual exclusiveness (technical) and complementariness (technical) among the projects respectively. Constraint (3.6) ensures the decision variables to be binary.

3.2.2 An illustrative example:

In this subsection, a small hypothetical problem with five projects is considered to illustrate the mathematical model. Duration of each of the projects along with the resource requirement of two types of resources is given in Table 3.1. Table 3.2 shows the expected benefits for each project and Table 3.3 shows interdependencies among the projects.

Table 3.1: Project durations and resource requirements

Projects	Duration	Resource Requirement (Type 1)	Resource Requirement (Type 2)
1	5	4	3
2	6	3	2
3	4	4	1
4	3	2	3
5	5	1	3

In Table 3.3, C denotes the complementariness (technical) and E denotes the mutual exclusiveness (technical) among the projects. Eight units of each of the two types of resources are available during every period of the planning horizon. The planning horizon is considered to be eight time periods.

Table 3.2: The expected benefit in each time period

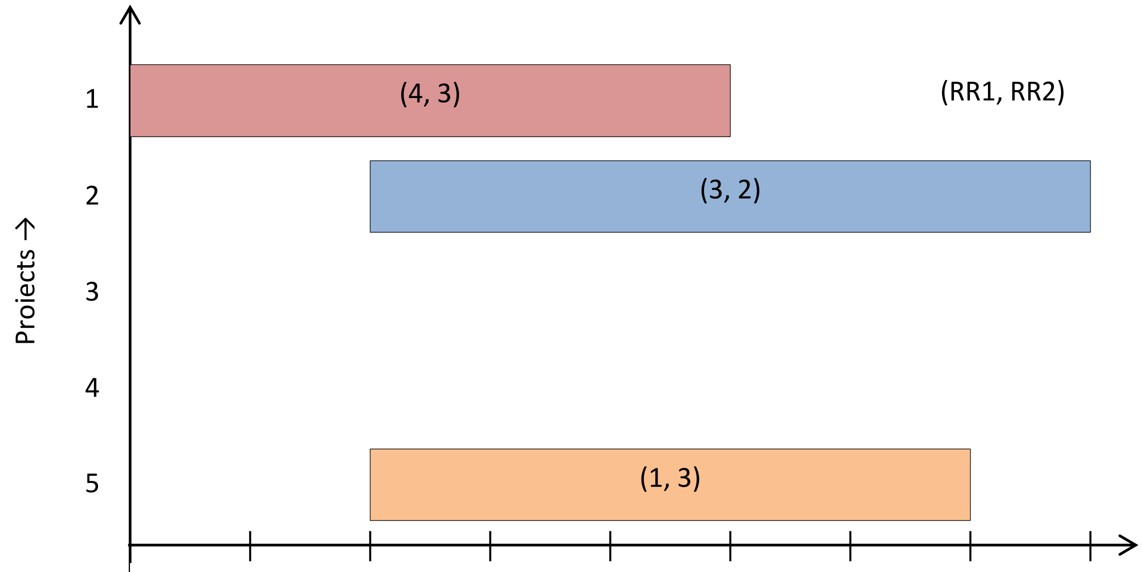
Projects	Time Periods							
	1	2	3	4	5	6	7	8
1	96	83	77	76	75	64	64	56
2	93	84	67	62	59	52	38	33
3	97	65	62	45	31	27	23	20
4	91	81	78	57	51	47	39	32
5	99	80	71	63	35	25	25	23

Table 3.3: Interdependencies between projects

Projects	Projects				
	1	2	3	4	5
1	-	C	E		
2	C	-		E	
3	E		-	C	
4		E	C	-	E
5				E	-

E = Mutual exclusiveness (technical), C = Complementariness (technical)

A feasible solution of the problem is $X_{1,1} = 1$, $X_{2,3} = 1$ *and* $X_{5,3} = 1$. Figure 3.1 shows the schedule of the selected projects. As can be seen, projects 1, 2 and 5 have been selected and began at the start of periods 1, 3 and 3 respectively. Projects 3 and 4 have not been selected as they have mutual exclusiveness (technical) with projects 1 and 2 respectively. Similarly, projects 1 and 2 are complementary (technical) to each other thus have been selected together. The total expected benefit from the portfolio equals 234 (96+67+71 = 234) units.



Time periods →	1	2	3	4	5	6	7	8
CRR1	4	4	8	8	8	4	4	3
CRR2	3	3	8	8	8	5	5	2
Max. Resource Availability	8	8	8	8	8	8	8	8

RR = Resource Requirement, CRR = Cumulative Resource Requirement

Figure 3.1: Project schedule and cumulative resource requirements

3.3 Proposed Algorithms for PPSSP

3.3.1 Teaching-learning-based optimization algorithm

Teaching-learning-based optimization (TLBO) algorithm, developed by Rao, Savsani, & Vakharia (2011, 2012), is one of the population-based optimization methods. The algorithm imitates the classroom interaction between a teacher and students. Here teacher and students represent solutions to the problem, and the best student (solution) is considered as the teacher. It works on two basic methods of learning- the interaction between teacher and students (known as teacher phase) and the interaction among the students (known as student phase). To improve upon the algorithm further, self-study phase is incorporated in which self-learning takes place.

The flowchart for the proposed TLBO algorithm for the PPSSP is shown in Figure 3.2. The encoding scheme and the steps involved in the algorithm, i.e. population initialization process, teacher phase, student phase and self-study phase are explained next.

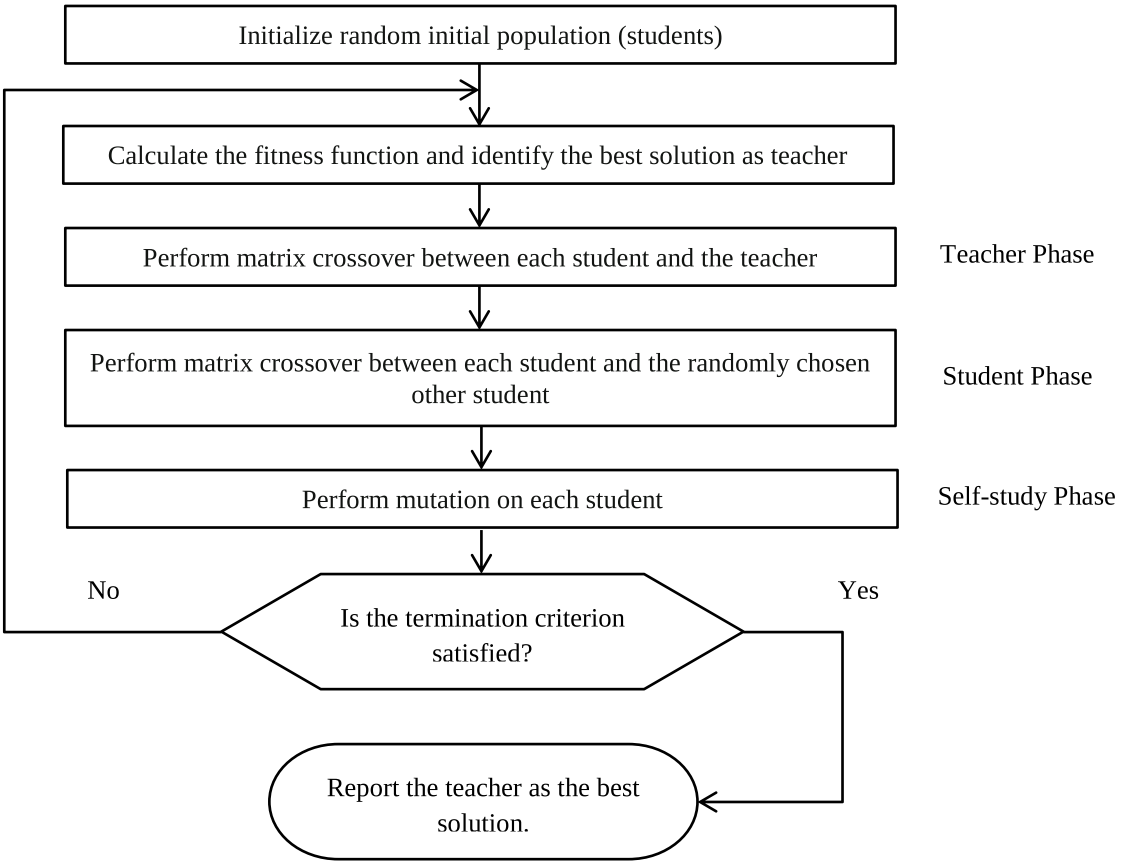

Initialize random initial population (students)
Calculate the fitness function and identify the best solution as teacher
Perform matrix crossover between each student and the teacher
Teacher Phase
Perform matrix crossover between each student and the randomly chosen other student
Student Phase
Perform mutation on each student
Self-study Phase
No
Yes
Is the termination criterion satisfied?
Report the teacher as the best solution.

Figure 3.2: TLBO Algorithm Flowchart for PPSSP

3.3.1.1 Encoding scheme

The encoding scheme is an essential element of any meta-heuristic algorithm. The scheme used for the proposed algorithm is the same as used by Ghorbani & Rabbani (2009). In this scheme, a solution to the problem (student or teacher) is represented by a P x T matrix in which each row represents a project, and each column represents a time period. If project i is selected and started in time period t then element a_{it} of the matrix will be 1 else 0. An example matrix (5 x 8) for a problem having 5 candidate projects with a planning horizon of 8 time periods is shown in Table 3.4. In given solution projects 1, 2 and 5 are selected and started at times 1, 3 and 3 respectively.

3.3.1.2 Population initialization

A population with N students (initial feasible solutions) is generated randomly. To generate a student, a project is selected randomly from the set of candidate projects, and then scheduled at the earliest possible time, satisfying the resource availability constraints. Once this project is scheduled, its mutually exclusive projects are excluded from the set of candidate projects. Then all its complimentary projects are scheduled in random order. If all the projects in the set of complementary projects are scheduled, all of the projects in the set are selected and set of the candidate projects is updated. This process continues till no more selection is possible. Pseudo code for the process is given in Figure 3.3.

Table 3.4: Encoding scheme for a feasible solution

Projects	1	2	3	4	5	6	7	8
P1	1	0	0	0	0	0	0	0
P2	0	0	1	0	0	0	0	0
P3	0	0	0	0	0	0	0	0
P4	0	0	0	0	0	0	0	0
P5	0	0	1	0	0	0	0	0

Time Periods

3.3.1.3 Teacher phase – crossover

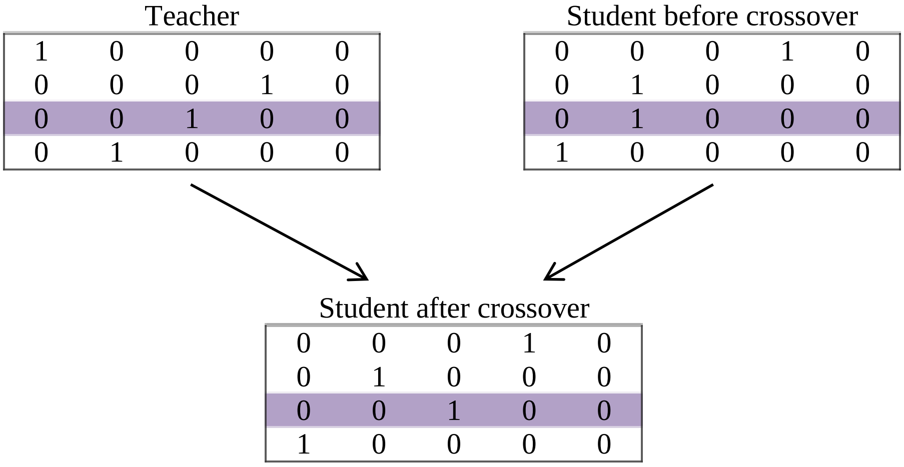
In this phase of TLBO, individual with the best fitness function value is identified as the teacher. The teacher tries to improve the performance of other individuals through the crossover. For a crossover, a predetermined number of rows from the selected projects are chosen randomly from the teacher and are transferred to the corresponding rows of remaining individuals maintaining the resource feasibility of the solution. The number of rows transferred in teacher phase is taken as a predetermined percentage of the size of the problem (number of projects) and is known as teacher-student crossover coefficient (TSC). The individuals with improved fitness function are accepted for inclusion in the population for student phase. Crossover between teacher and a student is shown in Figure 3.4.

3.3.1.4 Student phase – crossover operator

In this phase of TLBO, individuals improve their performance via interaction among themselves. An individual X interacts with individual Y chosen randomly where few rows of the individual X are replaced with corresponding rows of individual Y, maintaining the resource feasibility of the solution. The number of rows is taken as a predetermined percentage of the size of the problem and is known as the student-student crossover coefficient (SSC). The individuals with improved fitness function are accepted for inclusion in the population for the self-study phase. This crossover is performed between students in the same manner as shown in Figure 3.4.

```
Population initialization for TLBO
Let     $A$ = set of candidate projects; $i = \{1,2,\ldots\ldots..N\}$
        $C_i$ = set of projects complementary to project $i$;
        $E_i$ = set of projects mutually exclusive to project $i$;
        $S$ = set of selected projects, and
        $D$ = set of projects which are not feasible to be scheduled.
Initialization: $A = \{i = 1,2,\ldots\ldots N\}; S = D = \emptyset,$
Start
Schedule project $i = \{1,2,\ldots\ldots.N\}$ (selected randomly)
        If project $i$ is scheduled then
                If $C_i \neq \emptyset$
                        If all the projects in $C_i$ are scheduled
                                        $S = S \cup \{i \cup C_i\}$
                                        $A = A \backslash \{i \cup C_i \cup E_i\}$
                        Else    $D = D \cup \{i \cup C_i\}$
                                $A = A \backslash \{i \cup C_i\}$
                Else    $S = S \cup \{i\}$
                        $A = A \backslash \{i \cup E_i\}$
        Else    $D = D \cup \{i \cup C_i\}$
                $A = A \backslash \{i \cup C_i\}$
Continue till $A = \emptyset$
End
```

Figure 3.3: Pseudo code for population initialization



Teacher

1	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	1	0	0	0

Student before crossover

0	0	0	1	0
0	1	0	0	0
0	1	0	0	0
1	0	0	0	0

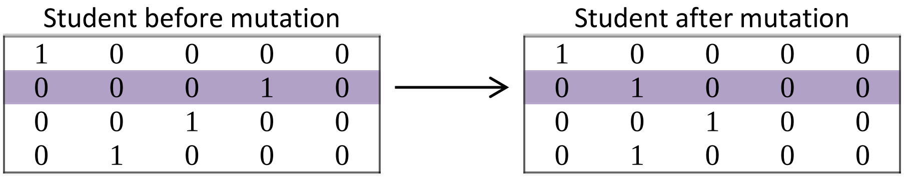
Student after crossover

0	0	0	1	0
0	1	0	0	0
0	0	1	0	0
1	0	0	0	0

Figure 3.4: Crossover in teacher phase

3.3.1.5 Self-study phase – mutation operator

In this phase of TLBO, a few rows of an individual are selected and their starting times are changed randomly maintaining the resource feasibility of the solution. The number of rows selected for self-study is taken as a predetermined percentage of the size of the problem and is known as self-study mutation coefficient (SSM). This operator helps the algorithm to explore the solution space. Individuals with better fitness function are accepted, and the population is updated. This new population becomes input to the teacher phase for the next iteration. Figure 3.5 shows the self-study phase applied to an individual. The algorithm terminates after a predetermined number of iterations.



Student before mutation

1	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	1	0	0	0

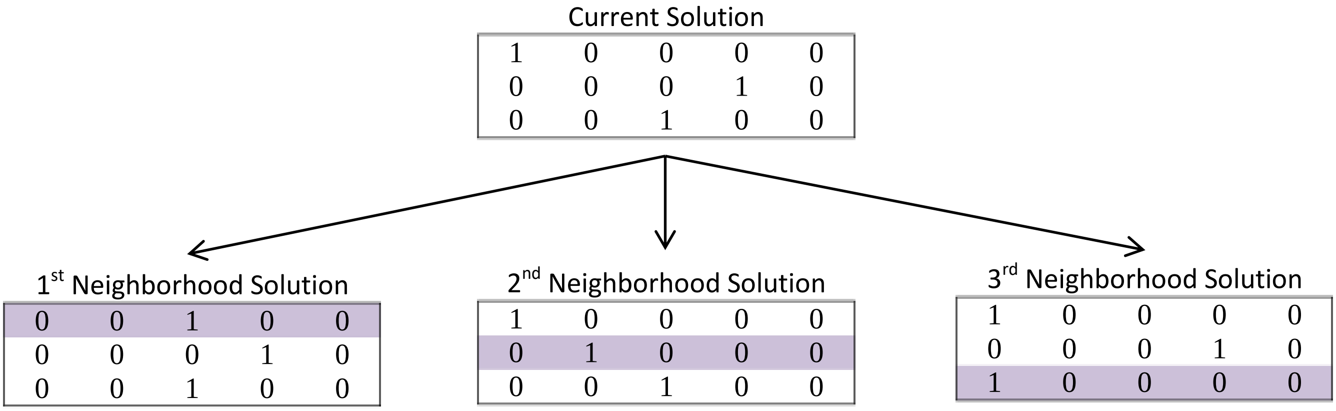
Student after mutation

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	1	0	0	0

Figure 3.5: Mutation in self-study phase

3.3.2 Tabu Search Algorithm for PPSSP

Tabu search algorithm is a powerful local search algorithm, developed by Glover (1986), and has been successfully used for solving a variety of combinatorial optimization problems including project scheduling (Lambrechts, Demeulemeester & Herroelen, 2008; Mika, Waligora & Weglarz, 2008; Pan, Hsaio & Chen, 2008; Waligóra, 2008; Li, Pan & Liang, 2010; Shen, 2014; Sukkerd & Wuttipornpun, 2016). The algorithm is known for strong local search which prevents its early convergence. The solution representation for TS is the same as that used for TLBO. Neighbourhood solutions (moves) of the current solution are generated by randomly changing the start time of one of the projects in the solution, so that new solution lies in the vicinity of the current solution. Changes in start times are applied respecting the time and resource constraints. This gives neighbourhood solutions equal to the number of projects in the current solution. Scheme for generating neighbourhood solutions is given in Figure 3.6.



Current Solution

1	0	0	0	0
0	0	0	1	0
0	0	1	0	0

1st Neighborhood Solution

0	0	1	0	0
0	0	0	1	0
0	0	1	0	0

2nd Neighborhood Solution

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0

3rd Neighborhood Solution

1	0	0	0	0
0	0	0	1	0
1	0	0	0	0

Figure 3.6: Creating neighborhood solutions for TS

The size of tabu list is taken to be a function of the size of the problem and selection of restricted move, if it offers a better solution than the best solution found so far, is considered as the aspiration criteria.

3.3.3 Hybrid TLBO-TS Algorithm for PPSSP

TLBO is a population-based algorithm which has very good exploration capability but lack in exploiting the solution space locally. To improve the exploitation capability of the algorithm the TLBO is combined with tabu search algorithm, having good local searchability, and a hybrid TLBO-TS optimization algorithm has been developed for PPSSP. The hybrid TLBO-TS optimization algorithm starts with an initial population of students then TLBO algorithm is applied as discussed in Section 3.3.1. The best solution identified after applying TLBO is considered as an initial solution for TS. After applying TS, the best solution is fed back to the TLBO algorithm as a new teacher. The algorithm is iterated for a predetermined number of times. The flowchart for the hybrid TLBO-TS optimization algorithm is given in Figure 3.7.

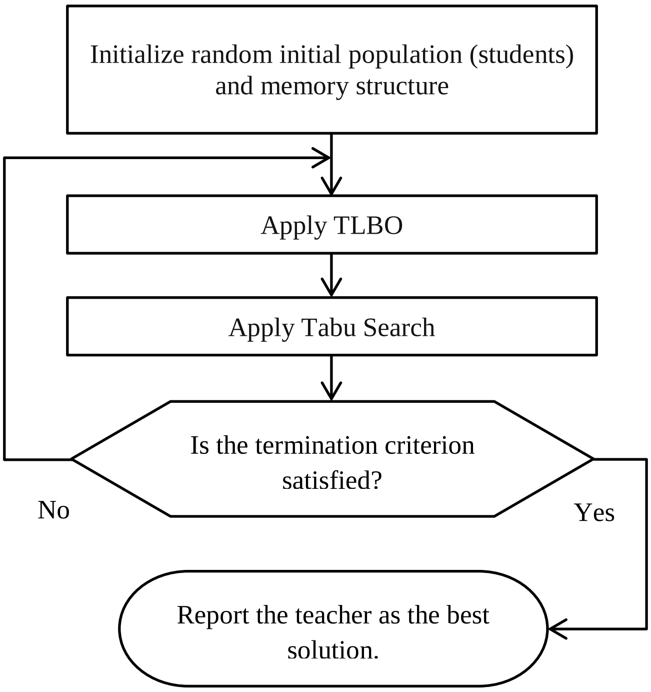

Initialize random initial population (students) and memory structure
Apply TLBO
Apply Tabu Search
Is the termination criterion satisfied?
No
Yes
Report the teacher as the best solution.

Figure 3.7: Hybrid TLBO-TS Algorithm Flowchart for PPSSP

3.4 Computational experiences

In this section, the performance of the proposed meta-heuristics for PPSSP - TLBO algorithm, TS algorithm, and hybrid TLBO-TS algorithm is evaluated. The three algorithms are compared with each other and are also compared with the Shuffled Frog Leaping Algorithm (SFLA) developed by Amirian & Sahraeian (2017). They developed SFLA for the project selection and scheduling problem with multiple objectives and grey project parameters hence the algorithms developed in this chapter are not directly comparable. To make SFLA comparable with proposed algorithms, SFLA is adapted by using simple sorting on the basis of maximum expected benefit instead of non-dominated sorting of the frogs in local and global search as well as in shuffling process. Additionally, the single frog is used in the censorship process instead of four. The algorithms are evaluated on four sets of problems each consisting of 16 instances of varying size and complexity. All these algorithms are coded in MATLAB 7.12 environment and executed on a laptop computer with Core i3, and Windows 8.1 using 4 GB of RAM. The generation of test problems, parameter setting and evaluation criterion are explained next.

3.4.1 Test problems

Four data sets with different complexity levels are generated to test the performance of the three algorithms. These datasets are generated using the generation scheme given by Tofighian & Naderi (2015) with the addition of complementarity (technical) between the projects. The data sets have been generated with varying complexity viz. low, moderate, high, and large sized problems. Large sized problems are designed to be with high complexity and large number of candidate projects. The complexity is increased by increasing number of interdependencies, number of resources required, project durations and decreasing the make-span and availability of resources. Table 3.5 shows the generation scheme for low, moderate, high and large-sized problem instances. Expected benefits for all projects are generated randomly in descending order using uniform distribution U (100,999).

Table 3.5: Instance generation schemes

Factor	**Generating Rule**			
	Low Complexity Instances	**Moderate Complexity Instances**	**High Complexity Instances**	**Large Size Instances**
Number of available projects	n = {3, 4, . . . , 10}	n = {7, 8, . . . , 14}	n = {7, 8, . . . , 14}	n = {15, 16, . . . , 22}
Durations	d = U(1, 3)	d = U(3, 7)	d = U(7, 10)	d = U(10, 15)
Number of different types of resources	m = {1, 2}	m = {2, 3, 4}	m = {4, 5}	m = {4, 5}
Required resources for each projects	a = U(10, 15)	a = U(10, 15)	a = U(10, 15)	a = U(10, 15)
Available amount of resource	r = 4U(10, 15)	r = 3U(10, 15)	r = 2U(10, 15)	r = 2U(10, 15)
Make-span	$T = \max\{\max(d_i),$ $\sum_i d_i$ x $U(0.8,1)\}$	$T = \max\{\max(d_i),$ 0.8 x $U(\max(d_i),$ $\sum_i d_i)\}$	$T = \max\{\max(d_i),$ 0.5 x $U(\max(d_i),$ $\sum_i d_i)\}$	$T = \max\{\max(d_i),$ 0.5 x $U(\max(d_i),$ $\sum_i d_i)\}$
Prob. of technical interdependency (Mutually exclusive projects)	10 %	20 %	30 %	30 %
Prob. of technical interdependency (Complementary projects)	5 %	10 %	15 %	15

3.4.2 Parameter setting

Tuning of parameters influences the effectiveness of a meta-heuristic algorithm. This section discusses the tuning of common control parameters and algorithm-specific parameters for the three algorithms-TLBO, TS and hybrid TLBO-TS developed in this research for PPSSP. As the number of parameters and their levels are quite large Taguchi approach is used to design the experiments.

3.4.2.1 Parameter setting for TLBO algorithm

Proposed TLBO algorithm has 4 factors: N (Number of students), TSC (teacher-student crossover coefficient), SSC (student-student crossover coefficient) and SSM (self-study mutation coefficient). Levels of the above factors are given in Table 3.6. The L9 orthogonal array is selected for experimentation as presented in Table 3.7. Figure 3.8 shows the main effects plot for means at each level. Optimum levels obtained from the experimentations are: N = 15, TSC = 0.5, SSC = 0.4 and SSM = 0.1.

