12-2016

CONSTRAINED MULTI-GROUP PROJECT ALLOCATION USING MAHALANOBIS DISTANCE

Abdulaziz Saud Alkabaa
University of Tennessee, Knoxville, aalkabaa@vols.utk.edu

Recommended Citation

Alkabaa, Abdulaziz Saud, "CONSTRAINED MULTI-GROUP PROJECT ALLOCATION USING MAHALANOBIS DISTANCE. " PhD diss., University of Tennessee, 2016.
https://trace.tennessee.edu/utk_graddiss/4121

To the Graduate Council:

I am submitting herewith a dissertation written by Abdulaziz Saud Alkabaa entitled "CONSTRAINED MULTI-GROUP PROJECT ALLOCATION USING MAHALANOBIS DISTANCE." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Industrial Engineering.

Alberto Garcia, Major Professor

We have read this dissertation and recommend its acceptance:

Hamparsum Bozdogan, Rapinder Sawhney, James L. Simonton

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

CONSTRAINED MULTI-GROUP PROJECT ALLOCATION USING MAHALANOBIS DISTANCE

A Dissertation Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Abdulaziz Saud Alkabaa
December 2016

DEDICATION

I dedicate this dissertation to my parents and family for their endless love and scarifies and to every torchbearer.

ACKNOWLEDGEMENTS

Foremost, I would like to thank our God Almighty for his guidance and generosity throughout my life. If not for His constant blessings, forgiveness, and grace, I would not have been able to accomplish my lifelong goal of continuing my education and earning a Ph.D. degree in a scientific discipline.

I would like to thank my beloved parents, Saud Alkabaa and Nadiah Montaser, for their praying, continuous support, and advising that enabled me to work hard and overcome many obstacles along my educational and life. A special thank to my parents in law, Shikh Ibrahim Alkharan and Latifa Alhazaa, for their praying and endless support.

A special thank goes to my loving wife, Aljawhara Alkharan, and my kids, Aljoori, Nawaf, and Jude, from the bottom of my heart who have been supporting and encouraging me tremendously and patiently to progress each step in this strenuous journey of mine. They were and are very understanding and cooperating whenever I succumb to irascible periods as a result of nervousness from both life and school. Indeed, they deserve tremendous credit for all my achievements that I obtained.

The first person that I would like to thank at the University of Tennessee is my academic and thesis supervisor, Dr. Alberto Garcia, who has been helping me before and after I have been admitted to the University of Tennessee. I am indebted to Dr. Garcia for his valuable advice in my education and life. Dr. Garcia has spent a significant amount of time and effort to increase my educational knowledge and skills. Working under the supervision of Dr. Garcia has undoubtedly provided me with an extraordinary contribution in my academic profession and stimulated my thinking in solving many

problems professionally. His intuitive suggestions will never be forgotten and will be engraved in my heart.

I especially want to express my sincerest gratitude to Dr. Hamparsum Bozdogan, who ingrained me the passion of Statistics. Dr. Bozdogan has supported and encouraged me whenever I talked with him about my dissertation and any new ideas.

I am very grateful to Dr. Rapinder Sawhney and Dr. James L. Simonton, for serving on my dissertation committee and for providing me with several ides that helped me through completing this dissertation.

I also want to thank my B.S. and M.S. supervisor, Dr. Mustafa Alidrisi, for his support and inspiration during my doctoral education.

Finally, my sincere appreciation and gratefulness goes to my sponsor, Saudi Arabian Ministry of Education and Saudi Arabian Cultural Mission to the United States, for sponsoring and supervising my doctoral progress. Without their efforts and financial support, this work could not have been completed.

ABSTRACT

Optimal allocation is one of the most active research areas in operation research using binary integer variables. The allocation of multi constrained projects among several options available along a given planning horizon is an especially significant problem in the general area of item classification. The main goal of this dissertation is to develop an analytical approach for selecting projects that would be most attractive from an economic point of view to be developed or allocated among several options, such as in-house engineers and private contractors (in transportation projects). A relevant limiting resource in addition to the availability of funds is the in-house manpower availability.

In this thesis, the concept of Mahalanobis distance (MD) will be used as the classification criterion. This is a generalization of the Euclidean distance that takes into account the correlation of the characteristics defining the scope of a project. The desirability of a given project to be allocated to an option is defined in terms of its MD to that particular option. Ideally, each project should be allocated to its closest option. This, however, may not be possible because of the available levels of each relevant resource.

The allocation process is formulated mathematically using two Binary Integer Programming (BIP) models. The first formulation maximizes the dollar value of benefits derived by the traveling public from those projects being implemented subject to a budget, total sum of MD, and in-house manpower constraints. The second formulation minimizes the total sum of MD subject to a budget and the in-house manpower constraints.

The proposed solution methodology for the BIP models is based on the branch-and-bound method. In particular, one of the contributions of this dissertation is the development of a strategy for branching variables and node selection that is consistent with allocation priorities based on MD to improve the branch-and-bound performance level as well as handle a large scale application. The suggested allocation process includes: (a) multiple allocation groups; (b) multiple constraints; (c) different BIP models. Numerical experiments with different projects and options are considered to illustrate the application of the proposed approach.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I
INTRODUCTION

Undoubtedly, the most significant factor affecting research and development (R&D) project selection and resource allocation in any organization (such as a government agency or a company) is the approach to decision-making. In R&D research perspective, successful decisions depend on the selection methodology as well as allocation criterion for prioritizing projects [1]. Therefore, a variety of methods and techniques have been developed and documented in the relevant literature for the last 50 years [1, 2]. A lack of methodology for making an optimal decision among competitive choices can lead to large organizational obligations and other serious manifold risks in business environments. Hence, project selection can be related to competitive markets of governmental agencies and companies [3].

Due to the limitation or scarcity of significant resources, numerous organizations do not have the capability to construct all engineering activities or projects in a given planning horizon. As a result, many organizations outsource a percentage of projects to one of several contractors or agencies. In this case, the classical problem of a single project allocation is neither applicable nor appropriate and thus will be considered as a multi-group project allocation decision making problem. Although this type of problem is very common in the real world, to the best of my knowledge, it has not been addressed in the research literature, resulting in our attention being focused on this critical issue.

The problem of developing an analytical approach for selecting and allocating the most suitable projects among a group of candidate projects, from an economic point of view and subject to limited resources, is a challenge faced by every decision maker. Typically, a project that generates high levels of benefits without exceeding available resources are selected for funding. This dissertation is concerned with developing an effective analytical decision-support approach for selecting and allocating projects among a group of different options in such a way that a specified performance measure is optimized, subject to some relevant constraints. Experiments show that this methodology provides precise allocation and great scalability with respect to variations in the size of instances processed and the degree of resource constraints.

Background

Many industrialized countries spend more than 2% of their Gross Domestic Product (GDP) on R&D activities [4]. For example, in 2013, the United States spent 2.742% (432.583 billion dollars) of its GDP; in 2014, Japan spent 3.583% (159.220 billion dollars); and China spent 2.046 % (344.678 billion dollars) [5]. Thus, it can be inferred that R&D is vital in the business world and is a key to many organizations' success [6]. From a R&D perspective, project selection and resource allocation are defined in terms of constrained optimization problems [1, 3, 7, 8]. Given a set of project proposals among several available projects, the goal is to develop a project selection methodology to aid a decision maker in selecting a subset of these projects subject to relevant limiting resources, such as budgets, manpower, and construction equipment, and facilities. Project selection problems may consider a variety of objectives dependeing on the decision making's purpose [3]. In most cases, the objective of this selection process's objective is to maximize benefits and minimize costs of the project [7]. To address R&D

project selection and resource-allocation problems, various methods have been utilized, including benefit measurements, mathematical programming, decision and game theory, simulation models, heuristics, and cognitive emulation [9]. Fahrni and Spätig's study [10] provides different approaches for determining the most applicable technique for a specific situation.

Mathematical programming has been widely used for selecting R&D projects. This method can be represented and solved by means of different programming models, such as Linear Programming (LP), Non-Linear Programming (NLP), Dynamic Programming (DP), Goal Programming (GP), and Stochastic Programming (SP) [9]. These models have been used in many research studies involving single or multiple objective functions. Numerous integer programming (IP) applications are derived using LP where the divisibility assumption must be relaxed. Project selection decision problems involve numerous interrelated yes-or-no decisions, where the variables might be expected integer and considered binary (0 or 1). Therefore, these types of decision problems become BIP and are widely used to aid with project selection problems [11]. Some algorithms in the literature provide solutions to IP and to Mixed Linear Integer Programming (MLIP) models, such as cutting plane and B&B algorithms. According to Taha [12, pp. 370 and 384]:

> "Although neither method is consistently effective computationally, experience shows that the B&B method is far more successful than the cutting-plane method, …. To date, and despite over 40 years of research, there does not exist a computer code that can solve ILP consistently. Nevertheless…, B&B is more reliable.

> Cutting-plane methods are generally difficult and uncertain, and the round off error presents a serious problem."

B&B algorithms are coded by all ILP commercial solvers. The project-allocation problem considered in this dissertation is a Nondeterministic Polynomial (NP)-hard problem [13]. Therefore, the execution time required to solve the problem increases exponentially as the problem enlarges [14]. The critical rules affecting the B&B's solution performances are the variable branching and node selection strategies, which can have a significant impact on the solution algorithms' efficiency [15]. Many modern IP solvers (such as CPLEX, LINDO, MATLAB, and Xpress-MP) provide users the capability of changing these strategies. Although many R&D project selection and resource-allocation problems are discussed in the literature, no research literature has been devoted to the branching strategies, which indeed can help users achieve better solutions in terms of B&B quality and performance. This absence of research led to exploring novel branching and node-selection strategies in this dissertation.

The process of allocating a project among several available options is not precisely presented the literature; however, it has been shown as a project-management decision problem of assigning several projects to project managers. The literature in this area focuses on two streams: project-assignment methodologies and criteria [16]. It can be inferred that the methodologies' performance in terms of project assignment is mainly influenced by the assignment criteria, which contribute to the project's success and effectiveness in allocating decisions. Although a potential correlation between project characteristics affects classifier's performance and accuracy, no research in the literature has addressed this specific consideration in project-assignment and R&D problems.

Instead, some R&D studies have focused on the importance the interdependence and interaction between projects or resources [17, 18]. Subsequently, a reliable classification criterion or tool to deal with correlated variables is needed to handle this issue. For this reason, the concept of Mahalanobis Distance (MD) [19] was introduced for the classification scheme to deal with similarity of an observation to the other group. Many applications-based MDs have been widely proposed and studied in the field of decision and classification problems, such as manufacturing, business, healthcare, and ecology [20]. The concept of MD classification criterion has been compared with many classifiers in several studies for the purposes of classifying correlated multivariate data, classifier accuracy, and effectiveness improvements [21-24]. Chapter 2 examines various relevant studies in R&D project selection and classification, solution methodologies, B&B strategies, and MD and their applications.

Problem Statement

This study proposes an analytical constrained multi-allocation approach that allows project managers or decision makers to group several projects into predefined categories that represent most-appropriate options for developing the projects. Specifically, the proposed procedure identifies the projects to be developed and assign them to the best fitting option. Typically, these options may correspond to doing a project in-house or outsourcing it to one of a group of private contractors, as in transportation projects. It is assumed that a panel of experts provides joint expertise to identify ideal projects, from the projects' database, for each allocation option (in-house and private contractors). In other words, the purpose of identifying these ideal projects for each corresponding option is to provide an index for each specific option.

In general, each project has a cost and a dollar value of benefits. The scope of each project is defined in terms of several factors or characteristics that may be correlated with one another. In order to identify how each project best matches each allocation option, for allocating each project to one of the options, the statistical concept known as MD, which is a generalization of the Euclidean Distance (ED) taking into account the correlation between the project characteristics, will be used as the classification criterion. In brief, MD measures the statistical distance from each project to each allocation option that has been identified by the panel of experts. The reader is referred to Chapter 3 for MD overview and applications.

Because the project-allocation problems include determining or deciding whether or not a project is selected for developing, the allocation variables are represented by only two outcomes, (0 or 1), ; that is, the choices with decision variables restricted to two values: 1 for allocating and 0 for otherwise. Consequently, the allocation process in this dissertation is formulated as a binary integer programming (BIP) model that contains only binary variables.

Two different BIP formulations are proposed to aid in assigning projects to either in-house or private contractors. The first formulation maximizes the dollar value of benefits while ensuring that projects are allocated to the best option consistent with available funds and in-house manpower. The second model allocates projects to the best options subject to available budget and in-house manpower. These BIP models' proposed solution methodology is based on the branch-and-bound (B&B) algorithm. In particular, one of this dissertation’s contributions is in proposing novel strategies for branching variables and node selection consistent with allocation priorities among different projects to improve the solution performances’ efficiency in different measurements. For illustration purposes, the proposed project’s allocation methodology and the corresponding mathematical models are applied to the data of the State Department of Transportation (DOT) to aid project-management decision makers in optimizing project allocation (to either option: in-house or several private contractors). We run numerical experiments with several different combinations of projects and options to demonstrate our allocation approach and then analyze the solution results. The application study shows that efficiency of the solution performances using the proposed B&B branching variables and node selection strategies outperforms the best of the default strategies that MATLAB computational platform provides. Moreover, this approach can improve project-allocation decisions by using the two models that provide different conclusions.

Contributions of this Study

This dissertation is intended to add the following contributions:

- Development of a MD project selection and multi-choice classification approach with multiple-resource constraints. Specifically, this new approach is proposed to improve the quality and accuracy of the best fitting allocation process. As we will show in the Literature Review, Chapter 2.
- Improve the B&B methodology significantly by formulating a novel branching strategy based on a specific variables ranking according to MD values to enhance solution performances (i.e., absolute gap between the internally calculated upper

and lower bound on the objective function, relative gap in percentage, computer run time, and number of explored nodes in the B&B tree). As we will see in the Literature Review, Chapter 2 and in the Results and analysis, Chapter 5.

Organization of the Dissertation

This dissertation is divided into seven chapters. This first chapter has introduced this study's general problem and challenges of this study. Chapter 2 contains a review of the literature pertaining to R&D project selections and resource allocations, as well as project assignments, MD classifications and applications, and branching strategies in the B&B algorithm. Chapter 3 presents the overall conceptual approach of the multi-project allocation-based on MD, including the definition of MD and current B&B variable and node selection methods. Chapter 4 discusses this study's proposed methodology, including the mathematical model formulation and the proposed branching strategy. Chapter 5 describes the computational design and application of the developed models for selecting and allocating transportation projects among multiple options. Also included is an assessment of the solution procedure's performance for the mathematical models using the B&B algorithm. Results and analysis obtained from the methodology are presented in Chapter 6. Chapter 7 includes a summary and conclusions and suggests further related-research possibilities. References and appendices are available at the end of this thesis.

CHAPTER II
LITERATURE REVIEW

The intent of this chapter is to provide an understanding of the previous research and literature relevant to project selection and allocation problems and to discover methodologies, optimization means and criteria, modeling approaches, and solution procedures that have been considered in each related study. This chapter is divided into four major sections. The first section defines the problem of interest and describes the modeling approach and the solution methodology used. The second section discusses the classification approach using the concept of MD and its applications in different disciplines. The third section presents B&B strategies that have been used and their impact on the solution performance. The last section is the conclusion, which summarizes the chapter.

Project Selection and Resource Allocation

Since Mottley and Newton in 1959 [25] proposed the concept of project selection and resource allocation, an evolutionary process has resulted in developing and reporting numerous quantitative methods and models in the literature. Although several methodologies for R&D project selection exist, a vital way of formulating scientific decision-making problems is mathematical models [26]. Moreover, the majority of the resource-allocation problems have been widely studied using mathematical programming [13].

Usually, much of a modeling technique focuses on how to effectively propose a methodological approach to selecting a set of projects from a project portfolio that meets different resource requirements in order to arrive at an optimum portfolio of R&D projects. However, less attention has been devoted to how these models can provide the best-fitting allocation as well as perform in a sophisticated situation (i.e., when numerous variables and constraints increase a problem's size). The survey study in this section shades the light on various modeling approaches with different applications in R&D project selection and resource allocation.

Loosely speaking, any project selection and allocation model depends on the type of problem. Researchers have proposed many efficient approaches. For example, Beaujon et al. [27] develop a mathematical model that is formulated by using mixed- integer programming and solving it using specific heuristic methodology to select projects to include in an R&D portfolio that provides the maximum expected benefits, subject to a several types of constraints, e.g., capital, headcount, strategic intent, etc. The paper addresses two important issues regarding the formulation of the project selection models: (1) allowing for partly funded projects in the portfolio (i.e., projects that proceed at an interval due to insufficient funding with the rest being held for future funding), resulting in solving the problem using linear programming; and (2) assigning a score to each project for the purpose of valuing projects and accounting for potential risk. This model's distinguishing factor is the set of constraints; specifically, this model allows the possessed skills of the research staff to be considered, such as research background in various categories, number of skilled staff needed to be involved in developing particular projects, and category of skills required.

Yavuz and Captain [26] propose a BIP model to select projects that maximize total net present value subject to budget constraints and other secondary conditions in the

Turkish armed forces' ten-year planning horizon. Different computational data sets were randomly created to simulate the real data and run the model for optimality and performance testing. Ten random problems were run; these problems varied from 908 projects to 1,067 projects, approximately 10,000 variables, and 2,500 constraints. CPLEX 4.0.7 software ran the simulated data models. CPLEX 6.5.3 software executed the actual data models. The average solution time for different model sizes is 140.26 seconds with a maximum difference of 1.97% between the global optimum LP relaxation and the integer optimal solution reached.

Hall et al. [28] present another BIP model for making project-funding decisions. This model is used to select projects related to reducing smoking for funding at the United States' National Cancer Institute. A federal targeted budget and private funds are allocated to some states in the U.S. over a seven-year period. In order to distinguish among the projects' importance, each project is ranked based on scoring criterion. This model is chosen to maximize the rank-function score for the projects subject to important constraints of available budget and other smoking-prevalence criteria. Containing 23 variables and 14 constraints, the model is solved in LINDO software version 5.0. The results provide lists of different solution based on various budgetary levels and preference-score ranges, thus enabling decision makers to choose the most appropriate solution.

Kozanidis and Melachrinoudis' paper [13] is one of the most interesting in the field of multi-choice constraint problems. The proposed model is a single-resource allocation problem used as a decision-making tool to assist transportation management in assigning limited funds to highway-safety projects classified as continuous and discrete improvements. This model is considered a generalization of the linear multiple-choice knapsack and the 0-1 knapsack problems. Numerous model properties are proposed to design a B&B solution algorithm. Specifically, the original B&B algorithm of solving the linear multiple-choice knapsack is adopted to fit with this specific problem, improve the solution efficiency, and lower the required memory space for the number of explored nodes in the tree. The algorithm solves the LP relaxation at each node in the B&B tree by changing the binary constraints for each variable with bound constraints; the bounding is between 0 and 1 and uses the relationship between the optimal solutions of parent and children nodes in branching order and node selection. These techniques for solving the B&B algorithm are used to reduce significantly the time required for branching and solving each node in the tree.

Liberatore [8] proposes an expert support system for industrial projects based on an analytical hierarchy process (AHP) designed to relate strategic planning, mission, and objectives to specific criteria for selecting R&D projects. To consider the tradeoffs between project benefits and costs, two methods are suggested for resource allocation. The first method is accounting for the ratio of the renormalized project priority, which represents the summation of the project's benefits, to the estimated cost of funding the project. The rated projects are ranked and then selected from high to low order based on their priority-cost ratio until the assigned R&D budget is consumed. The second method is formulating the resource allocation problem by using the BIP that maximizes the total project priority over all funded projects depending on a budget constraint. The author proposes other constraints that can be considered in the model to ensure certain factors, such as a maximum or minimum number of selected projects to be funded and based on different budgetary levels for every proposed strategy.

Technically speaking, the available methodologies for project selection and resource allocation can be applied to the majority of project types. However, relevant constraints, procedures, and requirements differ accordingly. For instance, Güney et al. [29] investigate various heuristic and algorithm solution quality and performance for selecting advertisements that influence social network members. BIP formulates the problem mathematically for the purpose of minimizing the total cost of selecting influences subject to lower bound on the probability of influencing the member number that can be assigned to a particular advertisement. The proposed model is BIP, which any commercial IP solver can solve. However, such a problem's difficulty is known to be NP-hard, and theoretical approximation's complexity is present. Moreover, one of the parameter coefficients may take any value other than a binary; in this case, obtaining an exact solution becomes computationally difficult. Accordingly, two heuristic approaches are proposed in the solution. Lagrangean relaxation and greedy heuristics are used and are compared to several optimum solutions that are obtained from a commercial IP solver, random selection method, and highest-degree selection method. Each method is computed with different problem sizes and evaluated in terms of results, percent deviation, and CPU times.

As discussed above, the techniques and approaches of the concept of constrained project allocation are widely applied in many disciplines. Various studies have extended this approach by accommodating specific conditions under which it is applied in education and human resources (HR). For example, Anwar and Bahaj [30] address the problem of allocating class projects to university students. In many university departments, a list of projects is offered to students from which they can select more than one project. Typically, each project is supervised by only one lecturer and assigned to only one student within a department. In addition, this paper considers another modeling approach of allocating a group project to students supervised by one or more lecturers. The paper shows two molding approaches and formulates them using BIP. The first model is formulated for individual project allocation to minimize the number of projects each lecturer supervises, subject to project-allocation restrictions. The second model is constructed for handling group projects that allocate a group of students to a project (i.e., a project may be allocated to more than one student). This model is intended to maximize the students' choice of project preferences subject to the number of project and lecturer constraints. The first model is solved by using dynamic programming and is implemented using LINGO 6.0.

Another well-known area of interest in HR is modeling a decision-support framework for project managers’ assignments. This problem investigates the development of different procedures and criteria for optimally allocating project managers to the best-matching projects, such as a project manager of developing new product projects in manufacturing companies. In addition to optimal allocation, studies provide significant information on performance and effects on projects, project managers in guiding projects to successfully accomplished, and organizations. Patanakul et al. [16] develop a decision support approach to assign new projects or reassign ongoing projects to project managers. The problem is formulated by BIP and considers project priorities, degrees of association between project managers and developing projects, and relevant organizational and personal limitations. The objective function in this model takes into account project requirements, project managers' effectiveness, and project contributions to the organizational mission.

Classification Approach using MD

Classification concept is defined as the process of separating or allocating various objects into several predefined classes, which can be defined by business rules, class restrictions, or several mathematical functions [31]. In general, two types of classification problems are considered in the research: (1) binary classification on one variable and (2) multiple classifications on more than one variable. In statistical analysis and data mining, many techniques are used for classification. However, lack of available implementation for many classifiers is a major drawback in any classification purpose [32]. In addition, the classification problem becomes very complex when numerous variables must be considered, especially when a correlation or dependency between variables may exist. Consequently, a more reliable criterion or measurement that handles correlated variables is necessary to address this issue. For this reason, in 1936, P. C. Mahalanobis introduced the concept of MD [19] for classifying a multivariate observation's similarity to the other group members' centroid or overall mean. The MD answers the question of how a multivariate observation is allocated to best matching groups or classes.

Since the MD concept originated, hundreds of studies have been conducted in many fields. For example, Huei-Chung Wang et al. [33] develop a classification method based on the Mahalanobis-Taguchi System (MTS) [34], initially proposed by Genichi Taguchi, who is well known for his extraordinary and controversial ideas and methods in the field of design and experiment analysis. The MTS combines MD, orthogonal arrays (OAs), and the signal-to-noise (SN) ratio. The study demonstrates MTS's effectiveness with two kinds of correlated multivariate datasets, Fisher iris and credit card, for the purpose of measuring the prediction accuracy in multi-dimensional data sets. By incorporating the MD, the results show that the MTS outperforms another well-known classifier by approximately 2.2 % in terms of classification accuracy rate for both datasets.

Bing Long et al. [35] present a novel approach for diagnosing initial faults in analog circuits using a classification algorithm of particle swarm optimization based on the MD. The MD is intended to categorize samples from two different fault predefined classes and to train and examine specific classifiers for any two fault classes of analog circuits. The near-optimal solution is achieved, by using MD in the classification algorithm, with a high recognition rate, and the time consumption is reduced by 98% compared to that of the optimal feature vector algorithm. The Euclidean distance (ED) is also compared to the MD in stability and optimality with an experimental study investigating MD's advantage in selecting the near-optimal feature vectors. The results show that ED's classification accuracy is 93.8%, whereas it increases in MD to 97.8%.

Cho et al. [36] propose a hybrid approach for predicting manufacturing firms' bankruptcy based on the combination of variable selection methods and MD. The algorithm of case-based reasoning approach is presented; and different experiments apply this algorithm with MD, ED, logistic regression, neural network, and decision tree models. For illustration purposes, 1000 Korean manufacturing firms are classified as bankrupt or healthy (non-bankrupt). The classification process uses the distance matrix of

MD and ED to measure the distance between the reference data set and the evaluation data set. The results show that the classifier algorithm using MD effectively outperforms other models in the correct classification ratios with an average of 73.6% and with the lowest misclassification error rate.

Srinivasaraghavan and Allada [22] develop a contemporary methodology that aids in providing a quantitative leanness measure by comparing a company against exemplar industry standards to facilitate continuous improvement. A survey is conducted to assess an organization's lean state by using several lean characteristics. The industries are classified into two groups: advanced lean (normal) and average/below (abnormal). Abnormality is considered to be due to either bad conditions or considerably perfect conditions. To distinguish between normal and abnormal categories with its abnormality conditions, the Mahalanobis Taguchi Gram-Schmidt System (MTGS) is used for this reason. The proposed MTGS measures the distance from a defined reference datum to each data point in each group and identifies those points that are out of Mahalanobis space, outliers. The study also presents an optimization approach to obtain the most effective information or variables needed in evaluating abnormal industries' level of lean implementation and then tests MD's impact as well.

Arathi and Govardhan [37] compare the classification accuracy and performance of MD and ED by incorporating them into a discriminate algorithm in time series data. The suggested technique is intended to evaluate the similarity or dissimilarity between two time series sequences (i.e., unlabeled time series should be allocated to a predefined group). The major advantages and disadvantages of ED and MD are described. For example, ED is unsophisticated in computation and does not require many details.

However, it has some shortcomings. It requires the data to be standardized when variables are measured in different scales; both sequences have the same length; it does not consider the correlation between the data variables; and it is sensitive to noise. On the other hand, MD accounts for data correlation by occupying the covariance in its equation; it does not require the data to be standardized, even if the variables are determined or measured in different scales; and it provides the same ED measurement when variables are uncorrelated. The study conducts comparison experiments on various datasets to assess how the classification algorithm with MD and ED is able to differentiate between any two classes. The experimental results show the proposed methodology with MD increases the accuracy around 10%-15% more than in ED and decreases the time complexity by 15%-22%.

MD Applications

The initial inspiration for proposing the MD was to analyze and classify human skulls into predefined categories, based on different characteristics or factors. Since then, many distinguished applications based on MD have been developed in different areas, such as, engineering, manufacturing, chemical research, healthcare engineering, pattern recognitions, businesses, etc. We list few applications in the literature as follows:

- Face Recognition [38]

In the purpose of classification face images that include frontal views of faces with different properties, such as facial appearances, lighting conditions, and occlusion dimensions, the MD is used to enhance the classification accuracy and performance in face identification or recognition and to measure the similarity between the faces and to classify and recognize the samples.

- Composition of Pharmaceutical Products [20]

Pharmaceutical companies produce drugs that consist of a number of active compounds developed for treating different diseases. These companies have historical data recorded on their database on the significant proper concentration amount of the different active ingredients based on patient records for each of the known diseases. With statistical analysis for these data, the pharmaceutical company can discover different MDs related to each different type of disease and classify the most appropriate ingredients to different diseases. Subsequently, the MDs are used in this application to categorize and optimize the drug products and finds which drugs are best for specific diseases.

- Financial Applications in Forecasting [39]

Information that can be used to asses forecasts can be provided by MD. In this application, a multivariate-observation forecast from a prediction model is to be assessed for its accuracy by comparing it with the known true observations. The suggested methodology chooses the mean squared error (MSE) that represents the loss function of the prediction model. The MSE-matrix determines the pattern or shape of the confidence region that denoted by the MD. Therefore, the lower the MD between observation and prediction points, the better the forecasting model and vise versa. The extreme points of the MD results might be considered such as an indication against the reliability of the forecasting model.

- Product Manufacturing [20]

Suppose a manufacturer produces a commercial product, for example, a car or computer that is described by a number of properties (cost, selling price, dimension, etc) with a verity of possible customers uses, for instance, work, family, etc. The manufacturer has datasets from the market research on a number of items purchased with specific parameters by customers with the intended use. The MD will be used in the product designing phases, that is: it measures how much close each intended use of different product items to the product's parameters based on the customer preferences.

- Breast Tumor classifications [40]

In the diagnostic processes of breast tumors, the Mammography is widely used for identifying tissue types, benignant or malignant. Every image for each tissue has common shape factors for barest tissues that are recognized by specialists and medical software. The MD concept is integrated with another method to be used in biomedical computer software to assign each breast image from the Mammography to lesion categories.

Branch-and-Bound Strategies and Their Impact on Solution Performance

In 1960, Land and Doig developed the first B&B algorithm for general MILP and pure ILP. Soon after, in 1965, Balas suggested an effective algorithm that can solve ILP problems with BIP. The B&B algorithm solves these problems by dividing the search space and generating a sequence of sub-problems. The branching strategy is the heart of the B&B algorithm and is considered one of the most significant factors affecting solution performances [41]. Therefore, we focus on two critical questions about the B&B algorithm that many research studies have explored but remain unanswered [12]:

1. Which *specific variable* in branching should be branched first?
2. In selecting the next branching node or sub-problem, *which node* should we begin to solve?

Because both questions are unanswered, the research in this area is still ongoing. However, ensuring computations could differ dramatically with different branching strategies. Consequently, the result in the B&B trees differs greatly in size and in computer time required to find an optimal solution [42]. The main reason we use the B&B algorithm is that its technique provides a flexible structure in which we can design any problem type, such as BIP. Moreover, many options are available in B&B that can be used in designing an efficient algorithm and that can substantially affect the solution performance.

Many studies and books cover B&B's concepts, applications, and techniques and address how this algorithm is flexible to accommodate different integer programming, such as pure IP, MIP, nonlinear MIP and IP, and BIP [41, 43]. In order to narrow down our study specifically in B&B branching strategies, the literature review in this section is restricted to B&B strategies that can improve the algorithm's solution performance in various ways. For example, Borchers [43] investigates how branching strategies with a proposed heuristic can effectively improve the B&B algorithm's speed in finding the optimal solution and reduce the memory storage the experimental codes use. Two branching strategies are used in the study: (1) a depth-first strategy, which is intended to choose the next sub-problem or node that was most recently created for obtaining a quick fathoming; and (2) a fixed-order strategy, which aims to rank branching variables in specific order before solving any nodes, thus reducing the constraints' size. The suggested techniques are coded in IBM's Optimization Subroutine Library for different examples of capacitated facility-allocation problems and other problems. Computational results show that the branching strategies for some problems enhance the CPU time by approximately 20% less than the algorithm without suggested strategies or default strategies.

Koch et al. [44] survey a comprehensive description of 87 real-world instances available in the Mixed Integer Problem Library (MIPLIB) (2010) and run those problems on different solvers to assess performance. All of the instances in the MIPLIB are combined into a large group of IP problems in academia and industries. The paper emphasizes that the most important contribution to the CPU time needed to solve the IP problems considered is attributed to the B&B tree’s size. Many factors may significantly influence the solver behavior, such as input format, constraint orders, differences in model formulation, problem size of adding or removing redundant variables or constraints, and bounds. Consequently, all the relevant research effort is done to propose an effective methodology or code that aims to reduce the number of nodes in B&B that must be evaluated. The paper also notes that the effective branching strategy can be implemented by determining the order in which candidate variables are considered to be branched.

Wang et al. [45] propose two B&B approaches, among different methods, to model and solve the well-known supply chain problem of supplier selection and order quantity allocation. The MIP model formulation for this model is intended to minimize the expected total costs associated with order placement, order quantity, and penalties subject to order quantity upper- and lower-bound restrictions. The first B&B aims to branch on the binary decision variables of supplier selection choices. The second B&B is proposed to branch on the order quantity from the supplier. Both models are coded and solved in MATLAB R2011a with different problem sizes for modeling evaluation and performance testing. Computational studies show that the second B&B method is better than the first for various performance measures, such as the number of searched nodes and the computation time by performing approximately less than the first B&B method by 84% and 40%, respectively.

Achterberg et al. [41] present a new branching strategy that combines well-known branching rules. The new strategy's branching technique, reliability branching, aims to assign a score to un-branched variables that have not been used in the branching process a certain number of times. This technique's purpose is assessing the impact on the quality of the gap between sub-problem and parent. This gap ultimately affects the B&B tree's size and the solution time. The study compares this strategy with different branching rules, such as most infeasible branching, pseudo-cost branching, strong branching, hybrid strong/pseudo-cost branching, and pseudo-cost branching with strong branch initialization. A numerical experiment is conducted on instances from MIPLIB 2003 and Mittelmann and run on CEPLEX and SIP. The branching strategies' performance is evaluated in CPU time and B&B nodes and shows that these strategies provide different measures. The research concludes that the new branching rule remarkably improves the performance of the solution.

