View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by University of Tennessee, Knoxville: Trace

University of Tennessee, Knoxville
Trace: Tennessee Research and Creative Exchange

Doctoral Dissertations Graduate School

8-2018

Parallel Machine Replacement: An Analysis in Construction Industry with Considerations of Horizon Uncertainty, Multi-Purpose Machines and Transportation

Brett Allen Shields
University of Tennessee, bshield3@vols.utk.edu

Recommended Citation
Shields, Brett Allen, "Parallel Machine Replacement: An Analysis in Construction Industry with Considerations of Horizon Uncertainty, Multi-Purpose Machines and Transportation. " PhD diss., University of Tennessee, 2018.
https://trace.tennessee.edu/utk_graddiss/5043

This Dissertation is brought to you for free and open access by the Graduate School at Trace: Tennessee Research and Creative Exchange. It has been accepted for inclusion in Doctoral Dissertations by an authorized administrator of Trace: Tennessee Research and Creative Exchange. For more information, please contact trace@utk.edu.

To the Graduate Council:

I am submitting herewith a dissertation written by Brett Allen Shields entitled "Parallel Machine Replacement: An Analysis in Construction Industry with Considerations of Horizon Uncertainty, Multi-Purpose Machines and Transportation." 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.

Andrew Yu, Major Professor

We have read this dissertation and recommend its acceptance:

Trevor M. Moeller, James L. Simonton, Janice Tolk

Accepted for the Council:
Dixie L. Thompson

Vice Provost and Dean of the Graduate School

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

Parallel Machine Replacement:
An Analysis in Construction Industry with Considerations of Horizon Uncertainty, Multi-Purpose Machines and Transportation

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

Brett Allen Shields

August 2018

Copyright © 2018 by Brett A. Shields.
All rights reserved.

ACKNOWLEDGEMENTS

I would like to first acknowledge my advisor Dr. Andrew Yu for providing me the recourses to pursue my doctorate. I would also like to acknowledge my good friend Dr. Javad Seif for enduring the process with me, introducing me to the field of PMRP, and always being a great friend! Next I would like to acknowledge my committee members for their insight and input: Dr. James Simonton, Dr. Janice Tolk, and Dr. Trevor Moeller. Lastly, I would like to acknowledge my family for putting up with me for my 9 years in school; FYI nothing will change and I am still eating all of your food and drinking all of your wine, mom.

To my son (s), I dedicated my master's thesis to you Mr. Braeden Kye, and now I will dedicate this dissertation to my small fleet of sons: Braeden Kye and Declan Patrick. I claimed previously, Braeden, that you "kept me trekking on all of these years", and that is still true. I continuously strive to achieve goals of increasing difficulty, and I hope that one day you (and Declan) will do the same. I am exceptionally proud of the young man you are becoming. Also, the offer still stands: $1000 for the first son to take me down.

ABSTRACT

In this dissertation, the procurement and replacement of interdependent assets is considered in which the machines satisfy demand in parallel. A number of realistic scenarios are modelled that are current limitations of the Parallel Machine Replacement Problem (PMRP). Considerations prevalent in construction management provide new formulations of the problem. A stochastic planning horizon is considered which is in line with the direction of the research field. Likewise, multi-purpose challengers are presented to offer a solution to the current heterogeneous fleet limitations. Lastly, shipping considerations for multiple demand sites are studied. New mixed-integer programming models are presented for each problem formulation. Each model considers numerous aspects that are contributions to the current literature for the parallel machine replacement problem. The work integrates the PMRP into construction management. A new solution methodology is presented that offers a usable technique for solving larger systems when shipping is of concern, without the limitations of the current models. The contributions are: considering multiple demand sites with shipping, a heterogeneous fleet, stochastic demand and planning horizon, multi-purpose machines, the ability to work and purchase used assets, applications in construction management and a solution method that is realistic and computationally efficient.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE
INTRODUCTION

Background

Assets are demanded in both industry and government to provide a service or produce a product. As the assets age through time and utilization, it is often the case that operations and maintenance costs increase, while the salvage values decrease, leading to higher costs for demand satisfaction. Not only do salvage, operations and maintenance cost vary, but the purchase price of an asset also changes with time and technology. It is desired to determine the optimal policy for purchasing and salvaging one or many assets over a planning horizon. Many aspects need to be considered to determine the optimal schedule for this problem. Often decisions such as purchasing, renting, selling, holding, and operating need to be determined. In both the single and multiple asset case, utilization of the assets is often studied, as it has an effect on the value of the assets. The problem, known to be combinatorial in nature, is Nondeterministic Polynomial hard (NP-hard), shown by Esra Büyüktahtakin, Cole Smith, Hartman, and Luo (2014), and can be computationally difficult to solve for realistic cases. As seen in the literature, many models are subject to economies of scale, capital budget constraints, and rapid technology changes. The basis of this research is that of an engineering economic analysis in asset replacement, yet a number of realistic instances have been included. In general, models have been applied to optimize multiple types of equipment, such as: buses, construction machines, aircraft, and medical devices (J. C. Hartman & C. H. Tan, 2014; J. Seif, Yu, & Shields, 2017).

In 2017, the top ten highest valued construction projects totaled over 400 BN USD alone (ARCADIS, 2017). Much of the cost of these projects are in the management of the equipment. Ownership cost of machinery can vary from 20 to 200 times the purchase cost (Ryan, 1968). It is always economical to replace the machines after a certain amount of usage because as the machines age, salvage value decreases and operations and maintenance (O&M) costs increase (Verheyen, 1979). Therefore, it is very important to find the optimal schedule for replacement of machinery and the application of this research is presented in construction asset management. Optimal replacement of equipment has been studied within the past century (Joseph C Hartman & Chin Hon Tan, 2014). When the machines are economically interdependent, it is not optimal to schedule their salvage and purchase separately. Machine and asset replacement problems fall into two main categories: serial and parallel.

Serial Replacement

The serial replacement problem is concerned with finding an optimal replacement policy for a single asset, or multiple economically independent assets. In the case of multiple assets, it is just as beneficial to solve a model for each asset independently as it would be to solve for all assets at the same time. It is noted by Hartman that the multiple asset problem is decomposed into multiple single-asset problems and normally a decision is made in the first time period, in which the model is updated in each time period with new data. It can be seen in the literature that the serial replacement problem is often outperformed by a more realistic (in general) problem, parallel replacement Joseph C Hartman (2000).

Parallel Replacement

The Parallel Machine Replacement Problem (PMRP) is the problem of finding a minimum cost replacement schedule for a group of machines that are economically interdependent and operate in parallel to satisfy a demand. That is, it may be the case that the purchase price is subject to a constraint such as economies of scale or demand dependence. PMRP allows for the ability to make a keep or replace decision, traditionally. More realistic formulations of the PMRP include decisions on sending an asset into holding or satisfying demand with renting, as presented here. This problem formulation is combinatorial in nature, and grows quickly in complexity as the number of potential decisions increases. Parallel replacement analysis can be over a finite or infinite horizon, and can have stochastic and deterministic formulation.

Parallel replacement analysis fits well into the construction industry, and allows for some new problem formulations. Specifically, most construction projects have machines that satisfy demand in parallel. It could be the case that two excavators work on the same site, or two dump trucks haul rock away together. The planning horizon uncertainty of a construction project due to delays and run-overs can be modeled with PMRP. Furthermore, construction machinery is economically interdependent. Any decision for one machine has an effect on other assets. For instance, if demand for a piece of equipment is stationary and one asset is salvaged, either a new asset must be purchased, or the remaining machines must be utilized more. Lastly, in construction equipment management, the shipping of machines to and from projects is prevalent, and gives rise to a new PMRP formulation. It should be noted that in this work it is not desired to perform predictive analysis on equipment costs to determine the optimal replacement time, yet a schedule is being prescribed based on a number of available options that are combinatorial in nature in order to determine an optimal schedule.

Infinite and Finite Horizon Problems

There exist two main formulations of the Parallel Machine Replacement Problem: infinite and finite planning horizon. Finite planning horizons have a predetermined length, and are often considered stochastic. Infinite planning horizons are seen in the classical equipment replacement problems with stationary costs, in which it is optimal to replace a machine at its economic life. But, as noted in (J. C. Hartman & C. H. Tan, 2014), when the planning horizon is finite, this is not always the case. Therefore, finite horizon problems are studied more in the recent years. Furthermore, non-stationary costs give rise to the vast amount of work in finite horizon problems, as the complexity is much greater with infinite horizon assumptions, and requires more involved modelling techniques.

Problem Definition: Adapting PMRP for Construction Industry

There exist difficulties in the field of asset replacement in that many of the research methods that are currently applied account for a subset of the aspects that occur in practice. Specifically, in the construction industry, optimization methods and advanced analysis techniques are not as predominant. There are also limitations on the solution methods, such as stochastic Dynamic Programming (DP), commonly used in the literature for large problem size. As stated by Hartman in (J. C. Hartman & C. H. Tan, 2014) there is a need for the consideration of aspects such as multiple machine types and planning horizon uncertainty, which are not principal in the literature. Despite the plethora of literature on PMRPs, the current models leave out important aspects of the real world and few have integrated PMRP into the construction industry. Many authors have worked on analysis of asset replacement under realistic assumptions and have gained important managerial implications for assets. Yet, some aspects still need to be considered such as: optimal utilization level of the assets (extension of (Joseph C. Hartman, 2004)), multiple machine and operation types, and the impact of shipping when demand is at varying locations. The statement on optimal utilization level should not be confused with that of Hartman. In that work, Hartman showed the best way to utilize each machine based on the assumptions of the costs and salvage values. In this he showed that in some cases one should utilize each asset equally, and in others a set of assets should be utilized fully and others partially. Yet, this work aims to answer if each asset is to be utilized equally, what the level should be in the cases of multiple heterogeneous challengers, used purchases, and salvaging at any age. This is also an extension of (Jha, 2000a) where a different problem formulation is proposed for determining the utilization levels.

The specific problem in question is that of optimally replacing a set of physical assets that are economically interdependent, over a finite and potentially uncertain planning horizon. The assets in this work have a number of decisions at each time period, further complicating the combinatorial nature of the formulation. In each time period a current machine can be replaced or sent into holding, while new machines can be bought new, used, or rented to satisfy demand. These specific decisions are already an extension of the current machine replacement literature, yet this work also changes the general structure of the problem to include machines of various operation type, multiple demand sites, and uncertain planning horizon.

Likewise, this work intends to discover the impact of shipping, stochasticity, and operation type on the management of construction machinery. These aspects are not normally considered in PMRP, yet are prevalent in construction management and other industries. Finally, the presented work aims to provide general and specific solutions methods that are more applicable to industry and provide the capability to solve large real-world problem formulations, specifically for the case of transportation. Also, many small and large construction companies use suboptimal asset replacement schedules, utilizing machines without strategy, subject to daily needs. An analysis tailored to construction management is essential to optimizing the procurement and replacement in this setting. Without more insight into best policy practices for construction machine replacement, construction companies will suffer financially from higher procurement, maintenance, and operations costs. Furthermore, the lack of data collection in the average construction company facilitates a common culture of contentment with these costs, costs that may be unnecessary. This work also aims to provide insight into the importance of applying mathematical optimization to construction management and assists in creating a standard for equipment replacement.

In this paper, the PMRP is analyzed in construction projects using deterministic and stochastic programming. Mixed integer models are presented that are customized for construction projects and consider a non-uniform demand over an uncertain project duration, multiple operation and machine types, and machine shipping; this is demonstrated in three independent models. Methods or processes for minimizing solution time and implementation ease are presented.

Delays in Construction Projects

It is common for construction projects to be delayed due to various causes. In order to control the delays, contractors should avoid cost overruns and disputes by matching planning and scheduling processes with resources and time Assaf and Al-Hejji (2006, p. 356). Construction machinery is one of the main resources for contractors, therefore optimal planning for their purchase and salvage is important.

Causes of delay in construction industry can vary from one country, region, type of projects, or even period of time to another. Al-Momani (2000) concluded that the main causes in Jordan are related to designers, user changes, weather, site conditions, late deliveries, economic conditions and increase in quantity. Frimpong, Oluwoye, and Crawford (2003) conclude that the main causes of delay and cost overruns in construction of groundwater projects in Ghana include: monthly payment difficulties from agencies, poor contractor management, material procurement, poor technical performances, and escalation of material prices. "Change order" is considered as the main cause of delay in large construction projects in Saudi Arabia (Assaf & Al-Hejji, 2006). It seems that delay is inevitable regardless of what causes it. Assaf and Al-Hejji (2006) concluded that 70% of the construction projects experience time overrun with average of time overrun being between 10% and 30% of the original duration. Faridi and El-Sayegh (2006) reveal in their research that, in the UAE, 50% of the construction projects get delayed and are not completed on time. The prevalence of an uncertain project horizon begs for better solutions to demand satisfaction, to avoid unnecessary costs when delays occur.

Multiple Machine and Operation Types

Multiple machine and operation types are prevalent in many industries, yet the work related to a heterogeneous fleet of assets has been limited. In (Keles & Hartman, 2004) Hartman considered a heterogeneous fleet of busses to determine the optimal replacement schedule, yet the consideration had no effect on the structure of the existing modelling. There is a need to model the realistic nature of the machines as heterogeneous in which the assets can perform multiple operations. Likewise, the machines should be able to take on intersecting roles of operations. For example, an excavator may be able to perform the same operation as a backhoe, yet each machine also has unique operations they can execute. This, of course, may require excess fixed or operating costs and this is necessary for solving the problem. (Tosun, 2014) studied the best excavator type, considering the type of job that needed to be performed and the excavator aspects. This work presents a solution to this problem with a new way of modeling the issue to account for the machine skill set overlap.

Construction Equipment Shipping

A significant cost in the management of large assets is that of shipping. (Shane, Molenaar, Anderson, & Schexnayder, 2009) state that poor estimation and project schedule changes lead to escalation of project costs. It can be seen quite frequently that large construction equipment is constantly transported from project to project as either new demand is realized or an asset needs to return to a holding station. Without an accurate shipping and replacement schedule, it may be the case that excess costs will be incurred. In the PMRP there has not been shipping considerations thus far, leading to higher project costs for the companies, as well as the stake holders. The reason that these specific considerations have not appeared in this problem is that few have considered multiple demand sites, as seen in the construction industry. The ones who have considered location differences have simply allocated all demand and solved for one larger system, or have equivalently solved for each demand site independently. This means that without shipping considerations, the solutions are currently suboptimal. To solve this, a new modeling structure is proposed that allows for the assets to be used at various demand sites.