Table 3.6: Factors and levels for TLBO algorithm

Factor	Symbol	Level	Values		
Number of students (Pop Size)	N	3	10	15	20
Teacher-student crossover coefficient	TSC	3	0.30	0.40	0.50
Student-student crossover coefficient	SSC	3	0.20	0.30	0.40
Self-study mutation coefficient	SSM	3	0.10	0.20	0.30

Table 3.7: The Taguchi orthogonal array L9 for parameter setting for TLBO algorithm

Trial No.	N	TSC	SSC	SSM
1	10	0.3	0.2	0.1
2	10	0.4	0.3	0.2
3	10	0.5	0.4	0.3
4	15	0.3	0.3	0.3
5	15	0.4	0.4	0.1
6	15	0.5	0.2	0.2
7	20	0.3	0.4	0.2
8	20	0.4	0.2	0.3
9	20	0.5	0.3	0.1

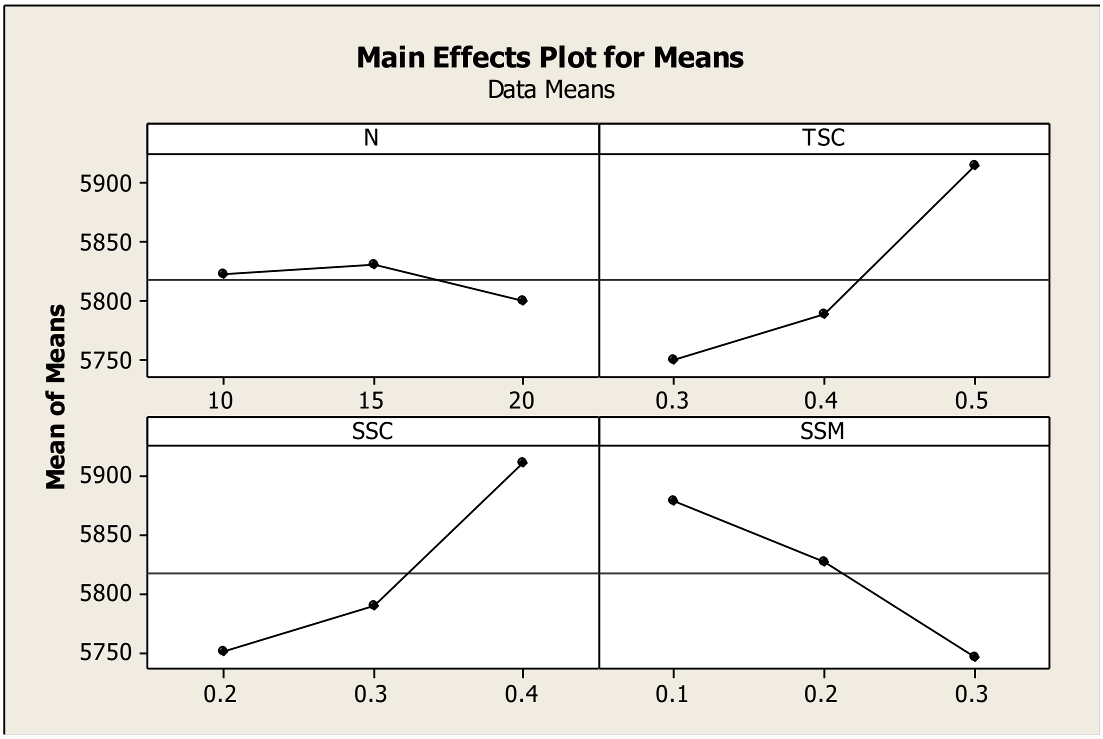


Figure 3.8: The mean of means ratio plot for each level of factors of TLBO

3.4.2.2 Parameter setting for Tabu search algorithm

For TS algorithm the size of tabu list (TL) is the only parameter to be controlled and needs to be tuned properly for the effectiveness of the algorithm. For the proposed TS algorithm the size of tabu list is taken as 0.4 times the size of the problem (number of candidate projects).

3.4.2.3 Parameter setting for Hybrid TLBO-TS algorithm

The proposed hybrid TLBO-TS algorithm has five parameters: N, TSC, SSC, SSM, and TL. Different levels of these parameters considered for experimentation are shown in Table 3.8. The L18 orthogonal array has been used for experimentation (Table 3.9). Figure 3.9 shows the main effects plot for means at each level of parameters. Optimum levels of parameters are obtained as: N = 10, TSC = 0.4, SSC = 0.3, SSM = 0.3 and TL = 0.4.

Table 3.8: Factors and levels for Hybrid TLBO-TS algorithm

Factor	Symbol	Level	Values		
Number of students (Pop Size)	N	3	10	15	20
Teacher-student crossover coefficient	TSC	3	0.30	0.40	0.50
Student-student crossover coefficient	SSC	3	0.20	0.30	0.40
Self-study mutation coefficient	SSM	3	0.10	0.20	0.30
Length of Tabu list	TL	3	0.40	0.50	0.60

Table 3.9: The modified Taguchi orthogonal array L18 for Hybrid TLBO-TS algorithm

Trial No.	N	TSC	SSC	SSM	TL
1	10	0.3	0.2	0.1	0.4
2	10	0.4	0.3	0.2	0.5
3	10	0.5	0.4	0.3	0.6
4	15	0.3	0.2	0.2	0.5
5	15	0.4	0.3	0.3	0.6
6	15	0.5	0.4	0.1	0.4
7	20	0.3	0.3	0.1	0.6
8	20	0.4	0.4	0.2	0.4
9	20	0.5	0.2	0.3	0.5
10	10	0.3	0.4	0.3	0.5
11	10	0.4	0.2	0.1	0.6
12	10	0.5	0.3	0.2	0.4
13	15	0.3	0.3	0.3	0.4
14	15	0.4	0.4	0.1	0.5
15	15	0.5	0.2	0.2	0.6
16	20	0.3	0.4	0.2	0.6
17	20	0.4	0.2	0.3	0.4
18	20	0.5	0.3	0.1	0.5

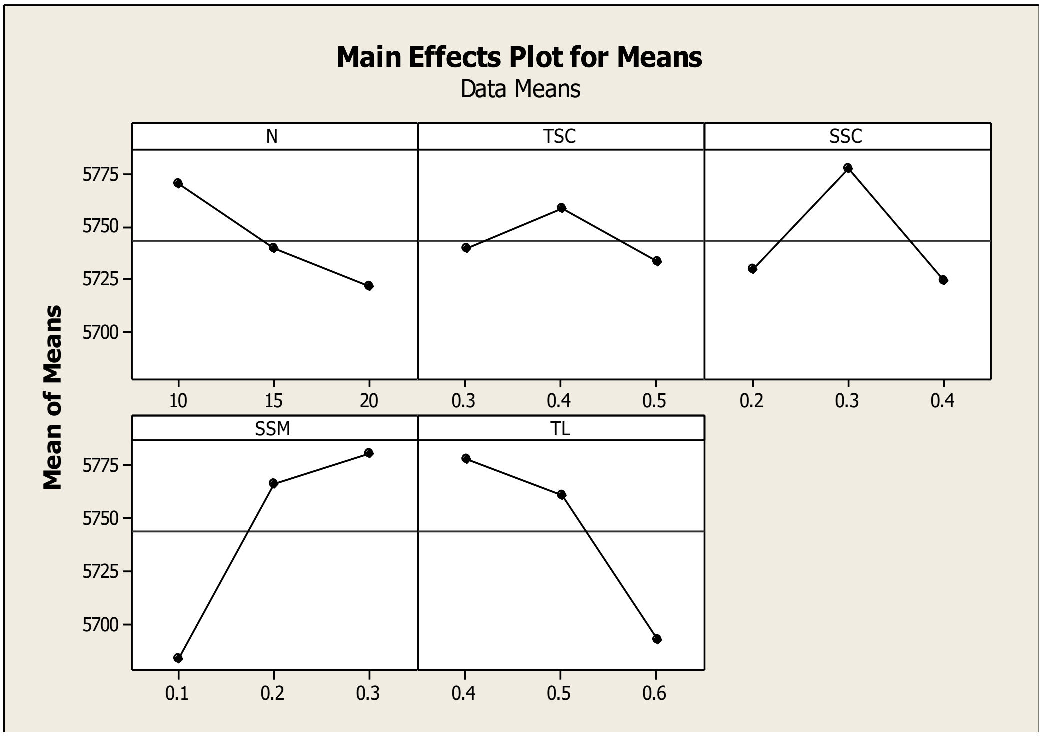


Figure 3.9: The mean of means ratio plot for each level of factors of Hybrid TLBO-TS

3.4.3 Comparison Metric

One of the ways to evaluate the performance of an algorithm is to compare the solutions by the algorithm with the optimum for a set of problems. But, as the PPSSP with both exclusiveness and complementariness between the projects has been considered for the first time in this research, the benchmark results are not available. Hence, the best value obtained from any of the three proposed algorithms and SFLA is used for comparison. For each problem instance, the percentage deviation is calculated by expression 3.7.

$$PD = \frac{(B-X)}{B} * 100 \qquad (3.7)$$

Where X is the objective function value obtained for a problem instance using an algorithm and B is the best value found for the problem by any of these algorithms.

3.4.4 Results

Each problem instance in the problem sets is solved 10 times by an algorithm, and the results are averaged to find percentage deviation for the problem. Each of the algorithms is run for 100 iterations. Table 3.10 shows the average of percentage deviations for each problem set when solved by the four algorithms. It is noted that performance of TLBO and SFLA is superior to TS irrespective of size and complexity of the problem. The performance of TLBO and SFLA is almost equal for low, moderate, and high complexity problems while SFLA outperforms TLBO for large sized problems. Hybrid TLBO-TS, however, outperforms the rest of the three algorithms on all four types of problems. It may be because of the improved exploration and exploitation capabilities of the algorithm due to hybridization.

In addition to the solution quality, the convergence of the algorithms has been considered for comparison. Figure 3.10 shows the typical convergence curves for one run of TS, TLBO, SFLA and hybrid TLBO-TS for solving an instance selected randomly out of the problems instances described earlier. It can be seen that hybrid TLBO-TS converges faster than the other three algorithms. Strong local searchability of the TS helps the hybrid algorithm to reach the best result faster. Thus, the proposed hybrid TLBO-TS algorithm not only produces better quality solutions but also does it faster.

Table 3.10: Average of Percentage Deviations for different sets of problems

Factor	Instance	Algorithms			
		TS	TLBO	SFLA	Hybrid TLBO-TS
Average Percentage Deviation	Low	14.6642	1.9339	1.7006	0.0057
	Moderate	9.5585	3.5724	3.7648	0.3258
	High	11.9557	4.2712	4.1907	0.3605
	Large	6.8698	2.7901	1.9173	0.0692

3.5 Conclusions

This chapter considers the problem of simultaneous selection and scheduling of interdependent projects termed as project portfolio selection and scheduling problem (PPSSP). A zero-one integer linear programming model for maximizing the total expected benefit of the portfolio has been formulated for the problem. Two types of technical interdependencies between the projects viz. mutual exclusiveness and complementariness have been considered. The complementariness (technical) has been introduced for the first time in the PPSSP to enforce the selection and rejection of a set of complementary (technical) projects together.

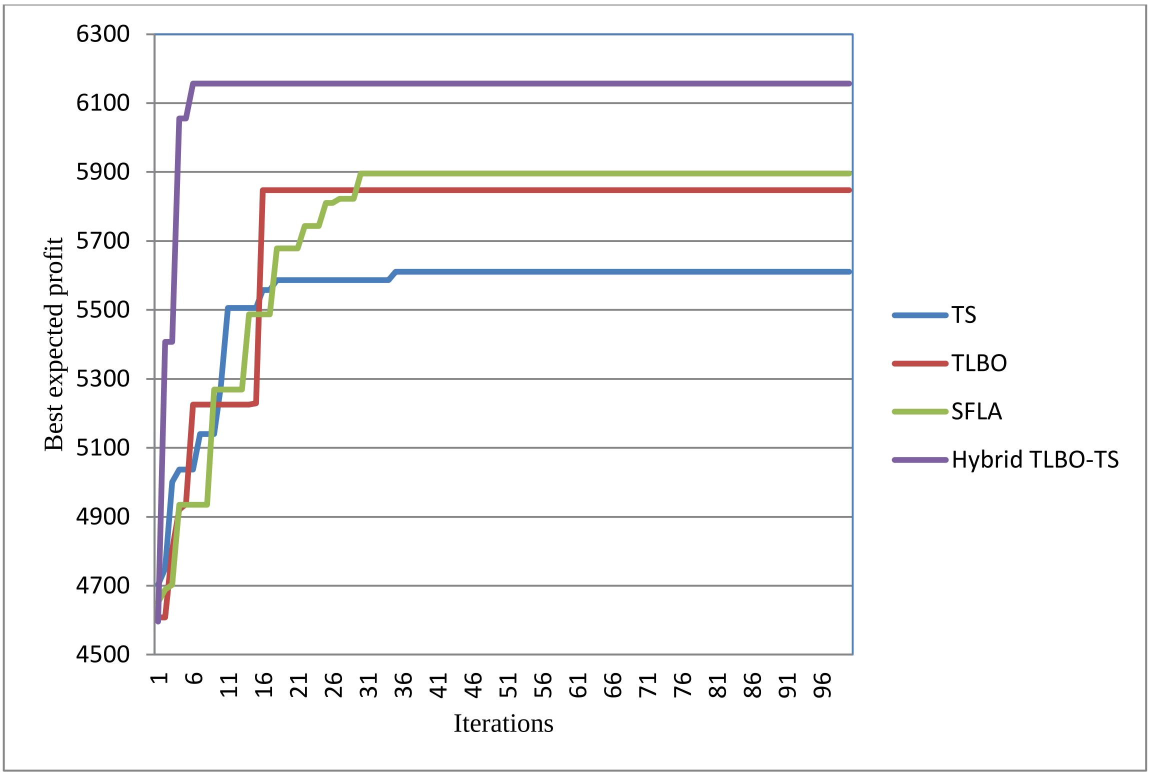


Figure 3.10: Convergence curves

Three meta-heuristics have been proposed for the solution of PPSSP viz. TLBO, TS and hybrid TLBO-TS. The optimal levels of parameters for the TLBO and hybrid TLBO-TS have been determined using Taguchi’s orthogonal array method. The three proposed algorithms are compared with each other and also with SFLA reported in the literature. TLBO and SFLA are found to perform better than TS for all types of problem instances. The performance of TLBO and SFLA is almost equal for low, moderate, and high complexity problems while SFLA outperforms TLBO for large sized problems. It is, however, noted that the hybrid TLBO-TS outperforms rest of the three algorithms. Further, it can be seen that hybrid TLBO-TS converges faster than the other three algorithms. Thus, the results obtained from the proposed hybrid TLBO-TS algorithm are found promising in terms of solution quality and convergence.

This model may be useful to project managers in the simultaneous selection of projects in the portfolio and scheduling when projects are technically interdependent in terms of exclusiveness and complementariness. Maximization of the total expected benefit has been considered as the objective function in this research. Other objectives such as maximization of the probability of success of the portfolio, minimization of resource usages variation may be considered in future research. Further, schedules of the projects are considered to be predetermined in this chapter, activity scheduling may be considered as an integral part of the process and suitable algorithms may be developed.

This research is further extended in chapter 4 by modelling the problem in a multi-objective environment with consideration of reinvestment of benefits from the completed projects in the portfolio.

Chapter-4

Multi-objective Project Portfolio Selection and Scheduling using Reinvestment Strategy

Chapter 4
Multi-objective Project Portfolio Selection and Scheduling using Reinvestment Strategy

4.1 Introduction

In the previous chapter, single-objective PPSSP was considered with benefit maximization as objective. The single objective problems hardly persist. One of the important objectives is optimal use of limited resources. In real life situations, underutilization of cost-intensive resources such as experts, machines etc. affect the expected benefit from the project. From literature analysis done in chapter 2, it is clear that minimization of the resource usage variation is the most promising method to handle the underutilization of cost-intensive resources. Further, the benefit from a completed project can be reinvested to add more projects for the higher expected benefit. This strategy provides an opportunity for the projects which otherwise do not get selected due to limited budget even when renewable resources are available. Besides increasing the benefit, this strategy also increases the utilization of renewable resources.

In this chapter, the problem of PPSSP is considered with two objectives: maximizing the total expected benefit and minimizing the resource usage variation. Benefit from a project is considered to be time sensitive and added to the budget for consideration of more projects. Two types of technical interdependencies: mutual exclusiveness and complementariness are considered. The problem is formulated as multi-objective MILP and a modified Non-dominated Sorting Teaching Learning Based Optimization (NSTLBO) algorithm has been proposed to solve the problem. The algorithm is hybridized with Tabu Search (TS) algorithm and Hybrid NSTLBO is also proposed. A grey-based Taguchi method is employed to tune the parameters of the algorithms. The proposed algorithms are compared with three well-known meta-heuristics, NSGA II, MOSS and SFLA developed for the problem by solving 96 randomly generated problems of different size and complexity.

The rest of the chapter is structured as follows. In Section 4.2 the problem is formally defined and a mathematical formulation along with an illustrative example is presented. Section 4.3 describes the proposed algorithms. Section 4.4 presents the computational experiences. Finally, the summary and conclusion of the chapter is presented in Section 4.5.

4.2 Problem statement and mathematical formulation

In this section, a mixed integer linear programming model (MILP) is presented for PPSSP with reinvestment strategy. There is a set of candidate projects with the predefined initial budget. The benefit from the projects, however, is reinvested facilitating selection of more projects (Belenky, 2012). Two types of technical interdependencies among the projects have been considered viz. mutual exclusiveness and complementariness. Let *K* types of renewable resources are needed for executing the selected projects. The model aims to select a subset of projects from a set of *N* candidate projects and simultaneously schedule them over the fixed planning horizon of *T* periods satisfying the resource availability and project interdependency constraints. Two conflicting objectives are considered. The first objective is to maximize the total expected benefit from the selected projects where the benefit from a project is considered to be time sensitive. The second objective is to minimize the sum of resource usage variation determined between two consecutive periods (Tofighian & Naderi, 2015).

4.2.1 Mathematical Model:

The indexes, parameters and variables for the proposed mathematical model are as below:

Indexes:

i = indexes for projects; (i = 1,2,3 ……. , N)

k = resources index; (k = 1,2,3 ……. , K)

t = time period indexes; (t = 1,2,3 ……. , T)

Parameters:

P_{it} = expected profit if project i starts in period t.

C_{it} = Budget requirement for project i in period t.

d_i = duration of the project i.

r_{ik} = resource requirement of type k for project i in each time period.

R_{kt} = resource availability of type k in period t.

B₁ = Initial Budget available in the beginning of period *t=1*.

h = project which is complementary (technical) to project *i*. *(h ϵ H*$_i$*)*

e = project mutually exclusive (technical) to project *i*. *(e ϵ E*$_i$*)*

Other variables:

B_t = Budget available in period *t*.

Decision variables:

X_{it} = 1 if project *i* is selected and starts in period *t*, 0 otherwise.

W_{tk} = variation in usages of *k*-th resource during periods *t* and *t*+1 or *t*-1.

Formulation:

Objectives:

Max $\sum_{i=1}^{N}\sum_{t=1}^{T-d_i+1} P_{it} * X_{it}$ (4.1)

Min $\sum_{t=1}^{T}\sum_{k=1}^{K} W_{tk}$ (4.2)

Constraints:

$\sum_{t=1}^{T-d_i+1} X_{it} \leq 1 \qquad \forall\, i$ (4.3)

$\sum_{i=1}^{N}(\sum_{j=\max\{1,t-d_i+1\}}^{\min\{t,T-d_i+1\}} X_{ij}) * r_{ik} \ \leq R_{kt} \qquad \forall\, k,t$ (4.4)

$\sum_{i=1}^{N}(\sum_{j=\max\{1,t-d_i+1\}}^{\min\{t,T-d_i+1\}} X_{ij}) * C_{it} \ \leq B_t \qquad \forall\, t$ (4.5)

$B_t = B_{t-1} - \sum_{i=1}^{N}(\sum_{j=\max\{1,t-d_i\}}^{\min\{t-1,T-d_i+1\}} X_{ij}) * C_{i(t-1)} + \sum_{i=1}^{N} P_{i(t-d_i)} * X_{i(t-d_i)}$

$\forall\, t > 1, t - d_i > 0$ (4.6)

$W_{tk} \geq \sum_{i=1}^{N}(\sum_{j=\max\{1,t-d_i+1\}}^{\min\{t,T-d_i+1\}} X_{ij}) * r_{ik} - \sum_{i=1}^{N}(\sum_{j=\max\{1,t+2-d_i\}}^{\min\{t+1,T-d_i+1\}} X_{ij}) * r_{ik} \qquad \forall\, k, t < T$

(4.7)

$$W_{tk} \geq \sum_{i=1}^{N}(\sum_{j=\max\{1,t-d_i+1\}}^{\min\{t,T-d_i+1\}} X_{ij}) * r_{ik} - \sum_{i=1}^{N}(\sum_{j=\max\{1,t-d_i\}}^{\min\{t-1,T-d_i+1\}} X_{ij}) * r_{ik} \qquad \forall\, k, t > 1 \qquad (4.8)$$

$$\sum_{t=1}^{T-d_i+1} X_{it} + \sum_{t=1}^{T-d_e+1} X_{et} \leq 1 \qquad \forall\, i, e \in E_i \qquad (4.9)$$

$$\sum_{t=1}^{T-d_i+1} X_{it} = \sum_{t=1}^{T-d_h+1} X_{ht} \qquad \forall\, i, h \in H_i \qquad (4.10)$$

$$X_{it} = [0,1] \qquad \forall\, i, t \leq T - d_i + 1 \qquad (4.11)$$

$$W_{tk} \geq 0 \qquad \forall\, t, k \qquad (4.12)$$

The first objective (4.1) maximizes the total expected benefit from the projects selected in the portfolio. The second objective (4.2) minimizes the sum of the resource usage variation. Constraint (4.3) guarantees that each project starts only once if selected. Constraints (4.4) and (4.5) enforce the resource and budget availability constraints during a period. The budget availability is updated by constraint (4.6) in each period. The resource usage variation is calculated by constraints (4.7) and (4.8). Interdependencies viz. mutual exclusiveness (technical) and complementariness (technical) are imposed by constraints (4.9) and (4.10) respectively. Lastly, the constraints (4.11) and (4.12) define the decision variables to be binary.

4.2.2 An illustrative example

In this section, a small numerical example is taken to illustrate the proposed model. Let there be five candidate projects available for selection. Project durations, interdependencies, renewable resource and budget requirements are given in Table 4.1. The planning horizon is considered to be 8 time periods. The initial budget availability is 130 units. The maximum availability of both the renewable resources is capped at 7 units for each period. Time-dependent expected benefits for each of the projects are given in Table 4.2.

A feasible solution to the problem and its schedule without reinvestment benefits are given in Table 4.3. Projects 1, 2 and 5 have been selected and started in time periods 5, 1 and 1 respectively. Projects 1 and 2 have been selected together as complementariness (technical) relationship exists between them. Project 3 has not been included in the portfolio due to mutual exclusiveness (technical) relationship with project 5. The value of the first objective, i.e. total expected benefit is 264 (74+92+0+0+98) units and value of the second objective, i.e. cumulative resource usage variation is 9 (5+4) units. Verification of resource, budget constraints and calculation of resource usage variation is reported in Table 4.4.