In one of the most significant application studies on branching rules and their considerable impact on solution performance, Murkute [14] investigates the capacitated multi-commodity fixed-charge network flow (MCFCNF) MIP problem. This research links a new branching strategy to the amount of information branching variables provide in the B&B tree's initial levels and determines if there is a relationship between the variables and a problem characteristic of arc utilization. The new rule of selecting next branching variables involves the following steps: (1) combining all feasible solutions, (2) selecting a group of candidate variables, (3) calculating each candidate’s percentage of contribution to the feasible solution, (4) assigning scores based on the contributions’ impact, and (5) branching according to the variables' scores. A comparison study was conducted on 20 instances and applied on two IP mathematical solvers, SCIP and GLPK, to evaluate the proposed approach compared to other branching strategies, using different measures, such as CPU time, searched nodes, and optimal solution gap. Computational results show substantial variations in the solution performance among different branching strategies.

Atamtürk and Savelsbergh [15] emphasize in their research study that the core of any efficient IP solver is the LP-based B&B method. This study provides various important options in the B&B algorithm that are available to the users and describes how its settings can significantly affect the B&B method’s behavior. These options include the choices of node selection, branching rules, cut generation, preprocessing, and heuristics. Furthermore, the research compares the following options in different IP solvers: CPLEX; LINDO; and Xpress-MP. Computational experiments are conducted on different instances from MIPLIP 2003 to compare and measure the B&B solution performance among the options provided in the solver’s default settings and the user’s options settings. For comparison purposes, various performance measures are considered to assess each solver’s capability, such as CPU time, number of searched nodes in the B&B tree, and solution gap between global optimal and best reached bound in the B&B tree. Each solver’s performance shows different results. Therefore, these results imply that the user control in the B&B options can significantly affect the algorithm performance. Users may improve the B&B method by selecting specific settings that best fit their problem types. The paper also addresses major challenges that might be encountered in the B&B algorithm and discusses possible features that may improve the solution quality. Additional references for relevant literature are provided in this paper.

There are many other branching strategies presented in the literature and commonly used, such as, fixed order, early, maximal pseudo-cost, most-fractional, most-infeasible, maximal-objective, and strong branching. Table 1 shows a list of references of widely used and proposed branching strategies.

Table 1: List of Branching Strategies

Branching Rule	Modeling type	Reference
Most infeasible, least infeasible Pseudo cost, strong branching, Hybrid, and Reliability branching	MIP	[15, 48]
Entropic Branching and Entropic Look-Ahead Branching	MIP	[67]
Early and Ordered branching	MIP	[43]
Partial sequence	MIP/ BIP	[13]
Problem-Oriented	BIP	[68]

Summary

This chapter provided a review of relevant literature in R&D project selection and resource allocation, MD classification criterion and its performance, and B&B strategies and their effects on the algorithm performance. As displayed in Figure 1, the relevant

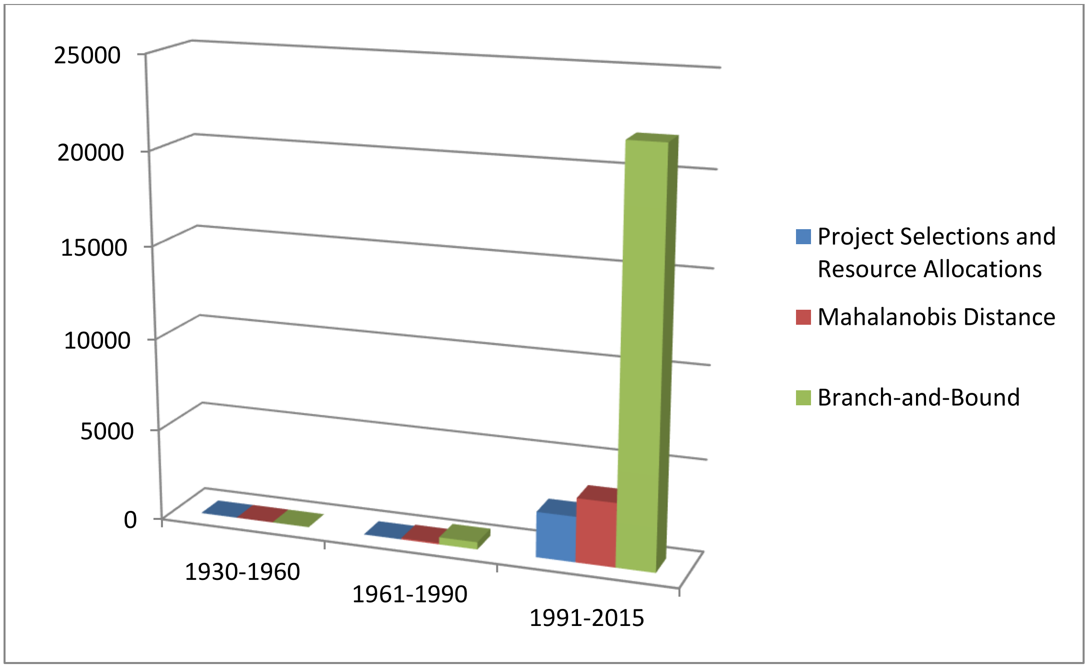


Figure 1: Number of Relevant Articles in Different Areas from the Year of 1930 to 2015

areas in our problem have remarkably increased over the last 25 years with total of 2458 articles were published in project selection and resource allocation, 3458 in the area of MD, and 21865 research studies in B&B from 1991 to 2015.

The literature review investigated the concept of selecting the most attractive projects; however, allocating them among different options subject to scarce resources was not addressed in the literature. Moreover, integrating classification criteria to be used in an optimization model for allocating projects to the best-fitting options is not applied or studied in the relevant works. Although many methods for solving IP are discussed in the literature, no one of them is computationally operative. In practice, the B&B method’s capability and reliability have been proven more than that of other IP methods or algorithms [12].

The key findings in the literature regarding the methods and models for R&D project selection and resource allocation involve two areas: (1) the methodologies that provide decision managers optimal project selection and allocation criteria and (2) the effective means of solving these methods [1]. Unfortunately, the majority, if not all, of the available proposed models have several gaps [3, 8, 9, 46]. First, they do not support the project allocation on more than one option with specific classification criteria; thus, no model is available for optimal project allocation to the best-fitting options. Second, many models do not explicitly incorporate the supporting staff's expertise and data from previously completed projects; hence, these kinds of models' decision tools do not realistically provide R&D managers the required flexibility to structure a relationship between allocation criteria and options. Finally, the models are ineffective for solving problems with both decision variables and constraints of appreciable size. That is, they are unable to handle large problems.

The literature review reveals that the majority of the research in the transportation project classification deals with binary choices. The constrained project classification problem in this dissertation is a typical combinatorial optimization problem, so it becomes a Nondeterministic Polynomial (NP)-hard problem. However, new technological improvements and developments in IP solvers and hardware processors provide the best ways to improve any IP algorithm's performance.

Based on this dissertation's literature review, the following gaps and findings exist:

- Despite the significant R&D project selection and resource allocation mathematical models and approaches available in the literature, no study has been developed or reported in the literature based on the MD minimization with multiple resource constraints in the R&D project selection area.

- All research in B&B strategy has focused on general MIP problems and has shown that these strategies can significantly affect solution times [15]. To the best of our knowledge, these strategies have not been computationally analyzed or compared in relation to pure BIP problems. Hence, we are among the first who propose and apply new B&B branching strategies in BIP and evaluate the algorithm performances with a comparison study.
- This dissertation adds to the literature of project management, B&B strategies, and MD applications by presenting a novel project selection and resource-allocation approach that not only finds the best-fitting project allocation among multiple options but also proposes another remarkable improvement in B&B algorithm performance based on specific branching variables with the minimum of MD measures.

CHAPTER III
OVERALL CONCEPTUAL APPROACH

As discussed in Chapter 2, all the methods that have been widely studied and applied to model the process of project selections and resource allocations totally depend on the type of the problem. We have also noted that these models can be applied with relevant adjustments in any kind of project and organization. In order to understand the procedures of assigning projects to multiple options based the MD concept, the process involved in our model must be described. The solution procedure for the constrained project selection and allocation in this dissertation involves major assumptions.

First, each project is represented by several properties or measurements, multivariate, which may be interdependent. Second, data bases containing recent project details that have been *developed* by either allocation option (in-house or private contractors) is available. Third, a panel of experts is intended to form or index *each allocation option* by a *selected project sample*, from the data bases, that is considered to be highly representative of the projects propose to be funded. The process of this assumption is outlined in Figure 2.

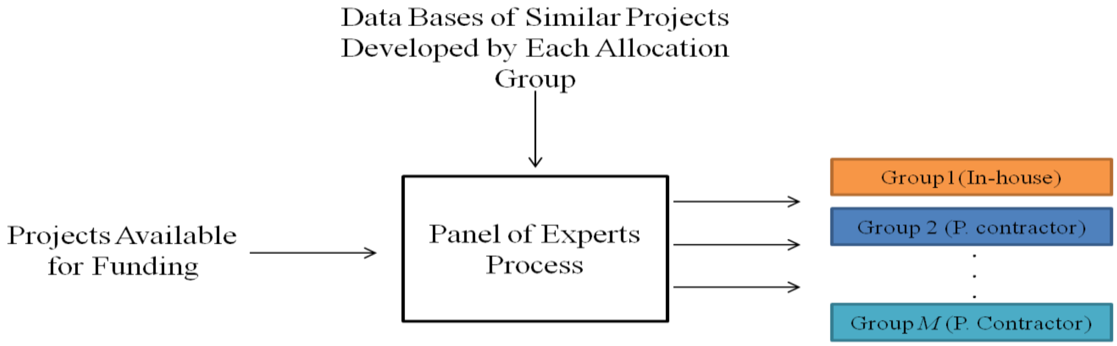

Data Bases of Similar Projects Developed by Each Allocation Group
Projects Available for Funding
Panel of Experts Process
Group 1 (In-house)
Group 2 (P. contractor)
Group M (P. Contractor)

Figure 2: Process of Determining Allocation Options

More specifically, the panel of experts, in the third assumption, consists of a group of people from different departments, such as purchasing, engineering, planning, etc. This panel uses the information available from the suggested projects for funding (inputs) with the information in the database containing similar projects most recently involving each option (in-house and contractors) to produce reference project samples for each option (output). These samples are used in the MD computation. Specifically, MD measures how each input project best matches every option (output) to be allocated.

Fourth, a percentage of project funding or completion is not allowed (i.e., a selected project is fully funded and completed by the available resources).

Fifth, the degree of risk and the possibility of failure in project completion are omitted in our study.

Finally, the interdependent chronological relationship, of project or resource between each other is not considered (i.e., starting and finishing any project is not necessarily reliant or depended the status of other projects or activities).

The department of project management handles projects in any business environment or governmental sector. Regardless of variations in project types, each project in R&D usually progresses in three primary phases [4] as follows:

1. Project proposal submittal
2. Project proposal and scope evaluation
3. Decision-making regarding project funding and relevant resources

This process is illustrated in Figure 3, which provides a clear picture of the step we concentrate on in this dissertation.

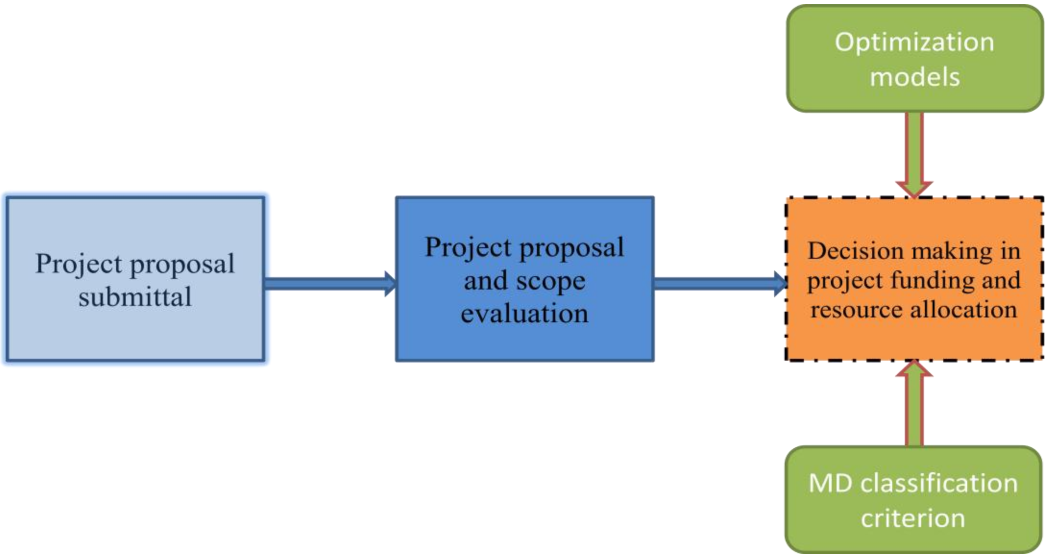

Optimization models
Project proposal submittal
Project proposal and scope evaluation
Decision making in project funding and resource allocation
MD classification criterion

Figure 3: R&D Project's Primary Phases

As shown in Figure 3, our contributions focus on providing decision makers and managers systematic methodology for selecting optimal projects to be funded subject to availability of resource and budget constraints, as well as identifying projects most suitable for either in-house development or private contractors' use.

This chapter will discuss the proposed tools (i.e., MD definition and equations) involved in the project-funding decision-making phase. Moreover, a compendium of the B&B algorithm is provided as a solution methodology for our proposed project-selection and resource- allocation models.

Mahalanobis Distance

The concept of the MD has been widely used in different statistical areas that consider multivariate observations, such as classification, cluster, and outlier detection techniques. MD is considered a more powerful distance measure than other methods, such as ED, because it considers different scales in multivariate observation variables or characteristics and the potential correlation among them.

Definition

The MD is a measurement of statistical distance between an *n*-dimensional point (multivariate) with coordinates, variables, given by a *p*-variant vector $\mathbf{x} = [x_1, x_2, x_3, \dots, x_p]$ and a space's center (overall mean) represented by a reference mean vector $\boldsymbol{\mu} = [\mu_1, \mu_2, \mu_3, \dots, \mu_p]$ of the data matrix **X**. The reference data matrix **X** consists of *n* observations, in rows, measured for *p* variables (characteristics), in the columns. The MD is defined in the following equation:

$$MD_{(\mathbf{x})} = \sqrt{(\mathbf{x} - \boldsymbol{\mu})\,\mathbf{S}^{-1}(\mathbf{x} - \boldsymbol{\mu})^T} \qquad \textbf{(1)}$$

where $\mathbf{S}$ is the sample covariance matrix of a reference sample **X**, which is given by:

$$\mathbf{S} = \frac{1}{n-1}(\mathbf{X}_\mathbf{c}^T \mathbf{X}_\mathbf{c}) \qquad \textbf{(2)}$$

where $\mathbf{X}_c$ is the centered data matrix $\mathbf{x} - \boldsymbol{\mu}$. For *p*=2, the sample covariance matrix $\mathbf{S}$ is represented in (3):

$$\mathbf{S} = \begin{bmatrix} s_1^2 & r_{12}s_1s_2 \\ r_{12}s_1s_2 & s_2^2 \end{bmatrix} \qquad \textbf{(3)}$$

where s_1^2 and s_2^2 are the variances of the values of the two vector variables, $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively; r_{12} is the correlation's coefficient between them; and $r_{12}s_1s_2$ is the covariance between the two variables (i.e., the covariance measures the level of relationship between two variables). The covariance matrix is a *p*-by-*p* square matrix that contains variables' variances in the diagonal elements of the **S** and the covariance between any two variables in the off-diagonal elements. The inverse of the covariance $\mathbf{S}^{-1}$ matrix is given in the following equation:

$$\mathbf{S} = \begin{bmatrix} s_2^2/\det(\mathbf{S}) & -r_{12}s_1s_2/\det(\mathbf{S}) \\ -r_{12}s_1s_2/\det(\mathbf{S}) & s_1^2/\det(\mathbf{S}) \end{bmatrix} \quad \textbf{(4)}$$

where $\det(\mathbf{S})=s_1^2s_2^2(1-r_{12}^2)$ is the determinant of the $\mathbf{S}$.

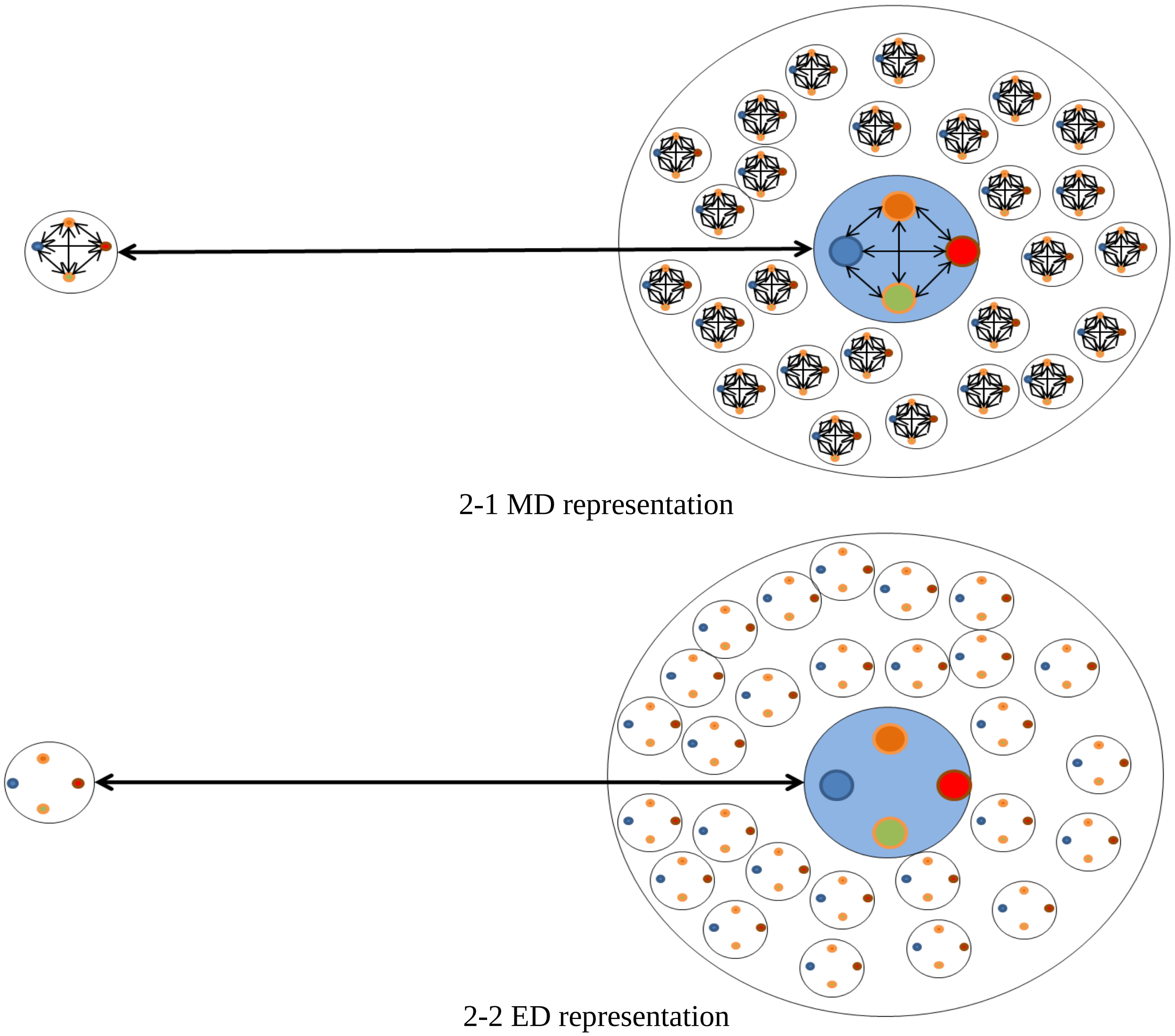
Note when $\mathbf{x}_1$ and $\mathbf{x}_2$ in Eq. (4) have no correlation (i.e., $(r_{12}=0)$ or $s_1^2=s_2^2)$), the Eq.(1) is shorten to the formula for the ED as in Eq. (5) and the covariance matrix $\mathbf{S}$ becomes the identity matrix. Thus the MD reduces to ED

$$ED(\mathbf{x}) = \sqrt{(\mathbf{x}-\mu)\,(\mathbf{x}-\mu)^T} \ . \quad \textbf{(5)}$$

Consequently, the ED does not consider the correlation between the random variable's components. As a result, the MD is commonly used in statistics and data mining fields because it is significantly better than the ED, especially when sample is not regularly distributed around the mean, in other words it is correlated.

Graphical Representation

In order to illustrate the concept of MD and provide better understanding of its function in project classification, the concepts of MD and ED are graphically displayed in Figure 4. Specifically, Figure 4 shows the major differences between these two distances regarding the correlation between the variables of an observation (multi-dimensional point). In general, each circle on the left and its typical circle shapes on the right represent a multivariate or multi-dimensional point (in our case, projects). More specifically, the colored small points inside these circles denote to the characteristics (variables) that identify each project scope.



2-1 MD representation

2-2 ED representation

Figure 4: MD and ED Graphical Representations

The biggest two circles on the right represent a multivariate space (reference samples in our scope). The circles in the center of the two biggest circles on right represent a special multivariate point, which is the overall mean or centroid of the multivariate space.

The colored small points inside the circles, in MD, are interconnected, representing a correlation between variables that is accounted for in this specific distance measure. On the other hand, the connections between variables in ED graph are not considered (i.e., the relationship between project's variables is not accounted for in this distance). Therefore, we can clearly describe the MD as a distance between a multivariate point and a multivariate space’s centroid accounting for a correlation with a multivariate point characteristic.

Computation Example

For illustration purpose, suppose we have two groups of datasets: *A* and *B*. *A* represents the dataset for multivariate observations and *B* the reference sample. The two groups can be represented in matrix form as $\mathbf{X}_A$ for group *A*, and $\mathbf{X}_B$ for group *B*. The data matrices of $\mathbf{X}_A$ and $\mathbf{X}_B$ is *n*-by-*p*, where *n* is the number of observations and *p* is the dimension of the data (variables or factors). The MD computation requires both matrices to have the same number of column dimension *p* (variables); nevertheless, they may have different numbers of rows *n* (observations). Furthermore, matrix $\mathbf{X}_B$ must contain more rows than columns.

The MD measures the distance between each observation *n* in $\mathbf{X}_A$ and the reference sample $\mathbf{X}_B$, which results in MD *n*-by-1 vector.

Computational steps

Let $\bar{\mathbf{x}}_B$be the vectors of means for group *B*, which contains the same number of elements and variables. The simulated data in Table 1 are be used in illustrating the steps in computing the MD.

Table 2: Simulated and Standardized Data for MD Illustration

Observation No.	**Dataset**				**Centered Data**	
	$\mathbf{X}_A$		$\mathbf{X}_B$		$\mathbf{X}_{\mathbf{c}A}$	
	$\mathbf{x}_{A1}$	$\mathbf{x}_{A2}$	$\mathbf{x}_{B1}$	$\mathbf{x}_{B2}$	$\mathbf{x}_{cA1}$	$\mathbf{x}_{cA2}$
1	1	1	0.53	-0.11	0.38	0.13
2	1	-1	1.83	2.98	0.38	-1.87
3	-1	1	-2.25	-1.72	-1.62	0.13
4	-1	-1	0.86	0.75	-1.62	-1.87
5			0.31	0.59		
6			-1.31	-1.26		
7			-0.43	-0.44		
8			0.34	0.95		
9			3.57	3.83		
10			2.76	3.11		
Mean			$\bar{x}_{B1}$= 0.6248	$\bar{x}_{B2}$= 0.8691		

Group *A* has n_A= 5 observations and group *B* has n_B = 10 observations. Each observation in either group has *two* values (two variables or factors are possibly correlated). In Table 2, bold-faced symbols represent vectors or matrices: $\mathbf{X}_A = [\mathbf{x}_{A1} \quad \mathbf{x}_{A2}], \mathbf{X}_B = [\mathbf{x}_{B1} \quad \mathbf{x}_{B2}]$,$\bar{\mathbf{x}}_B = [\bar{x}_{B1} \quad \bar{x}_{B2}]$.

The *raw data* in Table 1 have been transformed into *centered data* matrix $\mathbf{X}_\mathrm{c}$ and will be used in MD computation by subtracting from each value in $\mathbf{X}_A$ the mean of its corresponding column in $\bar{\mathbf{x}}_B$ (i.e., for observation in group *A*, $\mathbf{X}_{cA} = [\mathbf{x}_{cA1} \quad \mathbf{x}_{cA2}]$, $\mathbf{x}_{cAi} = \mathbf{x}_{Ai} - \bar{\mathbf{x}}_{Bi}$, ∀ i = 1, 2).

The MD's computational steps are outlined below:

Step 1

The covariance matrix for the reference sample, group *B*, is computed as in Eq. (2)

From the data given in Table 1, $\mathbf{S} = \begin{bmatrix} 3.11 & 3.24 \\ 3.24 & 3.58 \end{bmatrix}$

Step 2

The inverse of the covariance matrix $\mathbf{S}^{-1}$ is calculated as given in Eq. (4)

$\mathbf{S}^{-1} = \begin{bmatrix} 5.1297 & -4.6319 \\ -4.6319 & 4.4615 \end{bmatrix}$

Step3

The MD between each observation in *A* and the reference sample in *B* are ready to be computed from Eq. (1). For example, we pick the first observation ,row, in $\mathbf{X}_A(1) = [1 \quad 1]$

For the given data point,

$\mathbf{X}_{\mathbf{c}A}(1) = [0.38 \quad 0.13]$, $\mathbf{X}_{\mathbf{c}A}{}^T(1) = \begin{bmatrix} 0.38 \\ 0.13 \end{bmatrix}$, and $\mathbf{S}^{-1} = \begin{bmatrix} 5.1297 & -4.6319 \\ -4.6319 & 4.4615 \end{bmatrix}$

Therefore, using Eq. (1),

$$MD_{\mathbf{X}_{A(1)}} = \sqrt{(\mathbf{x}-\bar{\mathbf{x}})\mathbf{S}^{-1}(\mathbf{x}-\bar{\mathbf{x}})^T} = 0.5926$$

In order to understand the difference between the concepts of MD and ED, we display both measures in a graphical representation, as shown in Figure 5, and address how considering the correlation between variables affects on the distance measurements as well [47]. The results for MD and ED calculations are summarized in Table 3.

Table 3: Comparing between MD and ED

Observation Data $\mathbf{X}_A$	MD	ED
[1 1]	0.5926	0.4013
[1 -1]	4.7816	1.9061
[-1 1]	3.9419	1.6264
[-1 -1]	0.9979	2.4733

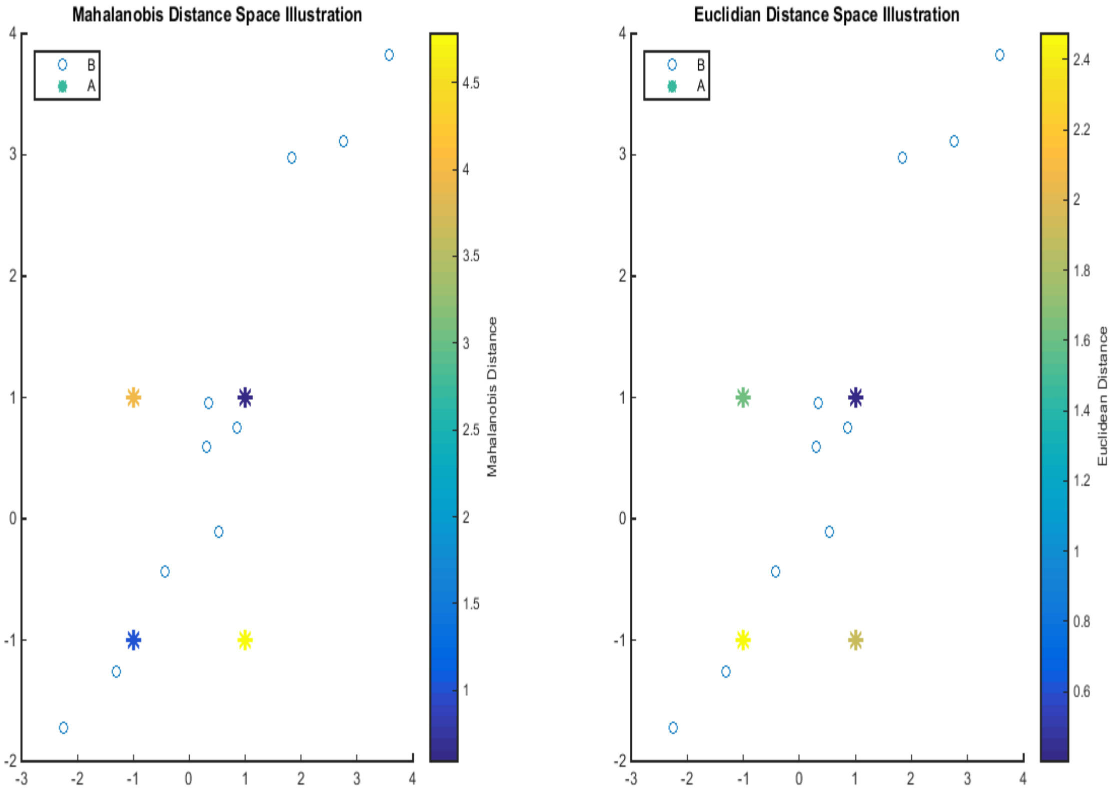


Figure 5: MD and ED Illustrations

In this example, all the MDs and EDs are calculated for $\mathbf{X}_A$ described in Table 2.

The results of MD and ED in Table 3 show that the observations in $\mathbf{X}_A$ with identical values in their variables have the minimum distances to the reference sample $\mathbf{X}_B$ in MD compared to the observations with different values in their variables. In contrast, the ED results with similar and opposite values in the observations' variables do not indicate any distance interpretation to the reference sample $\mathbf{X}_B$ in ED space. The MD results have more meanings in distance interpretation than ED because the MD accounts for the correlation and the scales between the reference's variables. As a result, the observations with identical or opposite values are clearly more distinguished and meaningful than the all results in ED regardless its observations' coordinates, similar or different. Moreover, Figure 5 further interprets the distances between every observation and the reference sample in MD and ED spaces. Consequently, MD has successfully shown how close each observation in $\mathbf{X}_A$ is to the reference sample $\mathbf{X}_B$; thus, it is very efficient and reliable classification criterion.

The overall conceptual approach of measuring the MD between each project and multiple allocation options in this dissertation is illustrated in Figure 6. The steps in measuring the MD between a given project and allocation options are summarized below:

1. Given a set of project characteristics or variables, define the scope of a project *n* in set *n*= 1, 2,…, *N*
2. Identify a group of typical projects *m* or reference samples (selected by a panel of experts) for each allocated option in set *m*= 1, 2, …, *M*
3. Compute the mean and the covariance matrix for each option in set *m*

4. Calculate the MD between each project *n* and the mean or centroid of the multivariate space in each option *m*.

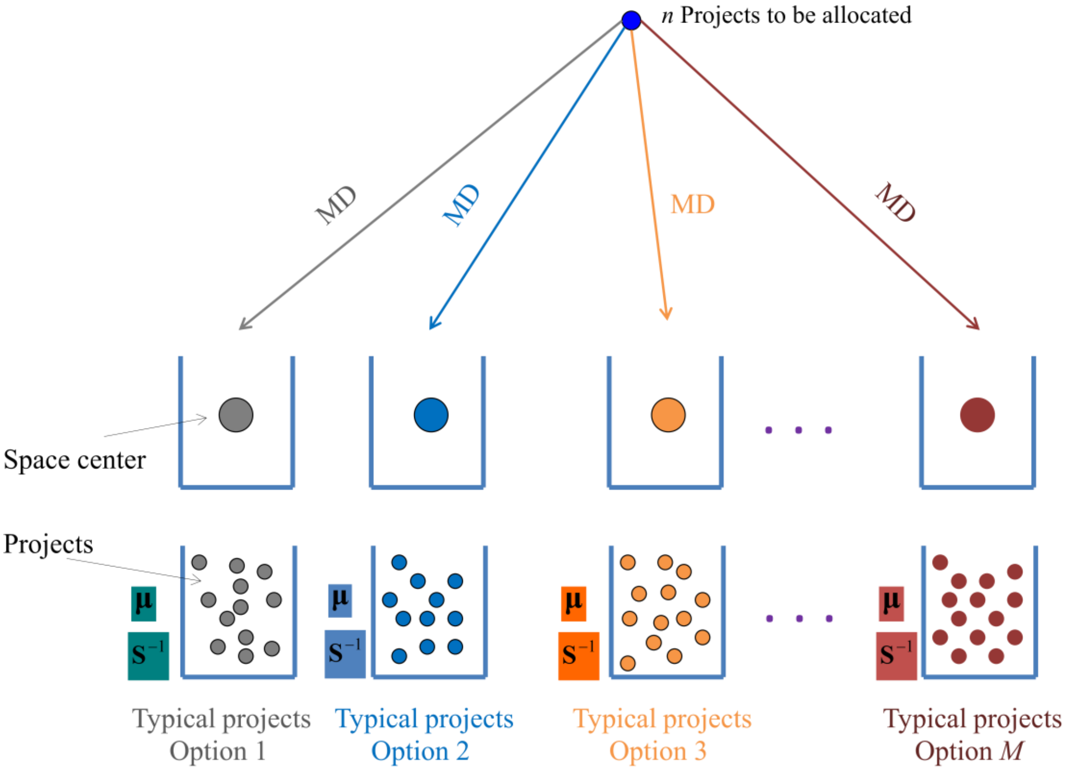

n Projects to be allocated
MD
MD
MD
MD
Space center
Projects
μ
S⁻¹
μ
S⁻¹
μ
S⁻¹
μ
S⁻¹
Typical projects Option 1
Typical projects Option 2
Typical projects Option 3
Typical projects Option *M*

Figure 6: Illustration of Project Allocation Using MD Classification Criterion

Typically, each project allocation option in this dissertation has different MD space (i.e., inverse covariance matrix, standard deviations, and overall mean in set *m*). Therefore, every project has an equivalent *m* numbers of MD, with a total of $(n \times m)$ total number of MDs for all projects. Generally speaking, the desirability of a given project *n* to be allocated to an option in set *m* is defined in terms of its MD to that option. In an ideal world, each project should be allocated to its closest option (minimum MD). This scenario, however, may not be possible depending on the available levels of each relevant resource that should be considered in the MD project-allocation process. The availability levels represent the MD as a constrained distance that should satisfy other conditions. Accordingly, the constrained MD is what we study in this dissertation.