Contributions and Motivation

The contributions of this research are that of solution methods, problem formulation, and application in construction management. The solutions methods proposed have not been considered in the literature and allow for more complex and larger problems to be solved for multiple demand sites. The problem formulations depict a more realistic PMRP formulation than those that appear in the literature and offer new managerial implications. Also, the tailored fit to construction management offers a practical approach for implementing the PMRP into any construction company, offering vast savings in the replacement of heavy equipment.

Limitations

A few limitations of the research stem from the industry and mathematics. In the construction industry specifically, data collection and advanced analysis techniques greatly depend on the size of the company. Most general contractors are not implementing mathematical optimization, and even the large construction companies seem to lack in certain data collection. One of the goals of this research is to create a solution method that is easier for the construction community to implement, and inform the community of the benefits of applying PMRP, as well as optimization. Yet, there has not been available data to test the problem from a company, and the data was compiled from various sources.

Furthermore, the modelling aspects of the problem have a limitation on how realistic one can make the models. Until a proof of $P = NP$, the nature of this combinatorial math problem is limited to only being able to consider a few of the aspects that happen in the real world at one time. For instance, it was not computationally economic to have both multiple machine and operation types in the same model as multiple projects (the number of variable indices would exceed that of 7). Unfortunately, this is a general limitation of integer programming and the community does the best it can, given the circumstances. Therefore, the balance between a mathematical model’s difficulty and its solution feasibility must be found.

Assumptions

The modelling and research assumptions vary based on the problem formulation. For the single asset case (the cases of stochastic planning horizon) the machines are assumed to be identical. Also, the machines are assumed to satisfy demand in parallel and are economically interdependent. These assumptions are that of the Parallel Machine Replacement Problem and without them, the Serial Replacement Problem would be solved. Assumptions specific to the case where there exists multiple machines and machine types is that if a set of machines can perform the same operation, they can do so with equal quality and the same utilization. Additionally, for all models, it is assumed that all demand must be satisfied, and partial satisfaction is not allowed. The machines are also assumed to have a limit on utilization and age for operation.

Summary

Equipment replacement has been studied over the past century (J. C. Hartman & C. H. Tan, 2014). As technology and mathematical abilities change, so has the research on asset replacement. Specifically, the Parallel Machine Replacement Problem has been adapted for various situations that have been realized. Here this work attempts to further that of previous works by creating and solving new instances of the PMRP. The work offers a realistic application tailored to the management of construction assets. It is known that the stochastic nature of the planning horizon is important to consider, both in replacement analysis and the construction industry. Given that the majority of construction projects extend beyond their initial planning timeframe, and that the replacement analysis experts have declared that the future direction of PMRP is involving stochastic planning horizon, this research solves a major real-world issue. The benefit and solution methodology of implementing this work is presented in the following sections. This work solves a number of issues:

- Integration of PMRP in construction management
- PMRP formulation limitations
    - Multiple machine and operation type
    - Limited stochastic planning horizon studies
    - A lack of shipping considerations

- Solution methodology for transportation decisions

CHAPTER TWO

LITERATURE REVIEW

History of PMRM

In replacement analysis, all of the assets within a group should be considered together for several reasons including economies of scale in the purchase price, demand constraints, and budgeting constraints (Keles & Hartman, 2004). PMRPs are concerned with simultaneous replacement of machinery in a group as opposed to individually. PRMPs have appeared in the literature since 1955 according to (J. C. Hartman & C. H. Tan, 2014). The term “parallel” replacement was coined by (Vander Veen, 1985; VanderVeen, 1985). In the PMRP models, the number of machines required in each time period (demand) is usually known and the length of the planning horizon is constant (see, for example, Jones, Zydiak, and Hopp (1991)). These models can vary based on the applications.

Multiple Asset Replacement

Multiple asset replacement is concerned with replacing multiple assets utilized in an integrated system, therefore the replacement decisions of all assets in that system must be considered simultaneously (Leung & Tanchoco, 1990). The earliest work in the replacement of a set of machines was (Leung & Tanchoco, 1986), where the authors provided the initial framework for such a problem. Early works in multiple asset replacement such as,(Jones et al., 1991), simplified the solution space using two rules: 1. The no splitting rule (NSR) and 2. The older cluster replacement rule. The NSR states that all same aged assets are kept or replaced at the same time. While the older cluster rule states that a machine of older age is always replaced before a newer one. As noted in (Hartman, 2014), this requires assumptions of non-decreasing O&M costs, and non-increasing salvage values, and their sum to be non-decreasing. STINSON and KHUMAWALA (1987) studied the multiple machine replacement problem considering lost production costs and machine downtime costs. The authors also present a heuristic algorithm for solving large problem size, and provides multiple ranked alternative solutions. As problems became more realistic, more advanced models and solution procedures arose. (Lotfi & Suresh, 1994) considered a nonlinear MILP formulation of the replacement of CNC machines, solved with a two-level exact solution method via Dynamic Programming. (Chand, McClurg, & Ward, 2000) extended the standard PRMR to include capacity expansion. That is, demand is assumed to increase over time, Chand also considered economies of scale. (McClurg & Chand, 2002) developed a forward-time algorithm that determines the optimal replacement plan for the generalized PMRP. A proof of the NSR and the OCRR was presented by (Tang & Tang, 1993), and extensions of these proofs are found in (Hopp, Jones, & Zydiak, 1993). Buy, lease, and rebuild decisions are studied by (Joseph C Hartman & Lohmann, 1997), where the authors use Integer Programming (IP) to solve computational experiments. In (S Rajagopalan, 1998; Sampath Rajagopalan, Singh, & Morton, 1998) capacity expansion is introduced into the equipment replacement literature. Here future demand changes and economies of scale are considered in a general model that solves efficiently.

More recent works include that of ((Keles & Hartman, 2004), (Büyüktahtakin & Hartman, 2009), (Joseph C. Hartman, 2004),(Esra Büyüktahtakin et al., 2014), (des-Bordes & Büyüktahtakın, 2017)). Keles and Hartman considered a mixed integer programming formulation for the replacement of a bus fleet. They determined implications of the model via sensitivity analysis and solved the problem with a commercial solver. Büyüktahtakin and Hartman considered the PMRP under technology change and deterioration and gave an integer programming formulation, and provide analysis on the effects the considerations had on the solution. A famous work by Hartman was that of replacement under stochastic horizon in 2004. Here, Hartman provided optimal utilization of machines given various assumptions of the input data. (Parthanadee, Buddhakulsomsiri, & Charnsethikul, 2012) studied the multiple machine replacement problem with assumptions that newer machines would be utilized more and studied: purchase-new-vehicles-only, no-splitting-in-selling, one-purchase-choice, older-vehicles-selling, and all-or-none rules. An aircraft replacement strategy was studied by (Bazargan & Hartman, 2012) in which the authors looked at optimal buying and leasing decisions for two US airline companies. Des-Bordes more recently considered the PMRP under economies of scale as well as a case study on MRI machines in 2017 with Büyüktahtakin. In this work the authors allowed for machines to satisfy demand that was service dependent by allowing for two sets of machines: a set whose demand can only be satisfied by their own asset style, and a set that can be satisfied by its own style as well as the first set. In 2014 Büyüktahtakin studied the parallel replacement problem under economies of scale (PRES). Here the author proved the NP-hardness of the problem formulation by transforming a 3SAT problem into the proposed formulation. More recently (Büyüktahtakın & Hartman, 2016) studied the PMRP under technology changes, considering capital gains where newer machines have a larger capacity.

Stochastic PMRP

While applying the existing PMRP models can be advantageous to the contractors in construction projects, it is very important to consider special characteristics of these projects so that the solutions become robust and trustworthy. The number of time periods in the planning horizon of a construction project is finite, yet uncertain. Also, even if the length of the planning horizon is certain, demand is not necessarily distributed equally; for example, the number of excavators required for each period in the beginning and closure phases of a project could be lower than the demand in mid phases. Such non-uniform demand has been studied mostly as an increasing demand called capacity expansion. Chand et al. (2000) consider PMR and capacity expansion jointly, which results in a non-decreasing demand in PMRP models. Non-uniformity of demand has appeared in the literature under various terminologies such as fluctuating demand, non-stationary or stochastic demands (see, for example, Jha (2000b); Joseph C. Hartman (2004); VanderVeen (1985)). One of the first stochastic works in the equipment replacement literature is (Childress & Durango-Cohen, 2005), where the authors prove the Worse Cluster Replacement Rule (WCRR) and the NSR for the stochastic formulation. More recently studied the replacement of aircraft under stochastic demand, using DP as a solution method. The most recent work to appear in the literature studies a stochastic planning horizon for a fleet replacement problem of alternative fuel type machines, see (Ansaripoor & Oliveira, 2018).

Tan and Hartman (2010) consider horizon uncertainty in replacement of a single asset where the length of horizon or service, T, is a finite variable, which is unknown, ranges between an upper bound and a lower bound, and has a probability distribution. They use dynamic programming as the solution methodology which is commonly used in the literature of replacement analysis. When multiple assets and their economic interdependence is considered, the replacement problem becomes a difficult, combinatorial problem (Keles & Hartman, 2004). Modeling of horizon uncertainty becomes challenging when multiple parallel machines are involved in replacement analysis and the number of machines required each time period fluctuates.

Utilization

According to Dhillon (2009, p. 2), "*life cycle cost* (LCC) is the sum of all costs incurred during the life span of an item or system." Javad Seif and Rabbani (2014) try to estimate LCC of construction machinery precisely by calculating the planned and unplanned maintenance activities based on the failure rates and failure probability distributions at a component level. They use this method in PMR and classify all components into three main categories based on their distributions. However, this method might not always be practical due to the high number of components and the complexity of realistic systems and machinery. Others have also tried to incorporate the cost of unplanned maintenance of critical or major equipment in LCC by considering failure distributions of such equipment (see, for example, Frangopol, Lin, and Estes (1997) and Sherwin (2000)). Joseph C. Hartman (2004) uses two indices that account for age and utilization level, making data collection easier and practical. In this research, two indices are also used for age and utilization level. This let us model PRM such that machinery can be purchased and salvaged at any age and any utilization level. Heavy construction machinery is an expensive asset that is traded at varying ages and utilization levels. Utilization is expressed as the cumulative number of hours a machine has operated. Salvage and purchase at different ages and utilization levels expands the solution space, which can lead to a lower total cost of procurement. In addition, because practitioners may explore other options such as renting the machinery, renting is incorporated into our model.

Literature Gap

The literature gap analysis is provided in Table 1. To fit the attributes of each model in the table, a symbol is used for each such that: PT is the problem type, MO is multiple operations (multi-purpose machines), MM is multiple machine types, HU is horizon uncertainty, U is utilization considerations, MDS is multiple demand sites, and T is transportation considerations. As seen in Table 1, multiple authors consider utilization considerations. This is a trend in the literature that addresses a realistic problem. Likewise, multiple machines and horizon uncertainty are current directions of the field. This is explicitly stated in ((J. C. Hartman & C. H. Tan, 2014). Hartman was one of the first whom considered utilization extensively, and this research aims to add to that aspect of the field. In terms of multiple operation types or multi-purpose machines, only two papers address this to date, (J. Hartman & Ban, 2002)and (des-Bordes & Büyüktahtakın, 2017). In the first paper, Hartman and Ban consider the Serial-Parallel replacement problem in which they have a number of machines that can perform various tasks in a production line. Although the context is slightly different, the idea of multiple machine and operation types stems here. Next, de-Bordes and Büyüktahtakin provide a realistic extension of multi-purpose machines in which the authors allow for demand to be satisfied in two categories, or sets. That is, one set of assets can have demand only met by themselves, while a second set of assets can have demand met by both sets of assets. The work presented here further weakens the assumptions and allows for any combination of machines and operations that a machine can perform. The biggest literature gap is that of multiple demand sites and transportation decisions. Only one paper to date has considered multiple demand sites mathematically, Hartman and Ban 2002. In this work, as mentioned previously, the machines are serial-parallel and have various operations. Yet, the way in which Hartman and Ban formulated their mathematical model, the structure is equivalent to multiple demand sites, without a distance matrix. This was modelled in such a way to account for the locations of the machines in various sites within a warehouse. No research to date has considered shipping integrated with PMRP.

CHAPTER THREE
METHODOLOGY

In this chapter a general process is discussed to determine which problem formulation to consider, as well as a specific algorithm for solving the instance where shipping decisions are made. Data gathering and generation is also discussed for a realistic depiction of heavy construction equipment. Parameter functions are derived from the literature and real data is fit to each to properly generate purchase, operations, and maintenance costs. Other inputs are determined easily from online sources. For each problem formulation, a mixed integer programming model (MILP) is presented. Each formulation is based on an extension of the standard PMRP by (J. Seif, Shields,B. and Yu,A., 2018). Two extensions of the base model are for multi-purpose machines and transportation considerations. The model's differences are discussed, and both stochastic and deterministic instances are considered. Lastly, a process for determining the optimal utilization level of all machines is given as an alternative to that of including utilization as a decision variable and further complicating the problems. It is also discussed how each formulation is analyzed by their Objective Function Values (OFV), the Value of Stochastic Solution (VSS), the Expected Value of Perfect Information (EVPI) and sensitivity analysis.

In the proposed field there are eight problem formulations that can occur, seen in Figure 1. Because of the limitations stated, only a subset of these are being studied. Problems P1, P4, P5, and P6 (denoted with a star) are those being considered. Most of the work is in the modelling and analysis of the problem formulations, where managerial implications and easier solutions are desired. Each of the four problems considered use data generated in the same way with functions derived from (Joseph C. Hartman, 2004) and fitted using online data. All models considered are deterministic or stochastic mixed integer programs, solved with Gurobi implemented in Python. A solution processe is presented to help minimize the solution time and to promote ease of implementation. Lastly, thorough sensitivity analysis is presented for each problem, giving insight and practical use for companies to use by evaluating modelling quality metrics.

Modelling

The life cycle cost (LCC) of a machine is calculated from the beginning of acquisition to its salvaging. Initial costs are that of purchasing machines of specific ages and operating hours (cumulative utilization level). Mid-life costs would be that of operating the machines, possibly holding the machine as inventory, and maintaining the machines. Renting a number of machines has also been considered as an alternative. When renting, the LCC would only be that of renting a machine, which is considered per year, and operating the rented machines. End-life costs for purchased machines would include the negative cost of salvaging a machine, in form of selling price, again of a specific age and cumulative utilization level.

A machine is salvaged when it reaches certain thresholds. Two thresholds are considered in the lifetime of a machine. The first threshold is the maximum number of years past the manufacturing date of the machine (L), i.e. a certain age. The second threshold is the maximum cumulative utilization level (U), e.g. a certain value for the total number of hours or miles a machine has operated. When a machine is owned, each time period (year) adds one unit to its age; however, the machine has to be operating in that period or the cumulative utilization level will not change. L is determined such that each year of operation adds one unit to the machine's cumulative operating hours. Consider a bulldozer that is operated 8 hours/day and 300 days per year. If the useful life of the bulldozer is 20000 hours, $\mathrm{L} = \lfloor 20000/(300 \times 8) \rfloor = 8.$ If a machine reaches either of the thresholds, it should not be considered for operation and should be salvaged.