Table 4.1: Project durations, resource requirements and interdependencies

Projects	Duration	Interdependencies		Renewable Resource Requirement		Budget Requirement
		Mutually Exclusive	Complementary	Type 1	Type 2	
1	3	-	2	3	2	10
2	4	-	1	2	2	12
3	2	5	-	2	3	10
4	5	-	-	2	2	11
5	4	3	-	3	2	12

Table 4.2: The expected benefit in each time period

Projects	Time Periods							
	1	2	3	4	5	6	7	8
1	95	82	76	75	74	63	63	55
2	92	83	66	61	58	51	37	32
3	96	64	61	44	30	26	22	19
4	90	80	77	56	50	46	38	31
5	98	79	70	62	34	24	24	22

Table 4.3: Schedule for a feasible solution without reinvestment

<table>
<tr><td></td><td></td><td colspan="8">Time Periods</td></tr>
<tr><td></td><td></td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr>
<tr><td rowspan="5">Projects</td><td>P1</td><td></td><td></td><td></td><td></td><td colspan="3">(3,2,10)</td><td></td></tr>
<tr><td>P2</td><td colspan="4">(2,2,12)</td><td></td><td></td><td></td><td></td></tr>
<tr><td>P3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>P4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>P5</td><td colspan="4">(3,2,12)</td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="10">Legend:- (RRR1, RRR2, BR)
RRR = Renewable Resource Requirement, BR = Budget Requirement</td></tr>
</table>

Table 4.4: Resource constraints verification without reinvestment

Factor	Time Periods							
	1	2	3	4	5	6	7	8
RR1 Usage	5	5	5	5	3	3	3	0
RR2 Usage	4	4	4	4	2	2	2	0
RA1 & RA2	7	7	7	7	7	7	7	7
RUV for RR1– W_{t1}	0	0	0	2	0	0	3	0
RUV for RR2– W_{t2}	0	0	0	2	0	0	2	0
Budget Usage	24	24	24	24	10	10	10	0
Budget Availability in the beginning of a period	130	106	82	58	34	24	14	4

RUV = Resource Usage Variation, RR = Renewable Resource, RA = Resource Availability

In order to show the effectiveness of reinvestment of benefits from the projects, the same problem is solved separately with reinvestment strategy. A feasible solution to the problem and its schedule are given in Table 4.5. Projects 1, 2, 4 and 5 have been selected and started in time periods 1, 4, 4 and 1 respectively. It can be noted that project 4 has also been selected which could not be selected due to lack of budget in the earlier case leading to the higher total expected benefit of 310 (95+61+0+56+98) units from the portfolio with the same level of resource usage variation of 9 units. Verification of resource, budget constraints and calculation of resource usage variation is reported in Table 4.6. From this comparison, it is clear that more projects can be accommodated in the portfolio by reinvesting benefit from the completed projects ultimately leading to a higher total expected benefit.

Table 4.5: Schedule for a feasible solution with reinvestment

<table>
<tr><td></td><td></td><td colspan="8">Time Periods</td></tr>
<tr><td></td><td></td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr>
<tr><td rowspan="5">Projects</td><td>P1</td><td colspan="3">(3,2,10)</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>P2</td><td></td><td></td><td></td><td colspan="4">(2,2,12)</td><td></td></tr>
<tr><td>P3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>P4</td><td></td><td></td><td></td><td colspan="5">(2,2,11)</td></tr>
<tr><td>P5</td><td colspan="4">(3,2,12)</td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="10">Legend:- (RRR1, RRR2, BR)
RRR = Renewable Resource Requirement, BR = Budget Requirement</td></tr>
</table>

Table 4.6: Resource constraints verification with reinvestment

Factor	Time Periods							
	1	2	3	4	5	6	7	8
RR1 Usage	6	6	6	7	4	4	4	2
RR2 Usage	4	4	4	6	4	4	4	2
RA1 & RA2	7	7	7	7	7	7	7	7
RUV for RR1– W_{t1}	0	0	0	3	0	0	2	0
RUV for RR2– W_{t2}	0	0	0	2	0	0	2	0
Budget Usage	22	22	22	35	23	23	23	11
Budget Availability in the beginning of a period	130	108	86	159	222	199	176	214
Benefit Reinvested				95 ↑	98 ↑			61 ↑
RUV = Resource Usage Variation, RR = Renewable Resource, RA = Resource Availability								

4.3 Proposed algorithms for PPSSP

This section describes the proposed Non-dominated Sorting TLBO (NSTLBO) algorithm employed to solve the problem under study. Further, the algorithm is hybridized with Tabu Search (TS) algorithm to enhance the intensification and diversification of the search.

4.3.1 Non-dominated Sorting TLBO (NSTLBO) algorithm

TLBO algorithm, developed by Rao, Savsani, & Vakharia (2011, 2012), is a population-based meta-heuristic algorithm which has been successfully used to solve various continuous and discrete optimization problems (Khalghani & Khooban, 2014; Yu et al., 2018; Aich & Banerjee, 2016; Ji et al., 2017; Khare & Kumar, 2016; Zheng et al., 2017; Zou et al., 2015) including project scheduling. This algorithm imitates the conventional classroom teaching-learning process. In the algorithm, a student represents a solution to the problem, and the best solution among the population is recognized as the teacher. The basic operators of the algorithm include teacher phase, student phase and self-study phase. Rao (2016) applied the concept of non-dominated sorting (NS) and crowding distance (CD), proposed by Deb et al. (2002), and developed NSTLBO for multi-objective optimization problems. The NS approach ensures that good solutions are selected, and population advances towards the Pareto front. The CD concept ensures diversity by selecting the teacher from the scattered area of the search space and hence prevents the algorithm from being trapped into local optima. This is a posteriori approach of decision making.

The NSTLBO algorithm starts with an initial population of randomly generated students (solutions). The population is then ranked using NS to identify the student with the highest rank (rank = 1) as a teacher. In the case of a tie, the student with the highest value of CD is considered as a teacher so that diversity is maintained in the search process. After this, the teacher, student and self-study phases are applied. The basic phases of the proposed NSTLBO are adapted from the TLBO which has been developed in chapter 3 for single objective PPSSP. The pseudo code for the proposed NSTLBO is given in Figure 4.1. The teacher phase, student phase and self-study phases are described after introducing encoding scheme and population initialization process next in this section.

```
1. Initialization: Randomly generate a population (students) of size N, termination
   criterion
2. Evaluation and Ranking: Sort and rank the students using NS and CD computation
Repeat
3. Teacher Phase: Identify teacher (rank=1) and apply teacher phase crossover to obtain
   N new students.
       3.1 Mix new students with previous population to get 2N students.
       3.2 Sort and rank the students using NS and CD.
       3.3 Select first N students to be the population for next phase.
4. Student Phase: Apply student phase crossover to obtain N new students.
       4.1 Combine new students with the students obtained after teacher phase to get 2N
           students.
       4.2 Sort and rank the students using NS and CD.
       4.3 Select first N students to be the population for next phase.
5. Self-study Phase: Apply student phase mutation crossover to obtain N new students.
       5.1 Combine new students with the students obtained after student phase to get 2N
           students.
       5.2 Sort and rank the students using NS and CD.
       5.3 Select first N students to be the population for teacher phase in next iteration.
Until a termination criterion is satisfied.
       Report the non-dominated set of solutions (students).
```

Figure 4.1: Pseudo code for NSTLBO algorithm for PPSSP

4.3.1.1 *Encoding scheme*

An encoding scheme is required to fit the solution to the meta-heuristic algorithm. The encoding scheme for PPSSP, introduced by Ghorbani & Rabbani (2009), is used for the proposed algorithm. In this scheme, a *P* x *T* matrix is used to represent a solution where *P* is the number of candidate projects, and *T* is number of time periods. Each row represents a candidate project, and each column represents a time period. The element a_{it} of a feasible solution will be 1 if project *i* is selected and started at period *t* otherwise 0. The encoding scheme for a problem with 5 projects and 8 time periods is presented in Table 4.7. In the solution, the value 1, appearing at (1, 1), (2, 3) and (5, 3) show that projects 1, 2 and 5 are selected and start at times 1, 3 and 3 respectively.

Table 4.7: Encoding scheme for a feasible solution

Projects								
P1	1	0	0	0	0	0	0	0
P2	0	0	1	0	0	0	0	0
P3	0	0	0	0	0	0	0	0
P4	0	0	0	0	0	0	0	0
P5	0	0	1	0	0	0	0	0
	1	2	3	4	5	6	7	8

Time Periods

4.3.1.2 *Population initialization*

An initial feasible population of N students (initial feasible solutions) is generated randomly. Population initialization scheme proposed in chapter 3 is used to generate the population. Generation of a student starts with the selection of a project randomly and scheduled to start at the earliest possible time subjected to resource and budget availability. Projects mutually exclusive to this project are excluded from the set of candidate projects, and its complementary projects are scheduled one by one in random order. Once this project is scheduled with all its complementary projects, the other projects are selected randomly following the same procedure till no more scheduling is possible. The set of candidate projects and budget availability are kept updated at each step. Figure 4.2 describes the pseudo code for the population initialization process.

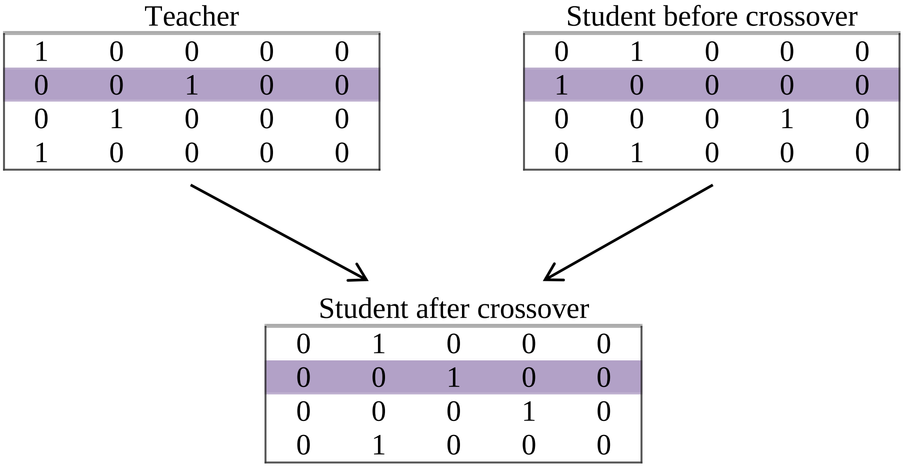
```
Population initialization for TLBO
Let     $A$ = set of candidate projects; $i = \{1,2, \dots\dots.. N\}$
        $C_i$ = set of projects complementary to project $i$;
        $E_i$ = set of projects mutually exclusive to project $i$;
        $S$ = set of selected projects, and
        $D$ = set of projects which are not feasible to be scheduled.
Initialization: $A = \{i = 1,2, \dots\dots N\}; S = D = \emptyset,$
Start
Schedule project $i = \{1,2, \dots\dots. N\}$ (selected randomly)
        If project $i$ is scheduled then
                If $C_i \neq \emptyset$
                        If all the projects in $C_i$ are scheduled
                                        $S = S \cup \{i \cup C_i\}$
                                        $A = A \backslash \{i \cup C_i \cup E_i\}$
                        Else    $D = D \cup \{i \cup C_i\}$
                                $A = A \backslash \{i \cup C_i\}$
                Else    $S = S \cup \{i\}$
                        $A = A \backslash \{i \cup E_i\}$
        Else    $D = D \cup \{i \cup C_i\}$
                $A = A \backslash \{i \cup C_i\}$
Continue till $A = \emptyset$

End
```

Figure 4.2: Pseudo code for population initialization

4.3.1.3 Teacher phase – crossover operator

In this phase of NSTLBO, the non-dominated sorting (NS) and crowding distance (CD) mechanism are applied to the initial population to identify the teacher. The student with the highest rank (rank = 1) is assigned as a teacher. In case of more than one student having the same rank, the student with the larger value of CD is assigned as a teacher. Once a teacher is assigned the crossover operator, designed for teacher phase (teacher-student interaction), is applied to improve the performance of rest of the students. For this, a number of rows corresponding to selected projects from the teacher are transferred to the corresponding rows of each student preserving the resource viability. The number of rows to be transferred varies with the size of the problem (number of projects) and is computed as a predefined fraction of the size of the problem. This predefined fraction is termed as teacher-student crossover coefficient (TSC). The new population obtained after teacher phase is mixed with the original population before teacher phase and sorted and ranked using NS and CD mechanism. First N students, based on the new ranking and CD, constitute the input for student phase. The mechanism of teacher phase for teacher-student crossover is shown in Figure 4.3.



Teacher

1	0	0	0	0
0	0	1	0	0
0	1	0	0	0
1	0	0	0	0

Student before crossover

0	1	0	0	0
1	0	0	0	0
0	0	0	1	0
0	1	0	0	0

Student after crossover

0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	1	0	0	0

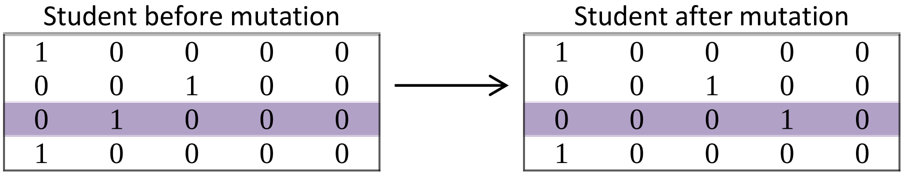
Figure 4.3: Crossover in teacher phase

4.3.1.4 Student phase – crossover operator

In this phase of NSTLBO, each student enhances its performance through interaction with another student. A student Y is chosen randomly for interaction with student X. A predetermined number of rows from the student Y are transferred to corresponding rows of student X, satisfying the resource availability. The number of rows to be transferred is decided in a similar way as done in case of teacher phase. The predefined fraction for this phase is termed as student-student crossover coefficient (SSC). Students obtained after student phase are merged with the old ones and sorted and ranked using NS and CD mechanism. First N students constitute the input for the self-study phase. The procedure for student phase is the same as shown in Figure 4.3.

4.3.1.5 Self-study phase – mutation operator

In this phase of NSTLBO, a student improves its performance by self-learning. The start time of the project is changed randomly in a few rows of the student selected randomly. The number of rows selected is computed as a predefined fraction of the size of the problem. This predefined fraction for this phase is termed as self-study mutation coefficient (SSM). The new population is merged with the old population, and first N students are considered after applying NS and CD mechanism. This new population is again fed to the teacher phase in the next iteration. This operator enhances the exploration capability of the algorithm. The self-study phase is explained in Figure 4.4. The algorithm is run for a predetermined number of iterations.



Student before mutation

1	0	0	0	0
0	0	1	0	0
0	1	0	0	0
1	0	0	0	0

Student after mutation

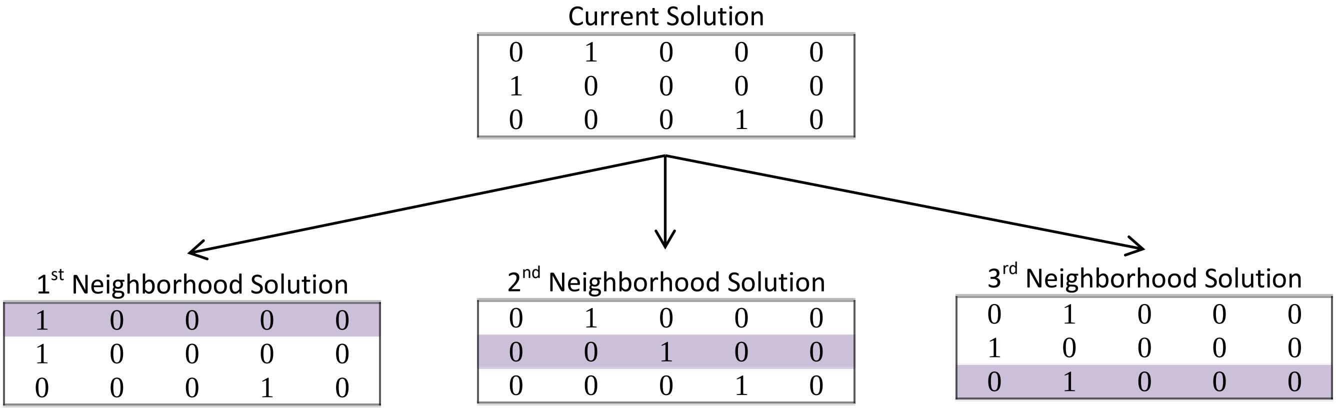
1	0	0	0	0
0	0	1	0	0
0	0	0	1	0
1	0	0	0	0

Figure 4.4: Mutation in self-study phase

4.3.2 Hybrid NSTLBO algorithm

Being a population-based meta-heuristic, TLBO is very good at exploration but is considered to be less effective in exploitation. To boost the local search (exploitation) and efficiency of the NSTLBO algorithm, it is combined with Tabu Search (TS), and a Hybrid NSTLBO algorithm is developed. TLBO is employed to find the region of the optimum while TS is utilized to exploit the region of the optimum locally. The use of TS for extensive intensification of search space enhances the superiority of hybrid meta-heuristics over basic meta-heuristics (Wu et al., 2015; Lin et al., 2016). TS is a very powerful local search algorithm with the ability to avoid trapping into local optima.

In the Hybrid NSTLBO algorithm, the TS algorithm is applied to the teacher obtained after self-study phase to improve the performance of the teacher. To get a neighbourhood solution, the start time of one of the projects is changed randomly satisfying the resource and time constraints. This gives a neighbourhood equal to the size of the problem, and each solution lies in the vicinity of the current solution. The neighbourhood solutions of the teacher for TS are generated using the scheme illustrated in Figure 4.5. The size of the tabu list is taken as the predetermined fraction of the size of the problem. The NS ranking and CD mechanism are applied to identify the best solution to be the teacher for next iteration in teacher phase. The algorithm is run for a predetermined number of iterations. The pseudo code for the Hybrid NSTLBO is given in Figure 4.6.



Current Solution

0	1	0	0	0
1	0	0	0	0
0	0	0	1	0

1st Neighborhood Solution

1	0	0	0	0
1	0	0	0	0
0	0	0	1	0

2nd Neighborhood Solution

0	1	0	0	0
0	0	1	0	0
0	0	0	1	0

3rd Neighborhood Solution

0	1	0	0	0
1	0	0	0	0
0	1	0	0	0

Figure 4.5: Creating neighborhood solutions for TS

4.4 Computational experiences

In this section, the performance of the proposed algorithms is evaluated by comparing it with NSGA-II and MOSS developed by Ghorbani & Rabbani (2009) and SFLA developed by Amirian & Sahraeian (2017) for PPSSP. Algorithms are tested on four different sets of randomly generated test problems of varying size and complexity. All algorithms are coded in MATLAB 7.12 environment and implemented using a laptop computer with Core i3, and Windows 10 using 4 GB of RAM. The test problems generation, assumptions and parameter setting of algorithms, comparison metrics and results obtained are described next.

4.4.1 Test problems

The proposed algorithms, NSGA-II, MOSS and SFLA, are evaluated on four sets of test problems, each containing 24 problems, with different complexity. The test instance generation scheme used in chapter 3 is further modified to contain the added information for complementariness, budget requirement per period and initial budget. Large sized test problems are generated with the same scheme as used for high complexity problems but with more candidate projects. The problems are designed to differ in complexity by increasing amount of interdependencies, project durations, number of resources required and decreasing the make-span, initial budget availability and availability of resources. The expected benefit for each project is generated randomly using uniform distribution U (50,150) in linearly decreasing order. The generation schemes for low, moderate, high and large sized problems are given in Table 4.8.

```
1. Initialization: Randomly generate a population (students) of size N, termination
   criterion
2. Evaluation and Ranking: Sort and rank the students using NS and CD computation

Repeat

3. Teacher Phase: Identify teacher (rank=1) and apply teacher phase crossover to obtain
   N new students.
      3.1 Mix new students with previous population to get 2N students.
      3.2 Sort and rank the students using NS and CD.
      3.3 Select first N students to be the population for next phase.
4. Student Phase: Apply student phase crossover to obtain N new students.
      4.1 Combine new students with the students obtained after teacher phase to get 2N
          students.
      4.2 Sort and rank the students using NS and CD.
      4.3 Select first N students to be the population for next phase.
5. Self-study Phase: Apply student phase mutation crossover to obtain N new students.
      5.1 Combine new students with the students obtained after student phase to get 2N
          students.
      5.2 Sort and rank the students using NS and CD.
      5.3 Select first N students to be the population for next phase.
6. Tabu Search: Identify teacher (rank=1) after self-study phase and apply Tabu Search.
      6.1 Find neighbourhood solutions (students) of teacher.
      6.2 Sort and rank the students using NS and CD.
      6.3 Identify the best student and combine it with the students obtained after self-
          study phase which yields (N+1) students.
      6.4 Again sort and rank the students using NS and CD.
      6.5 Select first N students to be the population for teacher phase in next iteration
Until a termination criterion is satisfied.
      Report the non-dominated set of solutions (students).
```

Figure 4.6: Pseudo code for Hybrid NSTLBO algorithm for PPSSP

4.4.2 Parameter setting

The effectiveness of a meta-heuristic depends on the proper tuning of its parameters. In this section, the parameter settings for the NSTLBO and Hybrid-NSTLBO algorithms developed for the PPSSP has been presented. As the operators of the algorithms are modified to suit the problem under study, the algorithm-specific parameters are introduced to the basic scheme of the algorithm. The grey-based Taguchi method is best suited to find optimal parameter settings for an algorithm for the multi-objective optimization problems thus it has been used in the current work. In this method, a multi-objective problem is transformed into a single-objective problem using Grey Relational Analysis, and then the procedure of the general Taguchi method is followed. For detailed procedure of grey-based Taguchi method, see Kuo, Yang and Huang (2008).

Table 4.8: Test problem generation schemes

Factor	**Generating Rule**			
	Low Complexity Instances	**Moderate Complexity Instances**	**High Complexity Instances**	**Large Size Instances**
Number of available projects	$N = \{5, 6, \dots, 12\}$	$N = \{9, 10, \dots, 16\}$	$N = \{9, 10, \dots, 16\}$	$N = \{17, 18, \dots, 24\}$
Durations	d_i = U(1, 3)	d_i = U(3, 7)	d_i = U(7, 10)	d_i = U(10, 15)
Number of different types of resources	$K = \{1, 2\}$	$K = \{2, 3, 4\}$	$K = \{4, 5\}$	$K = \{4, 5\}$
Required resources for each project per period	r_{ik} = U(10, 15)	r_{ik} = U(10, 15)	r_{ik} = U(10, 15)	r_{ik} = U(10, 15)
Available amount of resource	R_{kt} = 4U(10, 15)	R_{kt} = 3U(10, 15)	R_{kt} = 2U(10, 15)	R_{kt} = 2U(10, 15)
Required budget for each project per period	C_{it} = U(15, 25)	C_{it} = U(15, 25)	C_{it} = U(15, 25)	C_{it} = U(15, 25)
Initial budget available	B_1 = 10U(15, 25)	B_1 = 8U(15, 25)	B_1 = 6U(15, 25)	B_1 = 6U(15, 25)
Make-span	$T = \max\{\max(d_i), \sum_i d_i$ x U(0.8,1)$\}$	$T = \max\{\max(d_i)$, 0.8 x U($\max(d_i), \sum_i d_i)\}$	$T = \max\{\max(d_i)$, 0.5 x U($\max(d_i), \sum_i d_i)\}$	$T = \max\{\max(d_i)$, 0.5 x U($\max(d_i), \sum_i d_i)\}$
Probability of technical interdependency (Mutually exclusive projects)	10 %	20 %	30 %	30 %
Probability of technical interdependency (Complementary projects)	5 %	10 %	15 %	15

4.4.2.1 Parameter setting for NSTLBO algorithm

Parameters for proposed NSTLBO are same as used in chapter 3. It has 4 factors: number of students (N), teacher-student crossover coefficient (TSC), student-student crossover coefficient (SSC) and self-study mutation coefficient (SSM) each at 3 levels (Table 4.9).

Thus, L9 orthogonal array (Table 4.10) is used to design the experiments for grey-based Taguchi method. The distinguishing coefficient (ζ) is an important parameter in the grey-based Taguchi method which may be chosen by the decision maker, and for this study, it is set to 0.5. Considering objective 1 to be more important weights for the objective 1 & 2 are taken as 0.6 & 0.4 respectively. The results obtained for the average grey relational grade are presented in Table 4.11, and the main effects plot is shown in Figure 4.7. Optimum levels obtained from the experimentations are: N = 20, TSC = 0.5, SSC = 0.2 and SSM = 0.2.

Table 4.9: Factors and levels for NSTLBO algorithm

Factor	Symbol	Level	Values		
Number of students (Pop Size)	N	3	10	15	20
Teacher-student crossover coefficient	TSC	3	0.30	0.40	0.50
Student-student crossover coefficient	SSC	3	0.20	0.30	0.40
Self-study mutation coefficient	SSM	3	0.10	0.20	0.30

Table 4.10: The Taguchi orthogonal array L9 for parameter setting for NSTLBO algorithm

Trial No.	N	TSC	SSC	SSM
1	10	0.3	0.2	0.1
2	10	0.4	0.3	0.2
3	10	0.5	0.4	0.3
4	15	0.3	0.3	0.3
5	15	0.4	0.4	0.1
6	15	0.5	0.2	0.2
7	20	0.3	0.4	0.2
8	20	0.4	0.2	0.3
9	20	0.5	0.3	0.1

Table 4.11: Average grey relational grade by factor levels (mean value) for NSTLBO

Factor	Level 1	Level 2	Level 3
N	0.5679	0.5882	0.6537
TSC	0.5037	0.5845	0.7216
SSC	0.6269	0.5846	0.5983
SSM	0.5833	0.6471	0.5795

.