Branch-and-Bound Overview

The B&B algorithm is used in this dissertation as the methodology for solving our problem of interests. Specifically, B&B method solves the suggested mathematical models of allocating projects to either in-house or private contractors subject to relevant constraints as stated in Chapter 1. This section will present an overview of the major steps in B&B and show our concentration's position in this method. Comprehensive descriptions of B&B technique are available in various resources, including [11, 12, 48, 49].

Briefly, the B&B algorithm handles not only the problems of solution infeasibility, which the integrality constraint causes in an IP problem, but also the extremely large feasible points in the feasible region through splitting (branching) the solution region into two or more divisions (sub-problems or nodes).

Such branching reduces the whole region, for optimizing, into smaller ones; thus, the solutions in these sub-problems are much easier and efficient in CPU time and effort than solving the entire region at once. The further growth in the branching process by dividing the whole feasible region into smaller regions creates a B&B solution tree, which consists of the main root representing the original or initial problem and the nodes corresponding to the sub-problems. Each node is solved consecutively in a similar manner: by dividing the feasible region in each sub-problem (parent) into other sub-problems (child nodes) until the best integer solution is obtained. Figure 7 depicts the general BIP B&B search tree.

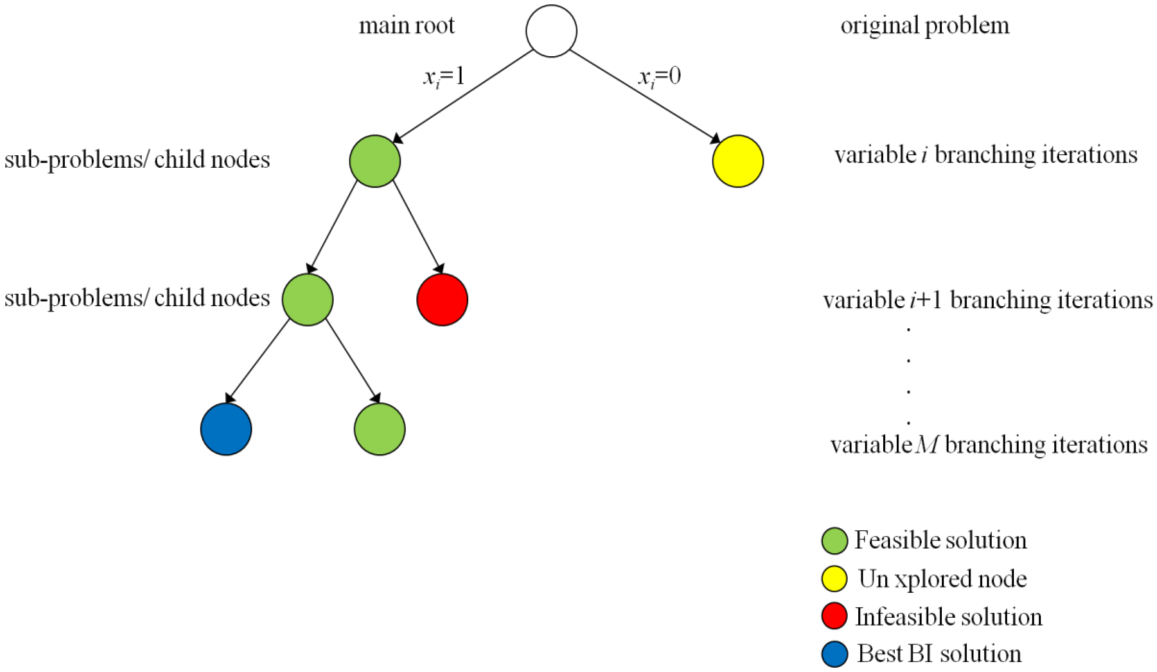

main root
original problem
x_i=1
x_i=0
sub-problems/ child nodes
variable i branching iterations
sub-problems/ child nodes
variable i+1 branching iterations
variable M branching iterations
🟢 Feasible solution
🟡 Un xplored node
🔴 Infeasible solution
🔵 Best BI solution

Figure 7: B&B Search Tree for BIP Problem in Two-Branching Case

To comprehend the B&B tree's construction for the BIP model, we represent the mathematical model form in Eq. (6) as follows:

$$\begin{aligned} &\text{Maximize} \quad z = \mathbf{c}^T \mathbf{x} \\ \text{Subject to}& \\ &\mathbf{Ax} \le \mathbf{b} \\ &\mathbf{x} \in \{0,1\} \end{aligned} \qquad \textbf{(6)}$$

where z is the objective function (it could be in a minimization form), **c** is the column vector of the variables' coefficients, **x** represents the column vector of the decision variables (binary), the matrix **A** is the constraint's coefficients, and the column vector **b** is the quantity of the resources available. The last constraint $\mathbf{x} \in \{0,1\}$ declares that all the variables must be binary.

The B&B method's first step is solving the original BIP problem's LP relaxation, which the *main root node* portrays in the tree. The LP relaxation solution involves *deleting* one set of constraints (BI) that makes the problem difficult to solve and substituting the decision variables $\mathbf{x}$ with other constraint that has an upper bound of 1 and lower bound of 0. Specifically, the optimal LP relaxation solution, for the main root node or the whole problem, provides the global upper bound (UB) for all feasible solutions, including the original BIP feasible solution (i.e., the optimal solution of the BIP original problem ≤ optimal solution for LP relaxation).

The next step is splitting (branch) the LP relaxation feasible region into two sub-problems by fixing *one* of the decision variables $\mathbf{x}$ at a value of (x_i=0) for one sub-problem and at a value of (x_i=1) for the other sub-problem and then solving each sub-problem using LP relaxation. Typically, continuing this process will generate parents that represent sub-problems and child nodes, which are also sub-problems derived from their parents in every branching iteration level as shown in Figure 7.

One of the most efficient and important steps in the B&B algorithm is called fathoming or pruning a sub-problem, (i.e., abandoning particular sub-problems from further exploration or branching). This step avoids solving every sub-problem in the B&B search tree; otherwise, an exponential number of sub-problems can result. There are three conditions for pruning a node:

1. If a sub-problem's alternative best known optimal BI solution is found at a specific iteration level, the bound that the LP relaxation finds for a sub-problem is

less than or equal to the best BI objective value. That is, a node should be pruned if its LP relaxation is found in the search tree, whether its solution is BI or continues feasible ≤ best- known BI optimal solution.

2. The LP relaxation for a sub-problem has no feasible solution.
3. The optimal solution of a sub-problem’s LP relaxation satisfies the BI constraint and its relaxed and non-relaxed objective solutions are equal.

As discussed in the aforementioned chapters, BIP mathematically models the project selection and resource allocation; namely, each variable in $\mathbf{x}$ or project takes only a binary value (1=allocate and 0=otherwise) also corresponding to the two branching. This scenario, however, may not be the case in this dissertation because we have multi-group project allocation (in-house and contractors) and each project should be allocated *at most* to only one classification option. In other words, the traditional B&B tree shown in Figure 7 is different in our allocation problem because every project is enumerated for each possible allocation among the available options. Therefore, the total number of branching in the B&B search tree equals ($M+1+(M+1)^N$), where M represents the total number of options, more than one, and N is the number total number of projects. To explain the B&B search tree's structure for the project allocation on multi-group problems, Figure 8 provides a snapshot of the tree with an example of allocating the total number of N projects on at most one of the three options (A, B, or C). In this Figure that each node has four branches corresponding to the possible allocation solution; not allocate at all; or allocated on option A, B, or C. The dramatic expansion and the search tree's complexity are clearly depicted as the problem's size increases (i.e., the number of projects and allocation options increases).

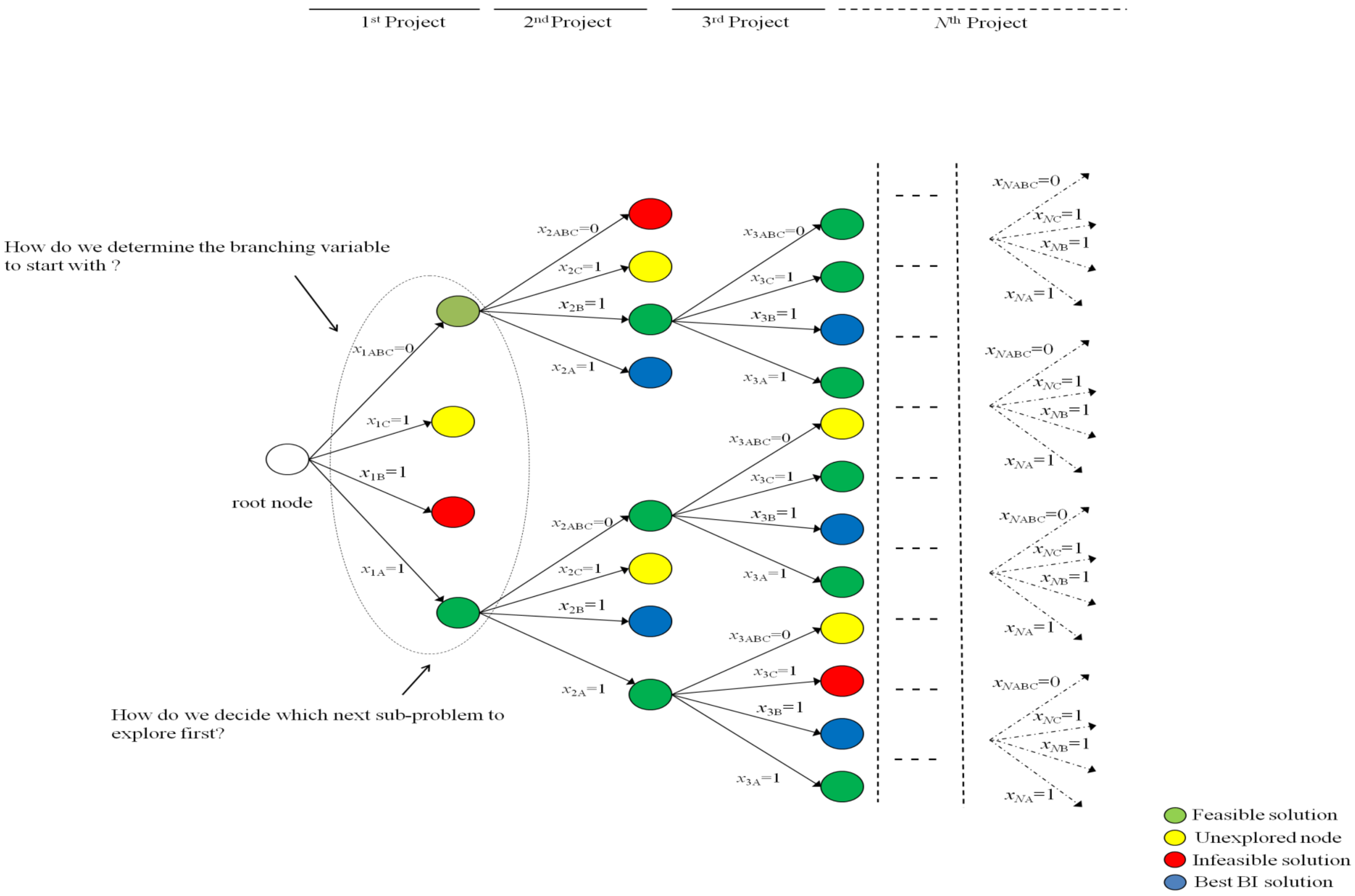


Figure 8: B&B Search Tree with more than Two Branches for BIP Problem in Multi-Group Project Allocation

The effects on CPU time, memory storage, and the solution gap's quality in the B&B method are obviously linked to the size of the tree. Undoubtedly, the branching strategies play a key factor in the B&B algorithm by significantly improving the solution performance and also speeding up the process of finding an optimal solution and node fathoming. No specific strategies work best for all problem types; otherwise no new strategies are required in this area. Therefore, creating new branching strategies on variables and choosing branching nodes to be explored during the next iteration greatly enhance the B&B algorithm's quality and bring our attention to contribute in proposing new branching strategy in this study.

Summary

This chapter has discussed MD, the classification criterion's cornerstone used to allocate projects among multiple options, and has introduced the structure of the algorithm applied in solving the models, B&B. The modeling approach for the constrained multi-group project allocation based on MD ensures assigning projects to their best option while maintains relevant resources. The core contributions of our solution approach in this dissertation are as follows:

1. Proposing models for selecting and allocating projects
2. Enhancing the optimal solution's quality by proposing clever branching strategies with node selections that can work efficiently with large problems.
3. Improving the B&B algorithm's solution performance

We will show in the following chapters our proposed branching strategy and its effect on the B&B solution performances.

CHAPTER IV
MATHEMATICAL MODEL AND PROPOSED METHODOLOGY

The analytical decision-support approach of selecting and allocating projects to either in-house or private contractors (multi-group) are subject to relevant resources or constraints. This approach is formulated mathematically using two different BIP models, which can be conceptualized as optimization procedures for constrained project selection and allocation problems. Each model, however, provides different types of performance measures or optimization criteria that will be discussed in this chapter. The B&B method is used to solve the mathematical models with a proposed branching strategy that can improve the algorithm performance's efficiency as well as handle a large size problem.

This chapter is divided in three sections. The first section describes the mathematical model formulation. The second section discusses the proposed methodology for branching strategy. The last section is this chapter's summary.

Model Formulation

Any decision-making problem concerning project selection and allocation involves only two choices corresponding to yes (allocate) or no (not to allocate) decisions. In this dissertation, these two choices involve whether or not to have available options develop individual projects. The two choices' corresponding mathematical description can be represented by decision variables that take binary or Boolean (0 and 1) values. Thus, our problem's mathematical formulation in this dissertation is BIP.

Modeling Assumptions

The required modeling assumptions for the problem of constrained project selections and allocations are as follows:

- Each project in set *n* (containing all projects for selection and allocation) has an implementation cost that is the same for in in-house and private contractors.
- Each project's cost includes the cost of labors (direct and indirect costs).
- Each project *n* has a dollar value of benefit, which is derived from the projects being developed that is the same for in in-house and private contractors.
- The in-house manpower cannot accomplish all projects *n*. Two levels of in-house manpower (man-hours) are assumed and varied when projects in set *n* are assigned to in-house or private contractors. This manpower can be used to either develop projects internally (in-house) or to supervise outsourced projects (private contractors).
- When project *n* is assigned to one of the private contractors, the in-house manpower becomes a supervision time and is assumed to be equal in all private contractor options.
- The budget available is not sufficient to fund all projects in set *n*.
- The number of projects to be allocated in any option is unlimited. However, each project *n* is allowed to be assigned to *at most* one of the option (in-house or private contractors).

Optimization Criteria

In this dissertation, the specified measure of performance drives the project selection for funding. We consider two types of performance measures to be optimized:

a) The total dollar value of benefits from those projects selected for funding.

b) The allocated projects' total value of MDs.

The first choice determines a subset of the projects to be funded that generates optimum benefits, while the second choice will result in a project classification that intending to allocate each project to its closest option.

As we mentioned in Chapter 1, this study proposes two BIP models for the project selection and allocation. The first mathematical formulation (Model 1) maximizes the dollar value of benefits the public derives from projects (transportation projects) being executed subject to a budget, total sum of MDs, and in-house manpower constraints. The second mathematical formulation (Model 2) minimizes the total sum of MDs subject to a budget and the in-house manpower constraints.

Notation and Definitions

The mathematical modeling for project selection and allocation in this dissertation is formulated using the following notation:

N number of projects

M number of private contractors

B available budget

c_{1n} cost of project n when completed in-house

c_{in} cost of project n when completed by private contractor i

b_n economic benefit from project n when completed in-house

b_{in} economic benefit from project n when completed by private contractor i

$$x_{1n} = \begin{cases} 1 & \text{if project} n \text{ is allocated to in - house} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{in} = \begin{cases} 1 & \text{if project} n \text{ is allocated to private contractor} i \\ 0 & \text{otherwise} \end{cases}$$

d_{1n} MD of project n to in-house option

d_{in} MD of project n to private contractor i option

W upper bound on total sum of MDs

T total manpower (man-hours) available

t_{1n} man-hours for finishing project n allocated to in-house option

t_{in} man-hours for supervising of project n allocated to private contractor i

Figure 9 displays the procedure in the proposed models' project allocation.

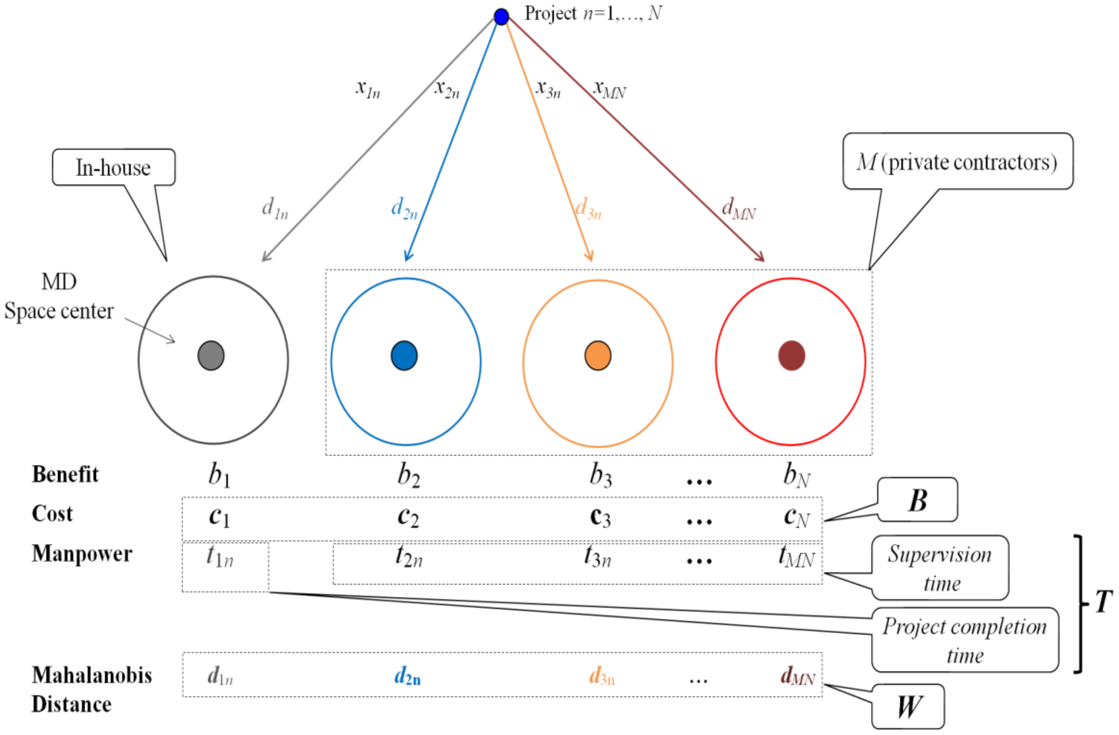


Figure 9: Representation of Constrained Multi-Group Project Allocation Using MD

In Figure 9, each project has the following characteristics:

- A binary decision variable that is represented by (x_{1n} =1 or x_{1n} =0) for in-house allocation and (x_{1n} =1 or x_{in} =0) for private contractor allocations.

- Total number of *M* +1 MDs that measure the MDs between each project *n* and allocation options (in-house d_{1n} and private contractors d_{in}). These MDs are minimized as an objective in model 2. The summation d_{1n} and d_{in} for all *N* projects and *M* +1 options represents the MDs upper bound *W*.
- Dollar value of project benefits b_n that is intended to be maximized in model 1.
- The summation of all project cost c_n for all *N* projects > the available budget represented by *B*.
- In-house man-hours required to finish project *n* when allocated to in-house option represented by t_{1n} and the amount of time needed to supervise project *n* when outsourced to private contractors denoted by t_{in}. The total sum of t_{1n} is > the available man-hours *T* in a transportation agency to complete all *N* projects.

Model 1

$$Maximize \quad \sum_{n=1}^{N} b_{1n}x_{1n} + \sum_{n=1}^{N}\sum_{i=2}^{M} b_{in}x_{in} \qquad \textbf{(7)}$$

Subject to:

$$\sum_{n=1}^{N} d_{1n}x_{1n} + \sum_{n=1}^{N}\sum_{i=2}^{M} d_{in}x_{in} \le W \qquad \textbf{(8)}$$

$$\sum_{n=1}^{N} c_{1n}x_{1n} + \sum_{n=1}^{N}\sum_{i=2}^{M} c_{in}x_{in} \le B \qquad \textbf{(9)}$$

$$\sum_{n=1}^{N} t_{1n}x_{1n} + \sum_{n=1}^{N}\sum_{i=2}^{M} t_{in}x_{in} \le T \qquad \textbf{(10)}$$

$$x_{1n} + \sum_{i=2}^{M} x_{in} \le 1 \qquad \forall (n = 1, 2, \ldots, N) \qquad \textbf{(11)}$$

$$x_{1n}, x_{in} \in \{0,1\} \quad \forall(\ n = 1, 2, \ldots, N \ \text{ and } \ i = \ 2, 3, \ldots, M) \qquad \textbf{(12)}$$

In the model 1 formulation, the objective function (7) gives the total benefit derived from the projects selected for funding to be maximized. Constraint (8) formulates upper-bound conditions for the total value associated with MDs for those projects selected for funding. The budget constraint (9) specifies the maximum level of funds available. Equation (10) enforces an amount of in-house manpower available to be used in in-house projects when the optimal solution allocates project *n* to in-house option, or supervising projects when the most attractive project *n* is assigned to any one of private contractors *i*. The restriction on assigning each project *n* to at most one allocation option is given in equation (11). Constraint (12) declares the binary decision variables.

To reach the most attractive optimal solution that ensures best project allocations, the model could be solved using an estimated initial value of *W*, for which a solution is feasible, and then continues solving the model by reducing *W* gradually until the solution becomes infeasible. *W's* last value that makes the solution feasible is considered to be the best upper bound of MDs summation that provides best optimal solution concentrating more on project allocations. Furthermore, this methodology can also involve a more complicated technique, such as Fibonacci search, search algorithm, and golden section search.

Model 2

$$Minimize \quad \sum_{n=1}^{N} d_{1n} x_{1n} + \sum_{n=1}^{N} \sum_{i=2}^{M} d_{in} x_{in} \qquad \textbf{(13)}$$

Subject to:

$$\sum_{n=1}^{N} c_{1n} x_{1n} + \sum_{n=1}^{N} \sum_{i=2}^{M} c_{in} x_{in} \geq B \tag{14}$$

$$\sum_{n=1}^{N} t_{1n} x_{1n} + \sum_{n=1}^{N} \sum_{i=2}^{M} t_{in} x_{in} \leq T \tag{15}$$

$$x_{1n} + \sum_{i=2}^{M} x_{in} \leq 1 \qquad \forall(n = 1, 2, \ldots, N) \tag{16}$$

$$x_{1n}, x_{in} \in \{0, 1\} \quad \forall(\ n = 1, 2, \ldots, N \ \ and \ \ i = \ 2, 3, \ldots, M) \tag{17}$$

The objective function (13) minimizes the all MDs total sum for allocated projects. This objective function cannot be minimized to zero because of the constraint structures in budget (14) and time (15). Equation (15) ensures that all selected projects in the optimal solution will be less than or equal the upper bound total time *T* available for the in-house manpower. Otherwise, the solution will be infeasible for both formulations. To be tight in an optimal binary integer solution, available budget *B* in constraint (14) must be less than or equal the total cost of all projects selected for funding. Otherwise, the actual budget spent will exceed the specified lower bound *B*.

In practical applications, various strategies of project selections and allocations are required over a specified range of *B*. The model for these strategies determines how a change *B*'s budgetary level can provide transportation departments with several project selection and allocation strategies for in-house versus private contractors. This strategic approach can also be considered when using model 1 formulation. Additionally, model 2 formulation can also include an upper bound on money that will be spent as in:

$$\sum_{n=1}^{N} c_{1n} x_{1n} + \sum_{n=1}^{N} \sum_{i=2}^{M} c_{in} x_{in} \leq B_1 \qquad \textbf{(18)}$$

where B_1 represents the budget amount available to sufficiently fund all projects.

Table 4 compares models 1 and 2, which are described above, to show what each model can provide.

Table 4: Comparison between model Formulations 1 and 2

Model Properties and Outputs	Model 1	Model 2
Focus on determining project subset that has optimal fitting allocation	No	Yes
Ensure best allocation	Yes	Yes
Select projects that produce maximum benefits	Yes	No
Used to select and allocate projects with limited budgets	Yes	Yes
Select and allocate projects for options that in-house manpower can accomplish	Yes	Yes

Proposed Branch-and-Bound Strategy

In our description of the B&B in Chapter 3, we highlighted two important questions regarding the branching strategy that considers the choice of the branching variable and node to solve. To the best of our knowledge, there is no specific or common branching rule can work best for any pure IP, MIP, and BIP problems. Subsequently, each state-of-art solver provides its own default or proposing setting that can decide which most suitable branching strategy works best based on the problem's structure and progress. Moreover, many state-of-arts solvers, such as CPLEX, LINDO, and MATLAB offer their users different choices to set up a specific strategy in variable and node selection. Many effective branching strategies have been studied and compared to others based on their performances and significant improvement on the search tree. For comprehensive descriptions of B&B strategies, we refer the reader to [15, 50-52], and the references their in. The proposed branching strategy for choosing the specific candidate variable (variable selection strategy) to split from each node and the particular node to explore (node selection strategy) in the B&B algorithm are described in the following details:

Branching Variable Strategy

To better understand the proposed strategy, the related assumptions and notations that will be used repeatedly when describing the proposed strategy on the B&B search tree must be identified. The assumptions and notations are as follows:

- Each node or sub-problem will be solved by its corresponding LP relaxation.
- Let k=1,2,…,K be the index of number of nodes. The feasible solution from each node k is maintained in a list L, which contains only the disbranched or unexplored feasible nodes and is updated automatically at each search-tree level.
- The feasible solution from k provides a lower bound (LB) on the global UB and is represented by z_k^{LB}.
- A specific node k found so far in L that provides a feasible solution z_k^{LB} satisfying BI constraint is called the incumbent solution (best solution) and denoted as $z^*_{x_n,\, x_{in} \in \{0,1\}}$.

- The B&B algorithm keeps solving each k and update best z^* based on maximum value, that is best $z^*=\max\left\{ \text{L}: {z^*}_{x,x_{in}\in\{0,1\}} \right\}$
- The algorithm will terminate when the gap between UB and LB equals 0 (UB-LB =0) or L= $\emptyset$. Also it can terminate when a certain value specifies the gap.

The conceptual structure of the proposed branching strategy in our problem is derived from the project allocation MD values as a classification criterion, which is highly important for project allocation, and the general guideline that suggests to branch initially on the most important variable [15, 53]. Therefore, we use the knowledge of this allocation problem based on MD to assign a branching priority to each candidate variable based on its minimum MDs and its corresponding project that also has minimum MD average. More formally, sort d_{1n} and d_{in} for each project n=1, 2, …, N in ascending order in set r=1,2,…,R as represented bellow:

$$d_{(i\text{-}1)11},\ d_{((i\text{-}2)21},\ldots,\ d_{(M)R1},\ d_{(i\text{-}1)12},\ d_{(i\text{-}2)22},\ldots,\ d_{(M)R2},\ldots,\ d_{(M)RN}$$

where the first subscript (i-1) represents the allocation option for either in-house or privet contractors, the second subscript (r) indicates the order position number of $d_{(i\text{-}1)}$, and the last subscript refers to the project number n. The next steps are to compute the arithmetic mean for each project n based on the sorted MD $d_{(i\text{-}1)rn}$, and then sort these averages in ascending order in set o=1,2,…,O. A representation of sorting the projects' MD averages is:

$$\text{Avg(MD)}_1,\ \text{Avg(MD)}_2,\ \text{Avg(MD)}_3,\ldots,\ \text{Avg(MD)}_O$$

To clarify the variable branching system's determination in the B&B tree, a pseudo code illustrating this strategy is shown below:

Variable Priority Selection Strategy Algorithm

```
For $n$=1,2,…,$N$                    % the projects repeatedly execute
        For $i$=1,2,…,$M$            % the options repeatedly execute
sort($d_{(m+1)n}$, ascend)           % sort each $d_{1n}$ and $d_{in}$ for each individual project
sort(Avg(MD)$_n$, ascend)            % sort each project MD's average in ascending order
        Branch on sorted $x_{in}$
        Return to $n$
Return to $i$
```

The proposed variable branching strategy based on the sorted $d_{(i\text{-}1)ro}$ is depicted in Figure 10. In this figure, the priority in branching starts with the candidate variable $x_{(i\text{-}1)11}$ corresponding to the least $d_{(i\text{-}1)ro}$ in both sets of r and o . Branching continues for the candidate variable $x_{(i\text{-}1)ro}$ that has the least $d_{(i\text{-}1)r}$ value in every sorted project in set o; the B&B search tree is directed to the most promising BI solution $z^*_{x_n,\, x_{in}\in\{0,1\}}$ that maximizes the LB (in a maximization problem). Therefore, the specific ordering in the proposed branching can significantly affect the speed of achieving good solutions in less time and with less searching. The B&B algorithm will solve the LP relaxation in every node k in each sorted project o and record the solution z_k^{LB} from each node k in the list L and then moves to the next sorted project o+1.

Node Selection Strategy

Before going to the next sorted project (o+1), the B&B algorithm has to be guided to choose a node k from the list L to branch. Several node-selection strategies are available in many solvers to be set up. These node-selection strategies can considerably affect to

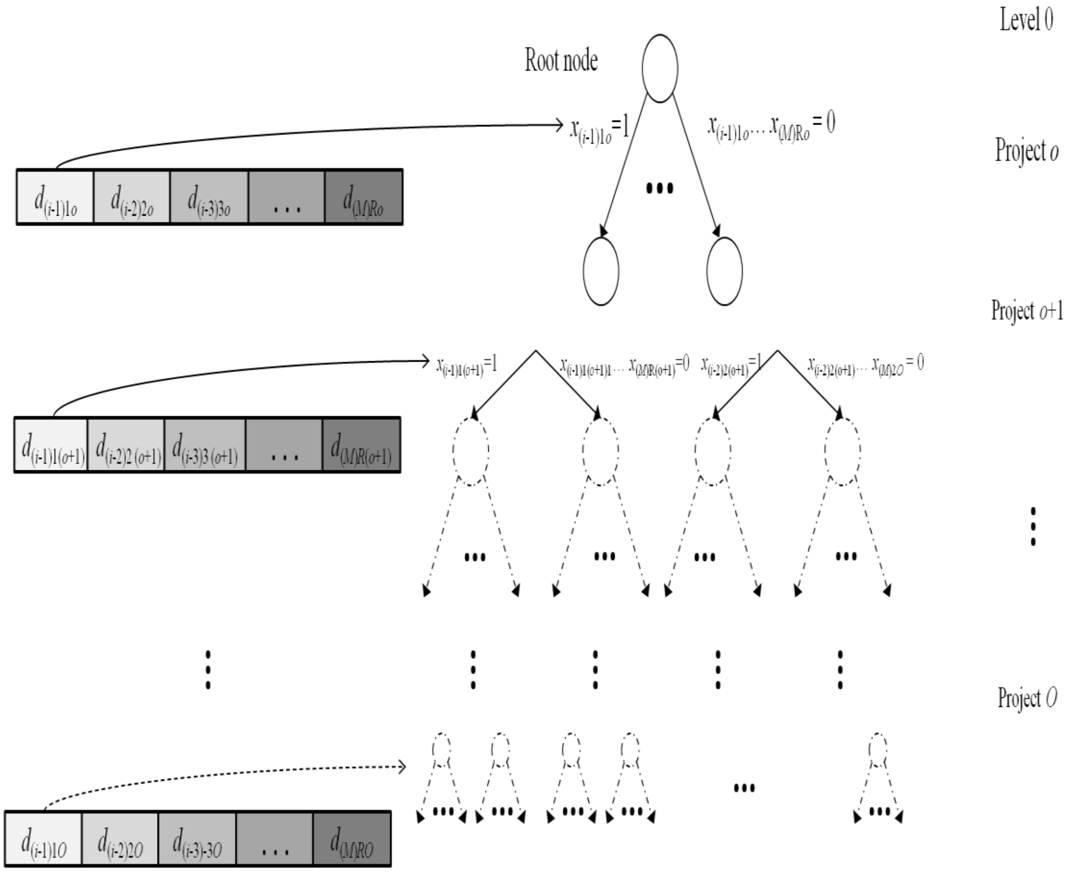


Figure 10: Proposed Variable Branching Scheme

the algorithm's performances and provide different improvements to the solution quality in reducing the gap between UB and LB. Once again, no specific strategy works best for all type of problems, and each IP solver provides various results.

The suggested node-selection strategy chooses a node *k* from the list L that has the most (or least if the problem is a minimization) z_k^{LB} value that is close to the root node's UB. This method is called best-bound search and focuses on the most promising node *k* that leads the search tree to obtaining the best $z^*_{x_n,\, x_{in} \in \{0,1\}}$ for the whole problem. This strategy is preferred to work efficiently with large problems [11, 53]. To better understand this strategy's representation, we show the selection process:

$$k_{j+1} = k_j \mid \max\{z_k^{LB}\} \qquad \textbf{(19)}$$

where k_j represents the branching node at sequence *j*.