Stochastic Programming

Stochastic Programming is a field of mathematical optimization concerned with finding the optimal solution to (mostly) linear programs, under which one or more of the parameters are uncertain. The formulation of a stochastic program is either two-stage or multi-stage. A first stage is composed of a decision variable in which a decision is made in the current time, and the remaining stages are determined in consequence to the first stage decision. A number of scenarios are considered, and the expected value over all optimal solutions (for each scenario) is the solution to a stochastic program, denoted the *Recourse Problem (RP)*. As an example, as seen in (Birge & Louveaux, 2011), a farmer is attempting to determine the optimal allocation of crops to a finite plot of land in order to maximize her profit. Each of her crops have varying yields and revenues dependent on the weather. Assume the crops can have a high, medium, and low yield, as three weather scenarios are realized. A first stage decision is to determine how much acreage of each crop to plant, and the second stage is determining the yields and sales. The optimal solution to the recourse problem finds the maximum expected profit, considering all scenarios.

Evaluation Metrics

To evaluate the solution quality and perform insightful analysis, a number of metrics are considered for the deterministic and stochastic problem formulations. For both cases, the objective function values (OFVs) are used as the primary metric. Also, in conjunction with Gurobi and Python, the solution times and time to retrieve a feasible solution are used to compare with the presented solution methodologies. Specific to stochastic models, the *Value of Stochastic Solution* (VSS), *Recourse Problem* (RP), Wait-and-See solution (WS), expected results from the expected value problem (EV and EEV), and *Expected Value of Perfect Information* (EVPI) are utilized in determining the performance of the problem formulations.

The WS solution is the expected value of the optimal solutions for each scenario. The EVPI is the RP minus the WS, commonly defined as the amount the stakeholder should be willing to pay for accurate future information. The EV solution is a low-hanging fruit option, in which the expected value of each uncertain parameter is used as input and the model is solved in a deterministic manner. The EEV is the expected result of using the EV solution. Lastly, the VSS is the expected value of using the EV solution minus the

RP. VSS, along with EVPI, are important measures of the necessity of considering the uncertainties in the model.

In this section, two stochastic PMRPs are presented (P5 and P6;Figure 1). P6 is a single project stochastic asset replacement and P5 is a stochastic asset replacement with multiple projects. Both are presented with their particular MILP models.

Monte Carlo Simulation for Stochastic Demand

In order to generate the scenarios for demand in the stochastic formulations, a Monte Carlo simulation was implemented in Excel and imported into python and then used to solve the optimization model with Gurobi. This sort of scenario generation was necessary because of the high number of instances and the limit on ability to code all scenarios into the stochastic program. First, the demand was modelled and the probabilities of each scenario was determined. A random number was generated and then an if statement was written that determined if the value was greater than the random number. If the number was greater than the probability of obtaining a project, the scenario existed, otherwise the scenario received zero demand. NORM.INV() was used to generate the project lengths and number of machines needed in each time period based on the initial demand. The simulation was then generated for a number of scenarios. Each scenario gave a demand site and project length, along with the required number of machines. This was used as the input into the MILP model. The code to connect the model to the simulation can be seen in Appendix A.

Single Asset - Single Project

The procurement of a particular construction machine over an uncertain planning horizon is modeled as a two-stage stochastic mixed integer program. A number of decision variables are introduced to model the schedule for purchasing, renting, holding in inventory, and salvaging the machines. In the first stage, a number of machines are purchased, and subsequently for the remaining time periods (second stage), machines are purchased, rented, set to idle (holding in inventory), or salvaged. The goal is to find an optimal schedule that chooses the values of first-stage variables such that the total cost (first-stage cost plus the expected cost of the second stage) is minimized.

In a real-world construction project, the number of periods in the project's planning horizon is stochastic. Additionally, the age and current cumulative utilization level of a machine impacts all various types of costs associated with the machine. The parameters and mixed integer program are presented below. This problem formulation in deterministic form is the basis of the transportation and multi-purpose machine extensions.

Parameters

c_{ijt}	Cost of purchasing a machine with age and utilization level (i,j) in period t
q_t	Cost of renting a machine for one time period in period t
h_t	Cost of holding a machine in period t
o_t	Operating costs of a machine in period t
m_{ijt}	Cost of maintaining a machine with age and utilization level (i,j) in period t
s_{ijt}	Salvage value of a machine with age and utilization level (i,j) in period t
ζ_t^{ω}	Number of required machines in time period $t \in [1, T^{\omega}]$ under scenario $\omega \in \Omega$
p_{ω}	Probability of scenario $\omega \in \Omega$ being realized
L	Maximum physical life (due to technological change)
U	Maximum cumulative utilization level (due to deterioration)
T^{ω}	Length of the planning horizon under scenario $\omega \in \Omega$

Decision Variables

X_{ij}	Number of machines with age and cumulative utilization level (i,j) to purchase in the first stage
y_{ijt}^{ω}	Number of machines with age and cumulative utilization level (i,j) to purchase in period t under scenario ω
z_t^{ω}	Number of machines to rent in time period t under scenario ω

S_{ijt}^{ω} Number of machines with age and cumulative utilization level (i, j) to sell (salvage) in period t under scenario ω

I_{ijt}^{ω} Number of machines with age and cumulative utilization level (i, j) that are idle in period t under scenario ω

a_{ijt}^{ω} Number of operating machines with age and cumulative utilization level (i, j) in period t under scenario ω

The two-stage stochastic mixed integer programming model is formulated as follows.

Minimize $TC = \sum_{i=1}^{L} \sum_{j=1}^{U} c_{ij1} X_{ij} + E[Q(\xi, Y)]$,

$$E[Q(\xi, Y)] = \sum_{\omega \in \Omega} p_{\omega} \left(\sum_{t=2}^{T^{\omega}} \left[q z_t^{\omega} + o_t a_t^{\omega} + \sum_{i=1}^{L} \sum_{j=1}^{U} \begin{pmatrix} c_{ijt} y_{ijt}^{\omega} + h_t I_{ijt}^{\omega} \\ + m_{ijt} a_{ijt}^{\omega} - s_{ijt} S_{ijt}^{\omega} \end{pmatrix} \right] \right) \quad (4)$$

The objective is to minimize the total expected cost of procuring a specific type of machine and meeting the demand for it over the stochastic duration of a construction project. This is given by the cost of purchasing initial machines (first stage) plus the expected cost of the operating, excess purchasing, holding, maintaining, renting, and salvaging costs in the following time periods (second stage). The objective function is two-stage. In the first stage a decision is made regarding the number of machines to purchase at the very beginning of the project. If the model is being used in the middle of the project, the first stage would be the current time period. The second-stage decision depends on the first stage and determines the costs of meeting the demand for machinery throughout the horizon. Because there can be multiple scenarios for the second stage, the expected value is minimized. The objective is subject to the following constraint sets:

$$z_t^{\omega} + \sum_{i=1}^{L} \sum_{j=1}^{U} a_{ijt}^{\omega} = \zeta_t^{\omega}, \qquad \forall t \in [2, T^{\omega}] \quad (5)$$

$$a_{iUt}^{\omega} = 0, \qquad \forall \omega \in \Omega, t \in [2, T^{\omega}], i \in [1, L] \quad (6)$$

$$a_{Ljt}^{\omega} = 0, \qquad \forall \omega \in \Omega, t \in [2, T^{\omega}], j \in [1, U] \quad (7)$$

$$a_{ij1}^{\omega} + I_{ij1}^{\omega} + S_{ij1}^{\omega} = X_{ij}, \qquad \forall \omega \in \Omega, i \in [1, L], j \in [1, U] \quad (8)$$

$$a_{ijt}^{\omega} + I_{ijt}^{\omega} + S_{ijt}^{\omega} = y_{ijt}^{\omega} + a_{(i-1)(j-1)(t-1)}^{\omega} + I_{(i-1)j(t-1)}^{\omega}, \quad \forall t \in [2, T^{\omega}], \omega \in \Omega, i \in [2, L], j \in [2, U] \tag{9}$$

$$a_{1jt}^{\omega} + I_{1jt}^{\omega} + S_{1jt}^{\omega} = y_{1jt}^{\omega}, \quad \forall t \in [2, T^{\omega}], \omega \in \Omega, j \in [1, U] \tag{10}$$

$$a_{i1t}^{\omega} + I_{i1t}^{\omega} + S_{i1t}^{\omega} = y_{i1t}^{\omega} + I_{(i-1)1(t-1)}^{\omega}, \quad \forall t \in [2, T^{\omega}], \omega \in \Omega, i \in [1, L] \tag{11}$$

$$S_{Ljt}^{\omega} = y_{Ljt}^{\omega} + a_{(L-1)(j-1)(t-1)}^{\omega} + I_{(L-1)j(t-1)}^{\omega}, \quad \forall t \in [2, T^{\omega}], \omega \in \Omega, j \in [2, U] \tag{12}$$

$$S_{iUt}^{\omega} = y_{iUt}^{\omega} + a_{(i-1)(U-1)(t-1)}^{\omega}, \quad \forall t \in [2, T^{\omega}], \omega \in \Omega, i \in [2, L] \tag{13}$$

$$S_{Lj1}^{\omega} = y_{Lj1}^{\omega}, \quad \forall \omega \in \Omega, j \in [1, U] \tag{14}$$

$$S_{iU1}^{\omega} = y_{i(L-1)1}^{\omega}, \quad \forall \omega \in \Omega, i \in [1, L] \tag{15}$$

$$S_{i11}^{\omega} = 0, \quad \forall \omega \in \Omega, i \in [1, L] \tag{16}$$

$$S_{1j1}^{\omega} = 0, \quad \forall \omega \in \Omega, j \in [1, U] \tag{17}$$

$$X_{ij}, y_{ijt}^{\omega}, z_t^{\omega}, I_{ijt}^{\omega}, S_{ijt}^{\omega}, a_{ijt}^{\omega} \in \mathbb{Z}_+, \quad \forall t \in [1, T^{\omega}], \omega \in \Omega, i, j = 1, \dots, L \tag{18}$$

Constraint set (5) ensures that the number of machines rented plus the number of machines operating in the current period satisfy the required number of machines in each time-period, under each scenario. Constraint sets (6) and (7) ensure those machines which have reached their thresholds (maximum age, L, or maximum cumulative utilization level, U) will not be operating in any time period. Constraint set (8) divides the total number of machines purchased in the first stage into different categories (operating, idle, and salvage). Similarly, and for each period, Constraint set (9) separates the total number of machines that are owned (machines purchased in the current period plus operating and idle machines from the immediate past period) into operating, idle, and salvage categories in the current period. Obviously, a machine that is new in terms of age ($i = 1$) cannot have been owned since the last period. Similarly, a machine that is new in terms of cumulative utilization level ($j = 1$) cannot have been operating since the last period.

Because of this, Constraint set (9) does not cover $i = 1$ and $j = 1$. Constraint sets (11) and (12) are the same as Constraint set (9) modified for $i = 1$ and $j = 1$, respectively. Constraint sets (13) and (14) ensure that all of the machines which are owned and reach their thresholds in the current period are salvaged. Similar to Constraints set (9), Constraints set (13) and (14) do not cover $j = 1$ and $i = 1$, respectively. Constraints set (15) and (16) follow the same logic as (13) and (14), but only cover $j = 1$ and $i = 1$, respectively. The final constraints make sure only integer solutions can occur. A rational solution is infeasible because buying half of a machine is not feasible.

Single Asset - Multiple Projects

Here, the considerations of replacement are integrated with transportation to and from various demand sites, denoted as projects. The base model is generalized to include, for the first time, network flow constraints that allow for shipping decisions to be made, as well as replacement, holding, renting, and operating. The premise of this model stems from a place of pure application in which, for this particular field of study, shipping is often part of the demand satisfaction. Although decisions are made at the end of the time period, shipping occurs in between time periods, presented as beginning of time period decisions in terms of notation. It is often that construction companies need to make annual, monthly, and even weekly decisions on where to send an asset to satisfy demand. Because the shipping of some assets is difficult, it is imperative that an accurate movement of machines is in place. Consider a 50,000lb excavator; this machine can take multiple days to transport, gain permits, schedule escorts, and may require alternative routes. Therefore, having an optimal schedule can mitigate time of arrival and potentially reduce delays in the projects in the future.

The model is presented as an extension of the base model, here in stochastic form. This formulation is a Parallel Machine Replacement Problem with Shipping, termed PMRP-S. In this problem, the uncertainty of project number, as well as planning horizon of each project is considered. For ease of understanding, the same parameter names are used for any that appear in the base model.