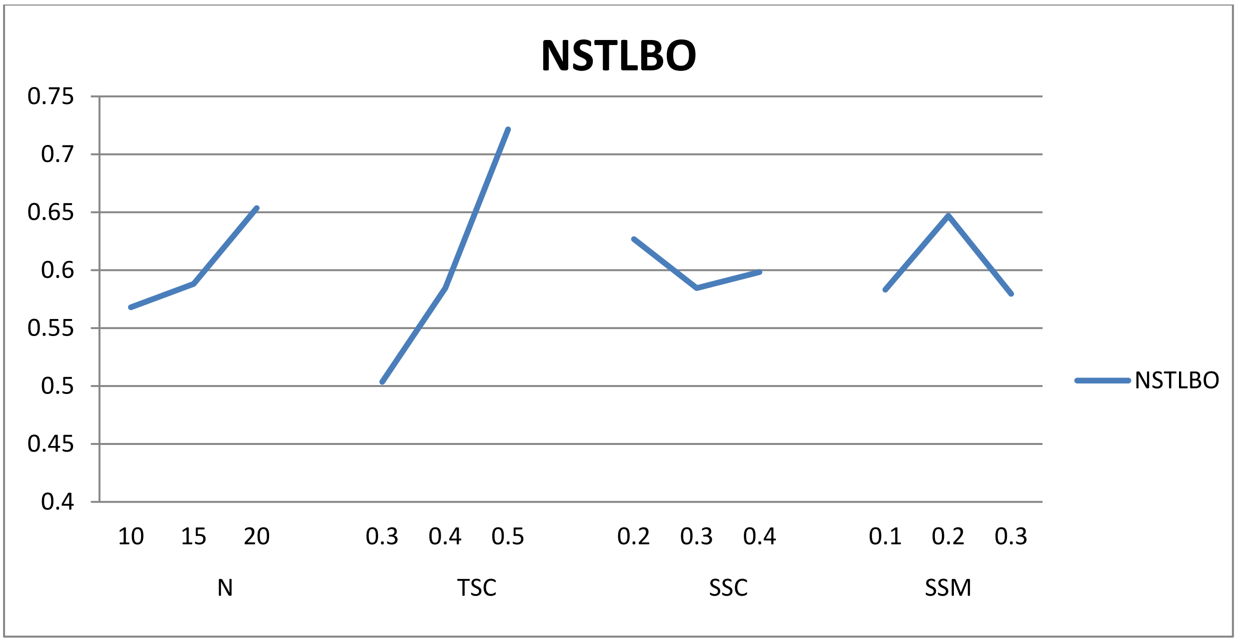


Figure 4.7: Grey-based Taguchi main effects plot (mean value) for NSTLBO

4.4.2.2 Parameter setting for Hybrid NSTLBO algorithm

Hybrid NSTLBO has tabu list (TL) as an extra factor in addition to the factors for NSTLBO each at three levels (Table 4.12), and L18 orthogonal array (Table 4.13) after modification is used for experimentation for grey-based Taguchi method. The distinguishing coefficient (ζ) and weights for the objective 1 & 2 are same as considered for NSTLBO algorithm. The results obtained for the average grey relational grade are presented in Table 4.14, and the main effects plot is shown in Figure 4.8. Optimum levels of parameters are obtained as: N = 20, TSC = 0.5, SSC = 0.2, SSM = 0.3 and TL = 0.4.

Table 4.12: Factors and levels for Hybrid NSTLBO algorithm

Factor	Symbol	Level	Values		
Number of students (Pop Size)	N	3	10	15	20
Teacher-student crossover coefficient	TSC	3	0.30	0.40	0.50
Student-student crossover coefficient	SSC	3	0.20	0.30	0.40
Self-study mutation coefficient	SSM	3	0.10	0.20	0.30
Length of tabu list	TL	3	0.40	0.50	0.60

Table 4.13: The modified Taguchi orthogonal array L18 for Hybrid NSTLBO algorithm

Trial No.	N	TSC	SSC	SSM	TL
1	10	0.3	0.2	0.1	0.4
2	10	0.4	0.3	0.2	0.5
3	10	0.5	0.4	0.3	0.6
4	15	0.3	0.2	0.2	0.5
5	15	0.4	0.3	0.3	0.6
6	15	0.5	0.4	0.1	0.4
7	20	0.3	0.3	0.1	0.6
8	20	0.4	0.4	0.2	0.4
9	20	0.5	0.2	0.3	0.5
10	10	0.3	0.4	0.3	0.5
11	10	0.4	0.2	0.1	0.6
12	10	0.5	0.3	0.2	0.4
13	15	0.3	0.3	0.3	0.4
14	15	0.4	0.4	0.1	0.5
15	15	0.5	0.2	0.2	0.6
16	20	0.3	0.4	0.2	0.6
17	20	0.4	0.2	0.3	0.4
18	20	0.5	0.3	0.1	0.5

Table 4.14: Average grey relational grade by factor levels (mean value) for Hybrid NSTLBO

Factor	Level 1	Level 2	Level 3
N	0.4868	0.4550	0.6115
TSC	0.5110	0.5154	0.5269
SSC	0.5645	0.5071	0.4817
SSM	0.5190	0.4978	0.5365
TL	0.5479	0.5402	0.4652

4.4.3 Comparison metrics

The comparison metrics are used to evaluate the performance of the proposed algorithms relative to other algorithms. In this chapter, two types of metrics are used to measure intensification and diversification of the search. Obtained non-dominated ratio proposed by (Ghorbani and Rabbani, 2009) is used to measure the intensification and spacing metric suggested by (Collette & Siarry, 2013) is used to measure diversification.

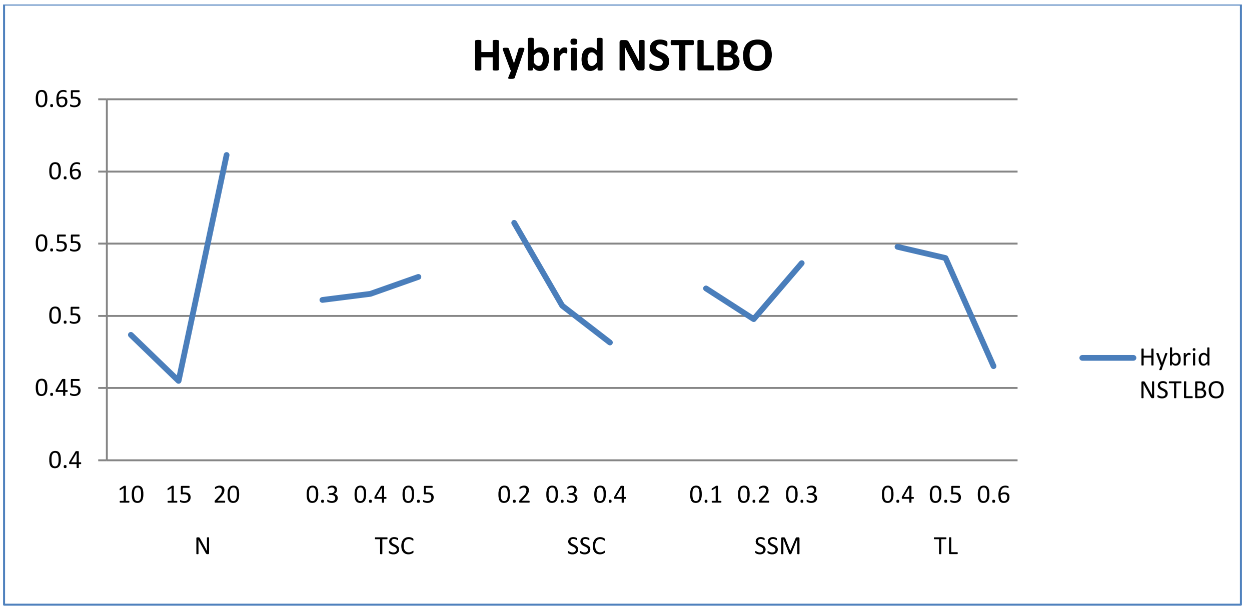


Figure 4.8: Grey-based Taguchi main effects plot (mean value) for Hybrid NSTLBO

4.4.3.1 Obtained non-dominated ratio

Non-dominated sorting (NS) is applied to the set of Pareto front solutions obtained by each of the algorithms considered here. The obtained non-dominated ratio is determined by dividing the number of non-dominated solutions (NDS) by each algorithm participating in the set by the number of solutions in the set. Higher values of the ratio show the greater ability of intensification of search by the considered algorithm. This ratio is considered as it is not sure about the Pareto optimality of the obtained NDS.

4.4.3.2 Spacing metric

This performance metric is used to quantify the uniformity of the spread of NDS obtained in the Pareto front. The formula for the spacing metric used in this chapter is given by equation 4.13 as follows:

$$S = \sqrt{\frac{1}{(n-1)}\sum_{i=1}^{n}\left(\bar{d} - d_i\right)^2} \qquad (4.13)$$

Where n is the number of NDS in the Pareto front. The distance between the obtained NDS (d_i) is calculated using the expression 4.14.

$$d_i = \min_{i,i\neq j} \sum_{m=1}^{k} \left| f_m^i - f_m^j \right|, i, j = 1,2, \dots \dots, n \qquad (4.14)$$

Where f_m^i is the objective function value of the m^{th} objective in the i^{th} solution and k denotes the total number of objectives. The mean of these distances ($\bar{d}$) is calculated using expression 4.15 as follows:

$$\bar{d} = \sum_{i=1}^{n} d_i / n \qquad (4.15)$$

Smaller values of *S* are desirable as smaller the value better the uniformity of the spread.

4.4.4 Comparative results

In this section, the performance of the proposed algorithms is evaluated by comparing the results with NSGA II, MOSS and SFLA. The performance is evaluated on four set of instances, having 24 instances in each set, of various complexity generated randomly in this research. Each algorithm is run 10 times on each instance, and the outputs are averaged to get the results for an instance. Further, the results are averaged for each instance set to get the final results.

The average number of NDS obtained in the Pareto front for each instance set is given in Table 4.15, and the average obtained non-dominated solutions ratio is given in Table 4.16. From Table 4.15 and 4.16, it is clear that the proposed Hybrid NSTLBO outperforms other four algorithms for all type of instances. The difference between the performance of the NSTLBO and SFLA is negligible however these two algorithms perform better than MOSS and NSGA II. Also, MOSS is better than NSGA II in performance which confirms the results obtained by Ghorbani & Rabbani (2009).

The results for the spacing metric are presented in Table 4.17. The value of the spacing metric for Hybrid NSTLBO is better than other algorithms for all the sets of instances which show that the solutions obtained by Hybrid NSTLBO are more diverse and uniformly distributed. It can be noted that for low and moderate complexity instances NSTLBO performs better than SFLA, but for high and large complexity instances SFLA performs better than NSTLBO. The difference, however, is almost negligible for moderate and high complexity instances. Moreover, it is important to mention that there is no significant difference between the performance of MOSS and NSGA II except for the low complexity instances where MOSS has better spacing values than NSGA II.

As a whole, it can be concluded that the proposed Hybrid NSTLBO outperforms other algorithms in all the comparison criteria for all type of instances.

Table 4.15: Results for the average number of non-dominated solutions

Factor	Instance	Algorithms				
		Hybrid NSTLBO	NSTLBO	SFLA	MOSS	NSGA II
Average number of non-dominated solutions	Low	4.666666667	4.2666667	4.2	3.075	2.7625
	Moderate	4.779166667	4.35	4.170833	3.320833	2.929167
	High	4.929166667	4.2166667	4.408333	3.4875	3.004167
	Large	4.970833333	4.3041667	4.475	3.516667	3.041667

Table 4.16: Results for the average obtained non-dominated solutions ratios

Factor	Instance	Algorithms				
		Hybrid NSTLBO	NSTLBO	SFLA	MOSS	NSGA II
Average obtained non-dominated solutions ratio	Low	0.279932735	0.226779	0.228491	0.164371	0.100426
	Moderate	0.296637808	0.227993	0.216382	0.153107	0.105881
	High	0.329303538	0.219407	0.206997	0.149154	0.095138
	Large	0.322381994	0.215993	0.221008	0.152161	0.088456

Table 4.17: Results for spacing metric values

Factor	Instance	Algorithms				
		Hybrid NSTLBO	NSTLBO	SFLA	MOSS	NSGA II
Spacing Metric	Low	0.736979167	0.793854	0.854475	0.895867	1.024879
	Moderate	0.680958333	0.801983	0.811233	0.955263	0.934983
	High	0.7258625	0.852375	0.842442	0.947721	0.985067
	Large	0.7606125	0.892308	0.860125	0.962392	0.974163

4.5 Conclusions

In this chapter, the project portfolio selection and scheduling problem (PPSSP) is considered with two objectives. The first objective is to maximize the total expected benefit and the second objective is to minimize the total resource usage variation. The benefit from a project is considered to be time sensitive and is reinvested in the portfolio. Reinvestment makes a selection of large number of projects possible ultimately leading to higher total benefit. Two types of technical interdependencies viz. complementariness and mutual exclusiveness between the projects are also considered to make the model more realistic. A MILP model is proposed for the problem. Since the problem under study is NP-hard, an NSTLBO algorithm is proposed to solve the problem. This meta-heuristic is further improved by hybridization with Tabu Search resulting in a Hybrid NSTLBO algorithm. The grey-based Taguchi method is used to tune the parameters of the algorithm. Proposed algorithms are tested on 96 randomly generated instances with different complexity and compared with well-known meta-heuristics: SFLA, MOSS, and NSGA II existing in the literature.

From the results, it is clear that Hybrid NSTLBO outperforms the other four algorithms (including the three existing and one proposed in this chapter) with respect to all performance criteria for all type of instances. The performance of the NSTLBO and SFLA is comparable as the difference is almost negligible. However, these algorithms perform better than MOSS and NSGA II. The MOSS performs better than NSGA II for obtained non-dominated ratio, but there is no significant difference in the performance for the spacing metric values.

This multi-objective model for the PPSSP presented in this chapter is further extended to consider benefit interdependencies among the projects in chapter 5.

Chapter-5

Multi-objective Project Portfolio Selection and Scheduling with Benefit Interdependencies

Chapter 5

Multi-objective Project Portfolio Selection and Scheduling with Benefit Interdependencies

5.1 Introduction

As discussed in earlier chapters, some projects may alter the outcome of other projects when selected simultaneously in the portfolio. This happens due to the existence of benefit interdependencies which are identified when benefits expected from the projects are non-additive. The benefit interdependencies play a vital role in selection & scheduling of new product development projects (e.g. automobiles and IT gadgets etc.). Such type of projects may have positive or negative synergies. Positive synergy arises when the sum of benefits from a set of projects implemented together is greater than the sum of benefits when implemented individually. Such projects are known as benefit complementary projects. On the other hand, negative synergy arises when the sum of benefits from a set of projects implemented together is less than the sum of benefits when implemented individually. Such projects are recognized as competitive projects. Thus, positive and negative synergies should be considered during the integrated selection and scheduling of projects to optimize the total expected benefit.

In this chapter, the problem of the multi-objective PPSSP presented in the previous chapter is extended to consider benefit interdependencies. The problem is formulated as multi-objective MILP considering two objectives: maximization of total expected benefit and minimization of resource usage variation. The total expected benefit consists of the benefits from the individual projects and the synergic benefit/loss due to benefit interdependencies. Benefit from a project is considered to be time sensitive and added to the budget for consideration of more projects. Technical interdependencies: mutual exclusiveness and complementariness are also considered. An improved NSTLBO (I-NSTLBO) algorithm is developed for the problem. The teacher phase of the I-NSTLBO algorithm is modified in a probabilistic way using the learning experience of other students to teach some of the students. The rest of the algorithm remains the same as proposed in chapter 4. The proposed I-NSTLBO algorithm is further hybridized with Tabu Search (TS) in order to improve the exploitation ability of the algorithm. A grey-based Taguchi method is used to optimize the parameters of the algorithms. Proposed algorithms are compared with three well-known meta-heuristics, NSGA

II, MOSS and SFLA developed for the problem by solving 96 randomly generated problems of different size and complexity.

The rest of the chapter is structured as follows. In Section 5.2 the problem is formally defined and a mathematical formulation along with an illustrative example is presented. Section 5.3 describes the proposed algorithms. Section 5.4 presents the computational experiences of the study. Finally, the summary and conclusion of the chapter is presented in Section 5.5.

5.2 Problem statement and mathematical formulation

In this section, a mixed integer linear programming model (MILP) is presented for PPSSP with reinvestment strategy. There is a set of candidate projects with the predefined initial budget. The benefit from the projects, however, is reinvested facilitating selection of more projects (Belenky, 2012). Technical and benefit interdependencies among the projects have been considered. Let *K* types of renewable resources are needed for executing the selected projects. The model aims to select a subset of projects from a set of *N* candidate projects and simultaneously schedule them over the fixed planning horizon of *T* periods satisfying the resource availability and project interdependency constraints. Two conflicting objectives are considered. The first objective is to maximize the total expected benefit from the selected projects which consists of two parts. The first part comes from the individual contributions of the projects while the second part comes from the additional benefits/losses generated due to the existence of benefit interdependencies. Further, the benefit from a project is considered to be time sensitive. The second objective is to minimize the sum of resource usage variation determined between two consecutive periods (Tofighian & Naderi, 2015).

5.2.1 Mathematical Model:

The indexes, parameters and variables for the proposed mathematical model are as below:

Indexes:

i = indexes for projects; $(i = 1,2,3 \ldots\ldots, N)$

k = resources index; $(k = 1,2,3 \ldots\ldots, K)$

t = time period indexes; $(t = 1,2,3 \ldots\ldots, T)$

Parameters:

P_{it} = expected profit if project *i* starts in period *t*.

C_{it} = Budget requirement for project *i* in period *t*.

d_i = duration of the project *i*.

r_{ik} = resource requirement of type *k* for project *i* in each time period.

R_{kt} = resource availability of type *k* in period *t*.

B_1 = Initial budget available in the beginning of period *t=1*.

h = project which is complementary (technical interdependency) to project *i*. $(h \epsilon H_i)$

e = project mutually exclusive (technical interdependency) to project *i*. $(e \epsilon E_i)$

v = project which is complementary (benefit interdependency) to project *i*. $(v \epsilon V_i)$

g_{iv} = gain coefficient for projects *i* and *v* having benefit complementariness.

u = project which is competitive (benefit interdependency) to project *i*. $(u \epsilon U_i)$

l_{iu} = loss coefficient for projects *i* and *u* having benefit competitiveness.

Other variables:

B_t = Budget available in period *t*.

Decision variables:

X_{it} = 1 if project *i* is selected and starts in period *t*, 0 otherwise.

W_{tk} = variation in usages of *k*-th resource during periods *t* and *t*+1 or *t*-1.

Y_i = 1 if project *i* and at least one of its complementary (benefit) projects is selected, and 0 otherwise.

Z_i = 1 if project i and at least one of its competitive (benefit) projects is selected, and 0 otherwise.

Formulation:

Objectives:

Max $\quad \sum_{i=1}^{N}\sum_{t=1}^{T-d_i+1} P_{it} * X_{it} + \ (S_B^{+} - S_B^{-})$ (5.1)

Min $\quad \sum_{t=1}^{T}\sum_{k=1}^{K} W_{tk}$ (5.2)

Constraints:

$$\sum_{t=1}^{T-d_i+1} X_{it} \le 1 \qquad \forall\, i \qquad (5.3)$$

$$\sum_{i=1}^{N}\left(\sum_{j=\max\{1,t-d_i+1\}}^{\min\{t,T-d_i+1\}} X_{ij}\right) * r_{ik} \ \le R_{kt} \qquad \forall\, k,t \qquad (5.4)$$

$$\sum_{i=1}^{N}\left(\sum_{j=\max\{1,t-d_i+1\}}^{\min\{t,T-d_i+1\}} X_{ij}\right) * C_{it} \ \le B_t \qquad \forall\, t \qquad (5.5)$$

$$B_t = B_{t-1} - \sum_{i=1}^{N}\left(\sum_{j=\max\{1,t-d_i\}}^{\min\{t-1,T-d_i+1\}} X_{ij}\right) * C_{i(t-1)} + \ \sum_{i=1}^{N} P_{i(t-d_i)} * X_{i(t-d_i)}$$
$$\forall\, t > 1, t - d_i > 0 \qquad (5.6)$$

$$W_{tk} \ge \sum_{i=1}^{N}\left(\sum_{j=\max\{1,t-d_i+1\}}^{\min\{t,T-d_i+1\}} X_{ij}\right) * r_{ik} - \sum_{i=1}^{N}\left(\sum_{j=\max\{1,t+2-d_i\}}^{\min\{t+1,T-d_i+1\}} X_{ij}\right) * r_{ik} \qquad \forall\, k, t < T$$
(5.7)

$$W_{tk} \ge \sum_{i=1}^{N}\left(\sum_{j=\max\{1,t-d_i+1\}}^{\min\{t,T-d_i+1\}} X_{ij}\right) * r_{ik} - \sum_{i=1}^{N}\left(\sum_{j=\max\{1,t-d_i\}}^{\min\{t-1,T-d_i+1\}} X_{ij}\right) * r_{ik} \qquad \forall\, k, t > 1$$
(5.8)

$$S_B^{+} = \sum_{i=1}^{N}\sum_{t=1}^{(T-d_i+1)}\sum_{v=1}^{N}\sum_{t'=1}^{(T-d_v+1)} g_{iv}(P_{it}X_{it} + \sum_{v=1}^{N} P_{vt}X_{vt})\, Y_i \qquad i \ne v, i < v, v \in V$$
(5.9)

$$S_B^{-} = \sum_{i=1}^{N}\sum_{t=1}^{(T-d_i+1)}\sum_{u=1}^{N}\sum_{t'=1}^{(T-d_u+1)} l_{iu}(P_{it}X_{it} + \sum_{u=1}^{N} P_{ut}X_{ut})\, Z_i \qquad i \ne u, i < u, u \in$$
U (5.10)

$$\sum_{t=1}^{T-d_i+1} X_{it} + \sum_{t=1}^{T-d_e+1} X_{et} \le 1 \qquad \forall\, i, e \in E_i \qquad (5.11)$$

$$\sum_{t=1}^{T-d_i+1} X_{it} = \sum_{t=1}^{T-d_h+1} X_{ht} \qquad \forall\, i, h \in H_i \qquad (5.12)$$

$$X_{it} = [0,1] \qquad \forall\, i, t \leq T - d_i + 1 \qquad (5.13)$$

$$W_{tk} \geq 0 \qquad \forall\, t, k \qquad (5.14)$$

$$Y_i = [0,1] \qquad \forall\, i \qquad (5.15)$$

$$Z_i = [0,1] \qquad \forall\, i \qquad (5.16)$$

The first objective (5.1) maximizes the total expected benefit from the selected projects. The first term is the sum of the benefits from each of the selected projects when implemented individually. The second term is the extra benefit due to the selection of interdependent projects. The second objective (5.2) minimizes the sum of the resource usage variation between each pair of two consecutive time periods for all the renewable resources. Constraint (5.3) determines the start time of the selected project and ensures that it starts only once. The resource limitations are imposed by constraint (5.4). Constraint (5.5) ensures that budget requirement per period does not violate the budget limitations. The budget availability is updated by constraint (5.6) in each period. The resource usage variation is calculated by constraints (5.7) and (5.8). Constraint (5.9) measures the extra benefit due to selection of interdependent projects caused by benefit complementariness. Constraint (5.10) measures the loss due to selection of interdependent projects caused by benefit competitiveness. Technical interdependencies viz. mutual exclusiveness and complementariness are imposed by constraints (5.11) and (5.12) respectively. Constraints (5.13) to (5.16) define the decision variables.

5.2.2 An illustrative example

A small numerical example with eight candidate projects is considered to illustrate the proposed model. Project durations, interdependencies, renewable resource and budget requirements are given in Table 5.1. The planning horizon is considered to be 8 time periods. The initial budget availability is 135 units. The maximum availability of both the renewable resources is capped at 8 units for each period. Both, gain coefficient (g_{iv}) for benefit complementary, and loss coefficient (l_{iu}) for benefit competitive projects are considered equal to 0.25. The gain coefficient is used to measure the synergic benefit if a set of projects have positive synergy between them. Similarly, the loss coefficient is used to measure the synergic loss if a set of projects have negative synergy between them. These are calculated as the fraction of the sum of their individual benefits. Time-dependent expected benefits for each of the projects are given in Table 5.2.