This strategy was selected carefully based on its remarkable performance after being compared with various node selection strategies that MATLAB (e.g., minimum infeasible and best projection) provides in numerical experiments with different problem sizes. This strategy is illustrated in Figure 11.

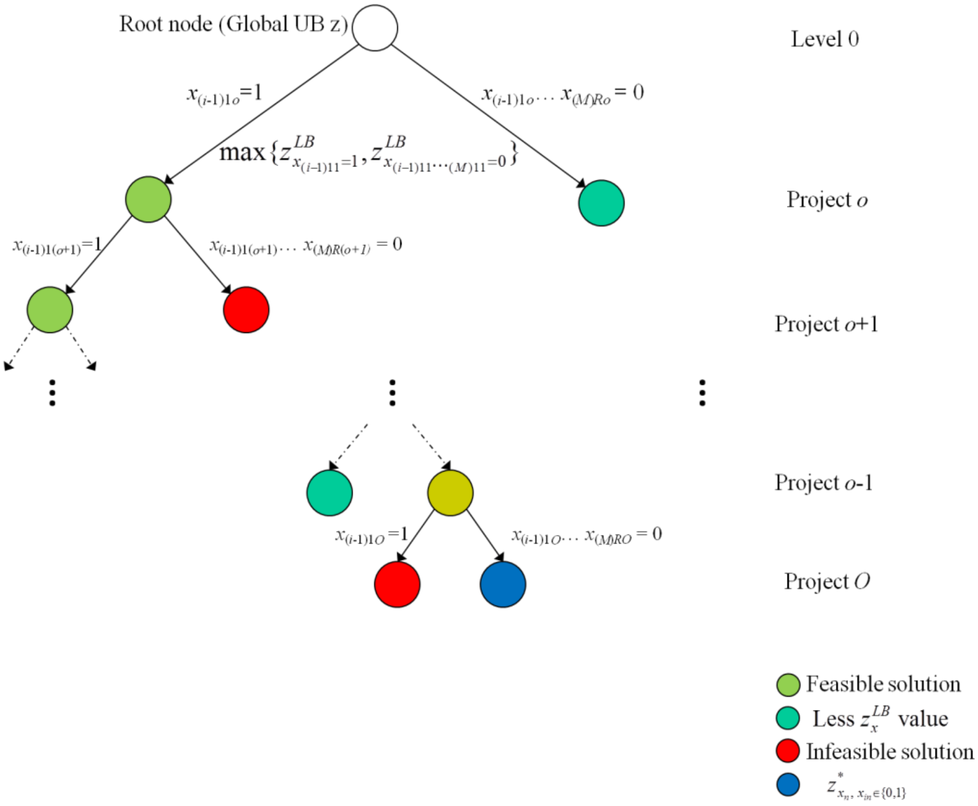

Root node (Global UB z)
Level 0
$x_{(i\text{-}1)1o}$=1
$x_{(i\text{-}1)1o} \cdots x_{(M)Ro} = 0$
$\max\{z^{LB}_{x_{(i-1)11}=1}, z^{LB}_{x_{(i-1)11\cdots(M)11}=0}\}$
Project *o*
$x_{(i\text{-}1)1(o+1)}$=1
$x_{(i\text{-}1)1(o+1)} \cdots x_{(M)R(o+1)} = 0$
Project *o*+1
Project *o*-1
$x_{(i\text{-}1)1O}$=1
$x_{(i\text{-}1)1O} \cdots x_{(M)RO} = 0$
Project *O*
Feasible solution
Less z_x^{LB} value
Infeasible solution
$z^*_{x_n,\, x_{in} \in \{0,1\}}$

Figure 11: Illustration of Best-Bound Node Selection Strategy

The B&B pseudo code for selecting nodes based on the best-bound strategy can be stated as follows:

Best-Bound Node selection strategy Algorithm

```
Input: BIP              %the BIP problem (Maximization for instance).
r=1,2,…,R               % the list of sorted $d_{1n}$ and $d_{in}$ for each project n.
o=1,2,…,O               % the list of sorted Avg(MD)$_O$ each project n.
  0. Initialize: set L={BIP}, z*= -∞ .
  1. Check: if L=∅, if so STOP and return to best z*.
  2. Branch on sorted variable in set r and Select k from L that has max $z_k^{LB}$ and
     remove k from L (L: =L\{k}).
  3. Solve LP relaxation for k , and check the solution feasibility of $z_k^{LB}$ , if
     infeasible go to step 1, otherwise set $z_k^{LB}$ as optimal solution for k .
  4. Apply the fathoming steps for step 3: if $z_k^{LB} \leq Z^*_{x_n,\, x_{in} \in \{0,1\}}$, go to step 1; if
     $z_k^{LB}$ is not BI go to the step 2, else set $z_k^{LB}$ as new best $Z^*_{x_n,\, x_{in} \in \{0,1\}}$ and go to
     step 1.
```

Summary

The mathematical models discussed in this chapter are intended to provide the project manages or decision makers with different conclusions regarding the objective function. These models, however, are consistent in providing optimal set of selected and allocated projects among different options based on the given constraints of the available resources' constraints. The proposed B&B strategy in selecting candidate variables for branching and nodes for exploration aims to effectively enhance the B&B algorithm within the search tree in the following ways:

- Reduce the gap between the UB and LB by increasing the LB and decreasing the UP.
- Speed up the convergence between the UB and LB.
- Concentrate on the nodes that reduce the gap between the UB and LB.
- Decrease the search tree's size, thus shortening CPU time.
- Expend less effort in finding the best incumbent solution in a shorter way and a smaller tree size.
- Reduce the dimension of the decision variables by avoiding their aggregate use through focusing on variables with the most important allocation options.
- Enhance the performance of nodes' fathoming speed.
- Avoid exploring superfluous nodes.
- Guide the algorithm for consistent search behavior by using the proposed combination of the branching variable and node-selection strategies.

In later chapters, we will numerically explain how the proposed B&B strategy improves the solution performances in both models.

CHAPTER V
COMPUTATIONAL DESIGN AND APPLICATION

This chapter discusses the application of the models and the B&B branching strategy to a state's Department of Transportation projects. The application purpose is to provide decision makers or project managers with the best sets of allocated projects among several options. An experimental design is discussed to compare the B&B strategy's solution performance with of the MTALB default strategy. This experimental design will be run with various combinations of several project numbers and allocation options.

To evaluate the B&B algorithm, we use various performance-assessment tools for measuring CPU solution time, absolute gap between the UB and LB on the objective function, percentage of relative gap, and number of explored nodes in model 2's B&B search tree.

The MD calculations, model formulation 1, and model formulation 2 are coded and solved using the MATLAB© language R2014a and are run on a PC with the following features: Intel Core i5 processor, 6.00GB RAM, and 64-bit Windows 7 operating system.

This chapter is divided into four sections. Section one describes the application's problem. Section two discusses the experimental design. Section three presents the application's solution procedures. Section four is the chapter's summary.

Problem Description

The US Department of Transportation (DOT) is sixth in the federal budget's top-ranking estimated total outlay with an average of 97.8 billion dollars in 2015 (out of 289.1 billion dollars spent by federal, state, and local governments combined) and with an estimated increase of 3% each fiscal year [54]. Moreover, the federal government's investments in improvements of highways, mass transit systems, airports, railroads, and water systems was approximately 65.85 billion dollars in fiscal year 2013 [55]. Figure 12 depicts the total combined spending in billion dollars on the US DOT for the last ten years [54].

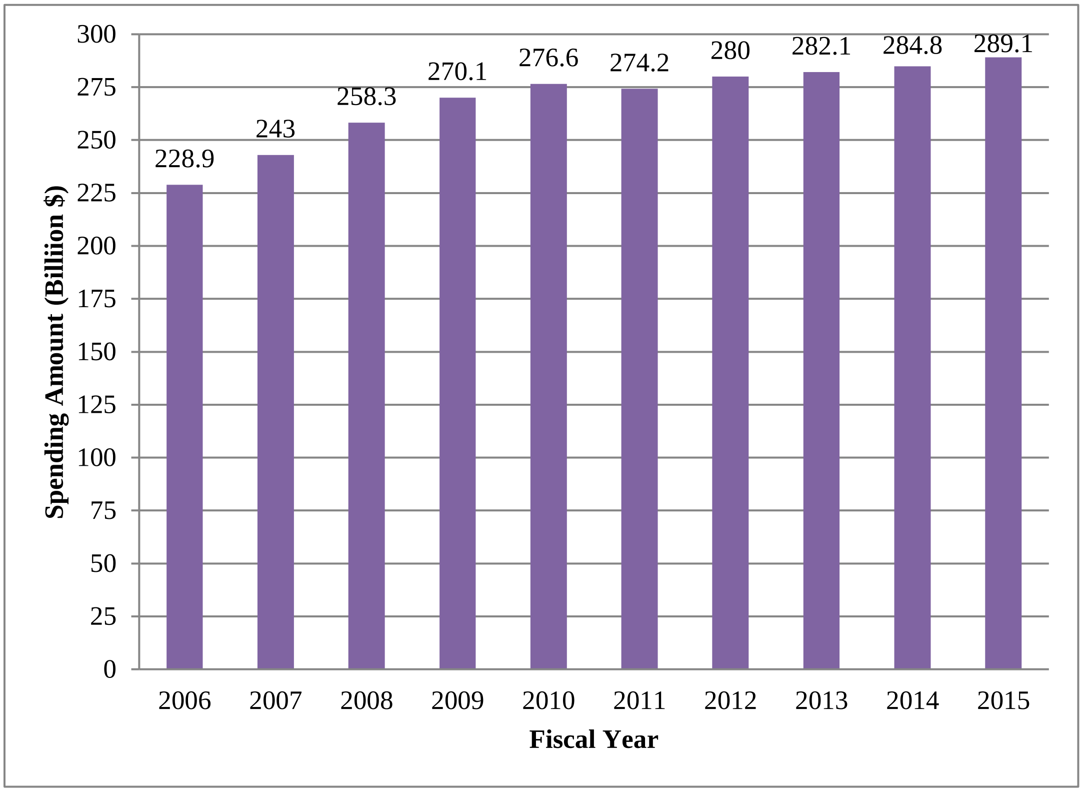


Figure 12: Total Spending Amount on US Department of Transportation from 2006 through 2015

As shown in Figure 12, the total spending on the US DOT has increased over the last ten years, with a slight dropping in 2011. Thus, to address budgetary concerns, optimal methodology should be used for allocating projects.

Application Study

In this dissertation, an application involving seal coat projects has been developed to aid states' DOT project managers in selecting transportation projects and allocating them to either in-house or private contractors.

Each year state DOTs are involved in such projects as construction, reconstruction, rehabilitation, preventive maintenance for roadways, safety enhancement, passenger and freight railroads, transit and commuter services, bikeways, aviation systems, waterways and canals, port districts, and inter-modal facilities. Considerations for common types of projects in State DOTs are as follows:

- Structure-replacement projects

  Replacement projects completely rework existing transportation systems (e.g., highways, bridges, railroads, and tunnels) that are considered functionally obsolete or structurally deficient.

- Safety and Improving Projects

  Safety and improving projects involve enhancing and maintaining quality in all areas for which State DOTs are responsible. For example, improve safety routes for school projects, install automatic railroad warning equipment at railroad crossings, replace existing grade crossings with new interchanges to improve safety and provide efficient traffic movement, and reduce traffic noise affecting communities living beside highways by installing noise-absorptive ceiling panels.

- Rehabilitation Projects

  Rehabilitation projects restore existing transportation facilities (e.g., pavements, bridges, and rest areas) to their original condition.

- Maintenance Projects

  Maintenance projects are either routine (e.g., sealing cracks and filling pot holes,) or preventive, working to preserve (e.g., asphalt concrete pavement covering roads 2-inch thick to resist pavement deterioration from accumulated traffic loads and environmental effects).

Several coordinates or factors measure each project *n* in states' DOTs. The following factors are significant in identifying ideal or most appropriate seal coat projects for either in-house or private-contractor options [56]:

- Average daily traffic (ADT)
- Gallons of asphalt used (GOFA)
- Number of cubic yards of aggregate (CYA)
- Length of project (in miles) (MIL)
- Total area sealed (in square yards) (TYS)
- Total direct cost of the project (TCOS).

These factors are essentially considered in constructing any seal coat project regardless of its locations. This dissertation is not limited for specific transportation project type. In other words, it can be applied on other project type of transportation project. In other words, it can be applied to other project types.

Models' Resources Estimation

Project's benefits

Each transportation-project category has benefits affecting public users, society, and economic productivity. These benefits are often defined and measured in terms of cost reduction resulting from their development (e.g., transportation accidents, air pollution, travel time, vehicle cost saving) [57]. Table 5 lists different project types' benefits [58-60].

Several models and software tools are available to estimate and calculate the transportation projects' benefits. These tools include Sketch Planning Analysis Spreadsheet Model (SPASM), Surface Transportation Efficiency Analysis Model (STEAM), and Life-Cycle Benefit-Cost Analysis Model. For more descriptions of and discussions about these models, refer to [58, 61] and references therein.

In this dissertation, we refrain from using a complicated model to evaluate the projects' benefits. Nevertheless, to estimate the project's benefits considered in the study's application, we approximate each project benefit b_n using a certain percentage of its cost, 25%, as a net profit. Specifically, each project n's benefit b_n is estimated from the derivation of its corresponding cost c_n by the following equation:

$$b_n = 0.25 \times c_n \qquad \forall(n = 1,2,\cdots,N) \tag{19}$$

The project cost c_n equals to the direct project cost (TCOS). We would like to emphasize that this percentage is arbitrarily suggested for the purpose of computing the application.

Available Budget

In all DOTs, the available budget for developing any type of projects is insufficient to fund all of them at once. In this application, the budget is assumed to be 80% of the total project costs as in the following equation:

$$B = 0.8 \times \sum_{n=1}^{N} c_n \tag{20}$$

Table 5: List of Transportation System Benefits

Targeted Beneficiaries	Project type	Derived Benefit
Public users and society	Roadway expansion	Reduce travel time, vehicle costs, and traffic congestion
	Roadway safety improvement	Reduce vehicle crashes, injures, and fatalities
	Construction of new routes providing increased transportation network connectivity	Reduce industrial transportation or logistic costs which affect economic productivity.
	New construction and transportation system improvements	Reduced air pollution from vehicle or transportation emissions and improved air quality by reduced time in congestion.
Economy and government	New construction in reducing automobile using	Reduce parking problems and related costs, roadway maintenance and facility costs, and energy costs.
	All project types	Increase jobs and reduce unemployment and increase tax revenue

As mentioned in Chapter 4's discussion of modeling formulation, different budgetary levels can be used to estimate different scenarios for project selection subsets.

Available man-hours

The required man-hours (hrs) to accomplish seal coat projects is estimated based on several real case studies and specialized contractors' websites [62-65]. To realistically estimate man-hours per sealed area in square yard (sy^2), we take the average of man-hour needed to complete 50 sy^2 in these studies, which approximately equals to 1 hr/50 sy^2.

Therefore, the unit man-hours required for in-house engineers to complete each project *n* in our application is determined by dividing each project *n*'s TYS by 1 hr/50 sy^2:

$$t_n = \frac{TYS \text{ sy}^2}{1 \text{ hr}/50 \text{ sy}^2} \qquad \forall \; n = (1, 2, \ldots, N) \tag{21}$$

The supervision time t_{in} is estimated as 10% of t_n. As it has been mentioned, that the time available *T* for DOT to develop all projects *N* is not enough; therefore, we estimate this resource as 60% of the aggregate time needed to complete all projects *N*:

$$T = 0.6 \times \sum_{n=1}^{N} t_n \tag{22}$$

This percentage is assumed subjectively, and can be treated as different time levels or percentages providing different scenarios in project selection and allocation sets.

Experimental Design

The application in this study is coded in the MATLAB coding language and applied in the MATLAB computing environment. MATLAB provides one of the fastest and most efficient optimization toolboxes (e.g., IP, MIP, nonlinear programming, stochastic programming, etc.) for solving various optimization problems. Additionally, MATLAB solves IP or MIP optimization problems with several stages to ensure achieving high reliable and efficient optimal solution. These stages test the optimization model structure's capability and solve the IP problems by several techniques before immediately solving the IP problems using the B&B algorithm. These stages are as follows:

1. LP processing which reduces or eliminates redundant decision variables and constraints

2. Solve the IP problem by LP relaxation
3. IP processing
4. Cutting generation or valid inequality constraints to be added to the IP problem
5. Using various significant heuristics to improve the UB
6. Using B&B algorithm.

For comprehensive discussion see [14, 15, 66].

MATLAB provides default settings on the B&B algorithm as shown in Table 6.

Table 6: List of Branching Rule Settings in the B&B Algorithm Provided in MATLAB

Setting	Description	Default setting
Variable Branching Rule	*Maximal pseudo-cost*: choose a branching variable with maximal pseudo-cost variable *Most fractional*: choose a branching variable furthest from being integral or its fractional part closest to 0.5 *Maximal corresponding objective value*: choose a branching variable that corresponds to a maximal objective component in the objective function	Maximal pseudo-cost
Node Selection Rule	*Best or simple projection criterion*: select the node to explore next that may lead to good or most promising feasible solution *Best-bound*: select the node that has the most(least if the problem is minimization) objective value *Minimum sum of infeasibilities*: choose the node with minimum sum of its integer infeasibilities (maximum when the problem is a maximization)	Simple projection

For complete available settings and options in MATLAB regarding the stages discussed for solving the IP problems, please see Appendix 1.

Two-Factor Factorial Experiments

To examine the best selected and allocated project sets' solution results and to compare the performance behavior of the proposed B&B strategy with that of the MATLAB's B&B default branching strategy in the suggested BIP models, we run factorial experiments involving two factors: (1) the number of projects (*N*); (2) the number of options (*M*), where the in-house option is included. There are five levels (groups) *n* of factor $N \in \{20,40,60,80,100\}$ and three levels *m* for factor $M \in \{5,10,15\}$. Each combination of *nm* has *n*×*m* BIP decision variables and is replicated three times.

In order to eliminate any error occurring because of computer processing or other related noises and enhancing the resulting quality, the computed results are based on each *nm* replication's averages.

In each replication we compute the following responses:

- Number of projects allocated in the best set (*B.set*) (i.e., number of projects allocated on in-house, number of projects allocated on private contractors, and number of non-funding projects)
- Computer computation time (CPU) in seconds
- The absolute gab (abgp) and relative gap (%abgp) measure the difference between the best LB and UB obtained in the B&B search.
- Number of explored nodes (*nds*) in the B&B (in model 2).

This two-factor factorial design is depicted in Table 7.

Table 7: Two-Factor Factorial Design for the DOT Application

	Factor *M*						
	Levels	5		10		15	
Factor *N*	20	*B.set* *abgp* *nds*	*CPU* *abgp%*	*B.set* *abgp* *nds*	*CPU* *abgp%*	*B.set* *abgp* *nds*	*CPU* *abgp%*
	40	*B.set* *abgp* *nds*	*CPU* *abgp%*	*B.set* *abgp* *nds*	*CPU* *abgp%*	*B.set* *abgp* *nds*	*CPU* *abgp%*
	60	*B.set* *abgp* *nds*	*CPU* *abgp%*	*B.set* *abgp* *nds*	*CPU* *abgp%*	*B.set* *abgp* *nds*	*CPU* *abgp%*
	80	*B.set* *abgp* *nds*	*CPU* *abgp%*	*B.set* *abgp* *nds*	*CPU* *abgp%*	*B.set* *abgp* *nds*	*CPU* *abgp%*
	100	*B.set* *abgp* *nds*	*CPU* *abgp%*	*B.set* *abgp* *nds*	*CPU* *abgp%*	*B.set* *abgp* *nds*	*CPU* *abgp%*

We run this experiment with each *nm* combination on both proposed models that containing the proposed branching strategy and MATLAB's default branching strategy. Therefore, the number of runs for each model is 90, with a total of 180 runs in the experiments.

Data Set Generation

One of the most importation components in this application is the data set since the specific project type described in this dissertation and other related modeling applications require a realistic data set to obtain reasonable results. Consequently, the data for this application is generated based on various descriptive statistics of real data set from a previous study. The data-generation process is one of the major difficulties in conducting this application because the linear relationships between the data's variables are found significantly correlated. Furthermore, it is important to ensure that the generated data set simulates the real data in terms of statistical parameters (e.g., mean vector, covariance matrix, and correlation coefficient). It is also essential to generate different data sets or samples for all allocation options (historical, reference, data for MD computation purposes) that have unequal covariance matrices (heterogeneous) for these samples to have accurate results.

Steps in Data Set Generation

The data set is generated by using the Statistics and Machine Learning Toolbox in MATLAB. The mathematical and statistical details for generating the data set are out of this dissertation's scope. Instead, the simulation procedures can be summarized as follows:

1. Hypothetically test the multivariate correlation between the variables to determine if linear relationships exist between each pair of variables.
2. Compute several descriptive statistics: the mean vector $\boldsymbol{\mu}$, the correlation coefficient matrix $\boldsymbol{\rho}$, and the covariance matrix $\mathbf{S}$ for the real data set.
3. Simulate the data set based on its mean vector and covariance matrix.
4. Ensure that the linear relationships between the variable in the simulated data set are significantly correlated and analogous to the correlation strength or coefficient of the original data set.
5. Ensure that the mean vectors across all selected samples of level *m* are not equals or significantly different, i.e., $\boldsymbol{\mu}_1 \neq \boldsymbol{\mu}_2 \neq \mathrm{K} \neq \boldsymbol{\mu}_M$. The equality of means from different level *m* can be tested by using one of the mean victors' homogeneity

tests in the Multivariate Analysis of Variance (MANOVA), such as Wilks' and Lawley-Hotelling's tests.

6. Ensure that the covariance parameters across all selected samples of level *m* have different covariance matrices (heterogeneous) (i.e., $\mathbf{S}_1 \neq \mathbf{S}_2 \neq \mathrm{K} \neq \mathbf{S}_M$). This testing can involve one of the covariance homogeneity tests, such as Box’s M and Levene's tests.

The linear correlation between the variables is tested through the Pearson correlation test at level of significance α=0.05 and found significantly correlated. Consequently, the data set are simulated randomly by using MATLAB from the correlated multivariate normal distribution with a specific mean vector $\boldsymbol{\mu}$, and a covariance matrix $\mathbf{S}$.

Application Solution Procedures

Solving the application's problem in models 1 and 2 involve the following procedures:

1. Select a random sample (observations) for each level *n* of factor *N* from the generated data set.
2. Select a random sample for each option in level *m* of factor *M* from the generated dataset.
3. Compute the *MDs* between each project in level *n* for factor *N* and all options in level *m* for factor *M*.
4. Compute the *MD* average for each level *n*.

5. For each combination of *nm*, compute project benefits by using Eq. (19), available budget by using Eq.(20) , man-hour time (in-house and supervision) by using Eq.(21), and total manpower (man-hours) available by using Eq.(22).
6. For each project in level *n*, sort all options in level *m* in ascending order based on the *MD* values.
7. Sort all projects in each level *n* in ascending order based on the computed *MD* average from step 3.
8. Reorder the models' parameters and decision variables that steps 7 and 8 affect.
9. Solve the models for every combination *nm*.

Figure 13 illustrate the process of selecting random samples for factor *N*'s levels *n* (step 1) and the generated data set for factor *M*'s levels *m* (step 2).

Figure 14 depicts the sorting process for four projects and three options and shows the resulting changes in models' parameters.

To facilitate sorting project *n*, a specific color for each project number is assigned to distinguish and trace the changing position.

As illustrated in Figure 14, each project's cost and benefit is not affected by option *m* sorting (step 7), which identify particular affected changes in the sorting process.

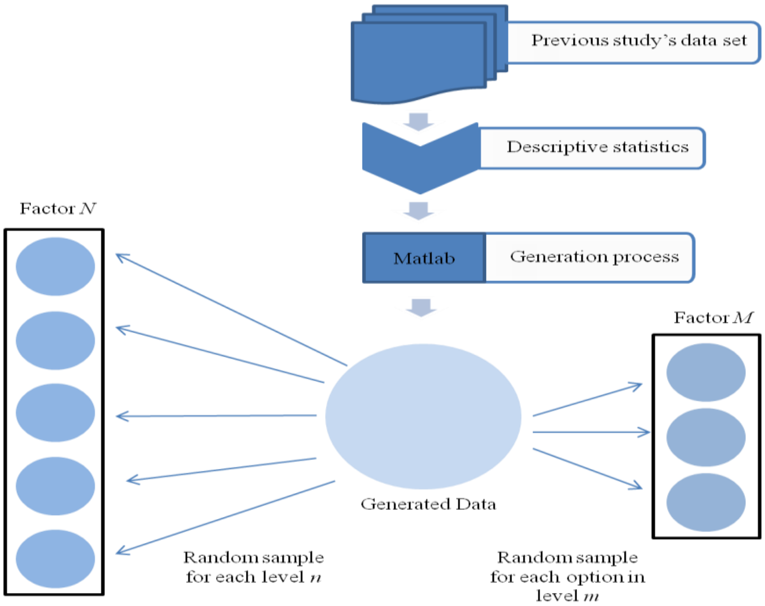
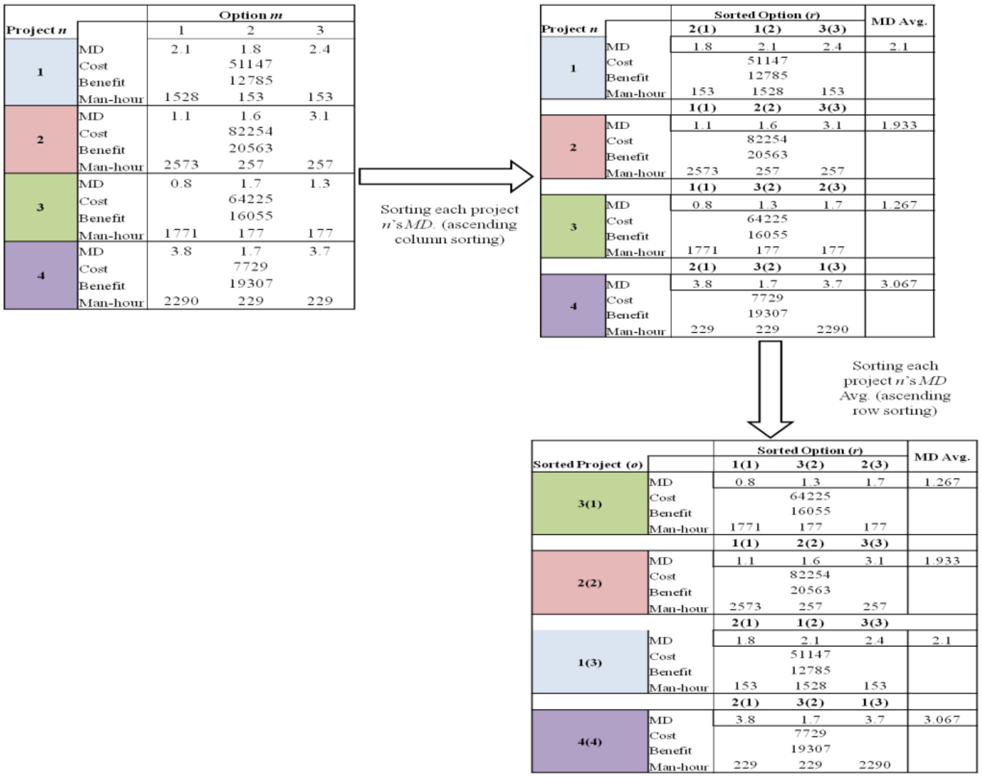

Previous study's data set
Descriptive statistics
Matlab
Generation process
Factor N
Factor M
Generated Data
Random sample for each level n
Random sample for each option in level m

Figure 13: The Process of Selecting Levels in Factor *N* and *M*



Project n		Option m		
		1	2	3
1	MD	2.1	1.8	2.4
	Cost		51147	
	Benefit		12785	
	Man-hour	1528	153	153
2	MD	1.1	1.6	3.1
	Cost		82254	
	Benefit		20563	
	Man-hour	2573	257	257
3	MD	0.8	1.7	1.3
	Cost		64225	
	Benefit		16055	
	Man-hour	1771	177	177
4	MD	3.8	1.7	3.7
	Cost		7729	
	Benefit		19307	
	Man-hour	2290	229	229

Sorting each project n's MD. (ascending column sorting)

Project n		Sorted Option (r)			MD Avg.
		2(1)	**1(2)**	**3(3)**	
1	MD	1.8	2.1	2.4	2.1
	Cost		51147		
	Benefit		12785		
	Man-hour	153	1528	153	
		1(1)	**2(2)**	**3(3)**	
2	MD	1.1	1.6	3.1	1.933
	Cost		82254		
	Benefit		20563		
	Man-hour	2573	257	257	
		1(1)	**3(2)**	**2(3)**	
3	MD	0.8	1.3	1.7	1.267
	Cost		64225		
	Benefit		16055		
	Man-hour	1771	177	177	
		2(1)	**3(2)**	**1(3)**	
4	MD	3.8	1.7	3.7	3.067
	Cost		7729		
	Benefit		19307		
	Man-hour	229	229	2290	

Sorting each project n's MD Avg. (ascending row sorting)

Sorted Project (o)		Sorted Option (r)			MD Avg.
		1(1)	**3(2)**	**2(3)**	
3(1)	MD	0.8	1.3	1.7	1.267
	Cost		64225		
	Benefit		16055		
	Man-hour	1771	177	177	
		1(1)	**2(2)**	**3(3)**	
2(2)	MD	1.1	1.6	3.1	1.933
	Cost		82254		
	Benefit		20563		
	Man-hour	2573	257	257	
		2(1)	**1(2)**	**3(3)**	
1(3)	MD	1.8	2.1	2.4	2.1
	Cost		51147		
	Benefit		12785		
	Man-hour	153	1528	153	
		2(1)	**3(2)**	**1(3)**	
4(4)	MD	3.8	1.7	3.7	3.067
	Cost		7729		
	Benefit		19307		
	Man-hour	229	229	2290	

Figure 14: Illustration of Sorting Steps in the Solving Process

Summary

In this chapter, the application's problem was discussed, and the application's experimental design and the application solution procedures were described. The seal coat projects’data was obtained from an evaluation study for Texas Transportation Institute and simulated in MATLAB R2014b. We used several statistical solvers (including JMP 11.1, SAS Enterprise Guide 6.1, SPSS 22, and MINITAB 16) based on their statistical tools to examine the variables correlation, underlying destitution families, and covariance homogeneity.

CHAPTER VI
COMPUTATIONAL RESULTS AND ANALYSIS

This chapter presents the computational experiments' results for the solution to the purposed project allocation models used with the application presented in the previous chapter. The experiments' goal was to compare the models' performance measures in terms of CPU time, absolute gap, relative gap, and number of explored nodes. In addition, showing different allocation sets from the solution of the default and the purposed branching strategies.

The computational time limit for solving the models was not set to a specific amount time; however, the maximum number of explored nodes in the B&B search tree was set to 3 million nodes in model 1. All other performance measures (e.g., absolute, relative gaps, and other solvers' options) were unchanged.

Data Input and Model Parameters

The generated data set for the experimental computation were used similarly in both formulations; that is, the multivariate observations in each level *n* for factor *N* and the random sample that represents each option in level *m* for factor *M* were identical in computations for formulation 1 and 2. Consequently, the available number of resources in budget *B* and the total manpower *T* were equal in both models for each *nm* combination. The applications' model parameters or resources (*W*, *B*, and *T)* with the total number of variables (x_{1n} and x_{in}) for each combination in formulations 1 and 2 are shown in Tables 8 and 9, respectively.

Table 8: Quantity of Available Resources and Number of Variables in Formulation1

N	*M*	*W*	*B*	*T*	No. Variables
20	5	180	809,300	22,700	100
	10	430			200
	15	720			300
40	5	290	2,211,404	55,274	200
	10	1,200			400
	15	2,000			600
60	5	930	3,531,000	85,830	300
	10	2,000			600
	15	3,300			900
80	5	1,060	4,007,813	102,024	400
	10	2,300			800
	15	3,780			1,200
100	5	1,190	4,640,340	116,930	500
	10	2,550			1,000
	15	4,160			1,500

Table 9: Quantity of Available Resources and Number of Variables in Formulation2

N	*M*	*B*	*T*	No. Variables
20	5	809,300	22,700	100
	10			200
	15			300
40	5	2,211,404	55,274	200
	10			400
	15			600
60	5	3,531,000	85,830	300
	10			600
	15			900
80	5	4,007,813	102,024	400
	10			800
	15			1200
100	5	4,640,340	116,930	500
	10			1000
	15			1500

As shown in Tables 8 and 9, the data set for the level combinations *nm* in formulation 1 equals to its corresponding combination *nm* in formulation 2. Furthermore, when changing the factor *M* levels at each level *n*, the resources *B* and *T* remain unchanged because these resources are derived only from the number of projects (level *n*). On the other hand, changing in factor *M* levels at each level *n* changes the amount of resource *W* in formulation 1 because that resource increases as the number of options or contractors increases. All the data sets for the projects needing to be allocated are provided in Appendix 2.

Because a one-to-one mapping exists between model and formulation, we use these terms interchangeably.

Model Results and Analysis

For each problem formulation's result, we initially present the best project allocation subsets that are obtained from every experimental combination *nm* by solving or applying the default branching strategy, which the MATLAB solver provided, and the proposed branching strategy. Secondly, we show the solution performance comparison studies between the MATLAB default branching strategy and the proposed branching strategy.