Parameters

U	Maximum cumulative utilization of an asset
L	Maximum time-based life (lifetime) of an asset
Ω	Number of scenarios to be realized
P^{ω}	Number of demand sites in varying location under scenario $\omega \in \Omega$
T^{ω}	Length of the planning horizon under scenario $\omega \in \Omega$
c_{ijt}	Cost of purchasing an asset with age (i, j) in time period t
q_t	Cost of renting an asset for one time period t
h_t	Cost of holding an asset for one time t period sitting in a warehouse
o_t	Operating costs of an asset for one time period t
m_{ijt}	Cost of maintaining an asset with age (i, j) for one time period, t
s_{ijt}	Salvage/Selling profit for an asset with age (i, j) in time period t
ξ_{tp}^{ω}	Number of required assets in time period $t \in [1, T^{\omega}]$ for project p under scenario $\omega \in \Omega$
$tc_{(nm)}$	Cost of shipping an asset from n to m, where $(n, m) \in P$
pr_{ω}	Probability of scenario $\omega \in \Omega$ being realized

Decision Variables

X_{ijp}	Number of asset with age (i, j) to be bought in the first stage (before the first period in the planning horizon) for demand site p
y_{ijtp}^{ω}	Number of assets with age (i, j) to buy in time period t for demand site p under scenario ω
z_{tp}^{ω}	Number of assets to be rented in time period t for demand site p under scenario ω
S_{ijtp}^{ω}	Number of assets with age (i, j) to sell (salvage) in time period t for demand site p under scenario ω
I_{ijtp}^{ω}	Number of assets with age (i, j) that are idle in time period t for demand site p under scenario ω
a_{ijtp}^{ω}	Number of assets with age (i, j) in time period t for demand site p under scenario ω that are operating
$TC_{ijt}^{\omega(nm)}$	Number of assets with age (i, j) in time period t shipped from n to m under scenario ω

The two-stage stochastic integer model is formulated as follows:

$$Minimize\ \boldsymbol{\zeta} = \sum_{i=1}^{L} \sum_{j=1}^{U} \sum_{p=1}^{P} \boldsymbol{c_{ij} X_{ijp}} + \boldsymbol{E}[\boldsymbol{Q}(\boldsymbol{\xi}, \boldsymbol{Y})] \tag{19}$$

$$Q(\xi, Y) = \sum_{t=1}^{T^\omega} \sum_{p=1}^{P} \left[q z_{tp}^\omega + \sum_{i=1}^{L} \sum_{j=1}^{U} \left(c_{ijt} y_{ijtp}^\omega + h I_{ijtp}^\omega + (o + m_{ijt}) a_{ijtp}^\omega - s_{ijt} S_{ijtp}^\omega \right) \right] + \left[\sum_{t=1}^{T^\omega} \sum_{i=1}^{L} \sum_{j=1}^{U} \sum_{m=1}^{P} \sum_{n=1}^{P} tc_{(nm)} TC_{ijt}^{\omega(nm)} \right]$$

The objective is to minimize the total expected cost of managing over the stochastic duration of a project. This is given by the cost of purchasing initial assets (first stage) plus the expected value of the operating, excess purchasing, holding, maintaining, renting, shipping, and salvaging costs in the following time periods (second stage). The addition of shipping requires the added cost for sending an asset from one demand site to another. The objective function is two-stage. In the first stage, a decision is made on the number of assets to purchase at the very beginning of each project. The second stage decision is dependent on the first stage and determines the costs of meeting the demand for assets throughout the horizon. Because there can be multiple scenarios for the second stage, the expected value is minimized. As with the base model, the values are discounted to time zero. The objective is subject to the following constraint sets:

$$\sum_{k=1}^{M} \left(z_{tp}^\omega + \sum_{i=1}^{L} \sum_{j=1}^{U} a_{ijtp}^\omega \right) \geq \xi_{tp}^\omega, \qquad \forall p \in P, \forall t \in [2, T^\omega], \omega \in \Omega \tag{20}$$

$$a_{ijtp}^\omega + I_{ijtp}^\omega + S_{ijtp}^\omega = y_{ijtp}^\omega + \sum_{n \neq p}^{P-1} TC_{ijt}^{\omega(np)} - \sum_{m \neq p}^{P-1} TC_{ijt}^{\omega(pm)} + a_{(i-1)(j-1)(t-1)p}^\omega + I_{(i-1)j(t-1)p}^\omega, \qquad \forall t \in [2, T^\omega], \omega \in \Omega, i \in [2, L], j \in [2, U], \forall p \in P \tag{21}$$

$$a_{i(U-1)tp}^\omega = 0 \qquad \forall t \in [2, T^\omega], \omega \in \Omega, i \in [2, U], \forall p \in P \tag{22}$$

$$a_{(L-1)jtp}^\omega = 0 \qquad \forall t \in [2, T^\omega], \omega \in \Omega, j \in [2, U], \forall p \in P \tag{23}$$

$$a_{ij1p}^{\omega} + I_{ij1p}^{\omega} + S_{ij1p}^{\omega} = y_{ij1p}^{\omega} \qquad \omega \in \Omega, i \in [2, L], j \in [2, U], \forall p \in P \tag{24}$$

$$y_{ij1p}^{\omega} = x_{ijp} \qquad \omega \in \Omega, i \in [2, L], j \in [2, U], \forall p \in P \tag{25}$$

$$S_{(L-1)jtp}^{\omega} = y_{(L-1)jtp}^{\omega} - a_{(L-2)(j-1)(t-1)p}^{\omega} + I_{(L-2)j(t-1)p}^{\omega}, \qquad \forall t \in [2, T^{\omega}], \omega \in \Omega, j \in [2, U], \forall p \in P \tag{26}$$

$$S_{i(L-1)tp}^{\omega} = y_{i(U-1)tp}^{\omega} - a_{i(U-2)(t-1)p}^{\omega}, \qquad \forall t \in [2, T^{\omega}], \omega \in \Omega, i \in [2, L], \forall p \in P \tag{27}$$

$$X_{ijp}, y_{ijtp}^{\omega}, z_{tp}^{\omega}, I_{ijtp}^{\omega}, a_{ijtp}^{\omega} \in \mathbb{Z}^{+}, \qquad \forall t \in [1, T^{\omega}], \omega \in \Omega, i, j = 1, \ldots, U, \forall p \in P \tag{28}$$

Constraint (20) is the demand constraint, ensuring at least the number of required machines are placed in each project for each time period. Constraint (21) is an extension of Constraint (9) from the base model that requires the flow of machines entering and leaving a project to be balanced. This network flow constraint ensures that no shortage or excessive number of machines are in the system. The remaining constraints are derived from the base model, where all gain an index of dimension P. It can be seen in Constraint (21) that shipping decisions are made at the beginning of the time period, while other decisions are made at the end. This is to attempt to model the decision in between the time period, which is a more accurate depiction of reality. Modelling in this particular way causes no shipping between the first and second time periods, yet all remaining time periods work fluently. To account for this, it is possible to add a dummy initial starting year with demand zero, and an initial set of machines at each location, if so desired.

Multiple Operation and Asset Type

Here the base model is extended to include machines that can perform multiple operations. A number of parameters and constraints are added to model this particular scenario. Operations, maintenance, and an excess costs are modelled as operation dependent for each machine. The required number of machines in each time period is extended to include the specific operation to be performed. A binary parameter is added that determines if a machine can perform a particular operation or not. It is assumed that if two machines can perform the same operation, they can do so under the same utilization and quality, yet not necessarily under the same costs. The salvage value is left independent of operation, yet varies by machine. Similarly, holding and renting costs are dependent only on machine type. The maximum age and cumulative utilization level is extended to depend on the type of machine under consideration.

Parameters

M	Types of assets considered
O	Types of operations to be considered
U_k	Maximum cumulative utilization of asset k
L_k	Maximum time-based life (lifetime) of asset k
T	Length of the planning horizon
c_{kijt}	Cost of purchasing asset k with age (i, j) in time period t
uc_{ko}	Excess cost of using asset k for operation o
q_{tk}	Cost of renting asset k in time period t
h_{tk}	Cost of holding asset k in time period t
o_{kto}	Operating costs of asset k for performing operation o in time period t
m_{kijto}	Cost of maintaining asset k with age (i, j) in time period t for operation o
s_{kijt}	Salvage/Selling profit for asset k with age (i, j) in time period t
r_{to}	Number of required assets for operation o in time period t
op_{ko}	1 if asset k can perform operation o, 0 otherwise

Decision Variables

y_{kijt}	Number of assets k with age (i, j) to buy in time period t
z_{kto}	Number of assets k to be rented in time period t that are performing operation o
S_{kijt}	Number of assets k with age (i, j) to sell (salvage) in time period t
I_{kijt}	Number of assets k with age (i, j) that are idle in time period t
a_{kijto}	Number of assets k with age (i, j) in time period t that are performing operation o

The mixed-integer model is formulated as follows:

$$Minimize\ \boldsymbol{Z} = \tag{29}$$

$$\sum_{t=1}^{T}\sum_{k=1}^{M}\left[q_{tk}z_{kto} + \sum_{i=1}^{L}\sum_{j=1}^{U}\left(c_{kijt}y_{kijt} + h_{kt}I_{kijt} + \sum_{o=1}^{O}(uc_{ko} + o_{kto} + m_{kijto})a_{kijto} - s_{kijt}S_{kijt}\right)\right]$$

Here again, the objective is to minimize the total expected cost of procuring assets over the planning horizon, T. The objective is extended to include the costs of a machine for performing a particular operation, as well as any excess cost incurred. Renting is considered as an alternative, that is also operation dependent. The objective is subject to the following constraint sets:

$$\sum_{k=1}^{M} op_{ko}\left(z_{kto} + \sum_{i=1}^{L}\sum_{j=1}^{U} a_{kijto}\right) \geq r_{to}, \qquad \forall t \in [2, T\,], \forall o \in O \tag{30}$$

$$\sum_{o=1}^{O}\sum_{k=1}^{M}\left(z_{kto} + \sum_{i=1}^{L}\sum_{j=1}^{U} a_{kijto}\right) \geq \sum_{o=1}^{O} r_{to}, \qquad \forall t \in [2, T\,] \tag{31}$$

$$\sum_{o=1}^{O}\left(z_{kto} + \sum_{i=1}^{L}\sum_{j=1}^{U} a_{kijto}\right) \leq \sum_{o=1}^{O} op_{ko}r_{to}, \qquad \forall t \in [2, T\,], k \in M \tag{32}$$

$$\sum_{o=1}^{O} a_{kijto} + I_{kijt} + S_{kijt} = y_{kijt} + a^{\omega}_{k(i-1)(j-1)(t-1)o} + I^{\omega}_{k(i-1)j(t-1)}\,, \qquad \forall t \in [2, T\,], i \in [2, L], j \in [2, U], \forall k \in M, \forall o \in O \tag{33}$$

$$a_{ki(U-1)to} = 0 \qquad \forall t \in [2, T\,], i \in [2, U], \forall k \in M, \forall o \in O \tag{34}$$

$$a_{k(L-1)jto} = 0 \qquad \forall t \in [2, T\,], j \in [2, U], \forall k \in M, \forall o \in O \tag{35}$$

$$\sum_{o=1}^{O} a_{kij1o} + I_{kij1} + S_{kij1} = y_{kij1} \qquad i \in [2, L], j \in [2, U], \forall k \in M \tag{36}$$

$$y_{kij1} = x_{kij} \qquad i \in [2, L], j \in [2, U], \forall k \in M \tag{37}$$

$$S_{k(L-1)jt} = y_{k(L-1)jt} - \sum_{o=1}^{O} a_{k(L-2)(j-1)(t-1)o} + I_{k(L-2)j(t-1)}, \qquad \forall t \in [2, T\,], j \in [2, U], \forall k \in M \tag{38}$$

$$S_{ki(U-1)t} = y_{ki(U-1)t} - \sum_{o=1}^{O} a_{ki(U-2)(t-1)o}, \qquad \forall t \in [2, T\ ], i \in [2, L], \forall k \in M \qquad (39)$$

$$X_{kij}, y_{kijt}, z_{kto}, I_{kijt}, a_{kijto} \in \mathbb{Z}^{+}, \qquad \forall t \in T, (i, j) \in U, k \in M, \forall o \in O \qquad (40)$$

The constraints here gain an index of o and k. Three main demand constraints are added that ensure the proper number of machines is acquired. Constraint (30) ensures that no demand is satisfied by a machine that does not have the particular operation to satisfy said demand. Constrain (31) requires that the total number of machines for all operations is met, and Constraint (32) ensures that the machines can only satisfy up to the total amount of demand that the particular machine can take on. That is, each machine can only satisfy a subset of the demand, unless they can perform every operation, yet they are also not required to satisfy demand merely because they have a particular operation.

Solution Methods

In this section, the need for solution methodology for large problem size is considered, as well as the methodology for obtaining optimal utilization levels of the assets. A computational study is presented that shows the need for a solution method for multiple projects with shipping, while also demonstrating that the solver is computationally efficient for multiple machines. A clustering-based method is presented for the deterministic case for multiple project locations and can be used to solve the stochastic problem formulation as well. The clustering procedure decomposes the large problem size into multiple smaller instances that allow for a much more efficient computation time without giving up greatly on optimality. This procedure is compared to that of both solving the entire problem without shipping and the original formulation.

Complexity of PMRP

The parallel machine replacement problem is known to be NP-Hard (Esra Büyüktahtakin et al., 2014), and any of its generalizations are at least of the same complexity. The number of combinations of solutions grows quickly as the problem size increases. Yet, commercial solvers, such as Gurobi, can solve significantly large problems in a reasonable time. For instance, for the base model considered in this research, Gurobi can solve for monthly decisions ($U = L = 72$) for three years, $T = 36$ in under one minute, using a standard laptop. This would be considered a large problem size for the base model, as this is as realistic as would need to be considered for a large construction project. But, when the model is extended to multiple demand sites, the number of decision variables increases significantly. For small and medium size problems, the most common problem sizes a company would have, commercial solvers perform quite well. Yet, for sufficiently large problem size, even the advanced capabilities of the solvers succumb to long run times due to memory allocation problems.

Clustering Decomposition Algorithm

Because of the large number of possible routes available $\left(\frac{P(P-1)}{2}, where\ P\ is\ the\ number\ of\ projects\right)$ to satisfy demand while considering the optimal replacement schedule, a clustering of the projects is used to efficiently solve large problem size. The premise is to decompose the large problem size into smaller instances, with much fewer number of decisions. The process begins by determining a shipping frequency for the problem's distance matrix based on the expected demand in each time period, for each project. This frequency is determined by solving a number of small test problems by taking small steps forward in time. Because shipping is demand dependent, all other parameters in the model remain unchanged. To illustrate this, a shipping will occur when the demand of a project decreases while another project has a surplus of machines for a particular time period, and the shipping is more cost effective than that of a machine purchase. It should be noted that if demand is stationary, shipping may not occur.

After the shipping frequencies are determined, a clustering of the projects is created by iteratively adding projects to clusters based on the frequencies. This breaks the problem into multiple smaller complete graphs (clusters) with, at most, the original problem's number of edges. This process yields instances that can be solved with the same model, with shipping for any cluster that has frequency and without shipping for all remaining nodes in the network (eliminating many excess edges). This provides an efficient solution time, while allowing for a minimal tradeoff in optimality. To illustrate the process, a small general graphical representation is presented, as well as the pseudocode below.

Clustering Decomposition Algorithm (CDA)

Inputs:

T	Length of the planning horizon
s	Maximum cluster size
d	Length of the planning horizon in the sub-problems
$\boldsymbol{N}$	Set of all the nodes (projects)

Define $\boldsymbol{F} = \boldsymbol{0}_{|N|\times|N|}$ //The matrix of frequencies

Define $\boldsymbol{C} = \emptyset$ //Set of clusters

Define $\boldsymbol{\theta}$ a (0,1)-*matrix* of size $|N|$ by $|N|$ //$\theta_{i,j} = 1$ if shipping has happened between nodes i and j, and 0 otherwise.