Table 5.1: Project durations, resource requirements and interdependencies

Projects	Duration	Interdependencies				Renewable Resource Requirement		Budget Requirement
		Technical		Benefit				
		Mutually Exclusive	Complementary	Competitive	Complementary	Type 1	Type 2	
1	3	-	2	-	-	3	2	10
2	4	-	1	-	-	2	2	12
3	2	5	-	-	-	2	3	10
4	5	-	-	7	-	2	2	11
5	4	3	-	-	-	3	2	12
6	3	-	-	-	8	2	3	9
7	2	-	-	4	-	1	3	11
8	4	-	-	-	6	2	1	12

Table 5.2: The expected benefit in each time period

Projects	Time Periods							
	1	2	3	4	5	6	7	8
1	95	82	76	75	74	63	63	55
2	92	83	66	61	58	51	37	32
3	96	64	61	44	30	26	22	19
4	90	80	77	56	50	46	38	31
5	98	79	70	62	34	24	24	22
6	95	85	69	58	48	40	39	35
7	95	83	65	62	50	30	26	24
8	91	84	71	64	53	40	27	18

A feasible solution to the problem and its schedule with reinvestment are given in Table 5.3. Except for project 3, all the projects have been selected. Projects 1 and 2 have been selected together as technical complementariness relationship exists between them. Project 3 has not been included in the portfolio due to mutual exclusiveness relationship with project 5. Benefit complementariness relationship exists between project 6 and 8 which has resulted in a positive synergic benefit of 46.5 units. Further, benefit competitiveness relationship exists between project 4 and 7 which has resulted in a negative synergic benefit of 30 units. The resultant synergic benefit is accounted for 16.5 units. The total benefit from the individual contribution of the projects is 506 (74+92+90+34+95+30+91) units. Hence the value of the first objective, i.e. total expected benefit is 522.5 (506+16.5) units and value of the second objective, i.e. cumulative resource usage variation is 18 (8+10) units. Verification of resource, budget constraints and calculation of resource usage variation is reported in Table 5.4.

Table 5.3: Schedule for a feasible solution

		Time Periods							
		1	2	3	4	5	6	7	8
Projects	P1					(3,2,10)			
	P2	(2,2,12)							
	P3								
	P4	(2,2,11)							
	P5					(3,2,12)			
	P6	(2,3,9)							
	P7						(1,3,11)		
	P8	(2,1,12)							
Legend:- (RRR1, RRR2, BR) RRR = Renewable Resource Requirement, BR = Budget Requirement									

5.3 Proposed algorithms for PPSSP

This section presents the improved NSTLBO (I-NSTLBO) algorithms to solve the problem. The algorithm is further hybridized with Tabu Search (TS) algorithm to enhance the search process of the algorithm.

5.3.1 Improved NSTLBO algorithm

The improved NSTLBO (I-NSTLBO) algorithm developed in this section is the extended version of the NSTLBO algorithm proposed in Chapter 4. In TLBO algorithm, only the best student among the population is assigned as a teacher. This shows the greedy nature of the search process which might limit the exploration capability of the algorithm. The diversification can be introduced in the algorithm through the randomization in the teacher assignment of some of the students. The learning experience of other students in the population can be used to improve the performance of the students in the teacher phase.

Table 5.4: Resource constraints verification

Factor	Time Periods							
	1	2	3	4	5	6	7	8
RR1 Usage	8	8	8	6	8	7	7	3
RR2 Usage	8	8	8	5	6	7	7	2
RA1 & RA2	8	8	8	8	8	8	8	8
RUV for RR1– W_{t1}	0	0	2	0	2	0	4	0
RUV for RR2– W_{t2}	0	0	3	0	1	1	5	0
Budget Usage	44	44	44	35	33	33	33	12
Budget Availability in the beginning of each period	135	91	47	98	246	303	270	341
Benefit Reinvested				95 ↑	183 ↑	90 ↑		104 ↑
RUV = Resource Usage Variation, RR = Renewable Resource, RA = Resource Availability								

The proposed I-NSTLBO algorithm starts with an initial population of randomly generated students (solutions). The population is then ranked using NS to identify the student with the highest rank (rank = 1) as a teacher. In the case of a tie, the student with the highest value of CD is considered as a teacher so that diversity is maintained in the search process. After this, the teacher, student and self-study phases are applied. The teacher phase is modified to use the learning experience of the other students in the population to enhance the diversification in the search process (Zou et al., 2015). Two random probabilities are used in the teacher phase to realize the interaction between the teacher and the students. With this randomization some of the students get the chance to be updated based on the learning experience of other students instead of a teacher. The student and self-study phases are adapted from the NSTLBO proposed in chapter 4. The pseudo code for the I-NSTLBO is given in Figure 5.1. The teacher phase, student phase and self-study phases are described after introducing encoding scheme and population initialization process next in this section. The encoding scheme is the same as used in chapter 4.

```
1. Initialization: Randomly generate a population (students) of size N, termination
   criterion
2. Evaluation and Ranking: Sort and rank the students using NS and CD computation
Repeat
3. Teacher Phase: Identify teacher (rank=1) and apply teacher phase crossover to obtain
   N new students.
       3.1 Mix new students with previous population to get 2N students.
       3.2 Sort and rank the students using NS and CD.
       3.3 Select first N students to be the population for next phase.
4. Student Phase: Apply student phase crossover to obtain N new students.
       4.1 Combine new students with the students obtained after teacher phase to get 2N
           students.
       4.2 Sort and rank the students using NS and CD.
       4.3 Select first N students to be the population for next phase.
5. Self-study Phase: Apply student phase mutation crossover to obtain N new students.
       5.1 Combine new students with the students obtained after student phase to get 2N
           students.
       5.2 Sort and rank the students using NS and CD.
       5.3 Select first N students to be the population for teacher phase in next iteration.
Until a termination criterion is satisfied.
       Report the non-dominated set of solutions (students).
```

Figure 5.1: Pseudo code for I-NSTLBO algorithm for PPSSP

5.3.1.1 Encoding scheme

An encoding scheme is required to fit the solution to the meta-heuristic algorithm. The encoding scheme for PPSSP, introduced by Ghorbani & Rabbani (2009), is used for the proposed algorithm. In this scheme, a *P* x *T* matrix is used to represent a solution where *P* is the number of candidate projects, and *T* is number of time periods. Each row represents a candidate project, and each column represents a time period. The element a_{it} of a feasible solution will be 1 if project *i* is selected and started at period *t* otherwise 0. The encoding scheme for a problem with 5 projects and 8 time periods is presented in Table 5.5. In the solution, the value 1, appearing at (1, 1), (2, 3) and (5, 3) show that projects 1, 2 and 5 are selected and start at times 1, 3 and 3 respectively.

Table 5.5: Encoding scheme for a feasible solution

Projects	1	2	3	4	5	6	7	8
P1	1	0	0	0	0	0	0	0
P2	0	0	1	0	0	0	0	0
P3	0	0	0	0	0	0	0	0
P4	0	0	0	0	0	0	0	0
P5	0	0	1	0	0	0	0	0

Time Periods

5.3.1.2 Population initialization

An initial feasible population of N students (initial feasible solutions) is generated randomly. Population initialization scheme proposed in chapter 3 is modified to generate the population. Generation of a student starts with the selection of a project randomly and scheduled to start at the earliest possible time subjected to resource and budget availability. Projects mutually exclusive to this project are excluded from the set of candidate projects, and its complementary (technically interdependent) projects are scheduled one by one in random order. Once this project is scheduled with all its technical complementary projects, its benefit complementary projects are scheduled (if possible) one by one in random order. Similarly, other projects (which are not benefit competitive to any of the selected projects) are selected randomly following the same procedure till no more scheduling is possible. Projects which are not benefit competitive to any of the selected projects may be selected in case of budget and resource availability. The set of candidate projects and budget availability are kept updated at each step.

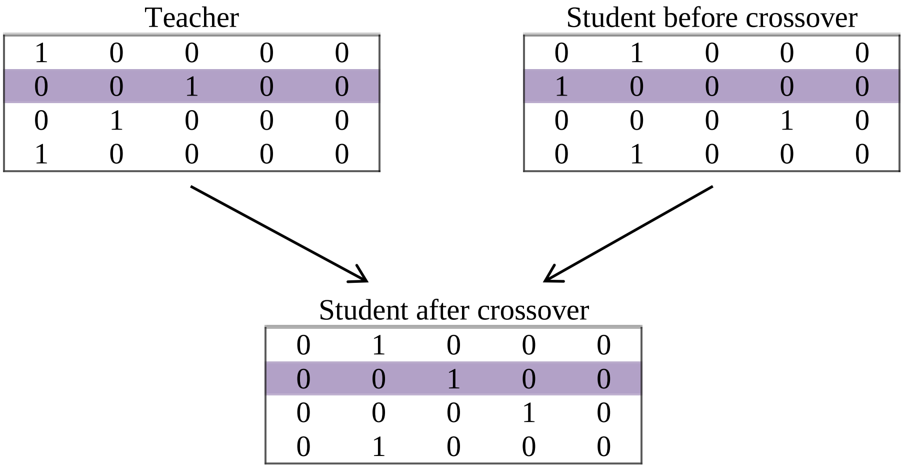
5.3.1.3 Teacher phase – crossover operator

In the teacher phase, the students undergo crossover in a probabilistic way using the learning experience of a teacher or other students. Few students learn from the other students instead of a teacher. Two random probabilities are used to decide that which student will perform crossover with another student.

Firstly, the teacher is identified by applying the NS and CD mechanism. The student with the highest rank (rank = 1) is assigned as a teacher. In case of more than one student having the same rank, the student with the larger value of CD is assigned as a teacher. Once a teacher is assigned the crossover operator, designed for teacher phase (teacher-student interaction), is applied to improve the performance of rest of the students.

For each student, two random numbers (*a*, *b*) are generated between [0, 1]. For i^{th} student, if *a* < *b*, perform the crossover between the teacher and the student. Conversely, if *a* > *b*, select another student *j* from the population randomly which is different from student *i*. If this randomly selected student *j* is better than student *i* then perform crossover with student *j* otherwise perform crossover with the teacher.

For crossover, a number of rows corresponding to selected projects from the teacher or the other student *j* selected randomly for a crossover with student *i* are transferred to the corresponding rows of student *i* preserving the resource viability. The number of rows to be transferred varies with the size of the problem (number of projects) and is computed as a predefined fraction of the size of the problem. This predefined fraction is termed as teacher-student crossover coefficient (TSC). The new population obtained after teacher phase is mixed with the original population before teacher phase and sorted and ranked using NS and CD mechanism. First N students, based on the new ranking and CD, constitute the input for student phase. The mechanism of teacher phase for teacher-student crossover is shown in Figure 5.2.



Teacher

1	0	0	0	0
0	0	1	0	0
0	1	0	0	0
1	0	0	0	0

Student before crossover

0	1	0	0	0
1	0	0	0	0
0	0	0	1	0
0	1	0	0	0

Student after crossover

0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	1	0	0	0

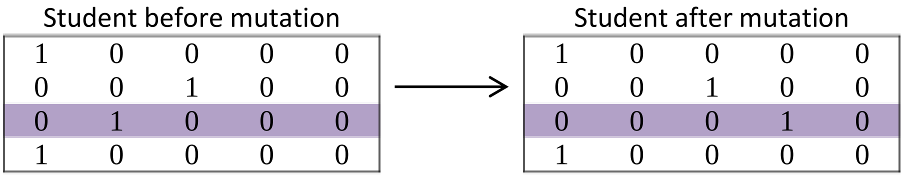
Figure 5.2: Crossover in teacher phase

5.3.1.4 Student phase – crossover operator

In this phase of I-NSTLBO, each student enhances its performance through interaction with another student. A student Y is chosen randomly for interaction with student X. A predetermined number of rows from the student Y are transferred to corresponding rows of student X, satisfying the resource availability. The number of rows to be transferred is decided in a similar way as done in case of teacher phase. The predefined fraction for this phase is termed as student-student crossover coefficient (SSC). Students obtained after student phase are merged with the old ones and sorted and ranked using NS and CD mechanism. First N students constitute the input for the self-study phase. The procedure for student phase is the same as shown in Figure 5.2.

5.3.1.5 Self-study phase – mutation operator

In this phase of I-NSTLBO, a student improves its performance by self-learning. The start time of the project is changed randomly in a few rows of the student selected randomly. The number of rows selected is computed as a predefined fraction of the size of the problem. This predefined fraction for this phase is termed as self-study mutation coefficient (SSM). The new population is merged with the old population, and first N students are considered after applying NS and CD mechanism. This new population is again fed to the teacher phase in the next iteration. This operator enhances the exploration capability of the algorithm. The self-study phase is explained in Figure 5.3. The algorithm is run for a predetermined number of iterations.



Student before mutation

1	0	0	0	0
0	0	1	0	0
0	1	0	0	0
1	0	0	0	0

Student after mutation

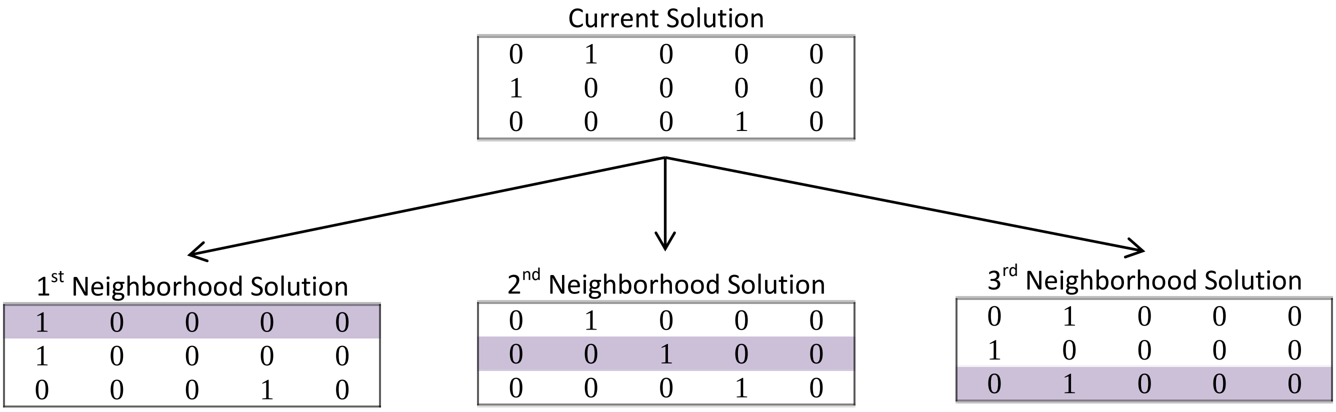
1	0	0	0	0
0	0	1	0	0
0	0	0	1	0
1	0	0	0	0

Figure 5.3: Mutation in self-study phase

5.3.2 Hybrid I-NSTLBO algorithm

Being a population-based meta-heuristic, TLBO is very good at exploration but is considered to be less effective in exploitation. To boost the local search (exploitation) and efficiency of the I-NSTLBO algorithm, it is combined with Tabu Search (TS), and a Hybrid I-NSTLBO algorithm is developed. TLBO is employed to find the region of the optimum while TS is utilized to exploit the region of the optimum locally. The use of TS for extensive intensification of search space enhances the superiority of hybrid meta-heuristics over basic meta-heuristics (Wu et al., 2015; Lin et al., 2016). TS is a very powerful local search algorithm with the ability to avoid trapping into local optima.

In the Hybrid I-NSTLBO algorithm, the TS algorithm is applied to the teacher obtained after self-study phase to improve the performance of the teacher. To get a neighbourhood solution, the start time of one of the projects is changed randomly satisfying the resource and time constraints. This gives a neighbourhood equal to the size of the problem, and each solution lies in the vicinity of the current solution. The neighbourhood solutions of the teacher for TS are generated using the scheme illustrated in Figure 5.4. The size of the tabu list is taken as the predetermined fraction of the size of the problem. The NS and CD mechanism are applied to identify the best solution to be the teacher for the next iteration in teacher phase. The algorithm is run for a predetermined number of iterations. The pseudo code for the Hybrid I-NSTLBO is given in Figure 5.5.



Current Solution

0	1	0	0	0
1	0	0	0	0
0	0	0	1	0

1st Neighborhood Solution

1	0	0	0	0
1	0	0	0	0
0	0	0	1	0

2nd Neighborhood Solution

0	1	0	0	0
0	0	1	0	0
0	0	0	1	0

3rd Neighborhood Solution

0	1	0	0	0
1	0	0	0	0
0	1	0	0	0

Figure 5.4: Creating neighborhood solutions for TS

```
1. Initialization: Randomly generate a population (students) of size N, termination
   criterion
2. Evaluation and Ranking: Sort and rank the students using NS and CD computation

Repeat

3. Teacher Phase: Identify teacher (rank=1) and apply teacher phase crossover to obtain
   N new students.
       3.1 Mix new students with previous population to get 2N students.
       3.2 Sort and rank the students using NS and CD.
       3.3 Select first N students to be the population for next phase.
4. Student Phase: Apply student phase crossover to obtain N new students.
       4.1 Combine new students with the students obtained after teacher phase to get 2N
           students.
       4.2 Sort and rank the students using NS and CD.
       4.3 Select first N students to be the population for next phase.
5. Self-study Phase: Apply student phase mutation crossover to obtain N new students.
       5.1 Combine new students with the students obtained after student phase to get 2N
           students.
       5.2 Sort and rank the students using NS and CD.
       5.3 Select first N students to be the population for next phase.
6. Tabu Search: Identify teacher (rank=1) after self-study phase and apply Tabu Search.
       6.1 Find neighbourhood solutions (students) of teacher.
       6.2 Sort and rank the students using NS and CD.
       6.3 Identify the best student and combine it with the students obtained after self-
           study phase which yields (N+1) students.
       6.4 Again sort and rank the students using NS and CD.
       6.5 Select first N students to be the population for teacher phase in next iteration
Until a termination criterion is satisfied.
       Report the non-dominated set of solutions (students).
```

Figure 5.5: Pseudo code for Hybrid I-NSTLBO algorithm for PPSSP

5.4 Computational experiences

In this section, the performance of the proposed algorithms is evaluated by comparing it with NSGA-II and MOSS developed by Ghorbani & Rabbani (2009) and SFLA developed by Amirian & Sahraeian (2017) for PPSSP. Algorithms are tested on four different sets of randomly generated test problems of varying size and complexity. All algorithms are coded in MATLAB 7.12 environment and implemented using a laptop computer with Core i3, and Windows 10 using 4 GB of RAM. The test problems generation, assumptions and parameter setting of algorithms, comparison metrics and results obtained are described next.

5.4.1 Test problems

The proposed algorithms, NSGA-II, MOSS and SFLA, are evaluated on four sets of test problems, each containing 24 problems, with different complexity. The test instance generation scheme used in chapter 4 is further modified to contain the added information related to technical interdependencies. Large sized test problems are generated with the same scheme as used for high complexity problems but with more candidate projects. The problems are designed to differ in complexity by increasing the amount of interdependencies, project durations, number of resources required and decreasing the make-span, initial budget availability and availability of resources. The expected benefit for each project is generated randomly using uniform distribution U (50,150) in linearly decreasing order. The generation schemes for low, moderate, high and large sized problems are given in Table 5.6.

5.4.2 Parameter setting

The effectiveness of a meta-heuristic depends on the proper tuning of its parameters. In this section, the parameter settings for the I-NSTLBO and Hybrid I-NSTLBO algorithms developed for the PPSSP has been presented. The grey-based Taguchi method is best suited to find optimal parameter settings for an algorithm for the multi-objective optimization problems thus it has been used in the current work.

5.4.2.1 Parameter setting for I-NSTLBO algorithm

Parameters for proposed I-NSTLBO are same as used in chapter 4. It has 4 factors: number of students (N), teacher-student crossover coefficient (TSC), student-student crossover coefficient (SSC) and self-study mutation coefficient (SSM) each at 3 levels (Table 5.7). Thus, L9 orthogonal array (Table 5.8) is used to design the experiments for grey-based Taguchi method. The distinguishing coefficient (ζ) is set to 0.5. Considering objective 1 to be more important weights for the objective 1 & 2 are taken as 0.6 & 0.4 respectively. The results obtained for the average grey relational grade are presented in Table 5.9, and the main effects plot is shown in Figure 5.6. Optimum levels obtained from the experimentations are: N = 20, TSC = 0.4, SSC = 0.2 and SSM = 0.3.

Table 5.6: Test problem generation schemes

Factor	**Generating Rule**			
	Low Complexity Instances	**Moderate Complexity Instances**	**High Complexity Instances**	**Large Size Instances**
Number of available projects	$N = \{5, 6, . . . , 12\}$	$N = \{9, 10, . . . , 16\}$	$N = \{9, 10, . . . , 16\}$	$N = \{17, 18, . . . , 24\}$
Durations	d_i = U(1, 3)	d_i = U(3, 7)	d_i = U(7, 10)	d_i = U(10, 15)
Number of different types of resources	$K = \{1, 2\}$	$K = \{2, 3, 4\}$	$K = \{4, 5\}$	$K = \{4, 5\}$
Required resources for each project per period	r_{ik} = U(10, 15)	r_{ik} = U(10, 15)	r_{ik} = U(10, 15)	r_{ik} = U(10, 15)
Available amount of resource	R_{kt} = 4U(10, 15)	R_{kt} = 3U(10, 15)	R_{kt} = 2U(10, 15)	R_{kt} = 2U(10, 15)
Required budget for each project per period	C_{it} = U(15, 25)	C_{it} = U(15, 25)	C_{it} = U(15, 25)	C_{it} = U(15, 25)
Initial budget available	B_1 = 10U(15, 25)	B_1 = 8U(15, 25)	B_1 = 6U(15, 25)	B_1 = 6U(15, 25)
Make-span	$T = \max\{\max(d_i), \sum_i d_i$ x U(0.8,1)}	$T = \max\{\max(d_i)$, 0.8 x U($\max(d_i), \sum_i d_i)\}$	$T = \max\{\max(d_i)$, 0.5 x U($\max(d_i), \sum_i d_i)\}$	$T = \max\{\max(d_i)$, 0.5 x U($\max(d_i), \sum_i d_i)\}$
Probability of technical interdependency (Mutually exclusive projects)	10 %	20 %	30 %	30 %
Probability of technical interdependency (Complementary projects)	5 %	10 %	15 %	15 %
Probability of benefit interdependency (Complementary projects)	5 %	10 %	15 %	15 %
Probability of benefit interdependency (Competitive projects)	5 %	10 %	15 %	15 %
Gain & loss coefficients for benefit interdependent projects	g_{iv}, l_{iv} = U(0.30, 0.50)	g_{iv}, l_{iv} = U(0.30, 0.50)	g_{iv}, l_{iv} = U(0.30, 0.50)	g_{iv}, l_{iv} = U(0.30, 0.50)

Table 5.7: Factors and levels for I-NSTLBO algorithm

Factor	Symbol	Level	Values		
Number of students (Pop Size)	N	3	10	15	20
Teacher-student crossover coefficient	TSC	3	0.30	0.40	0.50
Student-student crossover coefficient	SSC	3	0.20	0.30	0.40
Self-study mutation coefficient	SSM	3	0.10	0.20	0.30

Table 5.8: The Taguchi orthogonal array L9 for parameter setting for I-NSTLBO algorithm

Trial No.	N	TSC	SSC	SSM
1	10	0.3	0.2	0.1
2	10	0.4	0.3	0.2
3	10	0.5	0.4	0.3
4	15	0.3	0.3	0.3
5	15	0.4	0.4	0.1
6	15	0.5	0.2	0.2
7	20	0.3	0.4	0.2
8	20	0.4	0.2	0.3
9	20	0.5	0.3	0.1

Table 5.9: Average grey relational grade by factor levels (mean value) for I-NSTLBO

Factor	Level 1	Level 2	Level 3
N	0.5495	0.5478	0.5908
TSC	0.4541	0.6231	0.6109
SSC	0.6157	0.5140	0.5583
SSM	0.5149	0.5786	0.5946

.

5.4.2.2 Parameter setting for Hybrid I-NSTLBO algorithm

Hybrid I-NSTLBO has tabu list (TL) as an extra factor in addition to the factors for I-NSTLBO each at three levels (Table 5.10), and L18 orthogonal array (Table 5.11) after modification is used for experimentation for grey-based Taguchi method. The distinguishing coefficient (ζ) and weights for the objective 1 & 2 are same as considered for I-NSTLBO algorithm. The results obtained for the average grey relational grade are presented in Table 5.12, and the main effects plot is shown in Figure 5.7. Optimum levels of parameters are obtained as: N = 20, TSC = 0.4, SSC = 0.2, SSM = 0.3 and TL = 0.4.