Best Project Allocation Results with Model 1

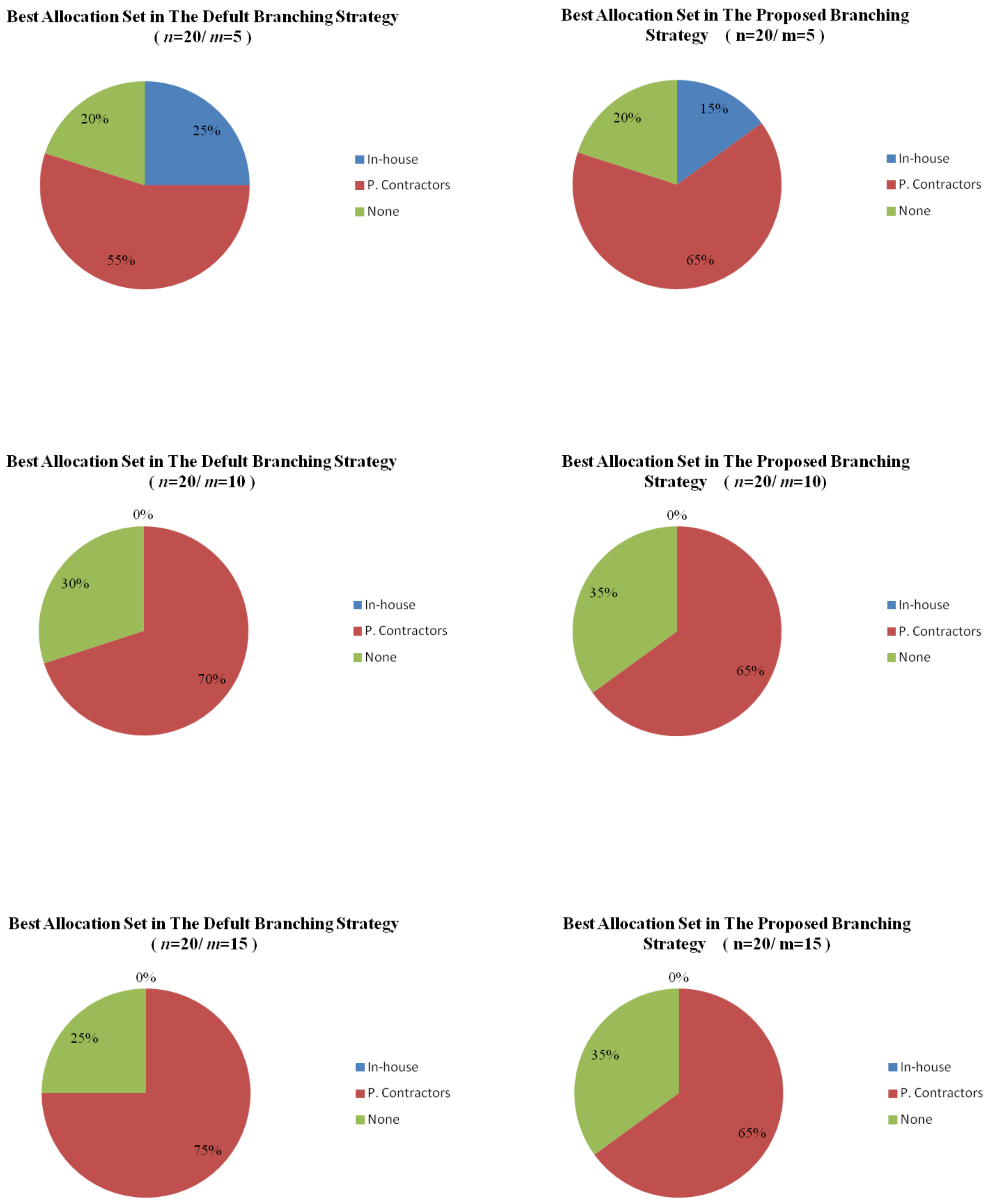
The results of the best project allocation subsets for each *nm* combination are shown in Table 10, a result from merging two tables to facilitate the comparisons. The first table displayed the default branching strategy's best project allocation outcomes; the second table presented the proposed branching strategy's best allocation results. The first and second columns in Table 10 indicate the factors *N* and *M* levels, respectively, which jointly indicate the project and option combinations *nm*. The number of the best projects allocated to the in-house option, private contractors, and non-allocated projects (infeasible to be funded) is represented in the third, fourth, and fifth columns, respectively.

Table 10: Results of the Best Allocated Project by Solving Formulation 1

N	*M*	Default Branching Strategy			Proposed Branching Strategy		
		In-house	**P. Contractors**	**None**	**In-house**	**P. Contractors**	**None**
20	5	5	11	4	3	13	4
	10	0	14	6	0	13	7
	15	0	15	5	0	13	7
40	5	9	23	8	4	24	12
	10	0	32	8	1	29	10
	15	0	24	16	0	29	11
60	5	12	37	11	10	35	15
	10	0	39	21	0	38	22
	15	0	36	24	0	38	22
80	5	14	53	13	23	45	12
	10	2	59	19	1	62	17
	15	4	56	20	0	57	23
100	5	21	52	27	13	55	32
	10	1	67	32	1	62	37
	15	0	65	35	1	61	38

As shown in Table 10, the allocation subset in each *nm* combination of the default branching strategy is slightly different from its corresponding combination in the proposed branching strategy. The inconsistency of obtaining different allocation sets with different branching strategy for the same *nm* combination is regarded as a phenomenon of our combinatorial problem.

To clarify the differences between the best allocations from both strategies for each combination, these allocations are depicted in Figures 15-19.



Best Allocation Set in The Defult Branching Strategy (*n*=20/ *m*=5)

Best Allocation Set in The Proposed Branching Strategy (n=20/ m=5)

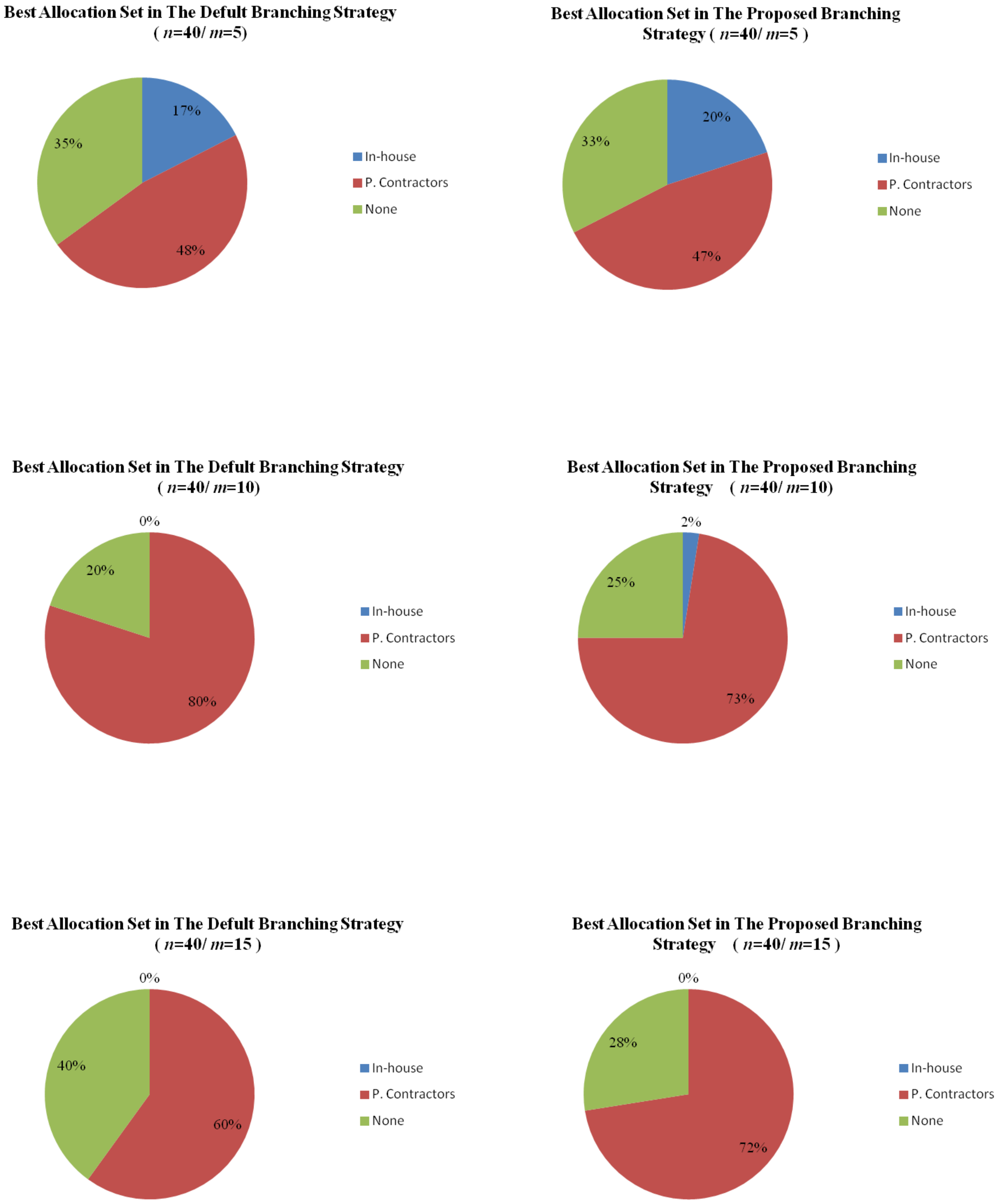
Best Allocation Set in The Defult Branching Strategy (*n*=20/ *m*=10)

Best Allocation Set in The Proposed Branching Strategy (*n*=20/ *m*=10)

Best Allocation Set in The Defult Branching Strategy (*n*=20/ *m*=15)

Best Allocation Set in The Proposed Branching Strategy (n=20/ m=15)

Figure 15: A Comparison between the Project Allocation for the Combination of *n*=20 and each Level in Factor *M* in Formulation 1



Best Allocation Set in The Defult Branching Strategy (*n*=40/ *m*=5)

Best Allocation Set in The Proposed Branching Strategy (*n*=40/ *m*=5)

Best Allocation Set in The Defult Branching Strategy (*n*=40/ *m*=10)

Best Allocation Set in The Proposed Branching Strategy (*n*=40/ *m*=10)

Best Allocation Set in The Defult Branching Strategy (*n*=40/ *m*=15)

Best Allocation Set in The Proposed Branching Strategy (*n*=40/ *m*=15)

Figure 16: A Comparison between the Project Allocation for the Combination of *n*=40 and Each Level in Factor *M* in Formulation 1

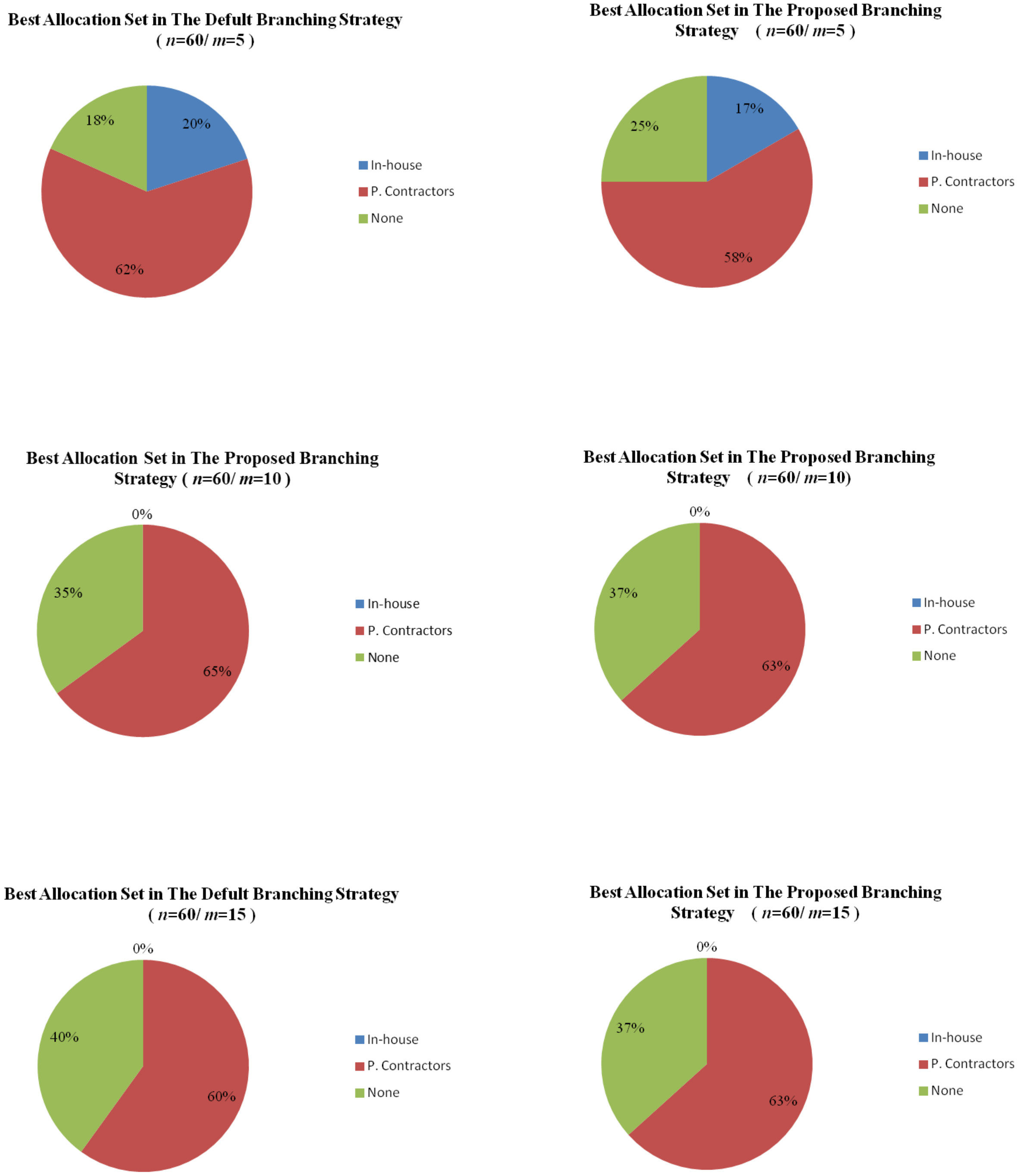
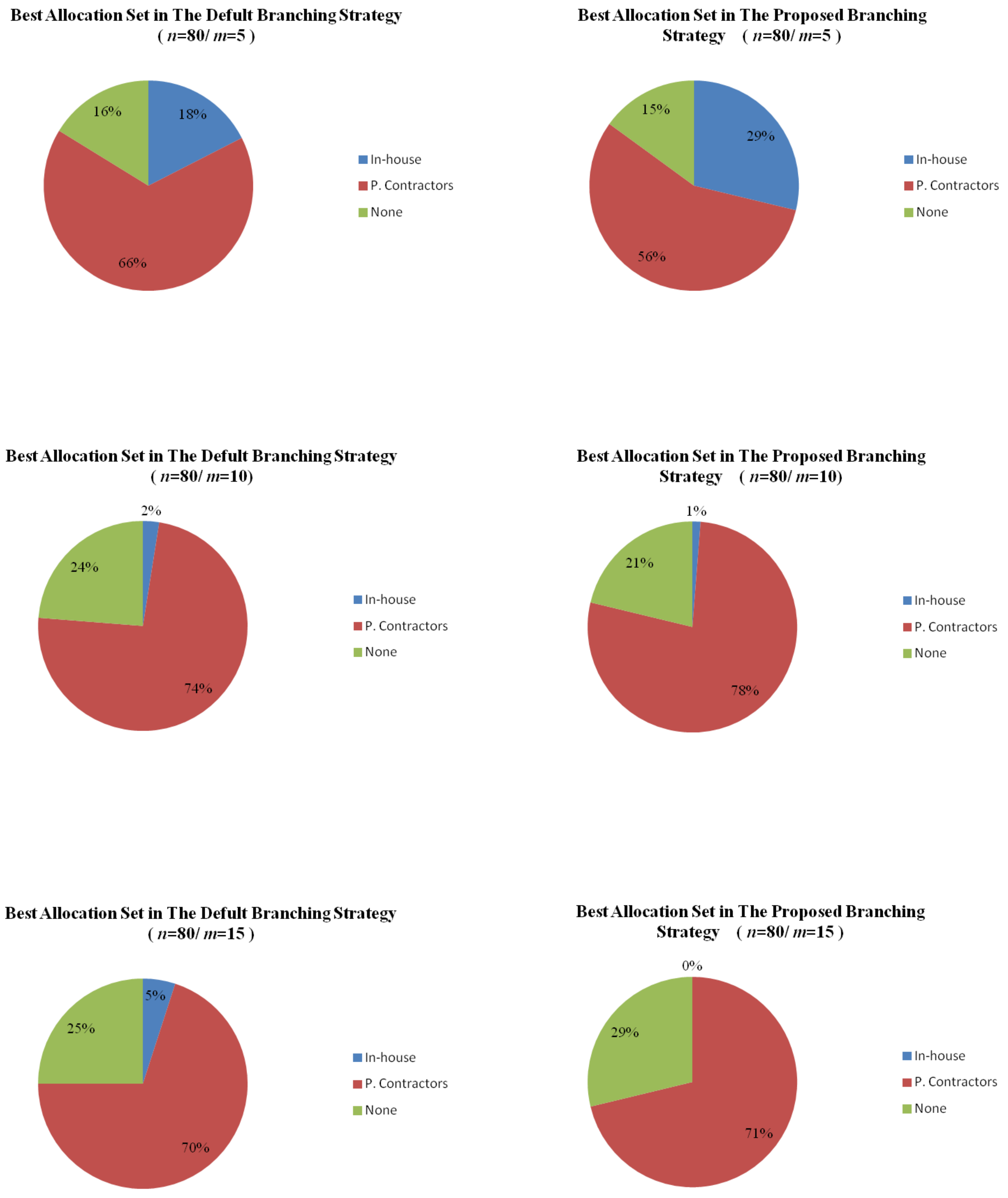


Figure 17: A Comparison between the Project Allocation for the Combination of *n*=60 and Each Level in Factor *M* in Formulation 1


Best Allocation Set in The Defult Branching Strategy (*n*=80/ *m*=5)

Best Allocation Set in The Proposed Branching Strategy (*n*=80/ *m*=5)

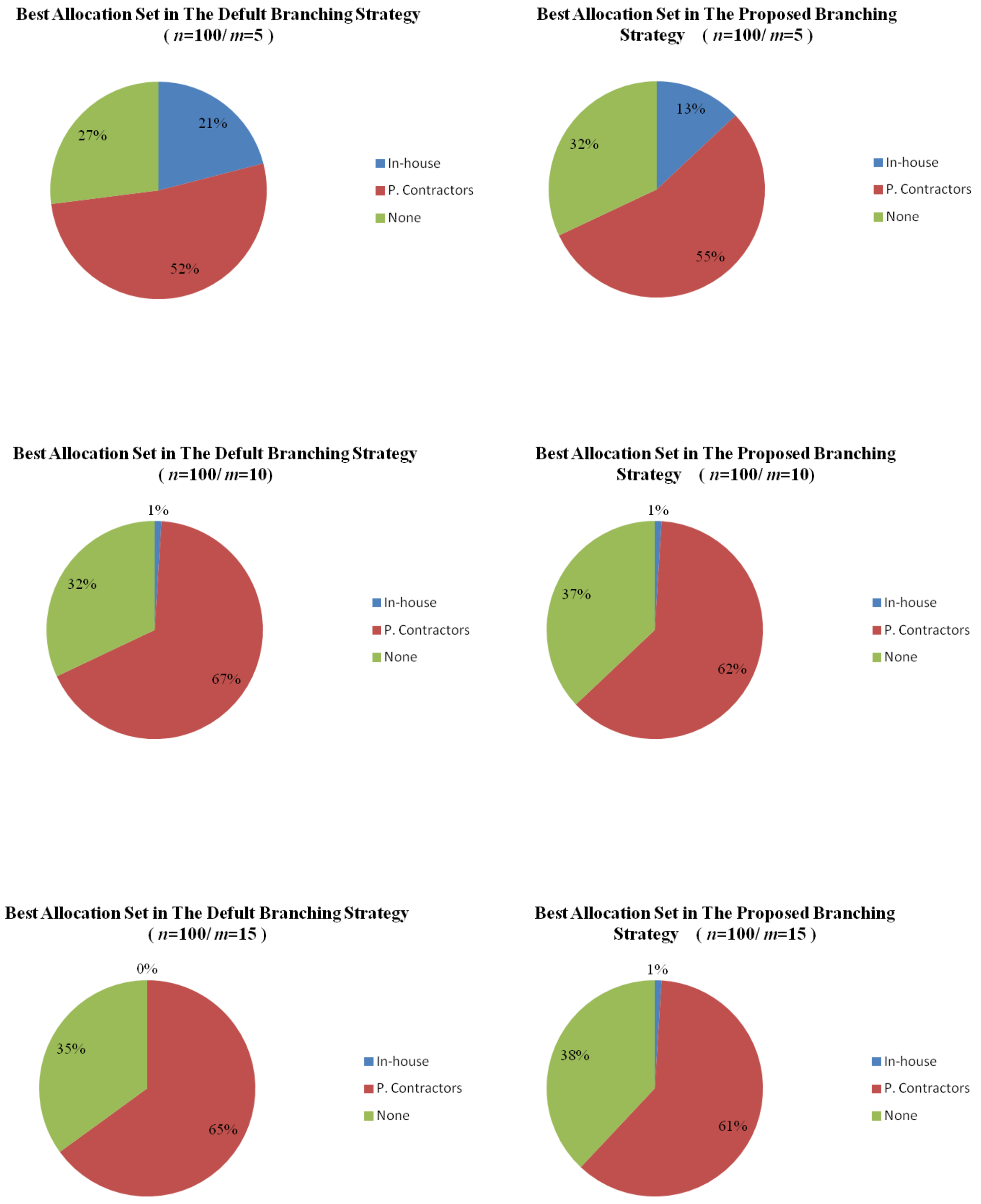
Best Allocation Set in The Defult Branching Strategy (*n*=80/ *m*=10)

Best Allocation Set in The Proposed Branching Strategy (*n*=80/ *m*=10)

Best Allocation Set in The Defult Branching Strategy (*n*=80/ *m*=15)

Best Allocation Set in The Proposed Branching Strategy (*n*=80/ *m*=15)

Figure 18: A Comparison between the Project Allocation for the Combination of *n*=80 and Each Level in Factor *M* in Formulation 1



Best Allocation Set in The Defult Branching Strategy (*n*=100/ *m*=5)

Best Allocation Set in The Proposed Branching Strategy (*n*=100/ *m*=5)

Best Allocation Set in The Defult Branching Strategy (*n*=100/ *m*=10)

Best Allocation Set in The Proposed Branching Strategy (*n*=100/ *m*=10)

Best Allocation Set in The Defult Branching Strategy (*n*=100/ *m*=15)

Best Allocation Set in The Proposed Branching Strategy (*n*=100/ *m*=15)

Figure 19: A Comparison between the Project Allocation for the Combination of *n*=100 and Each Level in Factor *M* in Formulation 1

These figures clearly depict how changing the number of allocation options (*m*-level) affects the allocation subsets (i.e., for each *n*-level the more the options are increased, the less likely projects are to be allocated in-house option). This scenario simulates real-life application in any constrained project allocation problem in order to receive more offers from different options, private contractors, thus increasing the competition among them and reducing the chance of allocating several projects to a specific option (in-house).

Branching Performance Results on Model 1

This section demonstrates the branching strategies' effect on the B&B's solution performance. The results of comparing the computational performance of the MATLAB's default branching strategy and the proposed branching strategy are summarized in Table 11. These results are an average based on running each *nm* combination three times (number of replications) as discussed in the previous chapter. Column one and two of Table 11 represent factor *N* and *M* levels, respectively. The computational measures evaluating the solution performance for MATLAB's default strategy and the proposed strategy are presented in Table 11 as follows: the third column indicates the computational time in seconds; the absolute gap is located in the fourth column; the fifth column shows the solution's relative gap. To assist in understanding the results of this experiment, it is helpful to create a graph of the average responses at each *nm* combination. These graphs are presented for each performance measure (average response) in Figure 20-22.

The complete results of each experimental run (replication) are in Appendix 2.

Table 11: The Effects of the MATLAB's Default and the Proposed Branching Strategies on the Performance Measurements for Solving Model 1

N	*M*	MATLAB Default Branching Strategy			Proposed Branching Strategy		
		Time (sec.)	Abs. Gap	Relative Gap (%)	Time (sec.)	Abs. Gap	Relative Gap (%)
20	5	271.675	8	0.0035	206.051	2	0.0004
	10	520.324	75	0.0367	411.512	12	0.0054
	15	471.709	69	0.0334	458.744	12	0.0054
40	5	425.83	46	0.0081	392.11	11	0.0017
	10	1244.537	23	0.0039	822.808	3	0.0003
	15	899.772	25	0.0044	33.614	2	0.0001
60	5	687.111	10	0.001	614.578	6	0.0005
	10	1330.217	138	0.0155	955.652	20	0.0021
	15	1413.858	4790	0.5454	539.062	21	0.0023
80	5	691.298	30	0.0032	562.437	0	0
	10	1673.504	20	0.0019	743.541	30	0.0033
	15	1748.609	20	0.0024	524.445	0	0
100	5	1512.149	0	0.0003	26.152	0	0
	10	1286.83	180	0.0151	738.107	0	0
	15	1591.461	60	0.0058	1152.914	0	0

Based on the results in Table 11, both strategies' CPU run time ("Time (sec.)") exhibit asymmetrical pattern as the problem size increases, excluding a positive trend in the combinations of (20 and 100)/*m* and (60 and 80)/*m* in the proposed and MATLAB default branching strategies, respectively. In contrast, the absolute gap ("Abs. Gap") and the relative gap ("Relative Gap (%)") do not show development in the majority of *nm* combinations, except a gradually drop in the proposed strategy of the combination of 40/*m*. Specifically, the absolute and relative gaps in factor *N* levels of 20, and 60 increase as changing from the low to the high levels of *M*, regardless of strategy type. However, the remaining *N* levels of 80 (proposed) and 100 with each *m* level show a fluctuated change in the absolute and relative gaps.

The overall comparison of the strategies' performance measurements show that the proposed branching strategy is clearly superior and outperforms the MATLAB default branching strategy, expect the 80/10 (*nm*) combination in the absolutes and relative gaps only.

The maximum performance improvement, reduction in the average CPU time, dramatically decreased in the combination 100/5 from 1512.149 seconds in the default strategy to 26.152 in the proposed strategy, representing a 98% time reduction. Furthermore, the proposed strategy has reached the optimal solution when the gaps equaled zero in seven combinations compared to only one combination of 100/5 in the default strategy. The worst performance in all measurements was in the default strategy for the combination 80/15 within 3 million nodes.

Figure 20 provides a clear picture of the average CPU time pattern in the default and the proposed strategies.

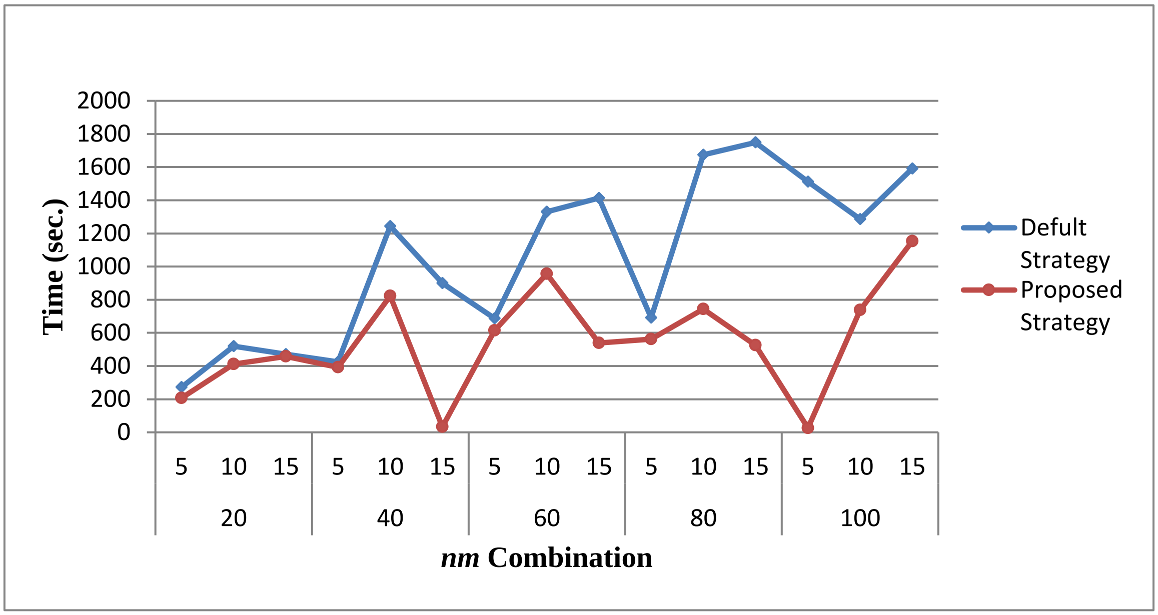


Figure 20: The Effect of the MATLAB Default and the Proposed Branching Strategies on the Average CPU Time in Model 1

In general, the average CPU time shows a fluctuated pattern from low to high levels for *M* and *N*' factors in both strategies, except an exponential increasing from low to high level in the proposed strategy in the last combination of 100/*m*. The fluttered behavior of the CPU time could be regarded to the irregular incremental increasing of the recourses. Interestingly, the most significance improvement in reduced CPU time happened in the proposed strategy in the combinations of 40/ (10, 15), 60/15, and 80 and 100/ (10, 15). These combinations' CPU time improvement is summarized below:

	Improved CPU Time (%)		
N ***M***	***5***	***10***	***15***
40		33.89	96.26
60			61.87
80		55.57	70.01
100	98.27	42.64	

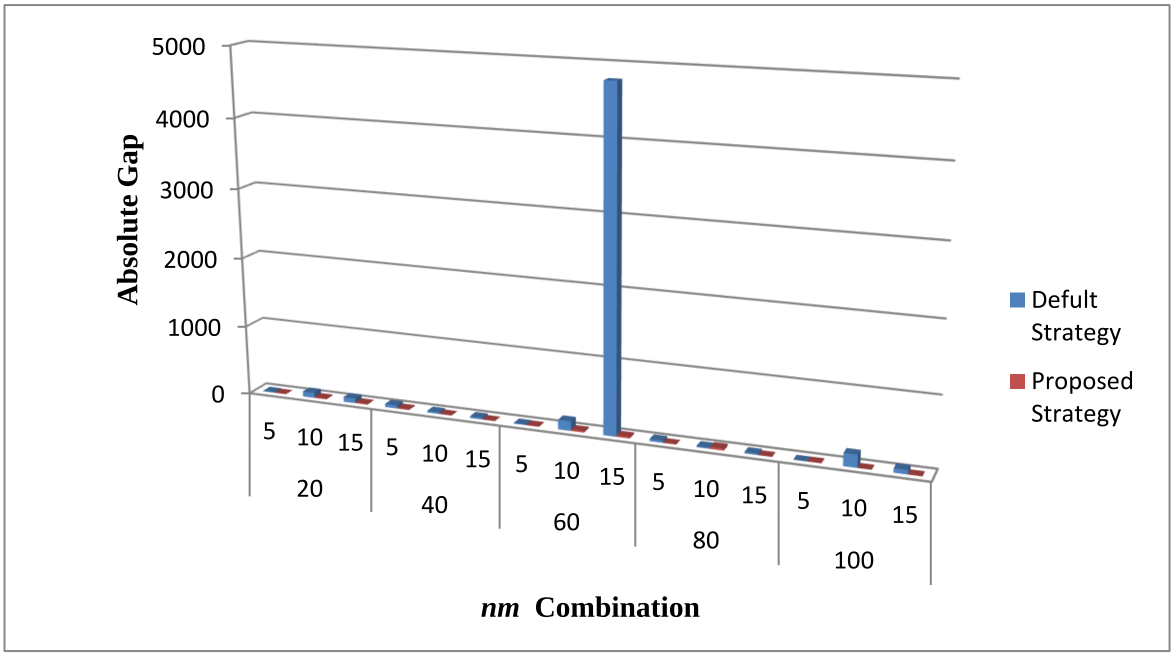


Figure 21: The Effect of the MATLAB Default and the Proposed Branching Strategies on the Absolute Gap in Model 1

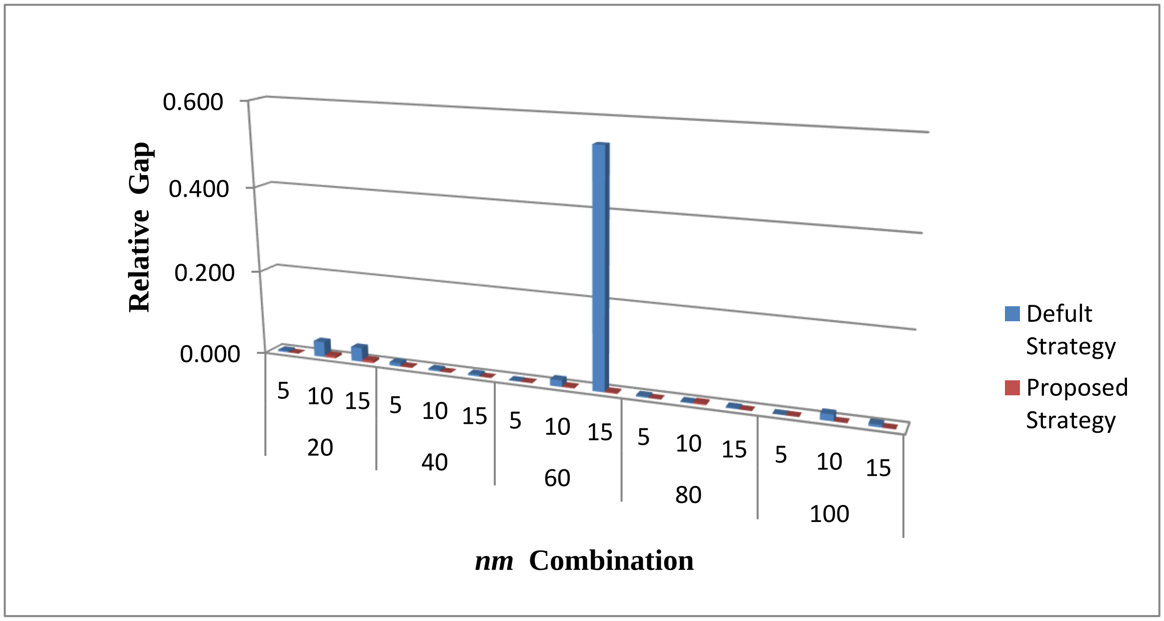


Figure 22: The Effect of the MATLAB Default and the Proposed Branching Strategies on the Relative Gap in Model 1

Figures 21 and 22 present the absolute and relative gaps, respectively. In general, these two figures are analogous and show similar patterns. More specifically, the proposed strategy displays a noticeable reduction in the gaps compared to the default strategy. Moreover, the gaps in the major experimental combinations are slightly increased in each level *n* as the level *m* moves to the higher level, which can be regarded to the computer exhausting and complexity from large problems. On the other hand, the gaps in the proposed strategy show a stabilized pattern across all the combinations. Thus, the proposed strategy is unaffected by the problems' sizes and can obtain optimal solutions for such problems regardless of their variables' and constraints' sizes and resource variations.