Define $\boldsymbol{N}' = \boldsymbol{N}$

Define $t = 1$

Solve the master problem for the planning horizon $[t, t + d]$

  Get the value of $\theta_{i,j}, \forall i \& j \in N$

Set $F_{i,j} = F_{i,j} + \theta_{i,j}, \forall i \& j \in N$

Set $t = t + 1$

While $t + d \leq T$ **do**

  Solve the master problem for the planning horizon $[t, t + d]$

    Get the value of $\theta_{i,j}, \forall i \& j \in N$

  Set $F_{i,j} = F_{i,j} + \theta_{i,j}, \forall i \& j \in N$

  Set $t = t + 1$

End While

Find $(i, j) \in (\boldsymbol{N}, \boldsymbol{N}), i \neq j$ such that $F_{i,j} \geq F_{k,l}, \forall (k, l) \in (\boldsymbol{N}, \boldsymbol{N})$ //Find the largest element of $\boldsymbol{F}$

Add the set $\{i, j\}$ to $\boldsymbol{C}$

Set $F_{i,j} = 0$

Set $\boldsymbol{N}' = \boldsymbol{N}' \setminus \{i, j\}$

While $|\boldsymbol{N}'| \geq 1$ **do**

Find $(m, n) \in (\boldsymbol{N}, \boldsymbol{N}), m \neq n, m\ or\ n \notin \boldsymbol{C}$ such that $F_{m,n} \geq F_{k,l}, \forall (k, l) \in \boldsymbol{N}$

For all $subset \in \boldsymbol{C}$ **do**

If $|subset| < s$ **AND** $\exists (a, b) \in (\{m, n\}, subset)$ such that $F_{a,b} > 0$ **do**

Add the qualified node to $subset$ \\break the ties by largest connection

Else do

Add the set $\{m, n\}$ to $\boldsymbol{C}$

End If

End For

Set $F_{m,n} = 0$

Set $\boldsymbol{N}' = \boldsymbol{N}' \backslash \{m, n\}$

End While

For all $subset \in \boldsymbol{C}$ **do**

Solve the master problem for the planning horizon $[1, T]$

End For

Aggregate the solutions

To understand the general clustering, consider a small network of four nodes and all possible edges, denoted a K_4, or a complete graph with four vertices, shown in Figure 2. In general, as defined in graph theory, a K_n is a complete graph of n number of nodes or vertices. Suppose that a number of smaller time periods is solved and shipping only occurs between nodes (1 and 2) and (2 and 3), as seen in Figure 3. In this case, we would aggregate these three nodes into one cluster, and node 4 would be set into a separate cluster, Figure 4.

In Figure 3, it can be seen that the bold edges represent the most frequent shipping routes. These frequencies determine that projects 1,2,3 should be clustered together (K_3), while project 4 is clustered separately (K_1). Subsequently, each cluster would be solved independently and the objective function values and solutions aggregated to form the final solution. It should be noted that if the frequencies yield a connected graph between all projects, this would be equivalent to the original problem, K_N, and the reason for the limit on cluster size, s. This problem is denoted the Master Problem. Similarly, if no significant frequencies are determined and every project is in its own cluster, then this case is equivalent to the problem without shipping considerations (N number of base model problems, denoted $N - Base$). More specifically, if the clustering was chosen in such a way that there were N number of K_1 graphs, the $N - Base$ problem is returned. Therefore, the optimality is expected to fall between these two problems, where solving without shipping would yield a faster and less optimal solution and including shipping would result in a much more expensive computation time, yet an optimal solution. This provides a valid upper and lower bound for the algorithm. The bounds on optimality would be:

$$Master\ Problem \leq CDA \leq N - Base$$

The premise behind the clustering is to be able to efficiently solve large problem size without giving up optimality significantly. This is discussed in the Computational Results section.

Proposition 1: The optimal solution to the Master Problem is less than or equal to N-base.

Proof

Because the solution space of the Master Problem contains all points of the N-base problem, we can say that the solution space of the master problem is a superset of the N-base problem's solution space. Therefore, by the properties of infima, if a set A is a subset of a set B, any function defined on the appropriate domain, with a codomain in R gives:

$$\min(f(B)) \leq \min(f(A))$$

Thus, with the objective functions defined, an optimal solution to the Master Problem is at most that of the N-base problem.

■

Proposition 2: The optimal solution of the Clustering Decomposition Algorithm (CDA) is greater than or equal to the Master Problem, and less than or equal to the N-base problem.

Proof

We have three cases:

Case a. The number of clusters is equal to the number of projects

Case b. The number of clusters is one

Case c. The number of clusters is strictly greater than one and strictly less than the number of projects.

For Case a: the number of clusters is equal to the number of projects, we return the N-base problem, and we are done.

For Case b: the number of clusters returned is one, then all projects are included into the problem, and we return the Master Problem. In this case we are done.

For Case c: Because all clusters have to be considered and the CDA utilizes the Master Problem and the N-base models only, then the least number of clusters is two and the most number of clusters is $P - 1$, with integer number of clusters required. Therefore, the solution space is at least a superset of the N-base problem and at most a subset for the Master Problem. Then, using the properties of infima as before, the minimum of the CDA is at most that of the N-base, and at least that of the Master Problem.

■

CHAPTER FOUR

NUMERICAL EXAMPLES AND COMPUTATIONAL RESULTS

Numerical Examples

In this section, numerical examples for each problem formulation are presented in detail. The data collected are also presented to represent as realistic of instances as possible. The base model is solved in stochastic form, showing the benefits of considering an uncertain planning horizon. Next, the multiple demand sites are solved in deterministic form, highlighting the influence of shipping on the cost savings and practicality. Lastly, multiple purpose machines are considered, presenting the potential benefit of considering machines that can perform various jobs in replacement.

Data Collection for Realistic Problem

Here the application of the presented models in construction projects is shown. The data for these numerical examples from various resources. The replacement of a number of excavators working in parallel in a hypothetical construction project, which has a stochastic length. There are three main components that form the data for this case study: length of the project, number of projects and costs. First, a discussion on how the scenarios for the length of the project in the stochastic case are formed, and then the cost-related data is presented. The number of projects is assumed to be one, unless specifically stated.

The data was collected from a number of online and literature sources. The parameter values were calculated from searching online to determine standard values, such as heavy equipment purchase prices, maintenance costs, and salvage values. Although, this information is desired to be as accurate as possible, the value of the presented work is in the modelling and analysis. The data give a realistic depiction of the operations of a construction company, mainly for the ease of comparison and understanding.

Formulation of Horizon Uncertainty

The demand for machinery is modelled in a stochastic planning horizon by having different distributions for demand over time. The length of the planning horizon changes as each scenario is realized. Each scenario, for the length of the planning horizon, has a determined probability of occurring. As an example, consider two potential scenarios for the length of a hypothetical project. In the first scenario, the project takes 3 years to complete and the demand for a certain type of machine is [5,7,5] (5 machines for the first and last periods and 7 machines for the second period). In the second scenario that takes 5 years, the distribution of the demand is [2,4,5,4,2]. Note that the total number of required machines, 17, does not change in either scenario, as the workload is the same and it is only the time frame that changes. Therefore, the distribution of demand is also stochastic.

Realistic Parameter Functions and Data

Let us assume that a heavy construction project is scheduled to be completed in 5 years. Because of the stochastic nature of large construction projects, probabilities are assigned for finishing the project early, on time, or extending beyond the 5-year deadline. Table 2 shows the possible scenarios that can be realized. These scenarios have been constructed as follows. The average ratio of actual time to the agreed time for road work varies between a maximum value of 2 and a minimum of 1/3 (Kaka & Price, 1991). Al-Momani (2000) conducted a quantitative analysis on the delays in 130 construction projects. The following implications from the data were observed:

- Compared to projects that were delayed, projects that finish early were close to the planned finish date; and
- Projects that have a planned duration of 2 years or more are more likely to get delayed instead of finish early.

Based on these observations, only one scenario is considered in which the projects finish one year earlier. In the second scenario, the project finishes on time. Because Assaf and Al-Hejji (2006) concluded that 70% of the projects get delayed, it is determined that the sum of these first two scenarios to be 0.30. Kaliba, Muya, and Mumba (2009) studied the delays in 13 road construction projects in Jordan, and it was observed that the projects that were finished could get extended up to four years; and in one case the project continued after four years of delay. Five more scenarios were considered in which the project finishes beyond the schedule. In Scenarios 3 to 6, the project gets delayed from one year to four years. In Scenario 7 (the last Scenario) the project gets delayed beyond four years.

Regarding the cost parameters, most of the data comes from various online sources, including Vorster (2014), WRS (2015), ZIEGLER (2017). Excavators are considered as a single type of machine for which parallel machine replacement analysis is performed. The cost of renting a machine was established by searching multiple websites for the average cost of renting a 50,000 lbs. excavator for a month, and then using that value to obtain the cost of renting the machine for one time period (one year). Similarly, the operations cost was derived from fuel usage and operator costs, assuming around 2,000 hours of annual work for each machine, which is an average level of utilization. In the next section, the optimal common utilization level is discussed, followed by all of the machines that are working in parallel. The maximum age of a machine (L) is assumed to be 6 years. Because the excavators are assumed to operate under 12,000 hours, the maximum cumulative utilization level (U) is $\left\lfloor \frac{12000}{2000} \right\rfloor = 6$. Note: if it is desired to make monthly decisions, U and L have to be multiplied by 12, giving $U = L = 72$. This greatly increases the complexity of the problem and is discussed in detail in Computational Results. Because it is not possible to operate the excavator more than 8,000 hours per year, the utilization level cannot be more than 4 ($1 \leq \mathrm{j} \leq 4$) when its age is one-year old ($\mathrm{i} = 1$). The utilization can be at most 16,000 hours for an excavator that is two-years old ($1 \leq \mathrm{j} \leq 8$ for $\mathrm{i} = 2$), and it cannot be more than 20,000 hours for older excavator because the assumed policy is that the useful life of the machine is 10 years, or 20,000 hours of operation.

The cost of purchasing a machine of specified age and cumulative utilization level is assumed to decrease as the age of the machine, as well as the cumulative utilization level, increases. Maintenance costs are assumed to increase when maintaining a machine of higher age and cumulative utilization level, while the salvage value is assumed to decrease as time passes. The rest of the data is as follows. Renting an excavator costs $\frac{\$90000}{year}$, holding cost is $\frac{\$5000}{year}$, and operating cost is $\frac{\$50000}{year}$. Table 3 shows the purchase price of a CAT 320E L hydraulic excavator that is dependent upon $i \in [1, L]$ and $j \in [1, U]$. This particular machine was chosen for the case study because of the vast amount of cost data and the observed popularity of the asset. Also, the E series from Caterpillar meets US EPA Tier 4 Interim emissions standards, which is an attractive attribute (Caterpillar, 2013). The values in this table are generated as follows. Joseph C. Hartman (2004) formulated the operations and maintenance costs, as well as salvage values as functions of time, interest rate, and utilization level. The following functions were derived from the mentioned functions:

$$c_{ijt} = \left(c_{11t} - \alpha \left[\frac{i}{\delta}\right] - \beta \left[\frac{j}{\delta}\right]^{u}\right)(1+r)^{t}\ , \tag{14}$$

$$m_{ijt} = \left(m_{11t} + \alpha \left[\frac{i}{\delta}\right] + \beta \left[\frac{j}{\delta}\right]\right)\left(\left(\left[\frac{j}{\delta}\right] + u\right)^{\gamma} - \left[\frac{j}{\delta}\right]^{\gamma}\right)(1+r)^{t}, \tag{15}$$

where α and β are scaling parameters that determine the change in costs for every step in (i, j). r is a defined rate that can be used to account for inflation and/or technology change for each time period, t. This rate is typically the market rate. The exponent γ is an input strictly greater than 1, as defined in (Joseph C. Hartman, 2004). The indices $i\ and\ j$ are divided by a parameter, $\delta = \{1,12,365\}$, depending on the planning horizon type (annual, monthly, or yearly). This is necessary to properly model the utilization of the machines over their respectful horizons. For the numerical example, $\delta = 1$, as the assumption is an annual replacement schedule. The data for this case study was generated based on these functions with a standard utilization factor set to Level 1, $u = 1$. For the purchase price, 217 data points were obtained for the mentioned excavator from MachineryTrader.com (2017), and fit the above functions to this dataset. Each data point includes the age (in years), cumulative utilization level (in hours), and purchase price (in USD), giving a real estimate of α and β. For the cost generation of the excavator $\alpha = 12000\ and\ \beta = 3000$, while for the maintenance cost $\alpha = 500\ and\ \beta = 5000$. A similar process was used for the maintenance cost. However, due to the scarcity of maintenance data found online or in the literature, the coefficients were determined based on the author's judgment. A base purchase price is assumed to be $\$22{,}000$ and a base maintenance cost is assumed to be $\$10{,}000$. These bases, along with functions (14) and (15) give a better representation of the cost behavior of the excavators over time and utilization. Salvage values were assumed to be 57% of the purchase price of a machine with the same age and cumulative utilization level. Table 3 and Table 4 show the generated data for the purchase price and maintenance costs for time periods 1 and 4, separated by a comma. Table 5 shows the assumed number of excavators required in each time-period, under various scenarios. The total number of required excavators over the planning horizon does not change from one scenario to another. This is because the workload (for example, the size of the land that is going to undergo earthmoving operations) is independent of how long it takes to complete the operations. In each scenario, the total number of excavators has been distributed across the planning horizon, while assuming that the respective length of the project is known in advance for that scenario. The following is the optimal solution of the case study, using the Gurobi optimization package implemented in Python. This solution is referred to as the recourse problem (RP), where all the scenarios are considered in optimizing the value of the first stage variables (X's). The first-stage solution is $X_{2,1} = 3, X_{3,1} = 1$, and $X_{i,j} = 0$, for all the remaining X's.

Table 6 shows the optimal values of the second-stage decisions. These values are read as follows. The first letter indicates the type of decision (P for purchase, I for inventory/idle, S for salvage/sell, R for renting, and O for operating) followed by the (i, j) values (age and cumulative utilization level) and the number of machines. For example, in the second time period of Scenario 1, three machines that are two years old with cumulative utilization level 1 are purchased (P(2,1)3), one three-year old machine which has been utilized up to Level 2 is kept in inventory as idle (I(3,2)1), and six machines are assigned to the operations (O(2,1)3, O(3,2)2, and O(4,2)1). Some trends in these values can be observed that might give general guidelines to the decision makers.

The optimal value for the objective function, total expected cost, is $\$2{,}196{,}599.30$. As seen in, the optimal solution algorithm of Gurobi chooses to purchase used machines in the first stage and subsequently fulfill the requirements by obtaining new machines and salvaging said machines throughout the project. The solver did not choose to rent machines. This result is intuitive considering the small data set provided in that there is no salvage value for renting. Likewise, it can easily be seen that purchasing used machines which have been lightly utilized is the best option.