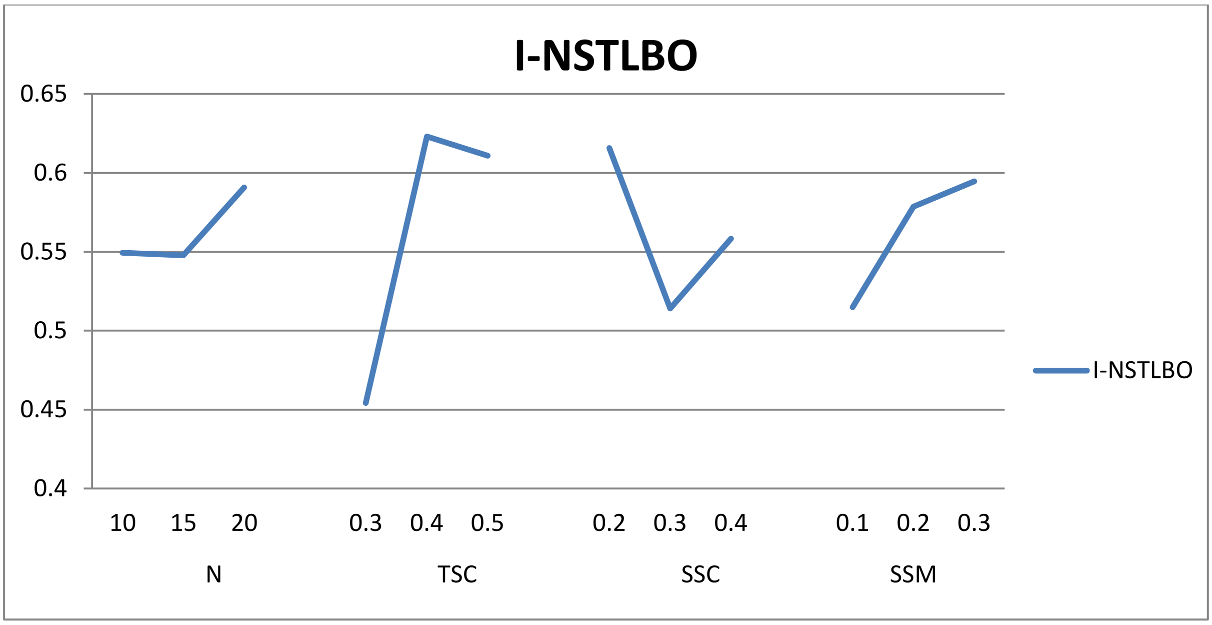


Figure 5.6: Grey-based Taguchi main effects plot (mean value) for I-NSTLBO

5.4.3 Comparison metrics

The comparison metrics are used to evaluate the performance of the proposed algorithms relative to other algorithms. The metrics used in chapter 4 are used here also to measure intensification and diversification of the search. Obtained non-dominated ratio proposed by (Ghorbani and Rabbani, 2009) is used to measure the intensification and spacing metric suggested by (Collette & Siarry, 2013) is used to measure diversification.

Table 5.10: Factors and levels for Hybrid I-NSTLBO algorithm

Factor	Symbol	Level	Values		
Number of students (Pop Size)	N	3	10	15	20
Teacher-student crossover coefficient	TSC	3	0.30	0.40	0.50
Student-student crossover coefficient	SSC	3	0.20	0.30	0.40
Self-study mutation coefficient	SSM	3	0.10	0.20	0.30
Length of tabu list	TL	3	0.40	0.50	0.60

5.4.3.1 Obtained non-dominated ratio

Non-dominated sorting (NS) is applied to the set of Pareto front solutions obtained by each of the algorithms considered here. The obtained non-dominated ratio is determined by dividing the number of NDS by each algorithm participating in the set by the number of solutions in the set. Higher values of the ratio show the greater ability of intensification of search by the considered algorithm. This ratio is considered as it is not sure about the Pareto optimality of the obtained NDS.

Table 5.11: The modified Taguchi orthogonal array L18 for Hybrid I-NSTLBO algorithm

Trial No.	N	TSC	SSC	SSM	TL
1	10	0.3	0.2	0.1	0.4
2	10	0.4	0.3	0.2	0.5
3	10	0.5	0.4	0.3	0.6
4	15	0.3	0.2	0.2	0.5
5	15	0.4	0.3	0.3	0.6
6	15	0.5	0.4	0.1	0.4
7	20	0.3	0.3	0.1	0.6
8	20	0.4	0.4	0.2	0.4
9	20	0.5	0.2	0.3	0.5
10	10	0.3	0.4	0.3	0.5
11	10	0.4	0.2	0.1	0.6
12	10	0.5	0.3	0.2	0.4
13	15	0.3	0.3	0.3	0.4
14	15	0.4	0.4	0.1	0.5
15	15	0.5	0.2	0.2	0.6
16	20	0.3	0.4	0.2	0.6
17	20	0.4	0.2	0.3	0.4
18	20	0.5	0.3	0.1	0.5

Table 5.12: Average grey relational grade by factor levels (mean value) for Hybrid I-NSTLBO

Factor	Level 1	Level 2	Level 3
N	0.4849	0.4520	0.5246
TSC	0.4944	0.5005	0.4665
SSC	0.5365	0.4702	0.4547
SSM	0.4786	0.4655	0.5172
TL	0.5181	0.4918	0.4515

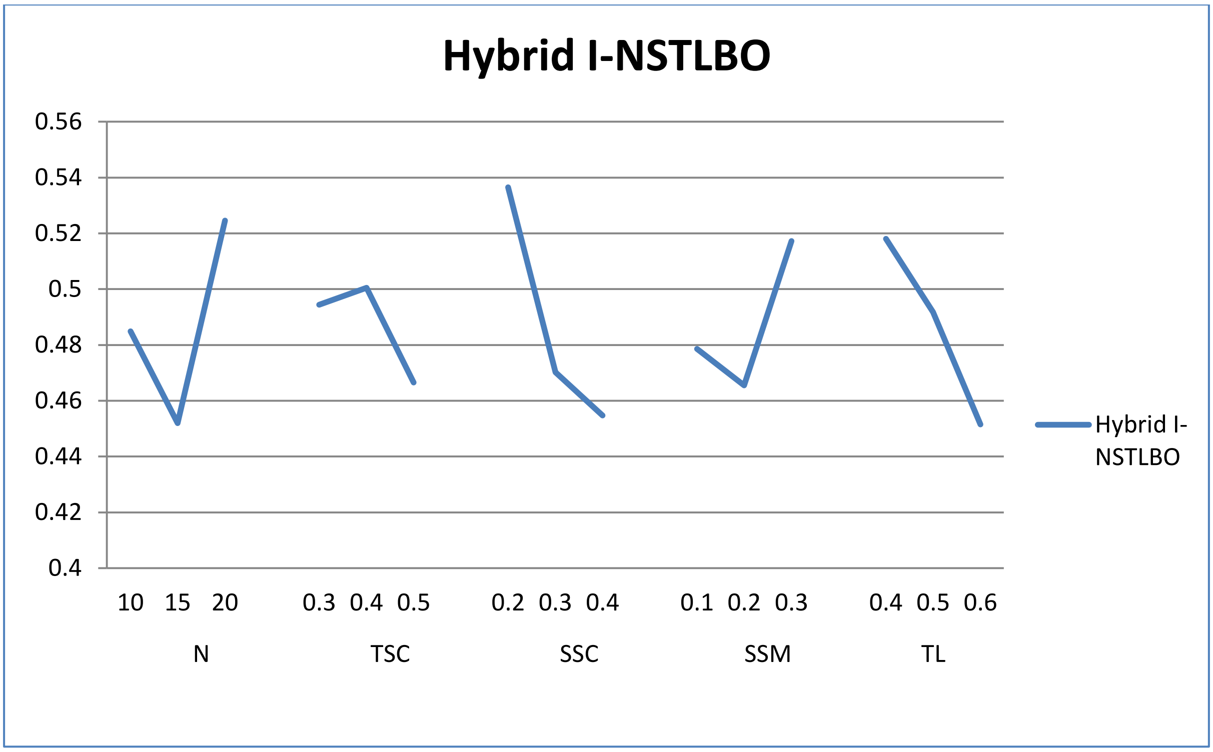


Figure 5.7: Grey-based Taguchi main effects plot (mean value) for Hybrid I-NSTLBO

5.4.3.2 Spacing metric

This performance metric is used to quantify the uniformity of the spread of NDS obtained in the Pareto front. The formula for the spacing metric used in this chapter is given by equation 5.17 as follows:

$$S = \sqrt{\frac{1}{(n-1)}\sum_{i=1}^{n}\left(\bar{d} - d_i\right)^2} \qquad (5.17)$$

Where n is the number of NDS in the Pareto front. The distance between the obtained NDS (d_i) is calculated using the expression 5.18.

$$d_i = \min_{i,i \neq j} \sum_{m=1}^{k}\left|f_m^i - f_m^j\right|, i, j = 1,2, \dots \dots, n \qquad (5.18)$$

Where f_m^i is the objective function value of the m^{th} objective in the i^{th} solution and k denotes the total number of objectives. The mean of these distances $(\bar{d})$ is calculated using expression 5.19 as follows:

$$\bar{d} = \sum_{i=1}^{n} d_i/n \qquad (5.19)$$

Smaller values of *S* are desirable as smaller the value better the uniformity of the spread.

5.4.4 Comparative results

In this section, the performance of the proposed algorithms is evaluated by comparing the results with NSGA II, MOSS and SFLA. The performance is evaluated on four set of instances, having 24 instances in each set, of various complexity generated randomly in this research. Each algorithm is run 10 times on each instance, and the outputs are averaged to get the results for an instance. Further, the results are averaged for each instance set to get the final results.

The average number of NDS obtained in the Pareto front for each instance set is given in Table 5.13, and the average obtained non-dominated solutions ratio is given in Table 5.14. From the results obtained for average number of NDS and average obtained non-dominated solutions ratio, it is clear that the proposed Hybrid I-NSTLBO outperforms other five algorithms for all type of instances. The performance of I-NSTLBO is slightly better than NSTLBO. This has happened due to the increased exploration during the search in teacher phase. However, it is difficult to decide between NSTLBO and SFLA that which algorithm is better. However, NSTLBO and SFLA perform better than MOSS and NSGA II.

The results for the spacing metric are presented in Table 5.15. Looking at the results obtained for spacing metric, it is observed that the value of the spacing metric for Hybrid I-NSTLBO is better than other algorithms for all the sets of instances. The difference between the performances of I-NSTLBO, NSTLBO and SFLA is negligible. On the other hand, the MOSS performs slightly better than NSGA II in terms of diversification. However, I-NSTLBO, NSTLBO and SFLA perform better than MOSS and NSGA II.

Overall it can be concluded that proposed Hybrid I-NSTLBO outperforms all other algorithms for all comparison criteria for all type of instances.

Table 5.13: Results for the average number of non-dominated solutions

Factor	Instance	Algorithms					
		Hybrid I-NSTLBO	I-NSTLBO	NSTLBO	SFLA	MOSS	NSGA II
Average number of non-dominated solutions	Low	4.728166	4.341063	4.309579	4.118023	3.075881	2.919549
	Moderate	4.809067	4.511066	4.399652	4.301597	3.320833	3.204511
	High	4.961357	4.362857	4.26166	4.490458	3.487533	3.016498
	Large	4.982142	4.401514	4.354709	4.516592	3.516667	2.903983

Table 5.14: Results for the average obtained non-dominated solutions ratios

Factor	Instance	Algorithms					
		Hybrid I-NSTLBO	I-NSTLBO	NSTLBO	SFLA	MOSS	NSGA II
Average obtained non-dominated solutions ratio	Low	0.2265165	0.19007	0.183057	0.185074	0.133338	0.081945
	Moderate	0.2407759	0.187694	0.184568	0.176181	0.124664	0.086117
	High	0.2697076	0.181292	0.179038	0.16944	0.122324	0.078199
	Large	0.2643344	0.1796	0.176709	0.181324	0.125177	0.072856

Table 5.15: Results for spacing metric values

Factor	Instance	Algorithms					
		Hybrid I-NSTLBO	I-NSTLBO	NSTLBO	SFLA	MOSS	NSGA II
Spacing Metric	Low	0.752833	0.781322	0.802625	0.841278	0.905355	0.998423
	Moderate	0.742316	0.790095	0.831171	0.835265	0.970934	0.975721
	High	0.773581	0.873673	0.885517	0.875034	0.930702	0.966588
	Large	0.808371	0.886495	0.913866	0.890347	0.972937	1.064451

5.5 Conclusions

In this chapter, the project portfolio selection and scheduling problem (PPSSP), considered in the previous chapter, is extended to consider benefit interdependencies. The problem has been formulated as multi-objective MILP considering two objectives: maximization of total expected benefit and minimization of resource usage variation. The total expected benefit consists of the benefits from the individual projects and the synergic benefit/loss due to benefit interdependencies. Benefit from a project is considered to be time sensitive and added to the budget for consideration of more projects. Technical interdependencies: mutual exclusiveness and complementariness are also considered.

An improved NSTLBO (I-NSTLBO) algorithm is developed for the problem. The teacher phase of I-NSTLBO algorithm is modified in a probabilistic way using the learning experience of other students to teach some of the students. The proposed I-NSTLBO algorithm is further hybridized with Tabu Search (TS) in order to improve the exploitation ability of the algorithm. The grey-based Taguchi method is used to tune the parameters of the algorithm. Proposed algorithms are tested on 96 randomly generated instances with different complexity and compared with well-known meta-heuristics: NSTLBO, SFLA, MOSS, and NSGA II existing in the literature.

From the results, it is clear that Hybrid I-NSTLBO outperforms the other five algorithms with respect to all performance criteria for all type of instances. The performance of the I-NSTLBO, NSTLBO and SFLA is comparable as the difference is almost negligible. However, these algorithms perform better than MOSS and NSGA II. The MOSS performs better than NSGA II for obtained non-dominated ratio, but there is no significant difference in the performance for the spacing metric values. Overall it can be concluded that proposed Hybrid I-NSTLBO outperforms all other algorithms for all comparison criteria for all type of instances.

This work does not consider detailed activity scheduling which may be taken up as future research. It would also be interesting to consider dynamic selection and scheduling of the projects for further research and appropriate algorithms may be developed to solve the problem.

Chapter-6

Summary and Conclusion

Chapter 6
Summary and Conclusion

With the increased globalization and competitiveness, most of the organizations deal with multiple projects at the same time. Such organizations need efficient decision making in selecting the right mix of the projects and scheduling them. Traditionally, these two activities were performed in a sequential manner. With this sequential approach, however, it may not be possible to schedule the selected projects within the stipulated time frame. Thus, scheduling needs to be considered as an integral part of the project selection process. Usually, the projects are highly interdependent in nature. Further, the limited availability of the resources and their efficient utilization are big issues to be kept in mind while decision making. The integration of selection and scheduling processes makes the decision making more complex but enhances the quality of the decision by including feasible and better projects.

To find the research gaps in the existing literature, a detailed literature review on the integrated selection and scheduling of projects was carried out in chapter 2. The review of the literature revealed that, in existing literature, the interdependencies among the projects have been considered in a limited way for the integrated problem of selection and scheduling of the projects. Out of the two technical interdependencies, only mutual exclusiveness has been considered as project interdependencies, but complementariness is yet ignored in the existing literature. There is a lack of research in the consideration of benefit interdependencies also which renders an increase in the overall benefit of the portfolio. Moreover, the reinvestment of benefits has been considered scantly in the literature. Lastly, there is a need of an efficient solution approach for such a complex problem. Keeping this in mind, this thesis aims at developing the models for the integrated selection and scheduling of the projects which handle the interdependencies among the projects and optimize the utilization of resources also.

Motivated by the abovementioned gaps, the proposed research work in this thesis is carried out in three parts. In the first part, a single-objective model for the problem is developed. In the second part, a multi-objective model is developed using the reinvestment strategy. In the third part, the multi-objective model is extended to consider benefit interdependencies.

In chapter 3, a single objective model is offered with two types of technical interdependencies: mutual exclusiveness and complementariness. The problem is formulated as a zero-one integer linear programming model. A modified TLBO algorithm has been proposed for the problem. In order to improve the performance of the algorithm, the hybridization of the TLBO with well-known tabu search algorithm is proposed. Taguchi approach is used to tune the parameters of the algorithms. The proposed algorithms are tested on four different complexity level data sets generated in this research. The proposed algorithms are compared with each other and are also compared with the SFLA existing in the literature. The results show that the proposed Hybrid TLBO-TS algorithm outperforms all other algorithms.

In chapter 4, an MILP model is presented for the problem with two objectives: maximization of total expected benefit and minimization of resource usage variation. Benefit from a project is considered to be time sensitive and added to the budget for consideration of more projects. Two types of technical interdependencies: mutual exclusiveness and complementariness are considered. The problem is formulated using a multi-objective MILP and a modified NSTLBO algorithm has been proposed to solve the problem. The algorithm is hybridized with Tabu Search (TS) algorithm, and Hybrid NSTLBO is also proposed. A grey-based Taguchi method is used to optimize the parameters of the algorithms. Proposed algorithms are compared with three well-known meta-heuristics, NSGA II, MOSS and SFLA developed for the problem by solving 96 randomly generated problems of different size and complexity. From the results, it can be concluded that the proposed Hybrid NSTLBO outperforms other algorithms in terms of diversification and intensification for all type of instances.

This research is further extended in chapter 5 to consider benefit interdependencies. The problem is formulated as multi-objective MILP considering two objectives: maximization of total expected benefit and minimization of resource usage variation. The total expected benefit consists of the benefits from the individual projects and the synergic benefit/loss due to benefit interdependencies. Benefit from a project is considered to be time sensitive and added to the budget for consideration of more projects. Technical interdependencies: mutual exclusiveness and complementariness are also considered. An improved NSTLBO (I-NSTLBO) algorithm is developed for the problem. The teacher phase of the I-NSTLBO algorithm is modified in a probabilistic way using the learning experience of other students to teach some of the students. The rest of the algorithm remains the same as proposed in chapter

4. This proposed I-NSTLBO algorithm is further hybridized with Tabu Search (TS) in order to improve the exploitation ability of the algorithm. A grey-based Taguchi method is used to optimize the parameters of the algorithms. Proposed algorithms are compared with three well-known meta-heuristics, NSGA II, MOSS and SFLA developed for the problem by solving 96 randomly generated problems of different size and complexity. From results, it can be concluded that proposed Hybrid I-NSTLBO outperforms all other algorithms in all the comparison criteria for all type of instances while the results for I-NSTLBO are also promising.

The results of the proposed algorithms have already been analysed and highlighted in chapters 3, 4 and 5. The performance of hybrid TLBO-TS, Hybrid NSTLBO and Hybrid I-NSTLBO developed in this study is quite promising. These algorithms are suitable for providing quick near optimal solutions to the complex and large-sized problems.

The PPSSP models developed in this study may be useful to project managers in the simultaneous selection of projects in the portfolio and scheduling when projects exhibit technical and benefit interdependencies. The models can be conveniently used by any organization dealing with multiple projects running at the same time. The models developed in the current study may help the project managers, dealing with big portfolios, to achieve increased benefit and enhanced resource utilization by reinvesting the benefit accrued from the completed projects. The model may find application in a wide variety of industries and service sectors such as the R & D based organizations, software industry, construction firms, contracting engineering firms, defence, pharmaceuticals, hospitals, chemicals, banking, information systems, accounting, advertising and governments. The developed models can suitably handle real-life large-sized problem commonly encountered in these organizations.

6.1 Major contributions of the present research

1) A 0-1 IP model has been developed for single objective project selection and scheduling problem with consideration of two technical interdependencies: mutual exclusiveness and complementariness. The complementariness (technical interdependency) among the projects for integrated project selection and scheduling problem has been considered for the first time in the literature.
2) Three new meta-heuristics have been developed to solve single objective integrated project selection and scheduling problem.

3) An MILP based model is developed for the multi-objective project selection and scheduling problem. This model considers technical interdependencies, time-dependent nature of project benefits and reinvestment of benefits in the portfolio.
4) Two meta-heuristics have been developed based on posteriori decision-making approach to find the optimal Pareto solutions to the above discussed multi-objective problem.
5) Extended the MILP model proposed earlier to consider benefit interdependencies. The benefit interdependencies among the projects for integrated project selection and scheduling problem has been considered for the first time in the literature.
6) The two meta-heuristics developed for the multi-objective problem are modified to improve its diversification capabilities.

6.2 Limitations and future research directions

1) Out of the three types of interdependencies, this thesis considers technical and benefit interdependencies. Resource interdependencies may be considered for the problem for further research.
2) In this thesis, single and multi-objective problems have been formulated for the integrated selection and scheduling of the projects. But, the proposed models do not consider the detailed activity scheduling. Activity scheduling also helps to determine the start and finish time of each project precisely. Also, the actual benefit of resources interdependencies can be realized with the detailed scheduling of activities.
3) The planning horizon is considered to be fixed in all the proposed models in this thesis. It would be an interesting extension to model the problem with flexible time horizon.
4) In the current research work, all the parameters are assumed to be deterministic. The current work may be extended to include uncertainties associated with the project parameters, and suitable solution approaches may be developed.
5) Dynamic selection and scheduling of projects could be another interesting extension to the problem for further research.

References

References

1. Aaker, D. A., & Tyebjee, T. T. (1978). A model for the selection of interdependent R&D projects. *IEEE Transactions on engineering management,* (2), 30-36.
2. Abbassi, M., Ashrafi, M., & Tashnizi, E. S. (2014). Selecting balanced portfolios of R&D projects with interdependencies: A Cross-Entropy based methodology. *Technovation, 34*(1), 54-63.
3. Aich, U., & Banerjee, S. (2016). Application of teaching learning based optimization procedure for the development of SVM learned EDM process and its pseudo Pareto optimization. *Applied Soft Computing, 39*, 64-83.
4. Amirian, H., & Sahraeian, R. (2018). A hybrid integer grey programming for an integrated problem of project selection and scheduling with interval data. *Journal of Industrial and Production Engineering, 35*(4), 199-213.
5. Amirian, H., & Sahraeian, R. (2017). Solving a grey project selection scheduling using a simulated shuffled frog leaping algorithm. *Computers & Industrial Engineering, 107*, 141-149.
6. Arratia M., N. M., López I., F., Schaeffer, S. E., & Cruz-Reyes, L. (2016). Static R&D project portfolio selection in public organizations. *Decision Support Systems, 84*, 53-63.
7. Baykasoğlu, A., Hamzadayi, A., & Köse, S. Y. (2014). Testing the performance of teaching–learning based optimization (TLBO) algorithm on combinatorial problems: Flow shop and job shop scheduling cases. *Information Sciences, 276*, 204-218.
8. Belenky, A. S. (2012). A Boolean programming problem of choosing an optimal portfolio of projects and optimal schedules for them by reinvesting within the portfolio the profit from project implementation. *Applied Mathematics Letters, 25*(10), 1279-1284.
9. Bhattacharyya, R., Kumar, P., & Kar, S. (2011). Fuzzy R&D portfolio selection of interdependent projects. *Computers & Mathematics with Applications, 62*(10), 3857-3870.
10. Carazo, A. F., Gómez, T., Molina, J., Hernández-Díaz, A. G., Guerrero, F. M., & Caballero, R. (2010). Solving a comprehensive model for multiobjective project portfolio selection. *Computers & operations research, 37*(4), 630-639.
11. Chen, J., & Askin, R. G. (2009). Project selection, scheduling and resource allocation with time dependent returns. *European Journal of Operational Research, 193*(1), 23-34.
12. Coffin, M. A., & Taylor, B. W. (1996a). Multiple criteria R&D project selection and scheduling using fuzzy logic. *Computers & Operations Research, 23*(3), 207-220.

13. Coffin, M. A., & TAYLOR III, B. W. (1996b). R&D project selection and scheduling with a filtered beam search approach. *IIE transactions, 28*(2), 167-176.
14. Collette, Y., & Siarry, P. (2013). *Multiobjective optimization: principles and case studies*. Springer Science & Business Media.
15. Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation, 6*(2), 182-197.
16. Demeulemeester, E. L., & Herroelen, W. S. (2006). Project scheduling: a research handbook (Vol. 49). Springer Science & Business Media.
17. Dokeroglu, T. (2015). Hybrid teaching–learning-based optimization algorithms for the Quadratic Assignment Problem. *Computers & Industrial Engineering, 85*, 86-101.
18. Fox, G. E., Baker, N. R., & Bryant, J. L. (1984). Economic models for R and D project selection in the presence of project interactions. *Management science, 30*(7), 890-902.
19. Ganji, M., Alinaghian, M., & Sajadi, S. M. (2016). A new model for optimising simultaneously projects selection and resource-constrained project scheduling problem. *International Journal of Productivity and Quality Management, 19*(4), 511-525.
20. Garcia, C. (2014). A metaheuristic algorithm for project selection and scheduling with due windows and limited inventory capacity. *Kybernetes, 43*(9/10), 1483-1499.
21. Ghahremani, P., & Naderi, B. (2015). Solution algorithms for the project selection and scheduling problem with resource constraints and time dependent returns. *International Journal of Industrial and Systems Engineering, 19*(3), 348-363.
22. Ghasemzadeh, F., Archer, N., & Iyogun, P. (1999). A zero-one model for project portfolio selection and scheduling. *Journal of the Operational Research Society, 50*(7), 745-755.
23. Ghorbani, S., & Rabbani, M. (2009). A new multi-objective algorithm for a project selection problem. *Advances in Engineering Software, 40*(1), 9-14.
24. Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & operations research, 13*(5), 533-549.
25. Gupta, S. K., Kyparisis, J., & Ip, C. M. (1992). Note—Project selection and sequencing to maximize net present value of the total return. *Management Science, 38*(5), 751-752.
26. Gutjahr, W. J., Katzensteiner, S., & Reiter, P. (2007, September). A VNS algorithm for noisy problems and its application to project portfolio analysis. In *International Symposium on Stochastic Algorithms* (pp. 93-104). Springer, Berlin, Heidelberg.