The most significant improvement in the gaps occurred in the proposed strategy for the following *nm* combinations:

	Improved Abs. Gap (%)			**Improved Relative Gap (%)**		
	M			***M***		
N	**5**	**10**	**15**	**5**	**10**	**15**
20	75	84	82.61	88.57	85.29	83.83
40	76.1	86.96	92	79	92.31	97.73
60	40	85.51	99.56	50	86.45	99.58
80	100		100	100		100
100		100	100		100	100

Best Project Allocation Results with Model 2

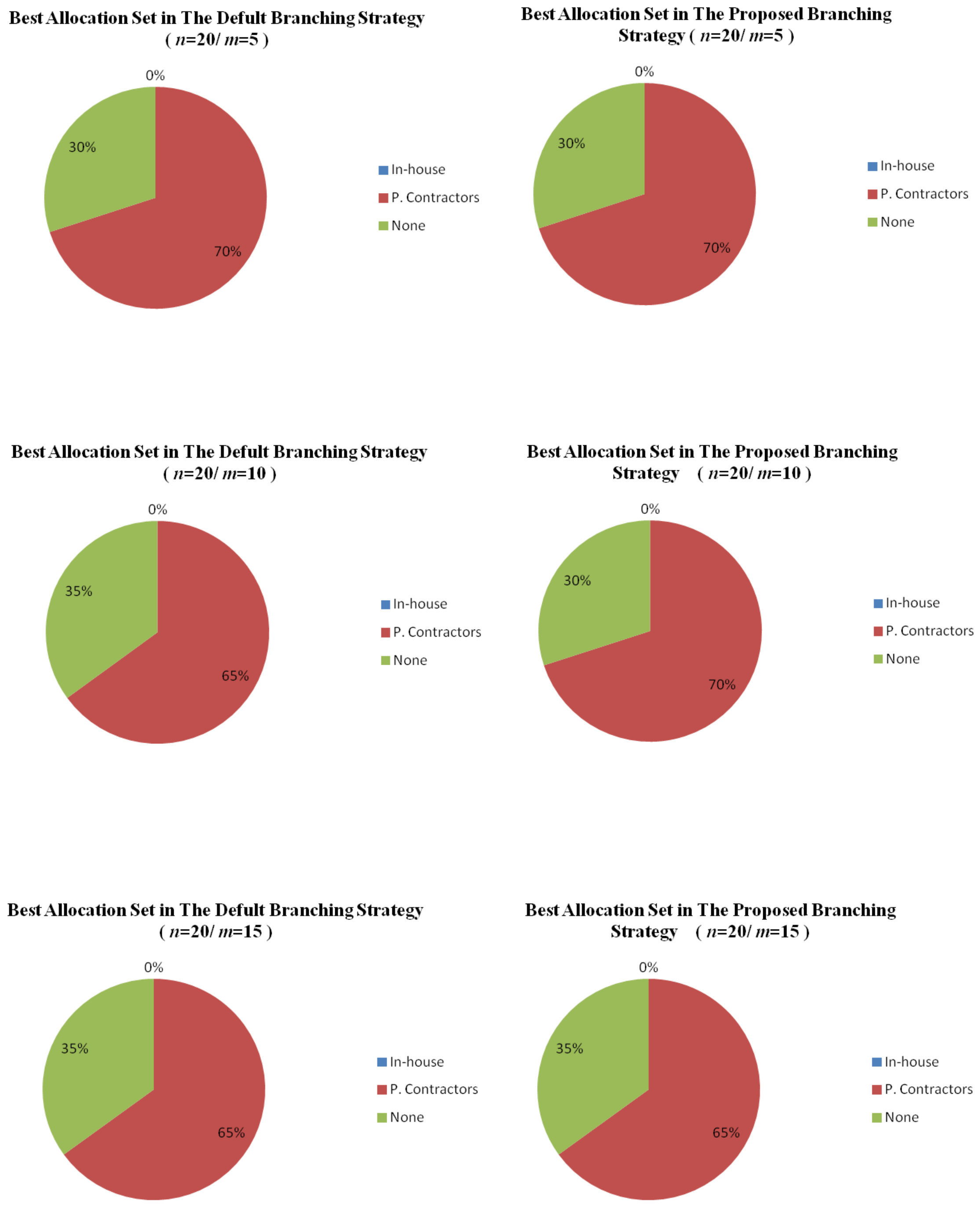
The best project allocation subset results by solving formulation 2 for each *nm* combination are shown in Table 12. This table's structure and description is analogous to Table 10’s in formulation 1.

Table 12: Results of the Best Allocated Project by Solving Formulation 2

N	***M***	**Default Branching Strategy**			**Proposed Branching Strategy**		
		In-house	**P. Contractors**	**None**	**In-house**	**P. Contractors**	**None**
20	5	0	14	6	0	14	6
	10	0	13	7	0	14	6
	15	0	13	7	0	13	7
40	5	2	22	16	2	23	15
	10	2	22	16	2	23	15
	15	0	24	16	0	24	16
60	5	2	33	25	2	33	25
	10	2	33	25	2	33	25
	15	0	40	20	0	40	20
80	5	3	43	34	2	44	34
	10	2	44	34	1	45	34
	15	0	45	35	0	46	34
100	5	5	53	42	5	53	42
	10	3	56	41	3	56	41
	15	1	64	35	1	64	35

The project allocation sets in Table 12 are different from the best allocations in model 1 because each formulation has different function and constraints. In addition, both models are proposed to offer different project allocation sets and schemes based on the project managers or the decision makers' project allocation objectives or strategies.

The project allocation results in both branching strategies for every combination are shown in Figures 23-27.


Best Allocation Set in The Defult Branching Strategy (*n*=20/ *m*=5)

Best Allocation Set in The Proposed Branching Strategy (*n*=20/ *m*=5)

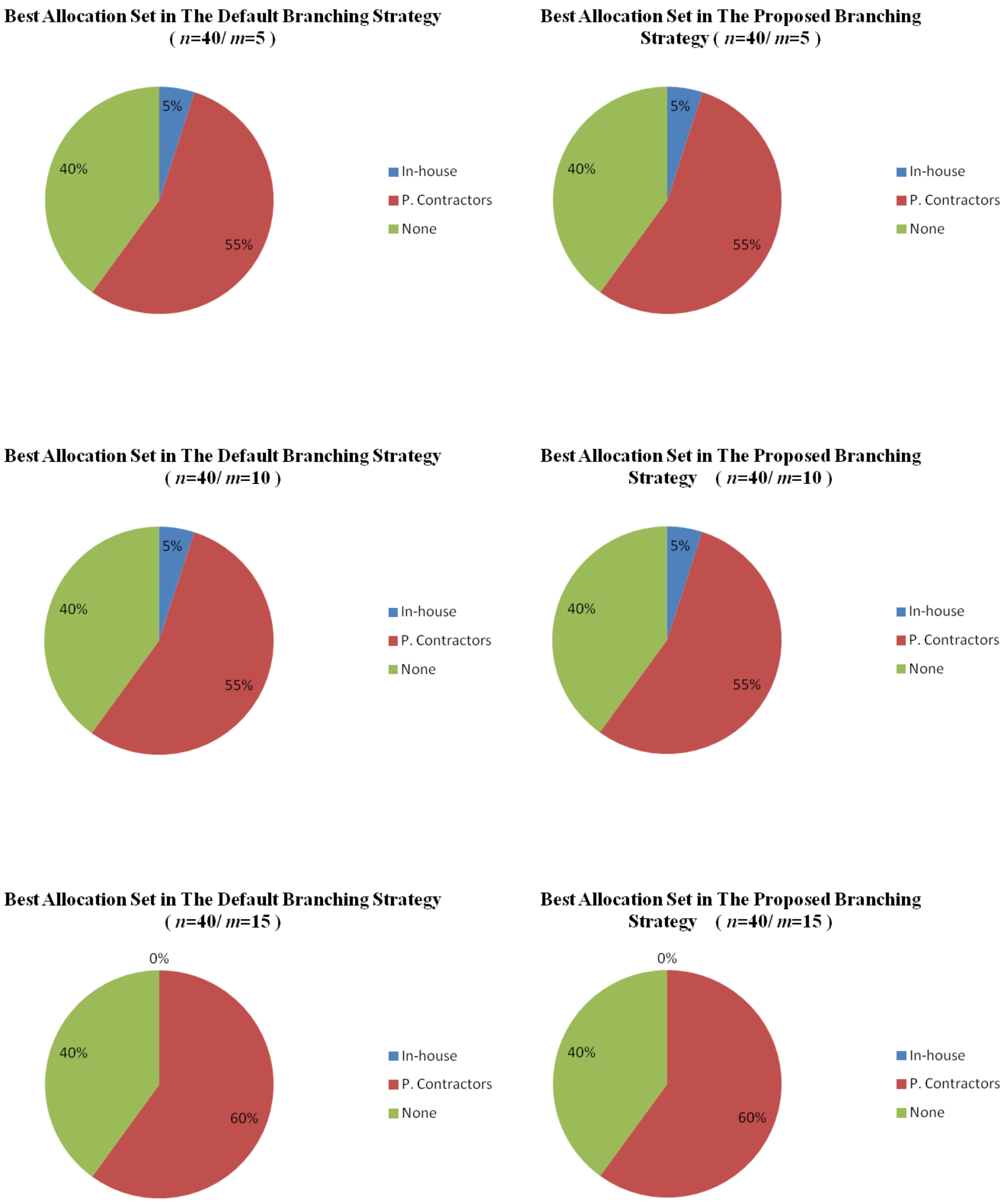
Best Allocation Set in The Defult Branching Strategy (*n*=20/ *m*=10)

Best Allocation Set in The Proposed Branching Strategy (*n*=20/ *m*=10)

Best Allocation Set in The Defult Branching Strategy (*n*=20/ *m*=15)

Best Allocation Set in The Proposed Branching Strategy (*n*=20/ *m*=15)

Figure 23: A Comparison between the Project Allocation for the Combination of *n*=20 and Each Level in Factor *M* in Formulation 2



Best Allocation Set in The Default Branching Strategy (*n*=40/ *m*=5)

Best Allocation Set in The Proposed Branching Strategy (*n*=40/ *m*=5)

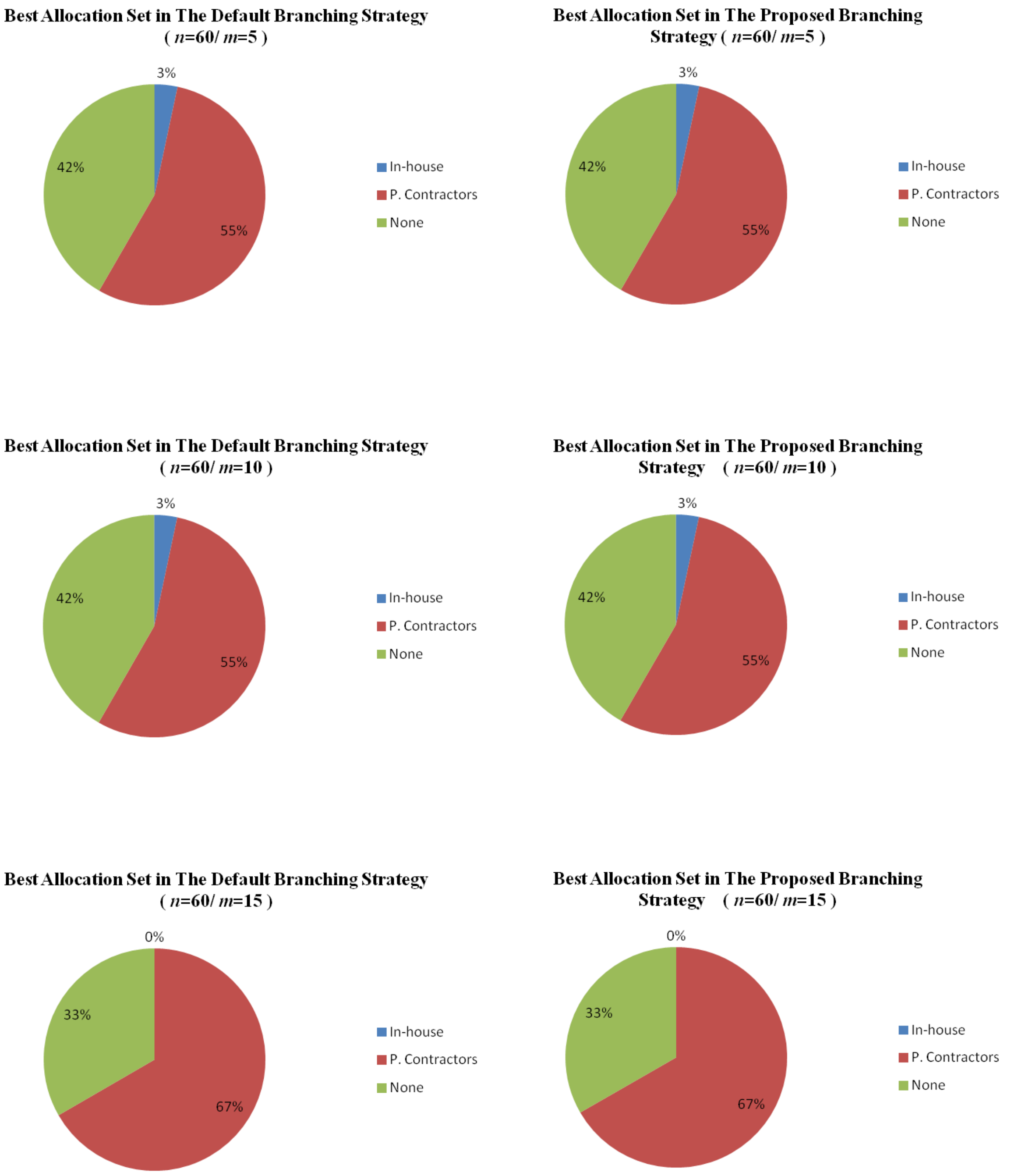
Best Allocation Set in The Default Branching Strategy (*n*=40/ *m*=10)

Best Allocation Set in The Proposed Branching Strategy (*n*=40/ *m*=10)

Best Allocation Set in The Default Branching Strategy (*n*=40/ *m*=15)

Best Allocation Set in The Proposed Branching Strategy (*n*=40/ *m*=15)

Figure 24: A Comparison between the Project Allocation for the Combination of *n*=40 and Each Level in Factor *M* in Formulation 2



Best Allocation Set in The Default Branching Strategy (*n*=60/ *m*=5)

Best Allocation Set in The Proposed Branching Strategy (*n*=60/ *m*=5)

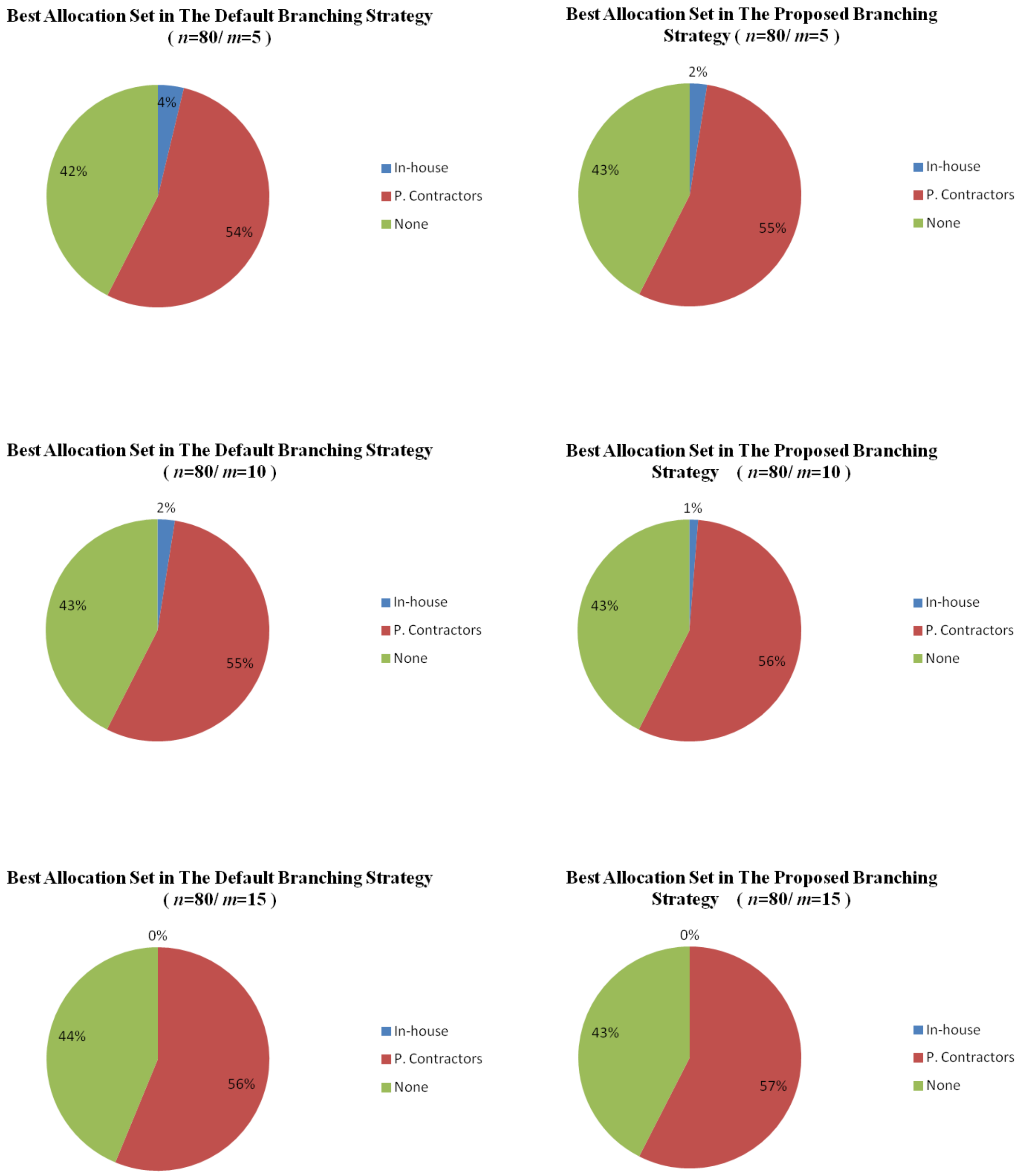
Best Allocation Set in The Default Branching Strategy (*n*=60/ *m*=10)

Best Allocation Set in The Proposed Branching Strategy (*n*=60/ *m*=10)

Best Allocation Set in The Default Branching Strategy (*n*=60/ *m*=15)

Best Allocation Set in The Proposed Branching Strategy (*n*=60/ *m*=15)

Figure 25: A Comparison between the Project Allocation for the Combination of *n*=60 and Each Level in Factor *M* in Formulation 2



Best Allocation Set in The Default Branching Strategy (*n*=80/ *m*=5)

Best Allocation Set in The Proposed Branching Strategy (*n*=80/ *m*=5)

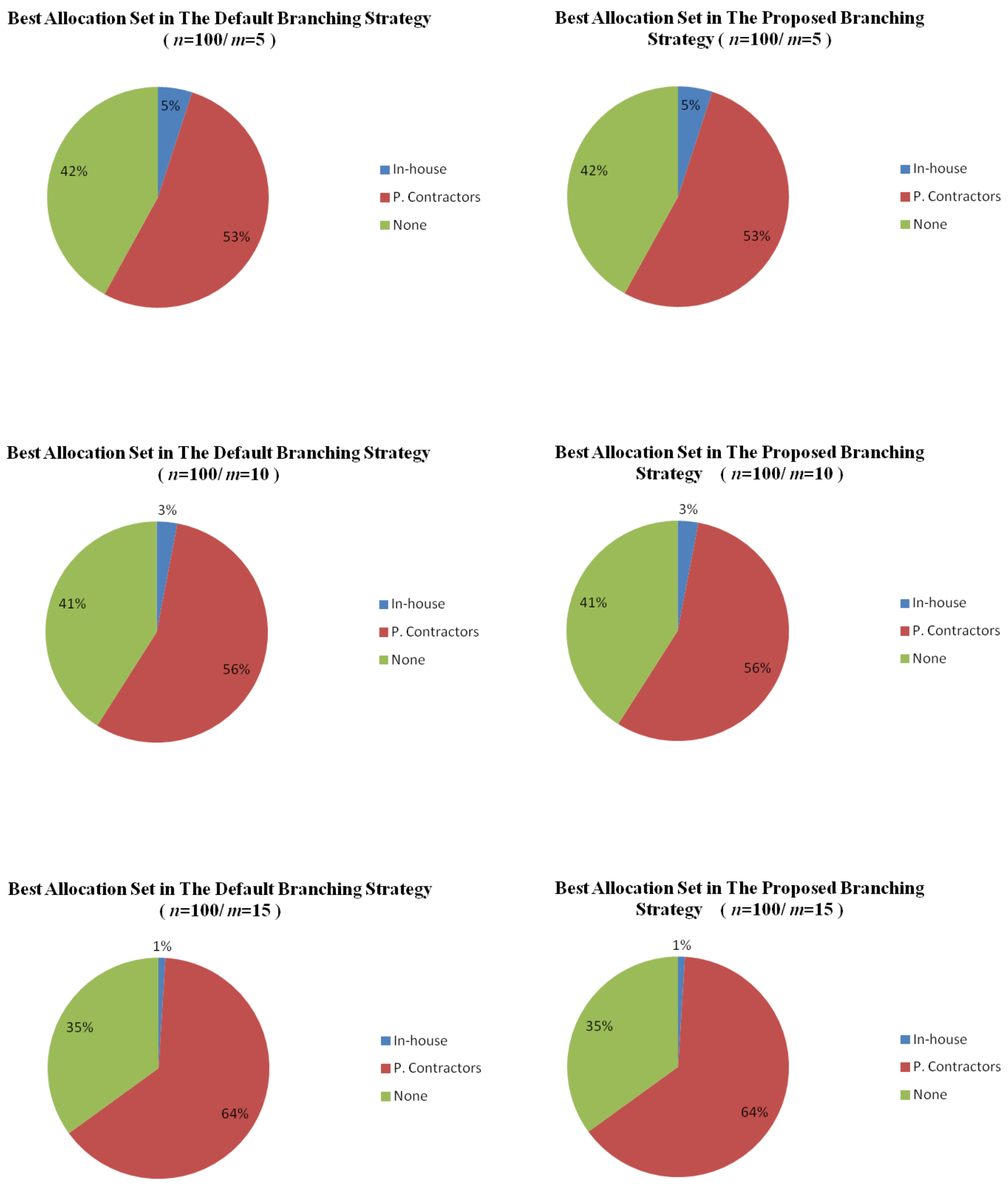
Best Allocation Set in The Default Branching Strategy (*n*=80/ *m*=10)

Best Allocation Set in The Proposed Branching Strategy (*n*=80/ *m*=10)

Best Allocation Set in The Default Branching Strategy (*n*=80/ *m*=15)

Best Allocation Set in The Proposed Branching Strategy (*n*=80/ *m*=15)

Figure 26: A Comparison between the Project Allocation for the Combination of *n*=80 and Each Level in Factor *M* in Formulation 2



Best Allocation Set in The Default Branching Strategy (*n*=100/ *m*=5)

Best Allocation Set in The Proposed Branching Strategy (*n*=100/ *m*=5)

Best Allocation Set in The Default Branching Strategy (*n*=100/ *m*=10)

Best Allocation Set in The Proposed Branching Strategy (*n*=100/ *m*=10)

Best Allocation Set in The Default Branching Strategy (*n*=100/ *m*=15)

Best Allocation Set in The Proposed Branching Strategy (*n*=100/ *m*=15)

Figure 27: A Comparison between the Project Allocation for the Combination of *n*=100 and Each Level in Factor *M* in Formulation 2

As it can be inferred from Figures 23-27, the best allocation sets in the default strategy are identical to allocation sets in the proposed strategy across all the *nm* combinations. In addition, increasing the number of allocation options (level *m*) reduces the opportunity for projects to be allocated in-house, representing the same situation as in model 1's project allocation sets.

Branching Performance Results on Model 2

The effects of the default and the proposed branching strategies on the solution performances are summarized in Table 13. Each performance measure in this table shows the average CPU time, absolute gap, relative gap, and number of nodes ("No. Nodes") needed to reach the optimal solution for every experimental combination *nm*. Because of the low degree of complexity level in this formulation, the number of explored nodes is included in assessing the strategies' performance. In another words, this formulation has less constraints than formulation 1. Consequently, MATLAB solves the BIP problem for this formulation to the optimality, where the absolute and relative gaps are trivial numbers. Moreover, the performance comparisons in this situation ideally focus on how the default and proposed branching strategies reach the optimal solution with fewer efforts and which strategy works better in large problem sizes.

Figure 28-31 presents each performance measure's representation behavior.

The complete results of each experimental run (replication) are in Appendix 2.

Table 13: The Effects of the MATLAB's Default and the Proposed Branching Strategies on the Performance Measurements for Solving Model 2

N	*M*	MATLAB Default Branching Strategy				Proposed Branching Strategy			
		Time (sec.)	Abs. Gap	Relative Gap (%)	No. Nodes	Time (sec.)	Abs. Gap	Relative Gap (%)	No. Nodes
20	5	0.065	0.000	0.000	N.A.*	0.064	0.000	0.000	N.A.*
	10	0.305	0.000	0.000	N.A.*	0.311	0.000	0.000	N.A.*
	15	0.144	0.000	0.000	N.A.*	0.133	0.000	0.000	N.A.*
40	5	0.817	0.021	0.010	4790	0.594	0.021	0.010	3720
	10	1.223	0.000	0.000	7042	0.339	0.021	0.010	1486
	15	0.432	0.021	0.010	1894	0.305	0.021	0.010	906
60	5	2.757	0.033	0.009	18388	1.503	0.000	0.000	11229
	10	11.400	0.000	0.000	59836	6.523	0.028	0.006	31783
	15	68.077	0.034	0.010	149490	93.920	0.000	0.000	265424
80	5	47.906	0.037	0.010	332890	25.470	0.031	0.007	159182
	10	188.138	0.026	0.004	799582	71.253	0.037	0.010	291458
	15	15.611	0.036	0.010	59394	8.651	0.028	0.005	34346
100	5	2.133	0.044	0.010	12150	0.865	0.044	0.010	4542
	10	1.687	0.000	0.000	3508	1.741	0.000	0.000	3469
	15	1354.747	0.1594	0.0613	>2000000	497.833	0.000	0.000	711288

*: Solved by the Cutting Generation Algorithm (The 4th stage in MATLAB IP solver), the number of nodes is not available in this case.

The average CPU time results for each *nm* combination in both strategies show flutter performance throughout changing *M* levels across all factor's *N* levels. At the first glance, it can be observed that both strategies work effectively in the high level of *M*, specifically with *N* levels of 20, 40, and 80 than other *nm* combinations.

As expected, both strategies' absolute gaps are relatively close to each other and very small, except the largest absolute and relative gaps in the default strategy occur with the combination of 100 projects and 15 options.

The experimental results for the level 20 in factor *N* with all *M* levels are solved by the cut generation methodology which is the fourth stage for solving the IP in the MATLAB, (i.e., the MATLAB did not solve this specific *nm* combination with the B&B algorithm because of the small problem), see the experimental design section in Chapter 5 for more information regarding the MATLAB IP solution stages. Therefore, the number of explored nodes in the search tree, in both strategies, for the combination of 20/*m* is not available in this formulation and experimental combination.

The number of explored nodes in all other combinations shows inconsistent pattern as going from low to high levels of factor *N* and *M*, in the default and proposed strategies.

In general, the proposed strategy surpasses the default strategy in all performance measurements and solves with remarkable quality the constrained project allocation problem in various experimental combinations.

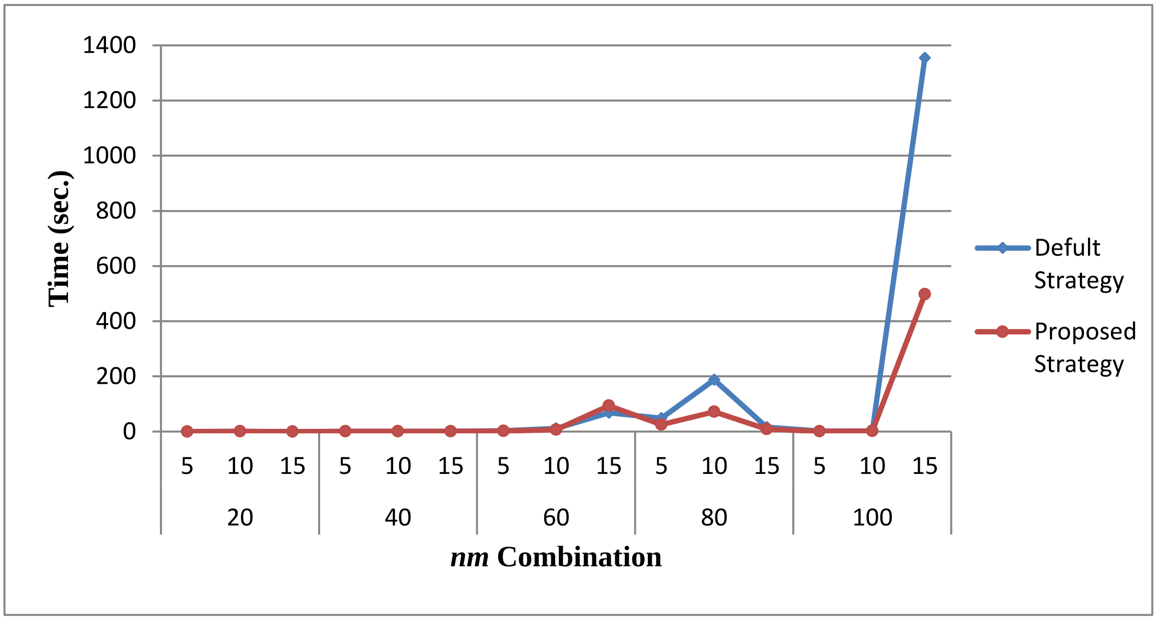


Figure 28: The Effect of the MATLAB Default and the Proposed Branching Strategies on the Average CPU Time in Model 2

Figure 28 shows a comprehensive overview of the average CPU run time for both strategies across all the *nm* combinations. The average CPU time in both strategies ran stably in a short time in levels *N* of 20 and 40 with all *M* levels and with the combination of 60/5 and 10. Then it started suddenly going up and down in the reaming *nm* combinations. This phenomenon is expected to occur as a result of increasing the problem size or moving to higher levels in factor *N* and *M*. Generally speaking, both strategies for this model indicate that B&B algorithm might suffer or run longer in a large problem of more than 1500 variables. The proposed strategy, however, works better with large problems compared to the default strategy. The most significance time reduction occurred in the proposed strategy for the combinations of 80/10 and 100/15 by 62.13% and 63.25%, respectively.