Here, Another variation of the model is solved according to Birge (1982), in order to have a better understanding of the value of the stochastic model. Wait–and-See (WS) solution is the expected value of the objective function when the problem is separately solved for each scenario. This value is:

$$\begin{aligned} WS \ = \ & 0.05 \ \times \$2{,}148{,}330.65 + \ 0.25 \ \times \$ \ 2{,}209{,}583.55 + \ 0.30 \ \times \$2{,}281{,}496.50 \\ & + \ 0.20 \ \times \$2{,}347{,}708.93 + \ 0.10 \ \times \$2{,}383{,}898.49 + \ 0.07 \\ & \times \$2{,}445{,}768.77 + \ 0.03 \ \times \$2{,}510{,}051.64 \ = \$2{,}060{,}308.78. \end{aligned}$$

The expected value of perfect information, $\text{EVPI} = \text{WS} - \text{RP} = \$136{,}290.52$. The WS solution is not very useful in practice as it does not give one solution for the whole problem and instead gives a solution for each individual scenario, assuming that the scenario certainly happens. Furthermore, it is not possible to know which scenario is being realized as we proceed through the project. We know which scenario has been realized only when the project ends. The EVPI is the price that one should be willing pay to know which scenario is going to be realized with certainty.

A much simpler problem has also been solved, which is called the Expected Value Problem (EV), where the expected value of demand for excavators is considered for each time period for the maximum length of the planning horizon. This problem gives a first-stage solution that can be fixed in the WS problem to obtain the expected value of using the EV solution (EEV), which allows the second-stage decisions to be chosen optimally as functions of the first-stage solution obtained from the EV problem. Table 7 shows the optimal solution for the EV problem. Unlike the recourse solution, here the solver chooses to satisfy the demand by only renting when the expected number of required excavators is less than three (Time Periods 5-10). By fixing the first-stage solution of the EV problem ($X_{4,1} = 4$, and $X_{i,j} = 0$, for all the remaining X's) the EVV is calculated as

$$EEV = 0.05 \times \$2148910.33 + 0.25 \times \$2209583.55 + 0.30 \times \$2282655.81 + 0.20 \times \$2348288.60 + 0.10 \times \$2488788.37 + 0.07 \times \$2665074.62 + 0.03 \times \$2834421.17 = \$2{,}334{,}762.56.$$

Value of the Stochastic Solution (VSS) is the difference between the EEV and the solution of the recourse problem, namely $VSS = EEV - RP = \$138{,}163.26$. This value is "the cost of ignoring uncertainty in choosing a decision" (Birge & Louveaux, 2011) and shows the value of the proposed model for the small case study. Obviously, the VSS increases as the problem size and the number of scenarios increases. It should be noted that, practically, the input data (purchase prices) should accurately reflect the prices and availability of machines in the used excavator market. However, if there is no machine available in the market with the age and utilization of those found in the optimal solution, it can be dealt with in two ways: 1) the price of the unavailable machines can be set to infinity so that they are not considered in the optimal solution, or 2) a constraint can be added that forces the associated decision variables to take on zero value. The problem can be solved again to find the next best solution.

PMRP-S

Here, the effect of shipping on the solution is considered in detail. The cost data from the base model is again used here, along with the consideration of six cities (demand sites): San Antonio, TX, Bozeman, MT, Los Angeles, CA, Orlando, FL, Chicago, IL, and San Francisco, CA. The cities where chosen to represent the practicality of using the model nationally. The assumed requirements for each city's construction site can be seen in Table 8. It can be seen that the length of project in each city varies from 5 years being the shortest duration in San Francisco, CA to 8 years in Orlando, FL. The distribution of demand was chosen to represent various instances that may occur: stationary demand periods, monotonically increasing or decreasing periods, and increasing to the middle of a project and decreasing until the end of the planning period.

The distances between each city were determined using Google Maps, taking only the routes which a large excavator could be sent (interstates), shown in Table 9. It should be noted that it is assumed that the shipping cost to and from each location is equivalent (symmetric), yet the model can easily handle various costs in each direction. The assumed cost per mile for shipping is $4.00 and all other data is the equivalent to the previous case, solving over a 10-year period (Transport, 2017). Figure 5 gives a visual representation of the city locations and all possible combinations of shipping routes available ($\frac{6(6-1)}{2} = 15\ edges$). It can be seen that even for a small network, the number of possible solutions is quite large for shipping alone. Here $P = 6, I = J = 6, and\ T = 10$ and the decisions being made include buying, holding, operating, selling, and shipping machines. Note: renting is not considered for ease of understanding, yet can easily be included if so desired.

The results in Table 10 show the schedule for purchasing, holding, salvaging and shipping the Caterpillar 320E excavator to satisfy demand. The solutions are represented as before, with the exception of shipping, which is represented as an age pair $T(i,j)$, then the number of machines shipped, followed by a pair of cities shipped from, $(p_g \to p_h)$, for some $g, h \in P$. Consider the first time period schedule for San Antonio, TX. The demand is one machine and the model simply obtains a new machine to satisfy the requirement. The decision also shows that the purchased machine is operated in that same time period. In the subsequent time period, demand stays uniform at one, yet another excavator is purchased to take advantage of the increase in demand for the next time period, which jumps to three. This means that the decision to obtain and hold the asset (un-utilized) is a cheaper option than to buy two new in the next time period. The schedule continues in this standard fashion until time period six, when an asset has reached the end of its useful life and must be salvaged. Notice that the sole machine required in the last time period is not salvaged until the next time period after it has operated to satisfy demand.

There are four shipping instances in this particular example. In time period two, one excavator is shipped from Los Angeles to San Francisco of age one-year-old and that has worked 2000 hours. Next, in year 5, one excavator is transported back to Los Angeles from San Francisco of age four years old and has worked around 6000 hours. Note: this is a different asset than was sent to that project originally. The next shipping occurrence happens in time period six, in which two machines near the end of their life are sent from San Francisco to Bozeman, MT to finish out their useful lives. Lastly, one machine is sent from Chicago to Orlando in year seven, also at the end of its life.

Considering the first shipping instance, it is easy to see that demand in San Francisco dropped, while the demand in Los Angles increased, allowing for the opportunity for transportation to satisfy demand. It can also be observed that demand increased in Chicago as well, yet the cheaper option was to ship to the cheaper city. In the next shipping schedule, it can be seen that demand increased in San Francisco, yet there were already excess machines in holding in said city from the previous time period, and the model took advantage of the idling ability to send the assets to Los Angles in the next time period. This demand increase was less than the number of machines available in San Francisco; therefore, the model would need to decide to hold for another year, sell the machine, or ship it. This instance would not be possible with the current modelling techniques that exist.

The third shipping happens in the sixth time period, when demand in San Francisco drops to zero (i.e. the project ends). In Bozeman, two machines are at the end of their lives and need to be salvaged. The model globally realizes the advantage and ships two machines to Montana. The last shipping occurrence is similar to that of the shipping in time period six. The Chicago project is coming to an end and although demand in Orlando is decreasing, it is advantageous to ship an excavator there and finish out the machines life. The shipping routes can be seen visually in Figure 6, where orange nodes are cities that have been transported to or from.

The benefit of considering the integration of shipping decisions into PMRPs can be quantified, even in this small example, as quite significant. The objective function value (OFV) of the Master Problem considering all shipping routes is $12,176,448.24; while, solving the problem with the current available models does not consider shipping and has an objective function value of $12,210,123.14, a $33,674.9 difference. This cost will obviously amplify with larger problem size, discussed in the Computational Results.

MP-PMRP

The presented model with multiple operations and machine types, denoted multi-purpose machines, is considered with a small problem size. The benefit of considering machines that can perform various job tasks is given numerically. The costs data from the previous numerical examples is again used with a few stated changes shown in the tables. Here, three machine types are considered and assumed to take on one or two available operations (one-to-many). The model is solved for this instance and compared to that of a one-to-one correspondence of machines to jobs. As seen in Table 11, machine 1 can perform operation 1, machine 2 can perform operation 2, and machine 3 can perform both operations. This is seen in practice, specifically for excavators. Excavators have been known to be used for digging, pile driving, brush cutting and material handling. Many operations are demanded in a construction project, and modelling with multi-purpose machines will allow for a cost-effective demand satisfaction.

Excess costs and varying O&M values are included for a machine performing a task. For instance, there may be no additional cost for a task that an excavator is normally used for, yet requires additional resources in order to perform a separate operation. To accurately represent this scenario, the machines have the assumed costs shown in Table 12. It may also be the case that a machine that can perform multiple tasks has a higher initial cost. Likewise, it may be that machines have varying holding and renting costs, seen in Table 12.

The holding cost for machines 1 and 2 is assumed to be $5000 annually, while machine 3 is assumed to have double the cost at $10,000. The renting cost of the multi-purpose machine is also more expensive by $30,000. Following the same premise, the purchase price of the multi-purpose machine is higher. Here it is assumed that the cost increase of machine 3 is due to the greater capabilities. Likewise, the assumed O&M cost of machine 3 is higher for operation 1, yet is lower for operation 2, seen in Table 13. Here it is assumed that operation 2 is easily completed by machine 3, yet any cost input can be entered and still retrieve an optimal solution.

The demand for this particular type of problem is necessarily operational dependent and requires the assumed extension given in Table 14. The demand for operation 1 is for an eight-year period, while operation 2 is only needed for six years. Demand for the first operation increases to a point and then decreases after (both monotonically), until the end of the planning horizon. For operation two, the same demand structure is used. Table 13 also gives the optimal schedule for the numerical example. In this table, in order to fit the solutions into the table space for representation, the decision variable that determines the number of machines operating of each type is represented by $O(k, o)$, removing the ages of the machines, for machine type k and operation, o. Purchasing, Idling, and salvaging are represented as $P(i, j, k)$, $I(i, j, k)$, and $S(i, j, k)$, giving the age (i, j) and machine type k obtained, held, or sold.

The optimal solution given in Table 14 utilizes both the single purpose machines, as well as the machine that can perform both operations. In the first time period, a total of four machines are needed to satisfy demand, and the model initially chooses to purchase two new machine 2's for performing operation 2, and one used machine 1 for performing operation 1. In the second time period, one machine 3 is purchased that is two years old and has worked less than 2000 hours, and a similarly used machine 1 is purchased. Machine 3 operates on operation 2, while machines 1 and 2 operate on their specific tasks. One older machine 1 is sent to holding with age three years old and has worked under 4000 hours. In time period three, another used machine 3 is purchased, and a used machine one is purchased. Here machine 3 only operates on operation 2, and all other demands are satisfied by machines 1 and 2. One machine of age three years old and that has worked under 6000 hours is idled. For time period four, two machine 2's are obtained of two years old and that have worked under 2000 hours. Again, the multi-purpose machines only work on operation 2 and remaining demand is satisfied by single purpose machines. A different machine is held for one time period. It should be noted that the model avoids idling machine number 3, most likely because of its significantly higher holding cost, even though it is sent to work with its higher operations cost. In time period 5, machine 3 works on both operations 1 and 2. A machine reaches the end of its useful life and must be salvaged. Demand satisfaction happens in the same manner in the subsequent years, until the end of the planning horizon.

The model was solved allowing for multi-purpose machines and compared to that of a more restrictive assumption, a one-to-one relationship between machines and operations. For this small example, the benefit of considering multi-purpose machines is the OFV of the model solved with a one-to-one assumption minus the presented formulation, shown here:

if machine 3 can only perform operation 1:

$$\$1{,}397{,}449.84 - \$1{,}385{,}973.84 = \$11{,}476$$

if machine 3 can only perform operation 2:

$$\$1{,}389{,}590.84 - \$1{,}385{,}973.84 = \$3617$$

Therefore, even for a small problem size, there can be significant cost savings in satisfying demand by considering the various operations a machine can perform. For this model to be implemented, the demand needs to be operation specific. Although the multi-purpose machine has much higher costs, the model globally determines a schedule that may be counterintuitive, especially for larger problem size as the number of machines and operations grows combinatorically.

Computational Results

In this section, small and large problems are solved to show the value of using the modeling techniques and solution methods presented. Also, sensitivity analysis and determination of the optimal utilization levels are given. Here the application of the presented model is shown in a construction project for each problem instance. The data for the case studies were obtained from various resources discussed previously. Parallel machine replacement of excavators is considered in construction projects, which have stochastic and deterministic lengths. It is also desired to not only consider the sensitivity of parameters, but also the efficiency and effectiveness of the Clustering Decomposition Algorithm.

Sensitivity Analysis

In this section, the managerial implications are presented from the sensitivity analysis performed on the model. Conventional sensitivity analysis determines the sensitivity of the objective function value associated with an optimal solution to available resources (budget) or demand. In the analysis presented in this section, the concern is with sensitivity of the solutions to the uncertainty in the input data. It is desired to see if the recommended optimal schedule for purchasing, selling, or holding the machines remains optimal if the data fed into the model changes up to a reasonable percentage.

Because the objective is a linear function of cost-coefficients, and according to Equations (14) and (15) these coefficients are themselves linear functions of the factors (α,β,etc.), it implies that the objective is also a linear function of the factors. Therefore, a linear relationship between a change in a factor and a change in the optimal OFV implies that the optimal solution does not change when the factor changes (when the optimal solution is not unique, the OFV associated with the set of optimal solutions does not change when the relationship is linear). If the optimal solution (the OFV associated with the set of optimal solutions) changes, a nonlinear relationship may be observed due to a shift in the growth of the optimal OFV.

The slope of a linear relationship determines the overall impact of the factor on the OFV. When the slope is 0 (a horizontal line), the decision variables associated with the factor under consideration have a value of 0 in the optimal solution. A larger slope indicates a higher overall impact on the objective function. If the functions in Equations (14) and (15) where nonlinear with respect to a certain factor, it could still be seen if the optimal solution changes by checking for sharp points due to shifts in the growth of the optimal

OFV, indicating a change in the optimal solution, or change in the OFV associated with the set of optimal solutions.

In order to gain insight into the model's behavior in a realistic manner, the utilization coefficients, age coefficients, and base costs were changed to determine the effect on the objective function. The utilization coefficient on the purchase price and O&M costs is a model of how much cost would be incurred if the impact utilization had on the cost changed. As an example, an increase in the utilization coefficient would imply that, if each utilization level previously changed the purchase or O&M costs by $10, now it changes by >$10. Likewise, for the age coefficients, if the amount of a used machine purchase decreased $10 for every year since the machine was new, then a larger age coefficient would mean the savings of purchasing the used machine would be >$10 per a year. Each of the factors were considered individually while the others were held fixed. The model was solved multiple times at each percentage increase of the factor in question. Figure 7 gives the results from the analysis. As seen in the figure, the base purchase price had the greatest impact on the objective function, intuitively with the steepest slope. That is, an 8% increase in the base cost of a new machine can alter the objective by 1%. Base maintenance cost and the utilization coefficient on maintenance cost showed the next most significant, with both having almost the exact same slope. Next, the utilization coefficient of the purchase price shows that a 12% variation can change the objective function by 0.5%. The utilization coefficient of a purchased machine has a slight effect on the objective, and the age coefficient for maintenance and purchase price have a far less significant impact.