27. Gutjahr, W. J., Katzensteiner, S., Reiter, P., Stummer, C., & Denk, M. (2008). Competence-driven project portfolio selection, scheduling and staff assignment. *Central European Journal of Operations Research, 16*(3), 281-306.
28. Hosseininasab, S. M., & Shetab-Boushehri, S. N. (2015). Integration of selecting and scheduling urban road construction projects as a time-dependent discrete network design problem. *European Journal of Operational Research, 246*(3), 762-771.
29. Huang, X., & Zhao, T. (2014). Project selection and scheduling with uncertain net income and investment cost. *Applied Mathematics and Computation, 247*, 61-71.
30. Huang, X., Zhao, T., & Kudratova, S. (2016). Uncertain mean-variance and mean-semivariance models for optimal project selection and scheduling. *Knowledge-Based Systems*, *93*, 1-11.
31. Jafarzadeh, M., Tareghian, H. R., Rahbarnia, F., & Ghanbari, R. (2015). Optimal selection of project portfolios using reinvestment strategy within a flexible time horizon. *European Journal of Operational Research*, *243*(2), 658-664.
32. Ji, X., Ye, H., Zhou, J., Yin, Y., & Shen, X. (2017). An improved teaching-learning-based optimization algorithm and its application to a combinatorial optimization problem in foundry industry. *Applied Soft Computing*, *57*, 504-516.
33. Keesari, H. S., & Rao, R. V. (2014). Optimization of job shop scheduling problems using teaching-learning-based optimization algorithm. *Opsearch, 51*(4), 545-561.
34. Khalghani, M. R., & Khooban, M. H. (2014). A novel self-tuning control method based on regulated bi-objective emotional learning controller's structure with TLBO algorithm to control DVR compensator. *Applied Soft Computing*, *24*, 912-922.
35. Khare, R., & Kumar, Y. (2016). A novel hybrid MOL–TLBO optimized techno-economic-socio analysis of renewable energy mix in island mode. *Applied Soft Computing, 43*, 187-198.
36. Kolisch, R., & Meyer, K. (2006). Selection and scheduling of pharmaceutical research projects. In *Perspectives in Modern Project Scheduling* (pp. 321-344). Springer, Boston, MA.
37. Kuo, Y., Yang, T., & Huang, G. W. (2008). The use of a grey-based Taguchi method for optimizing multi-response simulation problems. *Engineering Optimization, 40*(6), 517-528.
38. Lambrechts, O., Demeulemeester, E., & Herroelen, W. (2008). A tabu search procedure for developing robust predictive project schedules. *International Journal of Production Economics*, *111*(2), 493-508.

39. Larson, E. W., & Gray, C. F. (2000). *Project management: The managerial process*. International Edition, The Irwin/McGraw-Hill Series, Singapore.

40. Lee, J. W., & Kim, S. H. (2000). Using analytic network process and goal programming for interdependent information system project selection. *Computers & Operations Research, 27*(4), 367-382.

41. Li, J. Q., Pan, Q. K., & Liang, Y. C. (2010). An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems. *Computers & Industrial Engineering, 59*(4), 647-662.

42. Li, X., Fang, S. C., Guo, X., Deng, Z., & Qi, J. (2016). An extended model for project portfolio selection with project divisibility and interdependency. *Journal of Systems Science and Systems Engineering, 25*(1), 119-138.

43. Liesiö, J., Mild, P., & Salo, A. (2008). Robust portfolio modeling with incomplete cost information and project interdependencies. *European Journal of Operational Research, 190*(3), 679-695.

44. Lin, Y., Bian, Z., & Liu, X. (2016). Developing a dynamic neighborhood structure for an adaptive hybrid simulated annealing–tabu search algorithm to solve the symmetrical traveling salesman problem. *Applied Soft Computing, 49,* 937-952.

45. Liu, S. S., & Wang, C. J. (2011, April). Optimization for project selection and scheduling problems considering time-dependent resource constraints. In *Electric Technology and Civil Engineering (ICETCE), 2011 International Conference on* (pp. 530-533). IEEE.

46. Liu, S. S., & Wang, C. J. (2011). Optimizing project selection and scheduling problems with time-dependent resource constraints. *Automation in Construction, 20*(8), 1110-1119.

47. Mika, M., Waligora, G., & Węglarz, J. (2008). Tabu search for multi-mode resource-constrained project scheduling with schedule-dependent setup times. *European Journal of Operational Research, 187*(3), 1238-1250.

48. Musselwhite, W. C. (1990). Time-based innovation: the new competitive advantage. *Training & Development Journal, 44*(1), 53-57.

49. Pajares, J., López, A., Araúzo, A., & Hernández, C. (2009, April). Project Portfolio Management, selection and scheduling. Bridging the gap between strategy and operations. In *XIII Congreso de Ingeniería de Organización* (pp. 1421-1429).

50. Pajares, J., & López, A. (2014). New methodological approaches to project portfolio management: the role of interactions within projects and portfolios. *Procedia-Social and Behavioral Sciences, 119*, 645-652.

51. Pan, N. H., Hsaio, P. W., & Chen, K. Y. (2008). A study of project scheduling optimization using Tabu Search algorithm. *Engineering Applications of Artificial Intelligence, 21*(7), 1101-1112.

52. Pérez, F., Gómez, T., Caballero, R., & Liern, V. (2018). Project portfolio selection and planning with fuzzy constraints. *Technological Forecasting and Social Change, 131*, 117-129.

53. Project Management Institute, (2004). *A Guide to the Project Management Body of Knowledge (PMBOK®)*, 3rd edition, Newtown Square, PA: Project Management Institute.

54. Rao, R. V. (2016). Teaching-learning-based optimization algorithm. In *Teaching Learning Based Optimization Algorithm*. Springer, Cham.

55. Rao, R., & Patel, V. (2012). An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems. *International Journal of Industrial Engineering Computations, 3*(4), 535-560.

56. Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2012). Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. *Information sciences, 183*(1), 1-15.

57. Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-Aided Design, 43*(3), 303-315.

58. Santhanam, R., & Kyparisis, G. J. (1996). A decision model for interdependent information system project selection. *European Journal of Operational Research, 89*(2), 380-399.

59. Shariatmadari, M., Nahavandi, N., Zegordi, S. H., & Sobhiyah, M. H. (2017). Integrated resource management for simultaneous project selection and scheduling. *Computers & Industrial Engineering, 109*, 39-47.

60. Shen, L. (2014). A tabu search algorithm for the job shop problem with sequence dependent setup times. *Computers & Industrial Engineering, 78*, 95-106.

61. Shi, B., Wang, H., & Qi, L. (2011). A Hybrid Genetic Algorithm for Scheduling and Selecting a Project Portfolio. In *Foundations of Intelligent Systems* (pp. 69-75). Springer Berlin Heidelberg.

62. Shou, Y., Xiang, W., Li, Y., & Yao, W. (2014). A multiagent evolutionary algorithm for the resource-constrained project portfolio selection and scheduling problem. *Mathematical Problems in Engineering, 2014*.

63. Shou, Y. Y., & Huang, Y. L. (2010). Combinatorial auction algorithm for project portfolio selection and scheduling to maximize the net present value. *Journal of Zhejiang University SCIENCE C, 11*(7), 562-574.

64. Sukkerd, W., & Wuttipornpun, T. (2016). Hybrid genetic algorithm and tabu search for finite capacity material requirement planning system in flexible flow shop with assembly operations. *Computers & Industrial Engineering, 97*, 157-169.

65. Sun, H., & Ma, T. (2005). A packing-multiple-boxes model for R&D project selection and scheduling. *Technovation, 25*(11), 1355-1361.

66. Tofighian, A. A., & Naderi, B. (2015). Modeling and solving the project selection and scheduling. *Computers & Industrial Engineering, 83*, 30-38.

67. Tseng, C. C., & Liu, B. S. (2011). Hybrid Taguchi-genetic algorithm for selecting and scheduling a balanced project portfolio. *Journal of Science and Engineering Technology, 7*(1), 11-18.

68. Tuncel, G., & Aydin, D. (2014). Two-sided assembly line balancing using teaching–learning based optimization algorithm. *Computers & Industrial Engineering, 74*, 291-299.

69. Turner, J. R. (1993). *The handbook of project-based management: improving the processes for achieving strategic objectives*. 2nd edition. McGraw-Hill, London.

70. Waligóra, G. (2008). Discrete–continuous project scheduling with discounted cash flows-A tabu search approach. *Computers & Operations Research, 35*(7), 2141-2153.

71. Wang, B., & Song, Y. (2016). Reinvestment Strategy-Based Project Portfolio Selection and Scheduling with Time-Dependent Budget Limit Considering Time Value of Capital. In *Proceedings of the 2015 International Conference on Electrical and Information Technologies for Rail Transportation* (pp. 373-381). Springer Berlin Heidelberg.

72. Wu, Q., Wang, Y., & Lü, Z. (2015). A tabu search based hybrid evolutionary algorithm for the max-cut problem. *Applied Soft Computing, 34*, 827-837.

73. Xu, Y., Wang, L., Wang, S. Y., & Liu, M. (2015). An effective teaching–learning-based optimization algorithm for the flexible job-shop scheduling problem with fuzzy processing time. *Neurocomputing, 148*, 260-268.

74. Yu, D., Hong, J., Zhang, J., & Niu, Q. (2018). Multi-Objective Individualized-Instruction Teaching-Learning-Based Optimization Algorithm. *Applied Soft Computing, 62*, 288-314.

75. Yu, K., Wang, X., & Wang, Z. (2016). An improved teaching-learning-based optimization algorithm for numerical and engineering optimization problems. *Journal of Intelligent Manufacturing, 27*(4), 831-843.

76. Zhao, T. Y., & Huang, X. X. (2013). Chance-Constrained Programming Model for Optimal Project Selection and Scheduling. In *The 19th International Conference on Industrial Engineering and Engineering Management* (pp. 243-250). Springer, Berlin, Heidelberg.
77. Zheng, H. Y., Wang, L., & Zheng, X. L. (2017). Teaching–learning-based optimization algorithm for multi-skill resource constrained project scheduling problem. *Soft Computing, 21*(6), 1537-1548.
78. Zou, F., Wang, L., Hei, X., & Chen, D. (2015). Teaching–learning-based optimization with learning experience of other learners and its application. *Applied Soft Computing, 37*, 725-736.

Appendices

Appendix I: Problem Instance Generation

In this appendix, the MATLAB codes developed in the current study for generation of low, moderate, high and large complexity instances have been provided.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


               %%% 1. Codes for Low Complex Instance Generation %%%

clc;
clear all;
close;

for i = 1:2
    for projects = 3:10
        P = projects;                         % The number of competitive projects
        D = randi([1,3] , 1, P);              % Duration of projects
        K = randi([1,2]);                     % The number of resources
        T = fix((randi([8,10])/10) * sum(D)); % The number of time periods
        a = randi([10,15] , K,P);             % Resource requirement per period
        r = 4*max(max(a));                    % Maximum Resource availability
        B = zeros(P,T);                       % The Expected benefit in each time
                                              period(Benefit matrix)
        for alpha = 1:P
            B(alpha , :) = sort(randi([100 , 999] , 1, T) , 'descend' );
        end
        H = INDCS(P , 1);
        name = strcat('lowcomplex-',num2str(projects),'-',num2str(i),'.mat');
        save(name,'P','T','D','K','a','r','B','H');
        range = strcat('A' , num2str(10*(projects - 3) + i));
        xlswrite('indexes.xls' , [P T K] , 1 , range);
        names = strcat('lowcomplex-',num2str(projects),'-',num2str(i),'.xls');
        xlswrite(names , [P T K r] , 1);
        xlswrite(names , D , 2);
        xlswrite(names , a , 3);
        xlswrite(names , B , 4);
        xlswrite(names , H , 5);
    end
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


            %%% 2. Codes for Moderate Complex Instance Generation %%%

clc;
clear all;
close;

for i = 1:2
    for projects = 7:14
        P = projects;                         % The number of competitive projects
        D = randi([3,7] , 1, P);              % Duration of projects
        K = randi([2,4]);                     % The number of resources
        T = fix(0.8 * randi([max(D) sum(D)]));      % The number of time periods
        a = randi([10,15] , K,P);               % Resource requirement per period
        r = 3*max(max(a));                      % Maximum Resource availability
        B = zeros(P,T);                         % The Expected benefit in each time
                                                period(Benefit matrix)

        for alpha = 1:P
            B(alpha , :) = sort(randi([100 , 999] , 1, T) , 'descend' );
        end

        H = INDCS(P , 2);
        name = strcat('modcomplex-',num2str(projects),'-',num2str(i),'.mat');
        save(name,'P','T','D','K','a','r','B','H');
        range = strcat('A' , num2str(10*(projects - 3) + i));
        xlswrite('indexes.xls' , [P T K] , 1 , range);
        names = strcat('modcomplex-',num2str(projects),'-',num2str(i),'.xls');
        xlswrite(names , [P T K r] , 1);
        xlswrite(names , D , 2);
        xlswrite(names , a , 3);
        xlswrite(names , B , 4);
        xlswrite(names , H , 5);
    end
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

%%% 3. Codes for High Complex Instance Generation %%%

```
clc;
clear all;
close;

for i = 1:2
    for projects = 7:14
        P = projects;                           % The number of competitive projects
        D = randi([7,10] , 1, P);               % Duration of projects
        K = randi([4,5]);                       % The number of resources
        T = fix(0.5 * randi([max(D) sum(D)]));% The number of time periods
        a = randi([10,15] , K,P);               % Resource requirement per period
        r = 2*max(max(a));                      % Maximum Resource available
        B = zeros(P,T);                         % The Expected benefit in each time
                                                period(Benefit matrix)
        for alpha = 1:P
            B(alpha , :) = sort(randi([100 , 999] , 1, T) , 'descend' );
        end

        H = INDCS(P , 3);
        name = strcat('highcomplex-',num2str(projects),'-',num2str(i),'.mat');
        save(name,'P','T','D','K','a','r','B','H');
        range = strcat('A' , num2str(10*(projects - 3) + i));
        xlswrite('indexes.xls' , [P T K] , 1 , range);
        names = strcat('highcomplex-',num2str(projects),'-',num2str(i),'.xls');
        xlswrite(names , [P T K r] , 1);
        xlswrite(names , D , 2);
        xlswrite(names , a , 3);
        xlswrite(names , B , 4);
        xlswrite(names , H , 5);
    end
end
```

%%

%%% 4. Codes for Large Sized Instance Generation %%%

```
clc;
clear all;
close;

for i = 1:2
    for projects = 15:22
        P = projects;                        % The number of competitive projects
        D = randi([10,15] , 1, P);           % Duration of projects
        K = randi([4,5]);                    % The number of resources
        T = fix(0.5 * randi([max(D) sum(D)]));% The number of time periods
        a = randi([10,15] , K,P);            % Resource requirement per period
        r = 2*max(max(a));                   % Maximum Resource availability
        B = zeros(P,T);                      % The Expected benefit in each time
                                             period(Benefit matrix)
        for alpha = 1:P
            B(alpha , :) = sort(randi([100 , 999] , 1, T) , 'descend' );
        end
        H = INDCS(P , 3);
        name = strcat('largecomplex-',num2str(projects),'-',num2str(i),'.mat');
        save(name,'P','T','D','K','a','r','B','H');
        range = strcat('A' , num2str(10*(projects - 3) + i));
        xlswrite('indexes.xls' , [P T K] , 1 , range);
        names = strcat('largecomplex-',num2str(projects),'-',num2str(i),'.xls');
        xlswrite(names , [P T K r] , 1);
        xlswrite(names , D , 2);
        xlswrite(names , a , 3);
        xlswrite(names , B , 4);
        xlswrite(names , H , 5);
    end
end
```

%%

%%% Function file for Interdependencies %%%
%%% This function is common for all type of instance generation %%%

```
function H = INDCS(n , amt)
a = triu(randi([0 10],n) , 1);
b = zeros(n);
for i = 1:n
   for j = 1:n
       if (a(i , j) <=amt && a(i , j) ~=0)
          b(i , j) = 1;
       else
           b(i , j) = 0;
       end
   end
end
c = b+ b';
d = zeros(n);
e = [];
for i=1:n
    for j = 1:n
        d(i , j) = c(i , j)*j;
    end
    for k = 1:n
       if(d(i , k)~=0)
          e = [e d(i , k)];
       end
    end
    H{i} = e;
    e = [];
end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Appendix II: MATLAB Codes for Proposed Algorithms

In this appendix, the MATLAB codes developed in the current study for TLBO, TS and Hybrid TLBO-TS have been provided.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                  %%%%%%%%%% 1. MATLAB Codes for TLBO %%%%%%%%%%


clc
clear all

                              %% TLBO Parameters %%


MaxIt = 100;                     % Maximum Number of Iterations
N = 15;                          % Population size

load lowcomplex-3-1.mat
filename = 'TLBO-LCD-3-1-E1-SL.xls';
fig_name = 'TLBO-LCD-3-1-E1-SL.jpg';

TSC = round(0.4*P);              % Teacher student crossover coefficient
SSC = round(0.4*P);              % Student student crossover coefficient
SSM = round(0.3*P);              % Self-study mutation coefficient
Itr_4_con = 0;                   % Number of iterations for convergence


                           %% TLBO Algorithm Starts %%


R = ones(P,T,K);
for i = 1:K
  for j = 1:P
      R(j,:,i) = a(i,j);
  end
end
R;

excludedcount = D-1;
X = zeros(P,T);
Result = cell(1, N);
ResultB = cell(1, N);
for i = 1:N
    while(1)
        for row = 1:P
            rv = [1, zeros(1, T - excludedcount(row) - 1)];
            X(row, 1 : (T - excludedcount(row))) = rv(randperm(numel(rv)));
        end
        X;

% Interdependencies
mutexcl = randperm(2,1);
  switch mutexcl
```

```
    case 1
        if any(X(1,:))
            X(2,:)=0;
        end
    case 2
        if any(X(2,:))
            X(1,:)=0;
        end
  end

  % Fitness Calculation
        Q = X.*B;
        TF = sum(sum(Q));
        Y=X;
        for row = 1:P
            ic=find(Y(row,:));
            if ~isempty(ic)
              Y(row,ic:ic+D(row)-1)=ones(1,D(row));
            end
        end
        Y;

        R = ones(P,T,K);
        for k = 1:K
          for j = 1:P
              R(j,:,k) = a(k,j);
          end
           RR(:,:,k) = R(:,:,k).*Y;
           CRR(:,:,k) = sum(RR(:,:,k));
        end

        if ((max(CRR(:,:,1))>r)||(max(CRR(:,:,2))>r))
            continue;
        else
            break;
        end
    end

  ResultC{i} = X;
  ResultBC{i} = TF;
end

  ResultM = cat(3, ResultC{:});
  ResultTF = cat(3, ResultBC{:});

    [val,idx] = max(ResultTF);
    BestSol = ResultM(:,:,idx)
    BestFitness = val
    newResultM = ResultM;

% Initialize Best Profit Record
BestProfits = zeros(MaxIt,1);
```

```
                                %%%%% TLBO Main Loop %%%%%

for it=1:MaxIt


                               %%% Teacher Phase Starts %%%

    Teacher = BestSol;
while(1)
selectedrows = randperm(P, TSC);  %two random rows
%copy selected rows of selected page to all pages:
newResultM(selectedrows, :, ☺ = repmat(Teacher(selectedrows, ☺, 1, 1,
size(newResultM, 3));
newResultM;


for i=1:N
Q = newResultM(:,:,i).*B;
TF = sum(sum(Q));
Y=newResultM(:,:,i);
    for row = 1:P
            ic=find(Y(row,☺);
            if ~isempty(ic)
              Y(row,ic:ic+D(row)-1)=ones(1,D(row));
            end
    end
Y;

R = ones(P,T,K);
        for k = 1:K
          for j = 1:P
              R(j,:,k) = a(k,j);
          end
           RR(:,:,k) = R(:,:,k).*Y;
           CRR(:,:,k) = sum(RR(:,:,k));
        end
 ResultC1{i} = CRR(:,:,1);
 ResultC2{i} = CRR(:,:,2);
 ResultTF1{i} = TF;
end
ResultCC1 = cat(3, ResultC1{:});
ResultCC2 = cat(3, ResultC2{:});
newResultTF = cat(3, ResultTF1{:});

W = max(ResultCC1);
Z = max(ResultCC2);

        if ((max(W)>r)||(max(Z)>r))
            continue;
        else
            break;
        end
end
newResultM;
newResultTF;

% Compare and Count Number of iterations for convergence

for i=1:N
```

```
    if newResultTF(i) > ResultTF(i)
      ResultM(:,:,i)  = newResultM(:,:,i);
        if newResultTF(i) > BestFitness
            Itr_4_con = it;
        end

    end
end

% Compare and Update Solutions

for i=1:N
    if newResultTF(i) > ResultTF(i)
      ResultM(:,:,i)  = newResultM(:,:,i);
        if newResultTF(i) > BestFitness
            BestFitness = newResultTF(i);
            BestSol = newResultM(:,:,i);
        end
    end
end
ResultM;
newResultM = ResultM;

for i=1:N
Q = newResultM(:,:,i).*B;
TF = sum(sum(Q));
ResultTF1{i} = TF;
end
ResultTF = cat(3, ResultTF1{:});
newResultTF = ResultTF;


                           %%% Student Phase Starts %%%

while(1)

for I = 1:N
    idx = randperm(P,SSC);       % selecte two rows randomly

    while(1)
       matrices = randperm(N,1);  % select one matrix randomly
       if matrices == i
            continue
       else
            break
       end

        if newResultTF(i) < newResultTF(matrices)
            newResultM(idx,:,i) = newResultM(idx,:,matrices);    % replace random
rows in selected matrix
        else

        end

    end

end
newResultM;
```

```
for i=1:N
Q = newResultM(:,:,i).*B;
TF = sum(sum(Q));
Y=newResultM(:,:,i);
    for row = 1:P
            ic=find(Y(row,☺);
            if ~isempty(ic)
              Y(row,ic:ic+D(row)-1)=ones(1,D(row));
            end
    end
Y;
R = ones(P,T,K);
        for k = 1:K
          for j = 1:P
              R(j,:,k) = a(k,j);
          end
           RR(:,:,k) = R(:,:,k).*Y;
           CRR(:,:,k) = sum(RR(:,:,k));
        end

 ResultC1{i} = CRR(:,:,1);
 ResultC2{i} = CRR(:,:,2);
 ResultTF1{i} = TF;
end
ResultCC1 = cat(3, ResultC1{:});
ResultCC2 = cat(3, ResultC2{:});
newResultTF = cat(3, ResultTF1{:});

W = max(ResultCC1);
Z = max(ResultCC2);

        if ((max(W)>r)||(max(Z)>r))
            continue;
        else
            break;
        end

end
newResultM;
newResultTF;

% Compare and Count Number of iterations for convergence

for i=1:N
    if newResultTF(i) > ResultTF(i)
      ResultM(:,:,i)  = newResultM(:,:,i);
        if newResultTF(i) > BestFitness
            Itr_4_con = it;
        end

    end
end

% Compare and Update Solutions

for i=1:N
    if newResultTF(i) > ResultTF(i)
      ResultM(:,:,i)  = newResultM(:,:,i);
        if newResultTF(i) > BestFitness
```

```
            BestFitness = newResultTF(i);
            BestSol = newResultM(:,:,i);
        end
    end
end
ResultM;
newResultM = ResultM;

for i=1:N
Q = newResultM(:,:,i).*B;
TF = sum(sum(Q));
ResultTF1{i} = TF;
end
ResultTF = cat(3, ResultTF1{:});
newResultTF = ResultTF;


                    %%% Self Study Phase (Mutation) Starts %%%

while(1)

for i=1:N
    rowa = randperm(P,SSM);
    rowb = flip(rowa,2);
    newResultM([rowb],:,i) = newResultM([rowa],:,i);

    for row = 1:P
            ic=find(newResultM(row,T-D(row)+2:T,i));
            if ~isempty(ic)
              newResultM(row,:,i)=zeros(1,T);
            end
    end

end
newResultM;

% Interdependencies
mutexcl = randperm(2,1);
  switch mutexcl
    case 1
        if any(newResultM(1,:,:))
            newResultM(2,:,:)=0;
        end
    case 2
        if any(newResultM(2,:,:))
            newResultM(1,:,:)=0;
        end
  end

  % Fitness Calculation

for i=1:N
Q = newResultM(:,:,i).*B;
TF = sum(sum(Q));
Y=newResultM(:,:,i);
    for row = 1:P
            ic=find(Y(row,:));
            if ~isempty(ic)
              Y(row,ic:ic+D(row)-1)=ones(1,D(row));
            end
        end
Y;
R = ones(P,T,K);
        for k = 1:K
          for j = 1:P
              R(j,:,k) = a(k,j);
          end
           RR(:,:,k) = R(:,:,k).*Y;
           CRR(:,:,k) = sum(RR(:,:,k));
        end
 ResultC1{i} = CRR(:,:,1);
 ResultC2{i} = CRR(:,:,2);
 ResultTF1{i} = TF;
end
ResultCC1 = cat(3, ResultC1{:});
ResultCC2 = cat(3, ResultC2{:});
newResultTF = cat(3, ResultTF1{:});

W = max(ResultCC1);
Z = max(ResultCC2);

        if ((max(W)>r)||(max(Z)>r))
            continue;
        else
            break;
        end

end

newResultM;
newResultTF;

% Compare and Count Number of iterations for convergence

for i=1:N
    if newResultTF(i) > ResultTF(i)
      ResultM(:,:,i)  = newResultM(:,:,i);
        if newResultTF(i) > BestFitness
            Itr_4_con = it;
        end

    end
end

% Compare and Update Solutions
for i=1:N
    if newResultTF(i) > ResultTF(i)
      ResultM(:,:,i)  = newResultM(:,:,i);
        if newResultTF(i) > BestFitness
            BestFitness = newResultTF(i);
            BestSol = newResultM(:,:,i);
        end
    end
end
ResultM;
newResultM = ResultM;

for i=1:N
```

```
Q = newResultM(:,:,i).*B;
TF = sum(sum(Q));
ResultTF1{i} = TF;
end
ResultTF = cat(3, ResultTF1{:});
newResultTF = ResultTF;

BestFitness;
BestSol;

% Store Record for Current Iteration
    BestProfits(it) = BestFitness;

    % Show Iteration Information
    disp(['Iteration ' num2str(it) ': Best Expected Profit = '
num2str(BestProfits(it))]);
end

%% Results %%

figure;
fig = plot(BestProfits, 'LineWidth', 2);
% semilogy(BestProfits, 'LineWidth', 2);
xlabel('Iteration');
ylabel('Best Expected Profit');
grid on;
saveas(fig,fig_name);
BestFitness
BestSol;
Itr_4_con

TLBO_OP = {'Best Excepted Profit','No. of iterations to converge';
BestFitness,Itr_4_con};
xlswrite(filename, TLBO_OP, 1);
xlswrite(filename, BestSol, 2);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                   %%%%%%%%%% 2. MATLAB Codes for TS %%%%%%%%%%


clc
clear all

load lowcomplex-3-1.mat
filename = 'TS-LCD-3-1-E3-R5.xls';
fig_name = 'TS-LCD-3-1-E3-R5.jpg';
TLC = 0.6;            % Tabu length coefficient
UB = (T+1)-D ;        % Upper bound on start time of projects
Itr_4_con = 0;       % Number of iterations for convergence
                           %% Tabu Search Parameters %%


MaxIt=100;                      % Maximum Number of Iterations

TL=round(TLC*length(D));        % Tabu Length


                             %% TS Initialization %%

while(1)

sol = arrayfun( @(x) randi([0,x]) , UB );    % initial solution

X = zeros(length(D),T);
       for i = 1:length(D)
           ic=find(sol(i));
            if ~isempty(ic)
              X(i,sol(i))=1;
            end
       end

X;

% Interdependencies
mutexcl = randperm(2,1);
  switch mutexcl
    case 1
        if any(X(1,:))
            X(2,:)=0;
        end
    case 2
        if any(X(2,:))
            X(1,:)=0;
        end
  end

  % Fitness Calculation
        Q = X.*B;
        TF = sum(sum(Q));
        Y=X;
        for row = 1:length(D)
            ic=find(Y(row,:));
            if ~isempty(ic)
              Y(row,ic:ic+D(row)-1)=ones(1,D(row));
            end
        end
        Y;
        R = ones(P,T,K);
        for k = 1:K
          for j = 1:P
              R(j,:,k) = a(k,j);
          end
           RR(:,:,k) = R(:,:,k).*Y;
           CRR(:,:,k) = sum(RR(:,:,k));
        end
        if ((max(CRR(:,:,1))>r)||(max(CRR(:,:,2))>r))
            continue;
        else
            break;
        end
    end
  sol;

  % Initialize Best Solution Ever Found
BestSol=sol
BestFitness = TF

% Array to Hold Tabu Moves (Tabu List)
tabulist = zeros(TL,length(D));
newtabulist = zeros(TL,length(D));
tabulist(1,:) = sol;

% Array to Hold Best Profits
BestProfit=zeros(MaxIt,1);
```