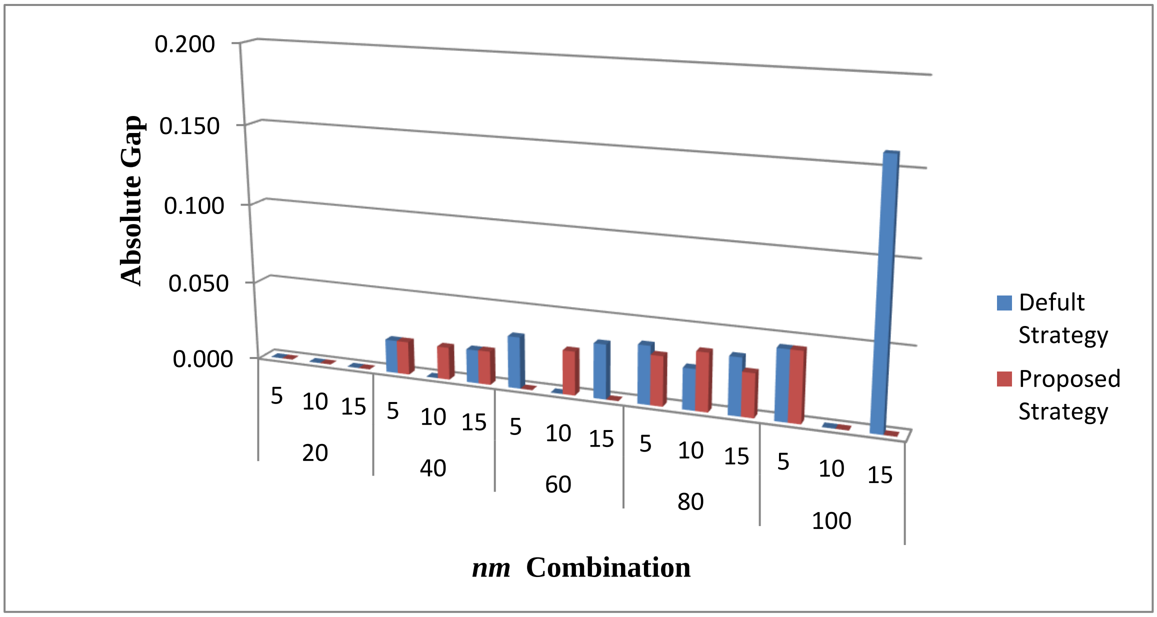


Figure 29: The Effect of the MATLAB Default and the Proposed Branching Strategies on the Absolute Gap in Model 2

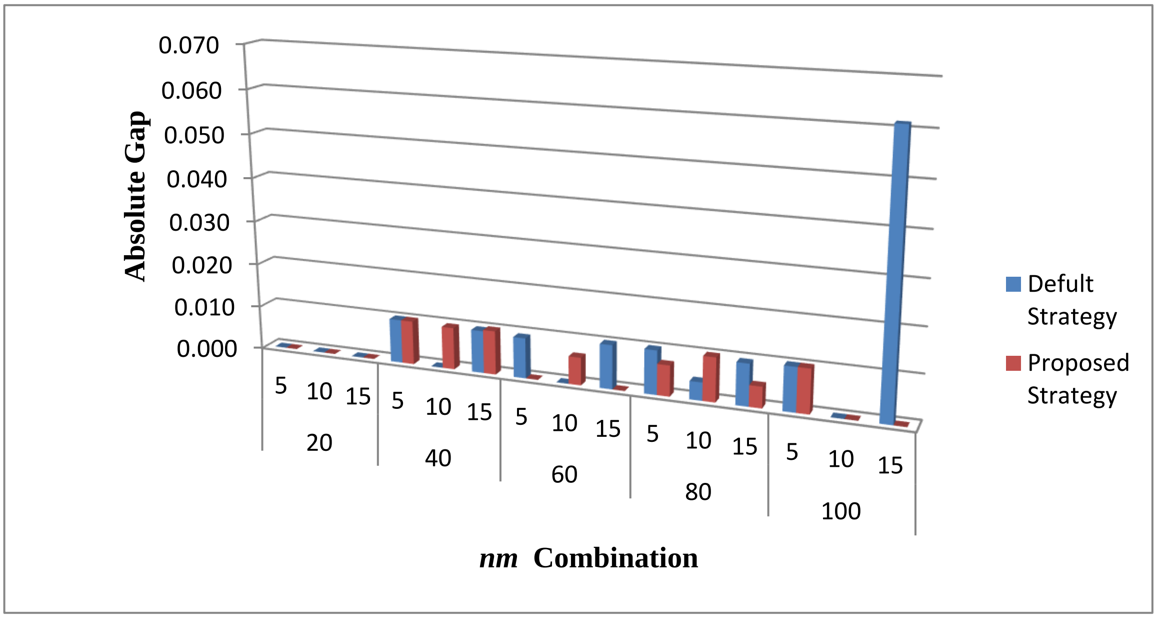


Figure 30: The Effect of the MATLAB Default and the Proposed Branching Strategies on the Relative Gap in Model 2

Figure 29 and 30 compare the performance measures of the absolute and relative gaps, respectively, between the default and proposed strategies. These performances are identical in both strategies. Excluding the 100/15, the absolute gaps in Figure 29 did not exceed the limit of 0.05 and 0.01 % in Figure 30 of the relative gaps, indicating that the B&B performs well. The most notable absolute and relative gaps happened in the combination 100/15 with 0.159 and 0.061%, respectively. Despite obtaining the largest amount of gaps in the 100/15 combination, the solution is still considered optimal, since this number of gaps is very small compared to the gaps in model 1.

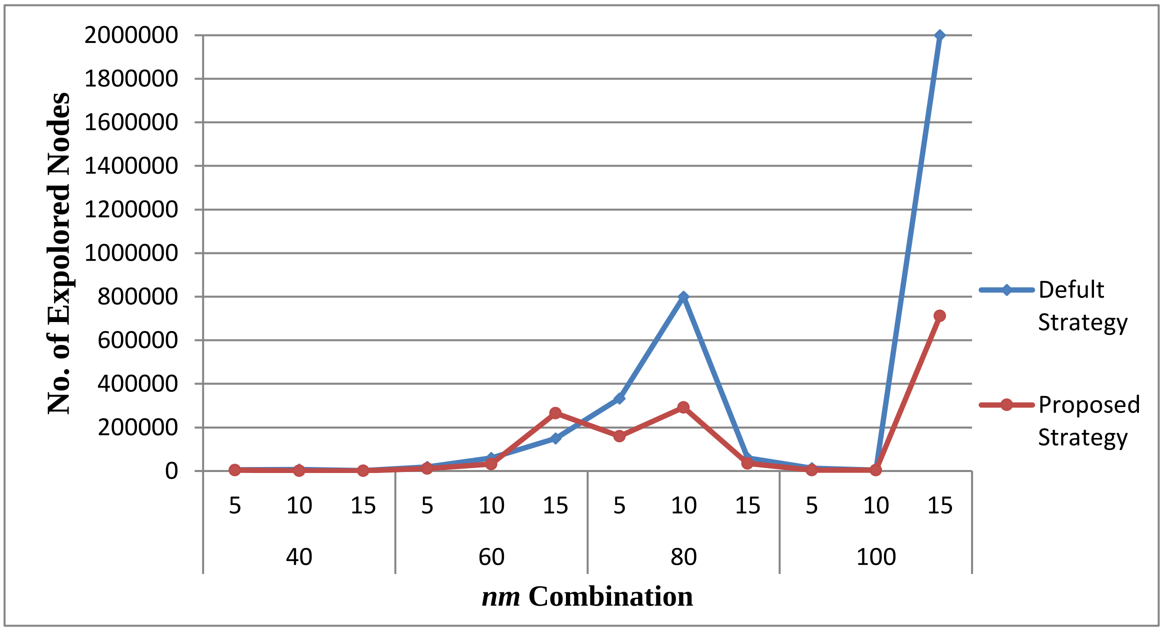


Figure 31: The Effect of the MATLAB Default and the Proposed Branching Strategies on the Number of Explored Nodes in Model 2

The number of the explored nodes or the B&B tree sizes, to reach the optimal solutions for each *nm* combination in both strategies is depicted in Figure 31. Furthermore, this figure shows the branching strategies' effect on the number of nodes needed to reach the optimal solution. Since the MATLAB solved the combination of 20/*m* using the cutting generation methodology instead of B&B algorithm, this figure exhibits the explored nodes from *N* levels of 40 to 100. The difference between both strategies can be clearly observed from the combination 60/5 through 100/15. During this interval, the number of nodes in both strategies displays a fluctuated pattern. By contrast, the default and the proposed strategies increase sharply in the combination 100/15 from 3508 to more than 2 million nodes and from 3469 to 711,288 nodes, respectively. Generally, the proposed branching strategy outperforms the default strategy outstandingly in the experiments, except for the combinations of 100/10 and 60/15.

The following results present significant improvement in the number of nodes reduced, by using the proposed strategy versus the default strategy.

N	***M***	**Improved No. of Explored Nodes.(%)**
40	10	78.90
	15	52.16
60	5	38.93
	10	46.88
80	5	52.18
	10	63.55
	15	42.17
100	5	62.62
	15	64.44

Summary

In this Chapter the data illustrating this study's application and drawing conclusions from the experiments were generated from a real case study based on descriptive statistics found to be more efficient and reliable in generating a data matching the real data. Each formulation with this study's branching strategies was examined separately by the suggested two-factor factorial design experiment. There were multivariate responses from the experiment conducted on each model (i.e., four responses for each *nm* combination on model 1 and five responses on model 2).These responses were averages of replicating each *nm* combinations three times for the purpose of obtaining accurate results. It is important to mention that all of the computed responses, when replicated, for each combination are identical, except the CPU run time. Best project allocation subsets from each model with the branching strategies were reported. In fact, going from a low to a high level of factor *M* for each level in factor *N* provides different allocation subsets. From this point of view, the results of the performed experiments with the generated data set can be considered to have a realistic credibility level. The B&B branching strategies' performance measures for the MATLAB default and the proposed strategy were examined and compared on both models. In general, the proposed branching outperformed the MATLAB default branching in both models. Finally one of the validate conclusions discovered while the experimental results were examination is the one that Atamtürk and Savelsbergh discussed [14, pp.73]:

> "It is important to observe that certain parameter settings can exhibit quite different computational behavior on instances from different sources. However, it is generally true to [sic] that certain parameter settings exhibit fairly consistent behavior on instances from the same source. Therefore, in practice tuning branching parameters can be extremely important. Even though a combination of parameters may not work well on a wide variety of instances, it may work extremely well on the class of problems that need to be solved."

CHAPTER VII
SUMMARY, CONLUSIONS, AND RECOMMENDATIONS

In this chapter, this study is summarized, conclusions are discussed, and recommendations for future research are made.

Summary

This dissertation proposed a novel decision-support framework based on an analytical approach aiming to select the most desirable number of projects, from an economic prospective, to be allocated to several options. The proposed approach included multiple groups, multiple constraints, two model formulations, and a B&B strategy for solving the models. The concept of MD, which accounts for the correlation between the project's characteristics, was used to allocate projects to the most fitting options. In developing this decision-support framework, a review of relevant literature focused on project selection and resource allocation, a classification approach using MD, and B&B strategies affecting the solution performances. Various numbers of considerable gaps in modeling project selection and B&B solution performances were identified and this dissertation's contributions to the literature have been addressed. Furthermore, light was shed on the MD and B&B and on how they can be used. Two novel BIP models providing different conclusions regarding best project-allocation sets were introduced. The first model maximizes the benefit s' dollar value while ensuring that projects are allocated to the best-fitting option consistent with available budget and in-house man-hour levels. The second model focuses on allocating projects to the best-fitting options subject to available budget and in-house man-hour levels. The solution methodology is based on a novel

B&B strategy for variables and node selections that aims to improve the solution procedure’s efficiency.

To illustrate the proposed BIP models and to assess the suggested B&B branching strategy’s solution performances compared to those of the MATLAB's default strategy, computational experiments were designed and run on various number of project and option combinations in terms of multiple responses. Each model with the proposed and MATLAB default branching strategies was run on a designed experiment with a total of 90 runs for both strategies. The computational experiments’ results for each model were analyzed, and the branching strategies were compared. These computational experiments were coded and solved in the MATLAB R2014b on a PC with of an Intel Core i5 processor, 6.00GB RAM, and a 64-bit Windows 7 operating system.

Conclusions

A decision-making approach to project selections in organizations intending to use their resources efficiently and economically has been widely studied. Numerous researchers have proposed a variety of models or methodologies to obtain optimal project portfolios or sets that maintain organizations' limited resources. These methodologies have been mathematically modeled based on such factors as different objectives, competing constraints, budgetary observances, and available manpower. This dissertation focused on proposing an analytical approach that selects multiple constrained transportation projects and allocates them to the best fitting options, such as in-house engineers and private contractors.

This study's application results, for both models, provide different allocation scenarios or sets corresponding to various experimental combinations. Since the correlation or dependency between variables, in a multivariate observation (project), can affect the allocation accuracy, the proposed models suggest more reliable allocation sets than others in terms of accounting for the correlation between project's characteristics through the MD concept. In addition, since the scope of each project is defined with respect to several factors or characteristics, these models provide several explanations of the association or matching between projects and options.

In comparing two models' best project-allocation sets, we observed that model 2 was more restricted than model 1 in assigning projects to the in-house and private contractors. This observation provides a distinguished conclusion regarding each model's objective. In other words, model 2 is emphasized more on allocating projects to the most matching options than model 1. Therefore, decision makers or the project managers will have the choice to choose one of the proposed models that accommodate their project allocation goal. These two models, however, can be used as optimization methodology for project allocation when a budget is insufficient to fund all projects.

Various significant performance measures were used to compare the proposed branching strategy's solution quality with that of the MATLAB default branching strategy. In both models, the proposed strategy outperformed the best MATLAB default strategy. The CPU run time's maximum performance improvement for models 1 and 2 (by 98.27% and 63.25%, respectively) was achieved by using the proposed strategy instead of the MATLAB default strategy. Absolute and relative gaps were approximately similar across all the combinations in model 2 for both strategies. On the other hand, the most improvement in these gaps was obtained in different combinations by using the proposed strategy in model 1. In comparing to the gaps obtained by MATLAB default strategy, the proposed strategy improved these gaps by 100% or reduced them to zero. One of the most effective performance measurements used in many studies is the number of explored nodes in the B&B search tree because this measurement has a positive relationship with the CPU run time (i.e., the number of nodes is explored more as the CPU run time increase). The proposed branching strategy in model 2 outperformed the MATLAB default strategy in the majority of combinations. The maximum improvement in this performance measure was approximately 80% in the number of nodes reduced compared to the MATLAB default strategy. It is important to mention that many, if not all, IP state-of-art solvers have designed the default branching strategy and other B&B settings to work best and accommodate a verity of instances in practice.

Finally, unlike most of the traditional studies discussed in the literature for the field of project selection and resource allocation, this dissertation not only offers a significant a quantitative tool that supports decision-making, but also provides an improved solution strategy that can work with large problems.

Recommendations and Future Research

To the best of our knowledge, the constrained multi-group project allocation using the MD and the proposed branching strategy in B&B algorithm has not been studied or considered in the literature. Consequently, any future research regarding the proposed models and solution methodology will contribute to the literature.

Further research efforts can be summarized as follows:

- Construct same project-allocation models based on ED, instead of MD, and compare the computational results based on both statistical distances. Comparison results will reveal several considerable observations in terms of optimal solutions, allocation sets, and performance measures.
- To improve the allocation results' reliability, add more constraints to the models, such as availability of project construction equipment, project completion date or scheduling, project interactions, and project completion risks or uncertainty.
- Determine which of the experimental factors or resources has the most significant effect on the multivariate responses of CPU run time, absolute gap, relative gap, and number of explored nodes. MANOVA is suggested for this purpose.
- Compare the proposed branching strategy with other branching strategy and branching settings.
- Propose decision-making software (instead of a panel of experts) for selecting the most matching projects for each allocation option to form each option's reference dataset. This software can be linked to the contracting database and can create the reference data set for each option based on specific selection criterion. Integrating this software with the proposed solution methodology could considerably improve the process of project selection.
- Suggest different resource levels for each experimental combination that offers several scenarios of best allocation subsets.
- Compare the solution performances between the proposed sorted and unsorted variables in the MATLAB B&B default strategy.

- Develop model selection techniques to choose the subset of factors that can best describe the projects from a pool of variables. The models can be selected according to several criteria, such AIC, BIC, and ICOMP [69, 70].

The proposed models can also be solved using dynamic programming. The general systematic procedures for solving our models is briefly stated as follows [12]:

1. Stage s_{im} is represented by the number of projects n=1, 2, ..., N
2. The alternatives at stage s_{in} are represented by $\sum_{i=1}^{M} x_{in} \leq 1$ defining that each project n is allowed to be assigned to *at most* one of the option i
3. Each state at stage s_{in} is represented by the amount of d_{in}, c_{in}, and t_{in} assigned to stages (projects) n, n+1, ..., N. This step states that the resource levels of W, B, and T are the only resections that relates all s_{in} stages or projects together.

   Define

   $f_n(s_n)$= maximum benefit or minimum MD for stages, s_{n+1}, s_{n+2},..., N given state s_{in}.

The backtrack procedure denotes the recursive equation that can be represented by two steps:

1. Write $f_n(s_{in})$ as a function of $f_n(s_{n+1})$ by:

$$f_n(s_{in}) = \max_{\substack{x_{in}=0,1 \\ \forall i=\{1,2,\ldots,M\}}} \{b_n x_{in} + f_{n+1}(s_{in+1})\}$$

$$\sum_{i=1}^{M} x_{in} \leq 1$$

$f_{n+1}(s_{in+1})$ is identical to zero in this step

2. Express s_{n+1} as a function of s_n to warrant the left side of the previous equation is a function of s_n only.

   By definition of the following equations:

$$s_{in} - s_{in+1} = d_{in}x_{in}$$

$$s_{in} - s_{in+1} = c_{in}x_{in}$$

$$s_{in} - s_{in+1} = t_{in}x_{in}$$

Represent the max MD, budget, and manpower time used at each stage s_n

Thus

$$s_{in+1} = s_{in} - d_{in}x_{in}$$

$$s_{in+1} = s_{in} - c_{in}x_{in}$$

$$s_{in+1} = s_{in} - t_{in}x_{in}$$

The recursive relationship for each constraint is written as:

$$f_n(s_{in}) = \max_{x_{in}=0,1 \quad \forall i=\{1,2,\ldots,M\}} \{b_n x_{in} + f_{n+1}(s_{in} - d_{in}x_{in})\}$$

$$\sum_{i=1}^{M} x_{in} \leq 1$$

$$f_n(s_{in}) = \max_{x_{in}=0,1 \quad \forall i=\{1,2,\ldots,M\}} \{b_n x_{in} + f_{n+1}(s_{in} - c_{in}x_{in})\}$$

$$\sum_{i=1}^{M} x_{in} \leq 1$$

$$f_n(s_{in}) = \max_{x_{in}=0,1 \quad \forall i=\{1,2,\ldots,M\}} \{b_n x_{in} + f_{n+1}(s_{in} - t_{in}x_{in})\}$$

$$\sum_{i=1}^{M} x_{in} \leq 1$$

For illustration purposes, Figure 32 depicts the graphical representation of the dynamic programming for the project allocation using Formulation 1.

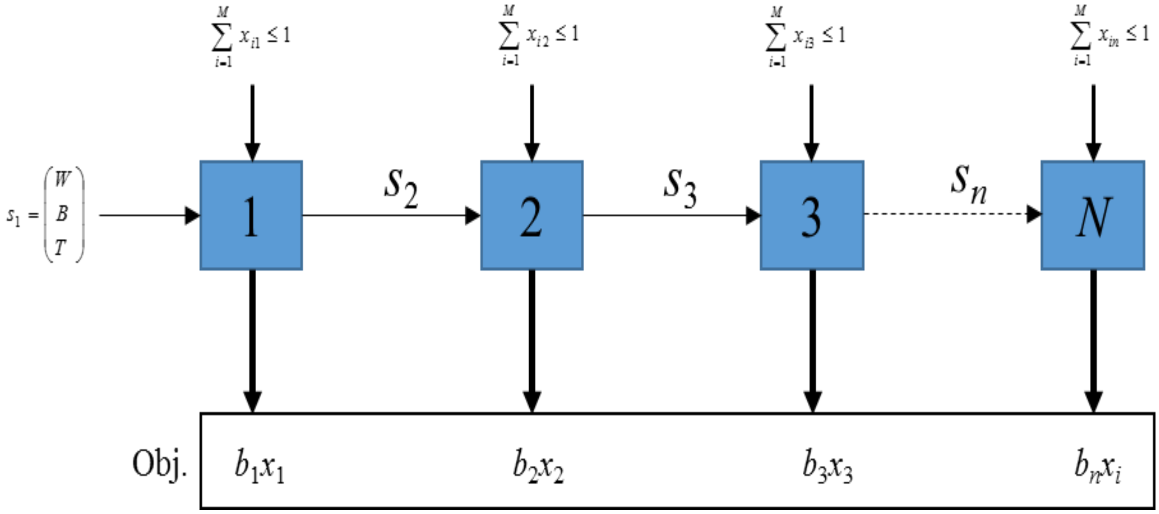

$\sum_{i=1}^{M} x_{i1} \leq 1$
$\sum_{i=1}^{M} x_{i2} \leq 1$
$\sum_{i=1}^{M} x_{i3} \leq 1$
$\sum_{i=1}^{M} x_{in} \leq 1$
$s_1 = \begin{pmatrix} W \\ B \\ T \end{pmatrix}$
1
s_2
2
s_3
3
s_n
N
Obj.
b_1x_1
b_2x_2
b_3x_3
b_nx_i

Figure 32: Dynamic Programming Representation for Model 1

Obviously, proposing a software or an algorithm based on this dynamic programing approach will undoubtedly add a considerable contribution to the literature and provide a great saving in computational’ s efforts and time.

At the end, we anticipate the decision-making framework proposed in this study can be further applied to different project types.

LIST OF REFERENCES

1. Liberatore, M.J., An extension of the analytic hierarchy process for industrial R&D project selection and resource allocation”. Engineering Management, IEEE, 1987. (1): p. 12-18.

2. Henriksen, A.D. and A.J. Traynor, A practical R&D project-selection scoring tool. Engineering Management, IEEE Transactions, 1999. **46**(2): p. 158-170.

3. Tofighian, A.A. and B. Naderi, Modeling and solving the project selection and scheduling. Computers & Industrial Engineering, 2015. **83**: p. 30-38.

4. Yaxu, L. and E. Zhang, R&D project portfolio Selection Model with Interaction Effect. [Accessed 2016 January 10]; Available from: http://www.seiofbluemountain.com/upload/product/200911/2006zxqyhy13a6.pdf.

5. Organisation for Economic Co-operation and Development (OECD). Gross domestic spending on R&D (indicator). [cited 2015 October 8]; Available from: https://data.oecd.org/rd/gross-domestic-spending-on-r-d.htm.

6. Ghorbani, S. and M. Rabbani, *A new multi-objective algorithm for a project selection problem.* Advances in Engineering Software, 2009. **40**(1): p. 9-14.

7. Chen, J. and R. G. Askin, Project selection, scheduling and resource allocation with time dependent returns, European Journal of Operational Research, 2009. **193**:p.23-34.

8. Liberatore, M.J., An Expert Support System For R&D Project Selection, Mathematical computational Modeling, 1988.**11**: p. 260-265.

9. Heidenberger, K. and C. Stummer, *Research and development project selection and resource allocation: a review of quantitative modelling approaches.* International Journal of Management Reviews, 1999. **1**(2): p. 197-224.

10. P. Fahrni and M. Sp¨atig, An application-oriented guide to R&D project selection and evaluation methods, *R&D Manage*, 1990. **20** (2): p. 155–171.

11. Hiller, S. F. and G. J. Lieberman, Introduction to Operations Research, 7th ed., New Jersey: McGraw-Hill, (2007).

12. Taha, A. H., Operations Research: An Introduction, 8th ed., New York: Pearson Prentice Hall, (2001).

13. Kozanidis, G. and E. Melachrinoudis, *A branch & bound algorithm for the 0-1 mixed integer knapsack problem with linear multiple choice constraints.* Computers & Operations Research, 2004. **31**(5): p. 695-711.

14. Murkute, S., *A Computational study of branching rules for multi-commodity fixed-charged network flow problems.* 2013.

15. Atamtürk, A. and M.W.P. Savelsbergh., "Integer-Programming Software Systems," Annals of Operations Research, 2005. **140**(1), p. 67-124.

16. Patanakul, P., D. Miloseviç, and T. Anderson. *Assigning projects to project managers in a multiple-project management environment: A pilot study of a decision support model*. in *Management of Engineering and Technology, 2003. PICMET'03. Technology Management for Reshaping the World. Portland International Conference on*. 2003. IEEE.

17. Carazo, A.F., T. Gómez, J. Molina, A. J. Hernández-Díaz, F. M. Guerrero, and R. Caballero, "*Solving a comprehensive model for multiobjective project portfolio selection*" Computers & operations research, 2010. **37**(4): p. 630-639.

18. Schmidt, R.L., *A model for R&D project selection with combined benefit, outcome and resource interactions.* Engineering Management, IEEE Transactions on, 1993. **40** (4): p. 403-410.

19. Mahalanobis, P. C., "The Generalised Distance in Statistics," Proceedings of the National Institute of Science of India, 1936. **2** (1): p. 49–55.

20. Criado, C., N. Alamo, H. Rabal, and N. Cap, *Decision and classification problems using Mahalanobis statistical distance. Fuzzy Systems and Knowledge Discovery* (*FSKD*), 2011. p. 1012-1016.

21. Su, C.-T. and Y.-H. Hsiao, *An evaluation of the robustness of MTS for imbalanced data.* IEEE Transactions on Knowledge & Data Engineering, 2007. (10): p. 1321-1332.

22. Srinivasaraghavan, J. and V. Allada, "Application of Mahalanobis Distance as a Lean Assessment Metric," Int'l J. Advanced Manufacturing Technology, 2006. (29): p. 1159-1168.

23. Riho, T., A. Suzuki, J. Oro, K. Ohmi, and H. Tanaka, "The Yield Enhancement Methodology for Invisible Defects Using the MTS Method," IEEE Trans. Semiconductor Manufacturing, 2005. **18** (4): p. 561-568.

24. Das P and S. Datta, “Exploring the effects of chemical composition in hot rolled steel product using Mahalanobis distance scale under Mahalanobis Taguchi system,” Computational Materials Science, 2007. **38** (4): p. 671-677.

25. Bard, J.F., R. Balachandra, and P.E. Kaufmann, An interactive approach to R&D project selection and termination. Engineering Management, IEEE Transactions on, 1988. **35**(3): p. 139-146.

26. Yavuz, S. and T. Captain, Making project selection decisions: a multi-period capital budgeting problem. International Journal of Industrial Engineering, 2002. **9**(3): p. 301-310.

27. Beaujon, G.J., S.P. Marin, and G.C. McDonald, Balancing and optimizing a portfolio of R&D projects. Naval Research Logistics (NRL), 2001. **48**(1): p. 18-40.

28. Hall, N.G., J.C. Hershey, L.G. Kessler, and R.C. Stotts, A model for making project funding decisions at the National Cancer Institute. Operations Research, 1992. **40**(6): p. 1040-1052.

29. Güney, E., S. Ercan, V. Çakir, and I. Düzdar, *A Mathematical Model and Solution Techniques for the Influential Selection Problem. IIE Annual Conference. Proceedings,* 2013. Institute of Industrial Engineers-Publisher.

30. Anwar, A.A. and A. Bahaj, *Student project allocation using integer programming.* Education, IEEE Transactions on, 2003. **46**(3): p. 359-367.

31. Miner, G., R. Nisbet, and J. Elder IV, *Handbook of statistical analysis and data mining applications*. 2009: Academic Press.

32. Fernández-Delgado, M., E. Cernadas, S. Barro, and D. Amorim, *Do we need hundreds of classifiers to solve real world classification problems?* The Journal of Machine Learning Research, 2014. **15**(1): p. 3133-3181.

33. Wang, H.-C., C.-C. Chiu, and C.-T. Su, *DATA CLASSIFICATION USING THE MAHALANOBIS—TAGUCHI SYSTEM.* Journal of the Chinese Institute of Industrial Engineers, 2004. **21**(6): p. 606-618.

34. Taguchi, G., Y. Wu, and S. Chodhury, *Mahalanobis-Taguchi System*. 2000: McGraw-Hill Professional.

35. Long, B., W. Xian, M. Li, and H. Wang, *Improved diagnostics for the incipient faults in analog circuits using LSSVM based on PSO algorithm with Mahalanobis distance.* Neurocomputing, 2014. **133**: p. 237-248.

36. Cho, S., H. Hong, and B.-C. Ha, *A hybrid approach based on the combination of variable selection using decision trees and case-based reasoning using the Mahalanobis distance: For bankruptcy prediction.* Expert Systems with Applications, 2010. **37**(4): p. 3482-3488.

37. Arathi, M. and A. Govardhan. *Effect of Mahalanobis Distance on Time Series Classification Using Shapelets*. in *Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India CSI Volume 2*. 2015. Springer.

38. Fan, Z., M. Ni, M. Sheng, Z. Wu, and B. Xu. *Principal Component Analysis Integrating Mahalanobis Distance for Face Recognition*. *Robot, Vision and Signal Processing (RVSP), 2013 Second International Conference on*. 2013. IEEE.

39. Stöckl, S. and M. Hanke, *Financial Applications of the Mahalanobis Distance.* Applied Economics and Finance, 2014. **1**(2): p. 78-84.

40. Rangayyan, R.M., N.M. El-Faramawy, J.L. Desautels, and O.A. Alim. *Measures of acutance and shape for classification of breast tumors.* Medical Imaging, IEEE Transactions on, 1997. **16**(6): p. 799-810.

41. Achterberg, T., T. Koch, and A. Martin, *Branching rules revisited.* Operations Research Letters, 2005. **33**(1): p. 42-54.

42. Winston, W.L. and J.B. Goldberg, *Operations research: applications and algorithms*. Vol. 3. 2004: Duxbury press Boston.

43. Borchers, B., *Improved branch and bound algorithms for integer programming*. 1992, Rensselaer Polytechnic Institute.

44. Koch, T., et al., *MIPLIB 2010.* Mathematical Programming Computation, 2011. **3**(2): p. 103-163.

45. Wang, Y., D.P. Warsing, E.E. King, A. Vila-Parrish, and S. Sebnem. *Unreliable Supplier Selection with Fixed Costs and Order Constraints*. *IIE Annual Conference. Proceedings*. 2012. Institute of Industrial Engineers-Publisher.

46. Baker, N. and J. Freeland, *Recent advances in R&D benefit measurement and project selection methods*. Management science, 1975. **21**(10): p. 1164-1175.

47. MathWorks. *Statistics and Machine Learning Toolbox: Mahalanobis Distance*. [cited 2016 January 9]; User's Guide (R2014b)]; Available from: http://www.mathworks.com/help/stats/mahal.html.

48. Achterberg, T., *Constraint Integer Programming*, in *Mathematical Department*. 2007, Zuse Institute Berlin.

49. Achterberg, T., *SCIP: solving constraint integer programs.* Mathematical Programming Computation, 2009. **1**(1): p. 1-41.

50. Fügenschuh, A. and A. Martin, *Computational integer programming and cutting planes.* Handbooks in Operations Research and Management Science, 2005. **12**: p. 69-121.

51. Land, A. and S. Powell, *Computer codes for problems of integer programming.* Discrete Optimization II, 1979: p. 221-269.

52. Linderoth, J.T. and M.W. Savelsbergh, *A computational study of search strategies for mixed integer programming.* INFORMS Journal on Computing, 1999. **11**(2): p. 173-187.

53. Wolsey, L.A., *Integer programming*. Vol. 42. 1998: Wiley New York.

54. The US Government Spending. *Total Budgeted Government Spending*. [cited 2016 February 7]; Available from: http://www.usgovernmentspending.com/.

55. The American Road & Transportation Builders Association (ARTBA). [cited 2015 November 3]; Available from: www.artba.org/about/transportation-faqs/.

56. Garcia-Dial, A. and F. Cediel-Franco, *Evaluation of In-house Maintenance Contract Costs for Guardrail.* Rest Areas, Pavement Marker, Striping and Seal Coats, Texas Transportation Institute, Texas A & M University System, 1988.

57. Transportation Economics Committee. *Transportation Benefit-Cost Analysis*. [cited 2016 February 2]; Available from: http://bca.transportationeconomics.org/.

58. Transportation Research Board. Estimating the Benefits and Costs of Public Transit Projects: A Guidebook for Practitioners, (TCRP Report 78), Washington, DC, 2002.

59. Government Accountability Office. Highway and Transit Investments: Options for Improving Information on Projects' Benefits and Costs and Increasing Accountability for Results(GAO-05-172), Washington, DC, January 2005.

60. Section 1909 Commission Staff. *Analysis of Future Issues and Changing Demands on the System* 2007; Part K. Planning and Implementation of Transportation Infrastructure Projects, Washington, DC, January 2007.

61. California Department of Transportation. Life-Cycle Benefit-Cost Analysis Model. [cited 2016 February 11]; Available from: http://dot.ca.gov/hq/tpp/offices/eab/LCBC_Analysis_Model.html.

62. Smith, J.R. and G.B. Way, *Asphalt-Rubber Chip Seal and Polymer Modified Asphalt-Rubber Two Layer System, Case Studies*, in *Asphalt Rubber Conference* 2012: Munich.

63. Homewyse. Cost to Apply Slurry Seal. [cited 2016 March 1]; Available from: http://www.homewyse.com/services/cost_to_apply_slurry_seal.html.

64. Padfield, J., J. Handy, and J. Stephens, Seal Coat Productivity. 2014.

65. Civil Engineer Support Office Naval Construction Battalion Center, *Seabee Planner's and Estimator's Handbook*. 1996.

66. MathWorks. *Optimization Toolbox: Mixed-Integer Linear Programming Algorithms*. [cited 2016 March 16]; User's Guide (R2016a)]; Available from: http://www.mathworks.com/help/optim/ug/mixed-integer-linear-programming-algorithms.html.

67. Gilpin, A. and T. Sandholm, *Information-theoretic approaches to branching in search.* Discrete Optimization, 2011. **8**(2): p. 147-159.

68. Arkat, J., H. Abdollahzadeh, and H. Ghahve, *A new branch and bound algorithm for cell formation problem.* Applied Mathematical Modelling, 2012. **36**(10): p. 5091-5100.

69. Bozdogan, H., Model selection and Akaike's information criterion (AIC): The general theory and its analytical extensions. Psychometrika, 1987. **52**(3): p. 345-370.