Considering the results from this analysis, it can be concluded that if the purchase price of the machines increases, the corresponding OFV increase is to be expected. The same goes for all aspects, yet it is arguable that a larger increase of age-based cost for purchase and maintenance is not of much concern. Note that all effects are linear and can easily be predicted. An interesting finding is that the amount of utilization of the O&M costs has more of an impact on the OFV than that of the purchase price, while the age changes have almost the same effect. This means that the amount of utilization has a much more significant influence on the final total cost of satisfying demand. Because of this, the second analysis performed was on the assumptions of utilization of the machines.

Utilization factor levels were used to determine the implications of varying levels of utilization assumptions of the machines on the optimal objective function values. These implications are extensions of work completed by (Joseph C. Hartman, 2004). Hartman showed the optimal way to utilize each machine under various operations, maintenance, and salvage value assumptions. Yet, here the assumption is of equal utilization of all machines based on Hartman's conclusions (exponentially increasing O&M costs and linearly decreasing salvage values), and determining *what* level all machines should be equally utilized. Hartman has considered this in (Jha, 2000a) and (Joseph C. Hartman, 2004), where they make utilization a decision variable, yet this work aims to simply the process by decomposing the work and solving multiple easier problems. That is, we know that because the O&M and salvage costs are structured in such a way, each machine should be utilized the same number of operation hours; yet, what should that uniform quantity be?

In order to perform the analysis, some changes to the model were implemented to account for the changing utilization amounts. A decision variable was added to the model, namely, $E_t = the\ number\ of\ machines\ being\ utilized\ in\ time\ t$. The new objective function is presented here:

$$\text{Minimize } TC = \sum_{i=1}^{L} \sum_{j=1}^{U} c_{ij1} X_{ij} + E[Q(\xi, Y)] \tag{42}$$

$$E[Q(\xi,Y)] = \sum_{\omega\in\Omega} p_\omega \left(\sum_{t=2}^{T^\omega} \left[scale * E_t + qz_t^\omega + o_t a_t^\omega \right. \right.$$

$$\left. \left. + \sum_{i=1}^{L} \sum_{j=1}^{U} \left((fcp + c_{ijt}) y_{ijt}^\omega + h_t \left(I_{ijt}^\omega + S_{ijt}^\omega \right) + m_{ijt} a_{ijt}^\omega + fcsS_{ijt}^\omega - s_{ijt} S_{ijt}^\omega \right) \right] \right)$$

The objective contains a penalty function that increases the objective when the capacity of each time period is exceeded. The function is assumed to be linear and adds cost with a scaling factor determined by the engineer. Also, the salvage and purchase values include a fixed cost for each purchase or sale. That is, the fixed costs for buying can include staffing for finding the machines, shipping prices for each purchase, or any other fixed cost associated. Likewise, a fixed cost for salvaging is given to account for the cost incurred for finding a buyer and finalizing a sale. For this reason, the machines that are salvaged are assumed to take an inventory cost, seen in objective with inventory. Also, a constraint was added to the model in order to perform the analysis, shown in inequality (43).

$$E_t \geq \sum_{i=1}^{L} \sum_{j=1}^{U} a_{ijt} - cap_t \qquad \forall t \in [2, T^\omega] \qquad (43)$$

Constraint (43) is the penalty constraint that works in conjunction with the objective function. The constraint forces the objective to add additional costs when the model chooses to exceed capacity. This can occur when it is desired to utilize multiple machines at a lower level each, versus fewer machines working harder, in each time period.

The utilization factor is a change of assumption for the utilization amount of all machines. When the utilization factor is at 1, we have the standard assumptions discussed previously (2000 hours of work annually). But, when the utilization factor changes to, say 2, all of the machines work twice as much and fewer machines will be needed to satisfy demand. Likewise, a utilization level of 0.5 means that all machines are utilized at half of the initial assumptions (3 hours/day), yet more machines will be required to fulfill all machine requirements. This is reflected in the model by changing the right-hand-side of Constraint set (2) to r_t^{ω}/u . Therefore, the question becomes what level of utilization is optimal for this particular data set? To test this, the case study was solved multiple times while changing the utilization factor levels. Note that the highest possible factor level is 4, as the machines cannot work more than 24 hours per day.

It should also be noted that the capacity restriction was set so the model does not attempt to choose infinitely many machines at a very low utilization level. As shown in Figure 8, the optimal utilization level should be that of 1, giving the minimum objective value when the excavators are used 6 hours a day. It can be seen that working the excavators slightly more or slightly less would yield a sub-optimal solution. It should be also noted that if the excavators are to be utilized more intensely, it is better to work them at u=2, which is the next cheapest option. This tool can aid the project managers in prescribing a common utilization level for the machinery of the same type.

For each unique problem solved using the presented model, this process can be performed to determine the optimal utilization of all machines, under the proper assumptions given in Hartman (2004). For the construction industry, it seems that the presented implications would hold true because of the way operations and maintenance costs increase and the salvage values decrease. The process would still need to be performed in order to understand what utilization level would be optimal, yet in most cases it is probable that utilizing all assets equally is optimal.

To determine the effects that interest rate assumptions have on the objective function, an interest rate or 1% was assumed and then a uniform percentage increase was considered, subsequently solving for the optimal solution at each instance. The percentage change in the OFV was recorded. As seen in Figure 9, as interest rates increase, the percent change in the objective function increases non-linearly. This is not a surprising result because of the effect that interest has on costs data as time increases.

Sensitivity of the multiple project model was also considered. The questions of when does a shipping happen and how do shipping prices affect the optimal solution are driven from managerial and realistic perspectives. As described in the numerical example, shipping is both demand and cost dependent. Because demand is especially sensitive, and any fluctuation of demand may yield an infeasible solution, the sensitivity analysis is performed on the distance matrix, changing the assumptions of the cost-per-mile of the shipping cost. This is performed starting with a base cost of $1/mile and increasing to over $20/mile. As seen in Figure 10, the increase in cost-per-mile has a linear effect on the value of the objective function. In fact, a large increase in shipping cost has a minimal effect on the percent objective increase.

For instance, even if the cost-per-mile of shipping an excavator increased to $20 a mile, the overall cost of satisfying demand only increases by 0.06%. This is due to the value of having an optimal shipping schedule incorporated into the replacement and demand satisfaction decisions. It is arguable that implementing this particular model formulation is imperative to any multiple demand site instance. That is, with the optimal shipping and replacement decision in place, outside influences that would normally effect shipping are mitigated.

To gain insight on the effects that multi-purpose machines have in the solution and objective function, the base cost of the machine that can take on two operations from the numerical example was changed, holding all other cost the same. Seen in Figure 11, as the multi-purpose machine’s base cost increases (closer to the costs of the other machines), the objective is affected in a nonlinear manner. The 20% difference in the multi-purpose machine yields around a 10% decrease in the OFV. A base cost of $240,000 was used. Also, as the machine’s value increases to the value of the single purpose machines, the model determines that it is not economical to continue purchasing the multi-purpose machine. In fact, at a 9% increase in the multi-purpose machine value, the optimal solution never buys one for use.

Next, because of the implications from the first sensitivity analysis, all values are held fixed for the single purpose machines and change the base maintenance cost of the multi-purpose machine to determine the effect on the solution as well as the percent change in objective. A base cost of $8000 was used for operation 2 and $12,000 for operation 1. The costs are again increased or decreased incrementally for the analysis. It can be seen that the optimal solution decreased by up to 3% and increased to almost 1%. As seen in Figure 12, the base cost of both operations for the multi-purpose machine have a nonlinear effect on the objective function. With a large percentage decrease in the value for both operations, it is observed that the percentage the OFV decreases grows quickly. An increase in the maintenance cost of operation 1 seems to increase linearly with an increase greater than 1, and in parabolic fashion for an increase of operation 2's maintenance cost.

Efficiency of CDA for Large Problem Size

In this section, a number of large problem sizes are solved for the instance where shipping is of concern. For the models that only consider one demand site, the solution time for any realistic problem size is instantaneous. For instance, solving the base model for over 100 years was less than a 60 second computation time. Even making monthly decisions, for any realistic case (monthly decisions for say three years) it can be seen that the solution time is also instant. Any monthly or annual decisions beyond these thresholds lose a realistic scheduling in that demand realization may change greatly over time, and when this is the case, Stochastic Programming is the best solution method. For the shipping instances where the problem size and solution time grows much faster as more demand sites are solved for, an efficient solution methodology is possibly required.

Twenty test problems of various medium and large size are presented to evaluate the performance in terms of optimality and solution time for this case. The experiments were performed on a computer cluster made up of 11 Dell DSS 1500 servers, each containing 2 Intel Xeon E5-2630 8-core processors running at 2.4 GHz. Each server has 32 GB of RAM. There are a total of 176 processor cores and 352 GB of RAM. Table 15 gives the sizes of the sets for each problem size solved. Up to 25 projects are solved because it was observed that the largest construction company in the world, Bechtel, has presented around this number of projects in the US that are currently under construction (although they have 100s of projects worldwide). It is desired to determine at what point the *CDA* is required for solution. Because the complexity grows significantly with U and L, it was decided to make monthly decisions, versus annual. If daily decisions were desired, these parameters would need to take on the values of $365x6 = 2190$; yet, this is not practical in this case, as most construction projects seem to leave assets in place for at least one month. It was also determined that the most practical planning horizon for making monthly decisions would most likely not exceed that of 3 years, and 36 months is the largest considered here. It should be noted that it is not desired to determine how large of a problem size that can be solve, yet more importantly how large of a realistic problem can be solved before a solution method is necessary. Furthermore, it may be the case that weekly decisions be made, yet to solve this level of complexity would require significant changes in the model and is not in the scope of this work.

The computational results can be seen in Table 16. The objective function value (OFV) and the solution time in seconds are presented for each problem type and size. The gap between the Master Problem and the CDA is presented as Gap 1, and the gap between the CDA and the model without considering shipping is presented as Gap 2. The problem size changes in number of projects every five instances; consider the first five. The number of projects is five, and the Master Problem solves the model efficiently for the largest size in that realm in under 500 seconds. The performance of the CDA for small problem size is much less efficient, although the gap is always within 1% of the optimal solution. As the problem sizes increase, it is observed that the gap of the CDA stays significantly closer to the Master Problem, as the advantage of shipping is still in place. The average of Gap 1 is 0.16% while the average Gap 2 is 1.06%, both of which are considered good. Yet, the CDA performs much better in terms of optimality as the problems become much more complex, in retrospect to not considering shipping.

The CDA allows for the possibility of obtaining the optimal solution, as seen in Problem size 7. For Problem Sizes 10,14,15,18,19, and 20, the Master Problem and the No Shipping was not able to find a solution, while the CDA provided a solution relatively fast. In one case, Problem 17, the Master Problem could not find a solution, yet the No Shipping formulation could. This shows the advantage of a less complex model on solution time, yet the CDA algorithm provided a solution that was 2.58% better. In general, the fastest of the three methods is without shipping considerations, yet when the problem size is sufficiently large, even this formulation cannot find a solution. This is due to the CDA taking advantage of the structure of the problem formulation and shipping frequencies, allowing the problem to be broken up into clusters of maximum defined size, here six.

CHAPTER FIVE

CONCLUSION

In this dissertation, the Parallel Machine Replacement Problem was tailored to the specifications of construction machinery. The models generalize that of previous works in the parallel case, while considering the ability to purchase and salvage used machines at any age and utilization level, keep the machines in the inventory, rent to satisfy demand, transport machines, and utilize multi-purpose machines. These are often the case in application, especially in the management of heavy equipment. An application for construction projects was presented. The data was gathered and generated based on functions appearing in the literature and the parameters were set based on various data sets from the literature and online. Specific cost functions were used that generated the purchase prices, maintenance costs, and salvage values based on the real data. These functions take into consideration inflation and technology changes for the assets. The stochastic nature of the construction projects was considered yielding an uncertain time horizon and demand. These considerations are imperative to the construction industry due to the fact that most large projects get extended beyond the initial planning periods. Therefore, the Expected Value of Perfect Information (EVPI) and the Value of Stochastic Solution (VSS) were calculated to show the need to consider uncertainty in these types of projects. The EVPI and VSS showed significant figures at $136,290.52 and $138,163.26, respectively. Data from six real cities were collected based on distances and shipping costs of a large excavator. To show the value and importance of considering shipping decisions in PMRPs, the problem was solved with and without shipping, giving a difference of dollar amounts in the millions (Computational Results). Likewise, the comparison of considering multiple purpose machines is presented, having a value of over $11,000 for a small problem size.

A sensitivity analysis was performed to determine any managerial implications of the work for the procurement and replacement of assets. The impact of a changing purchase price and maintenance costs on the objective function was studied. This considered the effect of which a change in age and utilization had on the objective for both parameters. It was observed that the base purchase price had the most effect, followed by that of base maintenance costs and the costs of maintenance as it is utilized. The value of purchase price as it was utilized at various levels had the next most influence on the optimal solution, yet the values of purchase price and maintenance as the machines aged had little effect. Utilization level is known to be an important aspect of the PMR problems, yet it was desired to find the optimal utilization level of the machines under the proper assumptions. If it is known that the assets should be utilized equally, which was the case, then it is beneficial to determine at what level they should operate. The most cost-effective utilization level out of a set of possible values was determined in the sensitivity analysis, while avoiding utilization level as a decision. For the particular case study tested, it was found that the machines should work in 6-hour shifts, yet the analysis provides the next best value when utilizing the machines at 12 hours, with fewer machines. Any deviation from the best utilization level would yield an increase in the costs for the case study that would be from $100,000 to greater than $1,000,000. This process can be performed to determine the most cost-effective level for each individual project.

Similarly, sensitivity analysis was performed on the interest rate, shipping cost-per-mile, and cost parameters of multi-purpose machines. It is seen that interest has a nonlinear effect on the objective function value, and with an optimal solution in place, the shipping cost has little effect on the cost of satisfying demand. The implications of shipping price's minimal impact imply the importance of using the presented modelling techniques. Large construction companies can benefit significantly from integrating transportation decisions with the replacement decisions simultaneously. It was also observed that the base purchase price and maintenance cost of the multi-purpose machines can have various effects on the objective function value, in a non-linear manner. Each of these sensitivity analyses provide a basis for companies to base their replacement studies on. Each alternative solution provides opportunity costs, if varying alternatives are chosen. If the optimal solution is chosen, the loss from choosing a separate alternative is minimized, yet in reality many external factors affect the costs, and the sensitivity analysis provides a framework for considering multiple alternatives, to mitigate the lost opportunity.