%%%%% Tabu Search Main Loop %%%%%

```
for it=1:MaxIt
 sol;

% Neighborhood Generation

while(1)

C=zeros(length(D));         %create an empty matrix
for i = 1:length(D)
    C(i,:) = sol;
    C(i,i) = randi(UB(i));
    ResultC{i} = C(i,:);
end
NS = cat(3, ResultC{:});     % Neighborhood Solutions

% Generation of Neighborhood Solutions Matrices

CM = zeros(length(D),T,length(D));     %create an empty matrix
for i = 1:length(D)
   for j = 1:length(D)
      ic=find(NS(:,j,i));
      if ~isempty(ic)
          CM(j,NS(:,j,i),i)=1;
      end
   end
   ResultCM{i} = CM(:,:,i);
end
NSM = cat(3, ResultCM{:});

% Interdependencies
mutexcl = randperm(2,1);
  switch mutexcl
    case 1
        if any(NSM(1,:,:))
            X(2,:,:)=0;
        end
    case 2
        if any(X(2,:,:))
            X(1,:,:)=0;
        end
  end

  % Fitness Calculation
for i=1:length(D)
Q = NSM(:,:,i).*B;
TF = sum(sum(Q));
Y=NSM(:,:,i);
    for row = 1:length(D)
            ic=find(Y(row,:));
            if ~isempty(ic)
              Y(row,ic:ic+D(row)-1)=ones(1,D(row));
            end
    end
Y;
R = ones(P,T,K);
for k = 1:K
  for j = 1:P
      R(j,:,k) = a(k,j);
  end
   RR(:,:,k) = R(:,:,k).*Y;
   CRR(:,:,k) = sum(RR(:,:,k));
end
ResultC1{i} = CRR(:,:,1);
 ResultC2{i} = CRR(:,:,2);
 ResultTF1{i} = TF;
end
ResultCC1 = cat(3, ResultC1{:});
ResultCC2 = cat(3, ResultC2{:});
newResultTF = cat(3, ResultTF1{:});

W = max(ResultCC1);
Z = max(ResultCC2);

        if ((max(W)>r)||(max(Z)>r))
            continue;
        else
            break;
        end
end

NS;
NSM;
```

```
newResultTF;

% Find new best solution
    [val,idx] = sort(newResultTF,'descend');
    for i = 1:length(D)
        if ~ismember(NS(:,:,idx(i)),tabulist,'rows')
            bestnewsol = NS(:,:,idx(i)) ;
            bestnewresultTF = newResultTF(idx(i));
            break
        end
    end
    bestnewsol;
    bestnewresultTF;

% Update Tabu List

    for i=1:TL-1
        newtabulist(i+1,:) = tabulist(i,:);
    end
    newtabulist(1,:) = bestnewsol;
    tabulist = newtabulist;
    tabulist;

% Count Number of iterations for convergence
    if bestnewresultTF > BestFitness
      Itr_4_con = it;
    end

% Compare and Update Solutions
    if bestnewresultTF >= BestFitness
        BestFitness = bestnewresultTF;
            BestSol = bestnewsol;
    end


% Update Current Solution
    sol=bestnewsol;

BestFitness;
BestSol;

% Store Record for Current Iteration
    BestProfit(it) = BestFitness;

% Show Iteration Information
    disp(['Iteration   '   num2str(it)   ':   Best   Expected   Profit   =   '
num2str(BestProfit(it))]);

end

  %% Results %%

figure;
fig = plot(BestProfit, 'LineWidth', 2);
% semilogy(BestProfit, 'LineWidth', 2);
xlabel('Iteration');
ylabel('Best Expected Profit');
grid on;
saveas(fig,fig_name);
```

```
BestFitness
BestSol
tabulist
Itr_4_con

TS_OP = {'Best Excepted Profit','No. of iterations to converge';
BestFitness,Itr_4_con};
xlswrite(filename, TS_OP, 1);
xlswrite(filename, BestSol, 2);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

             %%%%%%%%%% 3. MATLAB Codes for Hybrid TLBO-TS %%%%%%%%%%


clc
clear all

                               %% TLBO Parameters %%


MaxIt = 100;                    % Maximum Number of Iterations
N = 20;                          % Population size

load lowcomplex-3-1.mat
filename = 'HTLBO-LCD-3-1-E1-AL.xls';
fig_name = 'HTLBO-LCD-3-1-E1-AL.jpg';
TSC = round(0.4*P);              % Teacher student crossover coefficient
SSC = round(0.3*P);              % Student student crossover coefficient
SSM = round(0.3*P);              % Self-study mutation coefficient
TLC = 0.4;                       % Tabu length cofficient
UB = (T+1)-D;                    % Upper bound on start time of projects
Itr_4_con = 0;                   % Number of iterations for convergence



                           %% Tabu Search Parameters %%


TL=round(TLC*length(D));        % Tabu Length

% % Array to Hold Tabu Moves (Tabu List)
tabulist = zeros(TL,length(D));
newtabulist = zeros(TL,length(D));


                       %% Hybrid TLBO-TS Algorithm Starts %%

excludedcount = D-1;
X = zeros(P,T);
Result = cell(1, N);
ResultB = cell(1, N);
for i = 1:N
    while(1)
        for row = 1:P
            rv = [1, zeros(1, T - excludedcount(row) - 1)];
            X(row, 1 : (T - excludedcount(row))) = rv(randperm(numel(rv)));
        end
        X;
% PC = randperm([P] , 2);
% X([PC],:) = 0;

% Interdependencies
mutexcl = randperm(2,1);
  switch mutexcl
    case 1
        if any(X(1,:))
            X(2,:)=0;
        end
    case 2
        if any(X(2,:))
            X(1,:)=0;
        end
  end

  % Fitness Calculation
        Q = X.*B;
        TF = sum(sum(Q));
        Y=X;
        for row = 1:P
            ic=find(Y(row,:));
            if ~isempty(ic)
              Y(row,ic:ic+D(row)-1)=ones(1,D(row));
            end
        end
        Y;
        R = ones(P,T,K);
        for k = 1:K
          for j = 1:P
              R(j,:,k) = a(k,j);
          end
           RR(:,:,k) = R(:,:,k).*Y;
           CRR(:,:,k) = sum(RR(:,:,k));
        end
        if ((max(CRR(:,:,1))>r)||(max(CRR(:,:,2))>r))
            continue;
        else
            break;
        end
    end
  ResultC{i} = X;
  ResultBC{i} = TF;
end
  ResultM = cat(3, ResultC{:});
  ResultTF = cat(3, ResultBC{:});

    [val,idx] = max(ResultTF);
    BestSol = ResultM(:,:,idx)
    BestFitness = val
    newResultM = ResultM;

% Initialize Best Profit Record
BestProfits = zeros(MaxIt,1);


                            %%%%% TLBO Main Loop %%%%%

 for it=1:MaxIt


                           %%% Teacher Phase Starts %%%

  Teacher = BestSol;
while(1)
selectedrows = randperm(P, TSC);  %two random rows
%copy selected rows of selected page to all pages:
```

```
newResultM(selectedrows, :, :) = repmat(Teacher(selectedrows, :), 1, 1,
size(newResultM, 3));
newResultM;


for i=1:N
Q = newResultM(:,:,i).*B;
TF = sum(sum(Q));
Y=newResultM(:,:,i);
    for row = 1:P
            ic=find(Y(row,:));
            if ~isempty(ic)
              Y(row,ic:ic+D(row)-1)=ones(1,D(row));
            end
    end
Y;
R = ones(P,T,K);
for k = 1:K
  for j = 1:P
      R(j,:,k) = a(k,j);
  end
   RR(:,:,k) = R(:,:,k).*Y;
   CRR(:,:,k) = sum(RR(:,:,k));
end
ResultC1{i} = CRR(:,:,1);
 ResultC2{i} = CRR(:,:,2);
 ResultTF1{i} = TF;
end
ResultCC1 = cat(3, ResultC1{:});
ResultCC2 = cat(3, ResultC2{:});
newResultTF = cat(3, ResultTF1{:});

W = max(ResultCC1);
Z = max(ResultCC2);

        if ((max(W)>r)||(max(Z)>r))
            continue;
        else
            break;
        end
end
newResultM;
newResultTF;

% Compare and Count Number of iterations for convergence

for i=1:N
    if newResultTF(i) > ResultTF(i)
      ResultM(:,:,i)  = newResultM(:,:,i);
        if newResultTF(i) > BestFitness
            Itr_4_con = it;
        end

    end
end

% Compare and Update Solutions

for i=1:N
```

```
    if newResultTF(i) > ResultTF(i)
      ResultM(:,:,i)  = newResultM(:,:,i);
        if newResultTF(i) > BestFitness
            BestFitness = newResultTF(i);
            BestSol = newResultM(:,:,i);
        end
    end
end
ResultM;
newResultM = ResultM;

for i=1:N
Q = newResultM(:,:,i).*B;
TF = sum(sum(Q));
ResultTF1{i} = TF;
end
ResultTF = cat(3, ResultTF1{:});
newResultTF = ResultTF;


                           %%% Student Phase Starts %%%

while(1)

for i = 1:N
    idx = randperm(P,SSC);       % selecte two rows randomly

    while(1)
       matrices = randperm(N,1);  % select one matrix randomly
       if matrices == i
            continue
       else
            break
       end

        if newResultTF(i) < newResultTF(matrices)
            newResultM(idx,:,i) = newResultM(idx,:,matrices);    % replace random
rows in selected matrix
        else

        end

    end

end
newResultM;

for i=1:N
Q = newResultM(:,:,i).*B;
TF = sum(sum(Q));
Y=newResultM(:,:,i);
    for row = 1:P
            ic=find(Y(row,:));
            if ~isempty(ic)
              Y(row,ic:ic+D(row)-1)=ones(1,D(row));
            end
    end
Y;
R = ones(P,T,K);
```

```
for k = 1:K
  for j = 1:P
      R(j,:,k) = a(k,j);
  end
   RR(:,:,k) = R(:,:,k).*Y;
   CRR(:,:,k) = sum(RR(:,:,k));
end
ResultC1{i} = CRR(:,:,1);
 ResultC2{i} = CRR(:,:,2);
 ResultTF1{i} = TF;
end
ResultCC1 = cat(3, ResultC1{:});
ResultCC2 = cat(3, ResultC2{:});
newResultTF = cat(3, ResultTF1{:});

W = max(ResultCC1);
Z = max(ResultCC2);

        if ((max(W)>r)||(max(Z)>r))
            continue;
        else
            break;
        end

end
newResultM;
newResultTF;

% Compare and Count Number of iterations for convergence

for i=1:N
    if newResultTF(i) > ResultTF(i)
      ResultM(:,:,i)  = newResultM(:,:,i);
        if newResultTF(i) > BestFitness
            Itr_4_con = it;
        end

    end
end

% Compare and Update Solutions

for i=1:N
    if newResultTF(i) > ResultTF(i)
      ResultM(:,:,i)  = newResultM(:,:,i);
        if newResultTF(i) > BestFitness
            BestFitness = newResultTF(i);
            BestSol = newResultM(:,:,i);
        end
    end
end
ResultM;
newResultM = ResultM;

for i=1:N
Q = newResultM(:,:,i).*B;
TF = sum(sum(Q));
ResultTF1{i} = TF;
end
```

```
ResultTF = cat(3, ResultTF1{:});
newResultTF = ResultTF;



                    %%% Self Study Phase (Mutation) Starts %%%

while(1)

for i=1:N
    rowa = randperm(P,SSM);
    rowb = flip(rowa,2);
    newResultM([rowb],:,i) = newResultM([rowa],:,i);

    for row = 1:P
            ic=find(newResultM(row,T-D(row)+2:T,i));
            if ~isempty(ic)
              newResultM(row,:,i)=zeros(1,T);
            end
    end

end
newResultM;

% Interdependencies
mutexcl = randperm(2,1);
  switch mutexcl
    case 1
        if any(newResultM(1,:,:))
            newResultM(2,:,:)=0;
        end
    case 2
        if any(newResultM(2,:,:))
            newResultM(1,:,:)=0;
        end
  end

  % Fitness Calculation
for i=1:N
Q = newResultM(:,:,i).*B;
TF = sum(sum(Q));
Y=newResultM(:,:,i);
    for row = 1:P
            ic=find(Y(row,:));
            if ~isempty(ic)
              Y(row,ic:ic+D(row)-1)=ones(1,D(row));
            end
    end
Y;
R = ones(P,T,K);
for k = 1:K
  for j = 1:P
      R(j,:,k) = a(k,j);
  end
   RR(:,:,k) = R(:,:,k).*Y;
   CRR(:,:,k) = sum(RR(:,:,k));
end
ResultC1{i} = CRR(:,:,1);
 ResultC2{i} = CRR(:,:,2);
 ResultTF1{i} = TF;
end
ResultCC1 = cat(3, ResultC1{:});
ResultCC2 = cat(3, ResultC2{:});
newResultTF = cat(3, ResultTF1{:});

W = max(ResultCC1);
Z = max(ResultCC2);

        if ((max(W)>r)||(max(Z)>r))
            continue;
        else
            break;
        end

end

newResultM;
newResultTF;

% Compare and Count Number of iterations for convergence

for i=1:N
    if newResultTF(i) > ResultTF(i)
      ResultM(:,:,i)  = newResultM(:,:,i);
        if newResultTF(i) > BestFitness
            Itr_4_con = it;
        end

    end
end

% Compare and Update Solutions

for i=1:N
    if newResultTF(i) > ResultTF(i)
      ResultM(:,:,i)  = newResultM(:,:,i);
        if newResultTF(i) > BestFitness
            BestFitness = newResultTF(i);
            BestSol = newResultM(:,:,i);
        end
    end
end
ResultM;
newResultM = ResultM;

BestFitness;
BestSol;

sol = zeros(1,size(BestSol,1));

   for i=1:size(BestSol,1)
        icx = find(BestSol(i,:));
       if ~isempty(icx)
       sol(i) = icx;
       end
   end
   sol
   BestSolTS = sol;
```

```
tabulist(1,:) = sol;

                        %%%%% Tabu Search Main Loop %%%%%

                          %% Neighborhood Generation %%

while(1)

C=zeros(length(D));         %create an empty matrix
ResultC = [];
for i = 1:length(D)
    C(i,:) = sol;
    C(i,i) = randi(UB(i));
    ResultC{i} = C(i,:);
end
NS = cat(3, ResultC{:});     % Neighborhood Solutions

% Generation of Neighborhood Solutions Matrices

CM = zeros(length(D),T,length(D));     %create an empty matrix
for i = 1:length(D)
   for j = 1:length(D)
      ic=find(NS(:,j,i));
      if ~isempty(ic)
          CM(j,NS(:,j,i),i)=1;
      end
   end
   ResultCM{i} = CM(:,:,i);
end
NSM = cat(3, ResultCM{:});

% Interdependencies
mutexcl = randperm(2,1);
  switch mutexcl
    case 1
        if any(NSM(1,:,:))
            X(2,:,:)=0;
        end
    case 2
        if any(X(2,:,:))
            X(1,:,:)=0;
        end
  end

  % Fitness Calculation

for i=1:length(D)
Q = NSM(:,:,i).*B;
TF = sum(sum(Q));
Y=NSM(:,:,i);
    for row = 1:length(D)
            ic=find(Y(row,:));
            if ~isempty(ic)
              Y(row,ic:ic+D(row)-1)=ones(1,D(row));
            end
    end
Y;
R = ones(P,T,K);
for k = 1:K
  for j = 1:P
      R(j,:,k) = a(k,j);
  end
   RR(:,:,k) = R(:,:,k).*Y;
   CRR(:,:,k) = sum(RR(:,:,k));
end
ResultC1{i} = CRR(:,:,1);
 ResultC2{i} = CRR(:,:,2);
 ResultTF2{i} = TF;
end
ResultCC1 = cat(3, ResultC1{:});
ResultCC2 = cat(3, ResultC2{:});
newResultTF_TS = cat(3, ResultTF2{:});

W = max(ResultCC1);
Z = max(ResultCC2);

        if ((max(W)>r)||(max(Z)>r))
            continue;
        else
            break;
        end
end


NS;
NSM;
newResultTF_TS;
ResultCC1;
ResultCC2;

% Find new best solution
    [val,idx] = sort(newResultTF_TS,'descend');
    for i = 1:length(D)
        if ~ismember(NS(:,:,idx(i)),tabulist,'rows')
            bestnewsol = NS(:,:,idx(i)) ;
            bestnewresultTF = newResultTF_TS(idx(i));
            break
        end
    end
    bestnewsol;
    bestnewresultTF;
    BestFitness;
% Update Tabu List

    for i=1:TL-1
        newtabulist(i+1,:) = tabulist(i,:);
    end
    newtabulist(1,:) = bestnewsol;
    tabulist = newtabulist;
    tabulist;

 % Again Count Number of iterations for convergence
    if bestnewresultTF > BestFitness
      Itr_4_con = it;
    end

 % Compare and Update Solutions
    if bestnewresultTF >= BestFitness
        BestFitness = bestnewresultTF;
            BestSolTS = bestnewsol;
    end

BestSolTS_matrix = zeros(length(D),T);
       for i = 1:length(D)
           ic=find(BestSolTS(i));
            if ~isempty(ic)
              BestSolTS_matrix(i,BestSolTS(i))=1;
            end
       end
BestSolTS_matrix;
BestSol = BestSolTS_matrix;

BestFitness;
BestSol;

% Store Record for Current Iteration
    BestProfit(it) = BestFitness;

% Show Iteration Information
    disp(['Iteration   '   num2str(it)   ':   Best   Expected   Profit   =   '
num2str(BestProfit(it))]);

end

  %% Results %%

figure;
fig = plot(BestProfit, 'LineWidth', 2);
% semilogy(BestProfit, 'LineWidth', 2);
xlabel('Iteration');
ylabel('Best Expected Profit');
grid on;
saveas(fig,fig_name);
BestFitness
BestSol;
tabulist;
Itr_4_con

HTLBO_OP   =   {'Best   Excepted   Profit','No.   of   iterations   to   converge';
BestFitness,Itr_4_con};
xlswrite(filename, HTLBO_OP, 1);
xlswrite(filename, BestSol, 2)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

List of Publications

International Journals:

[1] **Kumar, M.**, Mittal, M. L., Soni, G., & Joshi, D. (2018). A hybrid TLBO-TS algorithm for integrated selection and scheduling of projects. *Computers & Industrial Engineering, 119*, 121-130. (SCI Indexed – Impact Factor: 3.195)

[2] **Kumar, M.**, Mittal, M. L., Soni, G., & Joshi, D. An NSTLBO algorithm for bi-objective integrated project selection and scheduling problem. (To be communicated)

[3] **Kumar, M.**, Mittal, M. L., Soni, G., & Joshi, D. An improved NSTLBO algorithm for integrated selection and scheduling of interdependent projects. (To be communicated)

[4] **Kumar, M.**, Mittal, M. L., Soni, G., & Joshi, D. Integrated project selection and scheduling of projects: a literature review. (Under Preparation)

International Conferences:

[1] **Kumar, M.**, Mittal, M. L., Soni, G., & Joshi, D. (2019). A Tabu Search Algorithm for Simultaneous Selection and Scheduling of Projects. In *Harmony Search and Nature Inspired Optimization Algorithms* (pp. 1111-1121). Springer, Singapore. (SCOPUS Indexed)

[2] **Kumar, M.**, Mittal, M. L., Soni, G., & Joshi, D. (2019). Selection and scheduling of interdependent projects using a modified genetic algorithm. IEOM-2019, Bangkok (Accepted for Scopus Proceedings)

[3] **Kumar, M.**, Mittal, M. L., Soni, G., & Joshi, D. (2016). Project portfolio selection and scheduling problem (PPSSP): A literature review. Proceedings of SOM 2017 December 22-24, 2017, Department of Management Studies, IIITM Gwalior, India

Author’s Biographical Sketch

Name	:	Manish Kumar
Date of Birth	:	03 February, 1989
Email(s)	:	m81manishkumar@gmail.com, 2014rme9540@mnit.ac.in
Mobile No.(s)	:	+91-9610700081, +91-8233461886
Qualifications	:	B. E. (Mechanical Engineering),
		Maharishi Dayanand University, Rohtak (2010)
		M. Tech. (Manufacturing System Engineering)
		Malaviya National Institute of Technology, Jaipur (2013)
Areas of interest	:	Project management & scheduling, Facility layout planning,
		Meta-heuristics and Artificial intelligence applications