70. Pamukçu, E., H. Bozdogan, and S. Çalık, *A Novel Hybrid Dimension Reduction Technique for Undersized High Dimensional Gene Expression Data*

Sets Using Information Complexity Criterion for Cancer Classification. Computational and mathematical methods in medicine, 2015. **2015**.

APPENDIX

APPENDIX 1

MATLAB's R2014 b optimization toolbox is designed to solve Mixed-integer linear programming (MILP) problems by the function (*intlinprog*). This function can be modified with its options and other related constraints to solve any pure IP and BIP linear problems. These options can be found in the MATLAB R2014 b manual. For the reader's convenience, all of options and default settings are shown in Table 15 [66].

Table 14: MATLAB R2014 b Available Options and Default Settings

Option	**Description**	**Default setting**
BranchingRule	Rule for choosing the component for branching: • 'maxpscost' — The fractional component with maximum pseudocost. • 'mostfractional' — The component whose fractional part is closest to 1/2. • 'maxfun' — The fractional component with maximal corresponding component in the absolute value of objective vector f.	'maxpscost'
CutGeneration	Level of cut generation (see Cut Generation): • 'none' — No cuts. Makes CutGenMaxIter irrelevant. • 'basic' — Normal cut generation. • 'intermediate' — Use more cut types. • 'advanced' — Use most cut types.	'basic'
CutGenMaxIter	Number of passes through all cut generation methods before entering the branch-and-bound phase, an integer from 1 through 50. Disable cut generation by setting the CutGeneration option to 'none'.	10
Display	Level of display (see Iterative Display): 'off' or 'none' — No iterative display 'final' — Show final values only 'iter' — Show iterative display	'iter'
Heuristics	Algorithm for searching for feasible points (see Heuristics for Finding Feasible Solutions): • 'none' • 'rss' • 'round' • 'rins'	'rss'

Table 14: Continued

Option	Description	Default setting
HeuristicsMaxNodes	Strictly positive integer that bounds the number of nodes intlinprog can explore in its branch-and-bound search for feasible points. See Heuristics for Finding Feasible Solutions.	50
IPPreprocess	Types of integer preprocessing (see Mixed-Integer Program Preprocessing): • 'none' — Use very few integer preprocessing steps. • 'basic' — Use a moderate number of integer preprocessing steps. • 'advanced' — Use all available integer preprocessing steps.	'basic'
LPMaxIter	Strictly positive integer, the maximum number of simplex algorithm iterations per node during the branch-and-bound process.	3e4
LPPreprocess	Type of preprocessing for the solution to the relaxed linear program (see Linear Program Preprocessing): • 'none' — No preprocessing. • 'basic' — Use preprocessing.	'basic'
MaxNodes	Strictly positive integer that is the maximum number of nodes intlinprog explores in its branch-and-bound process.	1e7
MaxNumFeasPoints	Strictly positive integer. intlinprog stops if it finds MaxNumFeasPoints integer feasible points.	Inf
MaxTime	Positive real that is the maximum time in seconds that intlinprog runs.	7200
NodeSelection	Choose the node to explore next. • 'simplebestproj' — Best projection. See Branch and Bound. • 'minobj' — Explore the node with the minimum objective function. • 'mininfeas' — Explore the node with the minimal sum of integer infeasibilities. See Branch and Bound.	'simplebestproj'
ObjectiveCutOff	Real greater than -Inf. During the branch-and-bound calculation, intlinprog discards any node where the linear programming solution has an objective value exceeding ObjectiveCutOff.	Inf

Table 14: Continued

Option	**Description**	**Default setting**
OutputFcn	Specify one or more functions that an optimization function calls at events, either as a function handle or as a cell array of function handles. @savemilpsolutions collects the integer-feasible points in the xIntSol matrix in your workspace, where each column is one integer feasible point.	[ ]
PlotFcns	Plots various measures of progress while the algorithm executes, select from predefined plots or write your own. Pass a function handle or a cell array of function handles. @optimplotmilp plots the internally-calculated upper and lower bounds on the objective value of the solution.	[ ]
RelObjThreshold	Nonnegative real. intlinprog changes the current feasible solution only when it locates another with an objective function value that is at least RelObjThreshold lower: (fold – fnew)/(1 + fold) > RelObjThreshold.	1e-4
RootLPAlgorithm	Algorithm for solving linear programs: • 'dual-simplex' — Dual simplex algorithm • 'primal-simplex' — Primal simplex algorithm	'dual-simplex'
RootLPMaxIter	Nonnegative integer that is the maximum number of simplex algorithm iterations to solve the initial linear programming problem.	3e4
TolCon	Real from 1e-9 through 1e-3 that is the maximum discrepancy that linear constraints can have and still be considered satisfied. TolCon is not a stopping criterion.	1e-4
TolFunLP	Nonnegative real where reduced costs must exceed TolFunLP for a variable to be taken into the basis.	1e-7
TolGapAbs	Nonnegative real. intlinprog stops if the difference between the internally calculated upper (U) and lower (L) bounds on the objective function is less than or equal to TolGapAbs: U – L <= TolGapAbs	0

Table 14: Continued

Option	**Description**	**Default setting**
TolGapRel	Real from 0 through 1. intlinprog stops if the relative difference between the internally calculated upper (U) and lower (L) bounds on the objective function is less than or equal to TolGapRel: (U – L) / (abs(U) + 1) <= TolGapRel.	1e-4
TolInteger	Real from 1e-6 through 1e-3, where the maximum deviation from integer that a component of the solution x can have and still be considered an integer. TolInteger is not a stopping criterion.	1e-5

APPENDIX 2

Factor *N* Levels Data set

The experimental data set, multivariate observation, for factor *N* levels are shown in Tables 15,16,17,18, and 19 for the levels of 20, 40, 60, 80, and 100, respectively.

Table 15: Seal Coat Project Data Set, *n*=20

Project No.	ADT	GOFA	CYA	MIL	TYS	TCOS
1	193	19950	601	6.160	72336	35197
2	1200	5675	167	0.895	18573	10390
3	300	48150	1204	10.669	126468	82887
4	5950	25525	658	2.850	67530	46419
5	1507	20640	585	4.469	59136	36643
6	1995	6978	160	1.000	18186	12900
7	450	28357	1026	7.343	86158	69100
8	474	56650	1430	7.250	194036	54023
9	1200	28005	854	6.060	73046	54132
10	600	55490	1700	10.890	155280	77590
11	2057	16780	468	2.984	46264	29538
12	310	10300	279	2.500	35090	21965
13	336	26368	650	7.509	89967	56684
14	935	29666	836	7.230	84832	66900
15	1712	32345	900	7.172	97640	62796
16	190	10218	250	2.270	26635	13500
17	4900	13140	480	2.840	40691	28575
18	500	47355	1107	12.788	150941	69163
19	369	71320	1680	18.202	218944	116348
20	623	30031	839	7.462	86240	66952

Table 16: Seal Coat Project Data Set, *n*=40

Project No.	ADT	GOFA	CYA	MIL	TYS	TCOS
1	220	12250	294	2.58	30792	17352
2	930	8675	266	2.782	32646	15245
3	11274	48415	1536	2.879	147367	114529
4	850	48955	1442	15.199	178340	84492
5	70	7500	210	2.17	25500	21700
6	1448	35955	1008	6.049	107156	69810
7	2000	99760	2714	16.225	260285	178048
8	390	18910	420	4.637	54416	29892
9	7500	1445	50	0.252	4184	3064
10	890	97882	3274	21.267	300778	203359
11	360	27366	772	6.672	78285	49960
12	730	52890	1850	7.4	157445	95000
13	1000	6590	168	2.082	22200	9452
14	850	7127	190	1.583	18574	12301
15	3500	45120	1572	3.857	96661	97268
16	769	16204	463	3.958	46441	38743
17	201	13075	303	3.454	40650	21021
18	1196	63970	2124	12.042	169563	123393
19	450	23250	560	4.817	58664	40833
20	1300	58840	1904	9.268	152250	102427
21	573	42380	1080	13.68	144977	120900
22	3318	99300	2376	6.108	257959	171704
23	1100	71550	1792	13.81	183271	107987
24	1400	51310	1716	9.459	132249	99393
25	3317	41005	1060	9.69	141402	95270
26	250	14385	490	4.213	55953	29804
27	1200	28005	854	6.06	73046	54132
28	767	9855	274	2.241	26635	18262
29	1875	20000	518	5.2	67972	37905
30	346	17637	503	4.305	50512	38500
31	4274	17200	450	3.18	58730	42047
32	6873	60465	1914	4.523	178170	115822
33	1896	62510	1884	13.227	184456	128811
34	724	20730	600	5.243	65137	37742
35	160	14370	368	2.952	36130	24511
36	700	52250	1330	11.819	140772	86537

Table 16: Continued

Project No.	ADT	GOFA	CYA	MIL	TYS	TCOS
37	474	56650	1430	7.25	194036	54023
38	251	24290	58	7.279	76621	39020
39	950	10225	255	2.303	27023	17845
40	3488	64990	1818	4.642	174577	116151

Table 17: Seal Coat Project Data Set, *n*=60

Project No.	ADT	GOFA	CYA	MIL	TYS	TCOS
1	220	12250	294	2.58	30792	17352
2	930	8675	266	2.782	32646	15245
3	11274	48415	1536	2.879	147367	114529
4	850	48955	1442	15.199	178340	84492
5	70	7500	210	2.17	25500	21700
6	1448	35955	1008	6.049	107156	69810
7	2000	99760	2714	16.225	260285	178048
8	390	18910	420	4.637	54416	29892
9	7500	1445	50	0.252	4184	3064
10	890	97882	3274	21.267	300778	203359
11	360	27366	772	6.672	78285	49960
12	730	52890	1850	7.4	157445	95000
13	1000	6590	168	2.082	22200	9452
14	850	7127	190	1.583	18574	12301
15	3500	45120	1572	3.857	96661	97268
16	769	16204	463	3.958	46441	38743
17	201	13075	303	3.454	40650	21021
18	1196	63970	2124	12.042	169563	123393
19	450	23250	560	4.817	58664	40833
20	1300	58840	1904	9.268	152250	102427
21	573	42380	1080	13.68	144977	120900
22	3318	99300	2376	6.108	257959	171704
23	1100	71550	1792	13.81	183271	107987
24	1400	51310	1716	9.459	132249	99393
25	3317	41005	1060	9.69	141402	95270
26	250	14385	490	4.213	55953	29804
27	1200	28005	854	6.06	73046	54132

Table 17: Continued

Project No.	ADT	GOFA	CYA	MIL	TYS	TCOS
28	767	9855	274	2.241	26635	18262
29	1875	20000	518	5.2	67972	37905
30	346	17637	503	4.305	50512	38500
31	4274	17200	450	3.18	58730	42047
32	6873	60465	1914	4.523	178170	115822
33	1896	62510	1884	13.227	184456	128811
34	724	20730	600	5.243	65137	37742
35	160	14370	368	2.952	36130	24511
36	700	52250	1330	11.819	140772	86537
37	474	56650	1430	7.25	194036	54023
38	251	24290	58	7.279	76621	39020
39	950	10225	255	2.303	27023	17845
40	3488	64990	1818	4.642	174577	116151
41	400	36520	840	10.934	115710	58620
42	4200	38900	1042	3.693	104019	70803
43	7500	193965	6216	15.636	526798	382165
44	5600	71155	2564	15.49	218104	154638
45	6100	10881	321	1.056	30976	19819
46	3450	55668	1320	9.65	145141	93942
47	2600	81010	2849	15.972	231810	164098
48	951	21400	564	5.688	77166	36666
49	1750	13000	575	3.065	43155	25900
50	12200	28510	750	5.6	97357	65308
51	836	48995	1440	12.267	143933	83695
52	1841	67130	1708	16.049	226573	113715
53	2063	18570	574	5.047	71318	34824
54	353	34560	1080	7.954	93280	60561
55	7486	51627	1667	6.13	173930	107617
56	336	18050	343	4	46933	32497
57	450	11462	394	2.031	28600	16710
58	492	22230	623	5.285	62011	41714
59	330	3700	70	0.766	10452	5467
60	1814	25920	1365	6.665	113446	81092

Table 18: Seal Coat Project Data Set, *n*=80

Project No.	ADT	GOFA	CYA	MIL	TYS	TCOS
1	6204	27445	859	3.375	77929	50871
2	9636	80480	2184	4.855	236291	143764
3	985	32740	924	9.618	119586	58052
4	1152	17327	410	3.85	45173	16783
5	350	32825	1008	9.362	110820	59953
6	1100	17290	518	3.654	44510	33213
7	370	11231	292	3.099	37951	29536
8	3318	99300	2376	6.108	257959	171704
9	1044	32420	832	5.978	82794	55361
10	645	19200	444	5.204	61211	29095
11	360	19280	434	4.828	56826	30784
12	336	26368	650	7.509	89967	56684
13	400	36520	840	10.934	115710	58620
14	3300	32780	812	2.9	84294	60117
15	550	18105	560	4.98	60822	33182
16	4700	42090	1572	9.562	145863	88680
17	5908	24915	664	6.53	95161	45164
18	730	52890	1850	7.4	157445	95000
19	5478	92570	2551	6.131	265838	178193
20	103	17710	420	4.545	53659	28985
21	180	41687	1367	9.83	117555	73036
22	265	37585	892	10.243	120494	61100
23	470	13700	376	3.328	40356	24236
24	6975	27540	774	2.16	69404	44935
25	216	13148	390	3.163	37113	28617
26	308	42763	994	11.886	140069	54971
27	280	14040	318	3.601	42372	22336
28	20	3032	88	0.673	7897	5400
29	70	9500	290	2.11	27116	13023
30	350	56650	1560	11.78	40069	79849
31	190	10218	250	2.27	26635	13500
32	6000	17029	515	3.6	42240	18642
33	3436	30790	924	8.003	119093	56821
34	1400	51310	1716	9.459	132249	99393
35	223	9440	281	2.29	26870	20500
36	6600	3050	89	0.7	9798	5297

Table 18: Continued

Project No.	ADT	GOFA	CYA	MIL	TYS	TCOS
37	5600	71155	2564	15.49	218104	154638
38	1875	20000	518	5.2	67972	37905
39	2700	6050	370	1.678	23626	16538
40	17400	34770	936	1.549	97939	60879
41	2834	57460	1755	13.183	178487	101099
42	12424	34475	969	4.038	98560	83500
43	1100	20890	624	5.415	80595	38142
44	700	52250	1330	11.819	140772	86537
45	1354	43940	1410	8.607	121580	81951
46	2500	56234	1800	10	164267	144700
47	2129	90990	2940	17.73	249640	176437
48	3688	43225	1148	5.554	122641	81482
49	411	47830	1140	10.678	148092	78092
50	338	18140	426	4.782	57049	26537
51	1100	71550	1792	13.81	183271	107987
52	5866	11950	336	0.825	32270	26009
53	188	25160	593	6.999	82131	40785
54	220	13217	310	3.575	43204	22628
55	210	23925	566	6.666	78300	38849
56	2246	33450	860	8.01	114759	68890
57	1696	102050	2786	15.435	282418	237262
58	200	7090	204	0.946	15556	10814
59	400	40673	965	8.8	103000	60090
60	3488	64990	1818	4.642	174577	116151
61	530	1381	53	0.352	4130	3400
62	2507	10600	270	2.3	35778	17269
63	1311	89863	3016	23.727	279098	187077
64	16155	45990	1665	2.82	146220	115524
65	689	15350	344	3.967	47503	23441
66	844	1500	245	0.805	10791	6570
67	346	17637	503	4.305	50512	38500
68	2000	68501	2545	14.226	203852	135341
69	3249	55915	1820	10.889	153322	108827
70	300	48150	1204	10.669	126468	82887
71	440	12900	378	3.212	37688	30200
72	70	9208	241	2.64	30999	23400

Table 18: Continued

Project No.	ADT	GOFA	CYA	MIL	TYS	TCOS
73	3600	4373	136	0.899	12658	9623
74	590	6530	196	1.373	17723	11267
75	5557	27925	736	2.559	69381	52398
76	6300	21668	850	4.788	67415	53900
77	1800	39560	1080	8.861	140652	68977
78	1039	29925	634	5.466	64134	44364
79	4200	38900	1042	3.693	104019	70803
80	5600	57469	1884	10.966	167725	82700

Table 19: Seal Coat Project Data Set, *n*=100

Project No.	ADT	GOFA	CYA	MIL	TYS	TCOS
1	260	12385	281	2.629	30845	22337
2	63	15349	390	4.44	52096	42900
3	10355	59200	1654	5.437	204178	147397
4	5908	24915	664	6.53	95161	45164
5	210	23925	566	6.666	78300	38849
6	11358	17080	488	2.331	48156	31019
7	500	19135	550	4.364	56332	32266
8	453	31450	825	6.503	89096	60203
9	568	18092	510	4.094	52096	42800
10	590	6530	196	1.373	17723	11267
11	201	13075	303	3.454	40650	21021
12	440	10595	288	2.73	32000	18647
13	1258	84760	2892	16.479	232036	173637
14	1300	58840	1904	9.268	152250	102427
15	401	12850	224	2.955	34672	20606
16	850	7127	190	1.583	18574	12301
17	347	13575	302	3.946	41732	19453
18	1000	12700	368	3.313	38876	21832
19	292	14275	333	3.87	46258	24241
20	6839	73948	2406	7.92	236976	154822
21	180	41687	1367	9.83	117555	73036
22	220	13217	310	3.575	43204	22628
23	2834	57460	1755	13.183	178487	101099

Table 19: Continued

Project No.	ADT	GOFA	CYA	MIL	TYS	TCOS
24	1200	5675	167	0.895	18573	10390
25	11309	18445	520	4.479	51822	38519
26	986	20625	588	5.1	59833	40773
27	1700	15590	522	2.966	41776	29956
28	4014	105080	3339	15.645	280816	272053
29	649	17680	408	4.277	54872	29648
30	11918	43835	1280	2.194	130670	86385
31	6872	66705	1870	6.76	190362	126900
32	500	47355	1107	12.788	150941	69163
33	450	23250	560	4.817	58664	40833
34	5404	128535	3374	14.071	365715	271611
35	1270	50015	1440	12.25	157092	94129
36	3600	4373	136	0.899	12658	9623
37	1689	9150	264	2.425	33870	16408
38	12200	28510	750	5.6	97357	65308
39	3436	30790	924	8.003	119093	56821
40	602	26060	764	6.199	73503	46794
41	20	3032	88	0.673	7897	5400
42	826	28310	552	6.456	76965	30568
43	2246	33450	860	8.01	114759	68890
44	479	36151	1021	8.851	103052	85600
45	2507	10600	270	2.3	35778	17269
46	2624	34596	979	4	98560	50144
47	4900	13140	480	2.84	40691	28575
48	1136	25300	714	6.568	92490	45188
49	7500	193965	6216	15.636	526798	382165
50	470	8060	224	1.695	20000	14355
51	220	12250	294	2.58	30792	17352
52	350	17050	747	4.015	56533	34243
53	6204	27445	859	3.375	77929	50871
54	455	25771	605	7.005	82244	41688
55	1600	14515	427	1.421	38980	26174
56	308	42763	994	11.886	140069	54971
57	193	19950	601	6.16	72336	35197
58	950	10225	255	2.303	27023	17845
59	12000	82885	2831	8.154	270440	171959

Table 19: Continued

Project No.	ADT	GOFA	CYA	MIL	TYS	TCOS
60	6000	17029	515	3.6	42240	18642
61	1882	40925	1160	11.935	153085	76820
62	290	17605	397	4.545	53333	25777
63	1400	51310	1716	9.459	132249	99393
64	2712	6150	200	0.513	15650	11472
65	152	36650	868	10.328	118022	56438
66	4500	33450	980	3.773	97601	58544
67	1000	79800	2448	12.53	221519	168402
68	310	30530	666	6.514	76431	40285
69	360	19280	434	4.828	56826	30784
70	530	5540	150	1.372	14488	11000
71	320	21244	601	5.19	60896	39149
72	2100	18710	559	3.811	58343	31408
73	400	5725	165	1.212	17064	11070
74	1814	25920	1365	6.665	113446	81092
75	350	32825	1008	9.362	110820	59953
76	935	29666	836	7.23	84832	66900
77	11942	21100	605	1.6	60075	39558
78	769	16204	463	3.958	46441	38743
79	210	27600	658	5.78	69769	39188
80	1370	56420	1522	10.375	152150	103586
81	1000	17866	428	3.5	47168	29373
82	600	55490	1700	10.89	155280	77590
83	4200	38900	1042	3.693	104019	70803
84	244	12340	356	3.35	35376	24213
85	520	15285	504	2.292	46490	31169
86	3500	45120	1572	3.857	96661	97268
87	4600	74225	2009	7.025	197297	139889
88	160	14370	368	2.952	36130	24511
89	12424	34475	969	4.038	98560	83500
90	104	30305	690	8.105	95450	43788
91	3000	12000	333	2.335	32877	22531
92	650	17950	420	3.962	46658	32022
93	952	9639	225	2.149	25125	13210
94	440	6150	144	0.987	19460	9959
95	4700	42090	1572	9.562	145863	88680

Table 19: Continued

Project No.	ADT	GOFA	CYA	MIL	TYS	TCOS
96	7500	1445	50	0.252	4184	3064
97	3680	28735	841	5.005	81189	43659
98	836	48995	1440	12.267	143933	83695
99	800	6737	141	1.5	17600	9803
100	1252	21740	636	4.084	62305	39776

Complete Experimental Results

Each experimental combination result, with replications, for both strategies and formulations is represented in Tables 20 and 21.

Table 20: Replicated Experimental Results for Solving Model 1

		Branching Strategy Type							
		Default				**Proposed**			
N	***M***	Time (sec.)	Abs. Gap	Relative Gap (%)	Best Sub-Set	Time (sec.)	Abs. Gap	Relative Gap (%)	Best Sub-Set
20	5	275.195	8	0.0035	(5,11,4)	207.506	2	0.0004	(3,13,4)
		268.371	8	0.0035	(5,11,4)	206.138	2	0.0004	(3,13,4)
		271.461	8	0.0035	(5,11,4)	204.511	2	0.0004	(3,13,4)
	10	522.99	75	0.0367	(0,14,6)	412.685	12	0.0054	(0,13,7)
		510.121	75	0.0367	(0,14,6)	411.468	12	0.0054	(0,13,7)
		527.863	75	0.0367	(0,14,6)	410.383	12	0.0054	(0,13,7)
	15	471.997	69	0.0334	(0,15,5)	458.837	12	0.0054	(0,13,7)
		471.048	69	0.0334	(0,15,5)	458.711	12	0.0054	(0,13,7)
		472.084	69	0.0334	(0,15,5)	458.685	12	0.0054	(0,13,7)
40	5	426.138	46	0.0081	(7,19,14)	392.19	11	0.0017	(8,19,13)
		425.622	46	0.0081	(7,19,14)	391.335	11	0.0017	(8,19,13)
		425.914	46	0.0081	(7,19,14)	392.396	11	0.0017	(8,19,13)
	10	1245.916	23	0.0039	(0,32,8)	823.001	3	0.0003	(1,29,10)
		1243.513	23	0.0039	(0,32,8)	823.279	3	0.0003	(1,29,10)
		1244.181	23	0.0039	(0,32,8)	822.145	3	0.0003	(1,29,10)
	15	898.495	25	0.0044	(0,24,16)	30.13	2	0.0001	(0,29,11)
		898.634	25	0.0044	(0,24,16)	35.766	2	0.0001	(0,29,11)
		902.187	25	0.0044	(0,24,16)	34.946	2	0.0001	(0,29,11)

Table 20: Continued

		Branching Strategy Type							
		Default				Proposed			
N	***M***	Time (sec.)	Abs. Gap	Relative Gap (%)	Best Sub-Set	Time (sec.)	Abs. Gap	Relative Gap (%)	Best Sub-Set
60	5	689.314	10	0.001	(12,37,11)	612.654	6	0.0005	(10,35,15)
		684.858	10	0.001	(12,37,11)	617.219	6	0.0005	(10,35,15)
		687.162	10	0.001	(12,37,11)	613.863	6	0.0005	(10,35,15)
	10	1326.616	138	0.0155	(0,39,21)	945.239	20	0.0021	(0,38,22)
		1332.763	138	0.0155	(0,39,21)	966.613	20	0.0021	(0,38,22)
		1331.271	138	0.0155	(0,39,21)	955.104	20	0.0021	(0,38,22)
	15	1412.114	4790	0.5454	(0,36,24)	539.847	21	0.0023	(0,38,22)
		1412.351	4790	0.5454	(0,36,24)	539.625	21	0.0023	(0,38,22)
		1417.109	4790	0.5454	(0,36,24)	537.713	21	0.0023	(0,38,22)
80	5	690.833	30	0.0032	(14,53,13)	562.569	0	0	(23,45,12)
		693.123	30	0.0032	(14,53,13)	562.43	0	0	(23,45,12)
		689.938	30	0.0032	(14,53,13)	562.314	0	0	(23,45,12)
	10	1674.168	20	0.0019	(2,59,19)	744.33	30	0.0033	(1,62,17)
		1673.158	20	0.0019	(2,59,19)	743.132	30	0.0033	(1,62,17)
		1673.187	20	0.0019	(2,59,19)	743.161	30	0.0033	(1,62,17)
	15	1747.586	20	0.0024	(4,20,76)	527.727	0	0	(0,57,23)
		1748.132	20	0.0024	(4,20,76)	522.715	0	0	(0,57,23)
		1750.109	20	0.0024	(4,20,76)	522.894	0	0	(0,57,23)

Table 20: Continued

		Branching Strategy Type							
		Default				**Proposed**			
N	***M***	Time (sec.)	Abs. Gap	Relative Gap (%)	Best Sub-Set	Time (sec.)	Abs. Gap	Relative Gap (%)	Best Sub-Set
100	5	1516.371	0	0.0003	(21,52,27)	26.234	0	0	(13,55,32)
		1510.759	0	0.0003	(21,52,27)	25.908	0	0	(13,55,32)
		1509.318	0	0.0003	(21,52,27)	26.315	0	0	(13,55,32)
	10	1286.558	180	0.0151	(1,67,32)	737.487	0	0	(1,62,37)
		1286.117	180	0.0151	(1,67,32)	737.119	0	0	(1,62,37)
		1287.816	180	0.0151	(1,67,32)	739.715	0	0	(1,62,37)
	15	1590.1246	60	0.0058	(0,65,35)	1143.075	0	0	(1,61,38)
		1593.143	60	0.0058	(0,65,35)	1159.81	0	0	(1,61,38)
		1591.117	60	0.0058	(0,65,35)	1155.856	0	0	(1,61,38)

Table 21: Replicated Experimental Results for Solving Model 2

		Branching Strategy Type									
		Default					**Proposed**				
N	***M***	Time (sec.)	Abs. Gap	Relative Gap (%)	No. Nodes	Best Sub-Set	Time (sec.)	Abs. Gap	Relative Gap (%)	No. Nodes	Best Sub-Set
20	5	0.066	0	0	N.A.	(7,7,6)	0.064	0	0	N.A.	(0,14,6)
		0.065	0	0	N.A.	(7,7,6)	0.059	0	0	N.A.	(0,14,6)
		0.064	0	0	N.A.	(7,7,6)	0.07	0	0	N.A.	(0,14,6)
	10	0.287	0	0	N.A.	(0,13,7)	0.242	0	0	N.A.	(0,14,6)
		0.35	0	0	N.A.	(0,13,7)	0.376	0	0	N.A.	(0,14,6)
		0.279	0	0	N.A.	(0,13,7)	0.316	0	0	N.A.	(0,14,6)
	15	0.125	0	0	N.A.	(0,13,7)	0.128	0	0	N.A.	(0,13,7)
		0.149	0	0	N.A.	(0,13,7)	0.131	0	0	N.A.	(0,13,7)
		0.158	0	0	N.A.	(0,13,7)	0.139	0	0	N.A.	(0,13,7)
40	5	0.831	0.021	0.01	4790	(2,22,16)	0.588	0.021	0.01	3720	(2,23,15)
		0.781	0.021	0.01	4790	(2,22,16)	0.62	0.021	0.01	3720	(2,23,15)
		0.84	0.021	0.01	4790	(2,22,16)	0.573	0.021	0.01	3720	(2,23,15)
	10	1.252	0	0	7042	(2,22,16)	0.358	0.0206	0.0096	1486	(2,23,15)
		1.21	0	0	7042	(2,22,16)	0.345	0.0206	0.0096	1486	(2,23,15)
		1.206	0	0	7042	(2,22,16)	0.314	0.0206	0.0096	1486	(2,23,15)
	15	0.43	0.0211	0.0097	1894	(0,24,16)	0.314	0.0213	0.01	906	(0,24,16)
		0.449	0.0211	0.0097	1894	(0,24,16)	0.303	0.0213	0.01	906	(0,24,16)
		0.418	0.0211	0.0097	1894	(0,24,16)	0.299	0.0213	0.01	906	(0,24,16)

Table 21: Continued

		Branching Strategy Type									
		Default					Proposed				
N	***M***	Time (sec.)	Abs. Gap	Relative Gap (%)	No. Nodes	Best Sub-Set	Time (sec.)	Abs. Gap	Relative Gap (%)	No. Nodes	Best Sub-Set
60	5	2.762	0.0326	0.0092	18388	(2,33,25)	1.497	0	0	11229	(2,33,25)
		2.804	0.0326	0.0092	18388	(2,33,25)	1.484	0	0	11229	(2,33,25)
		2.706	0.0326	0.0092	18388	(2,33,25)	1.528	0	0	11229	(2,33,25)
	10	11.378	0	0	59836	(2,33,25)	7.306	0.0277	0.0063	31783	(2,33,25)
		11.313	0	0	59836	(2,33,25)	6.108	0.0277	0.0063	31783	(2,33,25)
		11.508	0	0	59836	(2,33,25)	6.154	0.0277	0.0063	31783	(2,33,25)
	15	66.856	0.0344	0.01	149490	(0,40,20)	94.513	0	0	265424	(0,40,20)
		68.209	0.0344	0.01	149490	(0,40,20)	92.809	0	0	265424	(0,40,20)
		69.166	0.0344	0.01	149490	(0,40,20)	94.437	0	0	265424	(0,40,20)
80	5	47.253	0.0368	0.01	332890	(3,43,34)	24.965	0.0313	0.007	159182	(2,44,34)
		48.159	0.0368	0.01	332890	(3,43,34)	25.573	0.0313	0.007	159182	(2,44,34)
		48.305	0.0368	0.01	332890	(3,43,34)	25.871	0.0313	0.007	159182	(2,44,34)
	10	187.668	0.0258	0.0041	799582	(2,44,34)	71.077	0.0367	0.01	291458	(1,45,34)
		189.215	0.0258	0.0041	799582	(2,44,34)	71.77	0.0367	0.01	291458	(1,45,34)
		187.532	0.0258	0.0041	799582	(2,44,34)	70.913	0.0367	0.01	291458	(1,45,34)
	15	16.381	0.0361	0.0095	59394	(0,45,35)	8.765	0.0275	0.0048	34346	(0,46,34)
		15.126	0.0361	0.0095	59394	(0,45,35)	8.699	0.0275	0.0048	34346	(0,46,34)
		15.325	0.0361	0.0095	59394	(0,45,35)	8.489	0.0275	0.0048	34346	(0,46,34)

Table 21: Continued

		Branching Strategy Type									
		Default					Proposed				
N	***M***	Time (sec.)	Abs. Gap	Relative Gap (%)	No. Nodes	Best Sub-Set	Time (sec.)	Abs. Gap	Relative Gap (%)	No. Nodes	Best Sub-Set
100	5	2.107	0.0442	0.01	12150	(5,53,42)	0.866	0.0442	0.01	4542	(5,53,42)
		2.134	0.0442	0.01	12150	(5,53,42)	0.846	0.0442	0.01	4542	(5,53,42)
		2.159	0.0442	0.01	12150	(5,53,42)	0.882	0.0442	0.01	4542	(5,53,42)
	10	1.688	0	0	3508	(3,56,41)	1.723	0	0	3469	(3,56,41)
		1.656	0	0	3508	(3,56,41)	1.733	0	0	3469	(3,56,41)
		1.717	0	0	3508	(3,56,41)	1.768	0	0	3469	(3,56,41)
	15	1354.017	0.1594	0.0613	>2000000	(1,64,35)	497.573	0	0	711288	(1,64,35)
		1355.949	0.1594	0.0613	>2000000	(1,64,35)	497.808	0	0	711288	(1,64,35)
		1354.275	0.1594	0.0613	>2000000	(1,64,35)	498.119	0	0	711288	(1,64,35)

VITA

Abdulaziz Saud Alkabaa was born and grew up in Jeddah, Saudi Arabia. He got his B.S. in Industrial Engineering from King Abdulaziz University in 2006. After he received his B.S., he has worked as an engineer in the Saudi Electricity Company for six years. During his working time, he joined King Abdulaziz University for his graduate studies and received his Master Degree in Industrial Engineering in 2010. In recognition of his hard works, Abdulaziz revived the Excellence GPA awards during his undergraduate and graduate studies form King Abdulaziz University and the Employee of the month from the Saudi Electricity Company.

Abdulaziz applied to King Abdullah’s scholarship program to pursue his doctorate degree in Industrial and Systems Engineering in 2011. Based on Abdulaziz's CV and application, he was selected to continuo his educational journey for the Ph.D. Since joining the UT community, Abdulaziz received the Chancellor’s Extraordinary Professional Promise Award in 2016, Graduate Student Senate travel award, and Saudi Arabian Cultural Mission’s achievement award. He is also pursuing his master’s degree in statistics simultaneously with his Ph.D. Abdulaziz has served as a reviewer for numerous conference proceeding papers.