Because the PMRP is combinatorial in nature, an efficient solution method was presented to solve large problem size when a number of demand sites are introduced. The algorithm begins by taking small time steps forward through the planning horizon, solving smaller instances of the problem. Then, the algorithm aggregates cities (project demand sites) into clusters. The original problem is then efficiently solved for each cluster and the solutions combined to obtain the full schedule. The algorithm was observed to perform well for large problem size, yet when the number of projects is small, the Master Problem outperformed the CDA. The CDA consistently held a larger gap between itself and the base problem, compared to the Master Problem, showing the CDA perforce closer to the global optimal solution.

Future works can include that of more solution methods to the PMRP-S problem, as the formulation is new, applying the M-PMRP to additive manufacturing for optimal replacement, and considering more on horizon uncertainty for PMRP-S. More specifically, the impact of varying salvage and purchase prices in different locations should be considered in the PMRP-S formulation. This type of cost behavior was observed on multiple machine purchasing websites, and would be desired when applying this work in a company. This extension, although easily implemented, would slightly increase the complexity, and more comparative computational results would be desired.

It would also be beneficial to extend this research to include capital budgeting and capital gains considerations. If the machine benefit can be measured, a possible look at return rates for each project and each replacement schedule. A possible consideration would be to study the effects of investing the money saved on each project, back into the projects. This may change the project length, machine utilization, and replacement schedules.

Furthermore, it is necessary to develop solution methods that are independent of commercial solvers. For instance, the problem structure may benefit from metaheuristic techniques, which have not been applied to the PMRP. The CDA algorithm presented provides a good general solution method for when advanced optimization software are available, yet for problem sizes larger than what is considered in this work, a subsequent heuristic technique should be developed, to be used in conjunction with the CDA. Likewise, the formulation with multiple operations may need a heuristic developed if a large number of assets and operations is considered, as may be the case in other applications.

Lastly, it is desirable to implement this research into a large construction company. The ability to look at the problem formulation under more accurate and specific data can derive much more general managerial implications and realistic solutions. This work can be implemented systemically with other works in construction management, such as the optimization of maintenance and production, where solutions of one may have implications on the other.

LIST OF REFERENCES

Al-Momani, A. H. (2000). Construction delay: a quantitative analysis. *International journal of project management, 18*(1), 51-59.
Ansaripoor, A. H., & Oliveira, F. S. (2018). Flexible lease contracts in the fleet replacement problem with alternative fuel vehicles: A real-options approach. *European Journal of Operational Research, 266*(1), 316-327.
ARCADIS. (2017). INTERNATIONAL CONSTRUCTION
COSTS 2017: COST CERTAINTY
IN AN UNCERTAIN WORLD. Retrieved from https://www.arcadis.com/media/2/C/F/%7B2CFC680C-A57E-43EA-99B4-0F4A784F7D51%7DAG1059_ICC%202017_FINAL_WEB%20SINGLE%20PAGES_NA-Opt.pdf
Assaf, S. A., & Al-Hejji, S. (2006). Causes of delay in large construction projects. *International journal of project management, 24*(4), 349-357.
Bazargan, M., & Hartman, J. (2012). Aircraft replacement strategy: Model and analysis. *Journal of Air Transport Management, 25*, 26-29.
Birge, J. R. (1982). The value of the stochastic solution in stochastic linear programs with fixed recourse. *Mathematical programming, 24*(1), 314-325.
Birge, J. R., & Louveaux, F. (2011). *Introduction to stochastic programming*: Springer Science & Business Media.
Büyüktahtakin, I. E., & Hartman, J. C. (2009). *Parallel replacement under technological change and deterioration.* Paper presented at the IIE Annual Conference. Proceedings.
Büyüktahtakın, İ. E., & Hartman, J. C. (2016). A mixed-integer programming approach to the parallel replacement problem under technological change. *International Journal of Production Research, 54*(3), 680-695.
Caterpillar. (2013). 320E L Hydraulic Excavator. Retrieved from http://s7d2.scene7.com/is/content/Caterpillar/C719302
Chand, S., McClurg, T., & Ward, J. (2000). A model for parallel machine replacement with capacity expansion. *European Journal of Operational Research, 121*(3), 519-531. doi:https://doi.org/10.1016/S0377-2217(99)00052-1
Childress, S., & Durango-Cohen, P. (2005). On parallel machine replacement problems with general replacement cost functions and stochastic deterioration. *Naval Research Logistics (NRL), 52*(5), 409-419.
des-Bordes, E., & Büyüktahtakın, İ. E. (2017). Optimizing capital investments under technological change and deterioration: A case study on MRI machine replacement. *The Engineering Economist, 62*(2), 105-131.
Dhillon, B. S. (2009). *Life cycle costing for engineers*: CRC Press.
Esra Büyüktahtakin, İ., Cole Smith, J., Hartman, J. C., & Luo, S. (2014). Parallel asset replacement problem under economies of scale with multiple challengers. *Engineering Economist, 59*(4), 237-258. doi:10.1080/0013791X.2014.898113
Faridi, A. S., & El-Sayegh, S. M. (2006). Significant factors causing delay in the UAE construction industry. *Construction Management and Economics, 24*(11), 1167-1176.

Frangopol, D. M., Lin, K.-Y., & Estes, A. C. (1997). Life-cycle cost design of deteriorating structures. *Journal of Structural Engineering, 123*(10), 1390-1401.
Frimpong, Y., Oluwoye, J., & Crawford, L. (2003). Causes of delay and cost overruns in construction of groundwater projects in a developing countries; Ghana as a case study. *International journal of project management, 21*(5), 321-326.
Hartman, J., & Ban, J. (2002). The series–parallel replacement problem. *Robotics and Computer-Integrated Manufacturing, 18*(3-4), 215-221.
Hartman, J. C. (2000). The parallel replacement problem with demand and capital budgeting constraints. *Naval Research Logistics (NRL), 47*(1), 40-56.
Hartman, J. C. (2004). Multiple asset replacement analysis under variable utilization and stochastic demand. *European Journal of Operational Research, 159*(1), 145-165. doi:https://doi.org/10.1016/S0377-2217(03)00397-7
Hartman, J. C., & Lohmann, J. R. (1997). Multiple options in parallel replacement analysis: Buy, lease or rebuild. *The Engineering Economist, 42*(3), 223-247.
Hartman, J. C., & Tan, C. H. (2014). Equipment replacement analysis: A literature review and directions for future research. *Engineering Economist, 59*(2), 136-153. doi:10.1080/0013791X.2013.862891
Hartman, J. C., & Tan, C. H. (2014). Equipment replacement analysis: a literature review and directions for future research. *The Engineering Economist, 59*(2), 136-153.
Hopp, W. J., Jones, P. C., & Zydiak, J. L. (1993). A further note on parallel machine replacement. *Naval Research Logistics (NRL), 40*(4), 575-579.
Jha, A. (2000a). *Multiple assets parallel replacement analysis under stochastic demand.* (Master's of Science), Lehigh University. Retrieved from http://preserve.lehigh.edu/cgi/viewcontent.cgi?article=1666&context=etd
Jha, A. (2000b). Multiple assets parallel replacement analysis under stochastic demand.
Jones, P. C., Zydiak, J. L., & Hopp, W. J. (1991). Parallel machine replacement. *Naval Research Logistics (NRL), 38*(3), 351-365.
Kaka, A., & Price, A. D. (1991). Relationship between value and duration of construction projects. *Construction Management and Economics, 9*(4), 383-400.
Kaliba, C., Muya, M., & Mumba, K. (2009). Cost escalation and schedule delays in road construction projects in Zambia. *International journal of project management, 27*(5), 522-531.
Keles, P., & Hartman, J. C. (2004). Case study: bus fleet replacement. *The Engineering Economist, 49*(3), 253-278.
Leung, L. C., & Tanchoco, J. (1986). Multiple machine replacement within an integrated system framework. *The Engineering Economist, 32*(2), 89-114.
Leung, L. C., & Tanchoco, J. M. A. (1990). Multiple machine replacement analysis. *Engineering Costs and Production Economics, 20*(3), 265-275.
Lotfi, V., & Suresh, N. C. (1994). Flexible automation investments: A problem formulation and solution procedure. *European Journal of Operational Research, 79*(3), 383-404.

MachineryTrader.com. (2017). CATERPILLAR 320EL For Sale. Retrieved from https://www.machinerytrader.com/listings/construction-equipment/for-sale/list/category/1031/excavators/manufacturer/caterpillar/model/320el

McClurg, T., & Chand, S. (2002). A parallel machine replacement model. *Naval Research Logistics (NRL), 49*(3), 275-287.

Parthanadee, P., Buddhakulsomsiri, J., & Charnsethikul, P. (2012). A study of replacement rules for a parallel fleet replacement problem based on user preference utilization pattern and alternative fuel considerations. *Computers & Industrial Engineering, 63*(1), 46-57.

Rajagopalan, S. (1998). Capacity expansion and equipment replacement: A unified approach. *Operations Research, 46*(6), 846-857.

Rajagopalan, S., Singh, M. R., & Morton, T. E. (1998). Capacity expansion and replacement in growing markets with uncertain technological breakthroughs. *Management Science, 44*(1), 12-30.

Ryan, W. (1968). *Procurement views of life cycle costing.* Paper presented at the Annual Symposium on Reliability.

Seif, J., & Rabbani, M. (2014). Component based life cycle costing in replacement decisions. *Journal of Quality in Maintenance Engineering, 20*(4), 436-452.

Seif, J., Shields,B. and Yu,A. (2018). Parallel machine replacement under horizon uncertainty. *The Engineering Economist*.

Seif, J., Yu, A. J., & Shields, B. A. (2017). *Minimizing cost of parallel machine replacement in construction projects with horizon uncertainty.* Paper presented at the 67th Annual Conference and Expo of the Institute of Industrial Engineers 2017.

Shane, J. S., Molenaar, K. R., Anderson, S., & Schexnayder, C. (2009). Construction project cost escalation factors. *Journal of Management in Engineering, 25*(4), 221-229. doi:10.1061/(ASCE)0742-597X(2009)25:4(221)

Sherwin, D. (2000). A review of overall models for maintenance management. *Journal of Quality in Maintenance Engineering, 6*(3), 138-164.

STINSON, J. P., & KHUMAWALA, B. M. (1987). The replacement of machines in a serially dependent multi-machine production system. *International Journal of Production Research, 25*(5), 677-688.

Tan, C. H., & Hartman, J. C. (2010). Equipment replacement analysis with an uncertain finite horizon. *IIE Transactions, 42*(5), 342-353.

Tang, J., & Tang, K. (1993). A note on parallel machine replacement. *Naval Research Logistics (NRL), 40*(4), 569-573.

Tosun, A. (2014). Determination of excavator type according to rock and excavator characteristics in soft formations that can be excavated directly. *Journal of Mining Science, 50*(2), 349-361. doi:10.1134/S1062739114020185

Transport, N. E. A. (2017). Cost to Ship Heavy Equipment. Retrieved from https://nxautotransport.com/cost-to-ship-heavy-equipment

Vander Veen, D. (1985). *Parallel replacement under nonstationary deterministic demand.* PhD thesis, University of Michigan.

VanderVeen, D. (1985). Parallel replacement under nonstationary demand. *Unpublished doctoral thesis, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor*.

Verheyen, P. (1979). Economic interpretation of models for the replacement of machines. *European Journal of Operational Research, 3*(2), 150-156.

Vorster, M. (2014). Annual and Life-to-Date Costs. Retrieved from http://www.constructionequipment.com/annual-and-life-date-costs

WRS. (2015). RATES FOR EARTHMOVING EQUIPMENT. Retrieved from https://www.worldwidemachinery.com/WRS/assets/wrs_rates.pdf

ZIEGLER. (2017). CAT, 320EL HYDRAULIC EXCAVATOR - MEDIUM. Retrieved from https://www.zieglerrental.com/equipment/excavators/cat-hydraulic-excavators-medium/cat-320el-hydraulic-excavator-medium/

Appendices

Appendix A: Tables and Figures

Table 1: Gap analysis of problem formulations appearing and not appearing in the literature.

PAPER	PT	MO	MM	HU	U	MDS	T
Dissertation Base Model	MILP			✓	✓		
Dissertation MM-PMRP	MILP	✓	✓		✓		
Dissertation PMRP-S	MILP			✓	✓	✓	✓
(Hartman and Ban 2002)	DP	✓				✓	
(Hartman 2004)	DP			✓	✓		
(Keles and Hartman 2004)	IP		✓		✓		
(Childress and Durango-Cohen 2005)	IP		✓	(stoch)			
(Hartman 1999)	IP				✓		
(Tan and Hartman 2010)	DP\LP			✓			
(de-Bordes and Büyüktahtakin, 2017)	MIP	✓	✓				

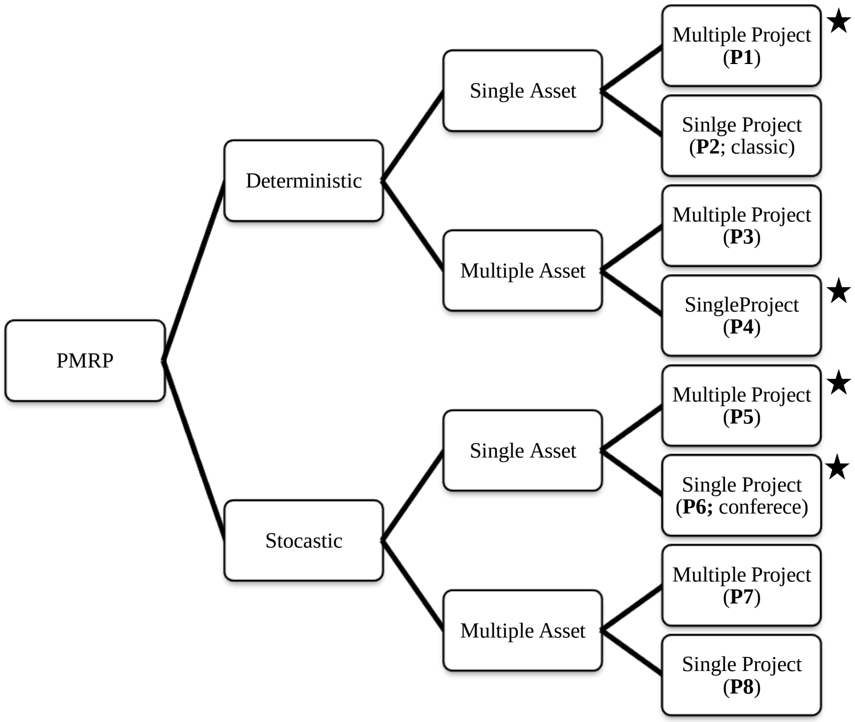

Figure 1: The heirarchical structure of the PMRPs (denoted P1-P8) being studied.

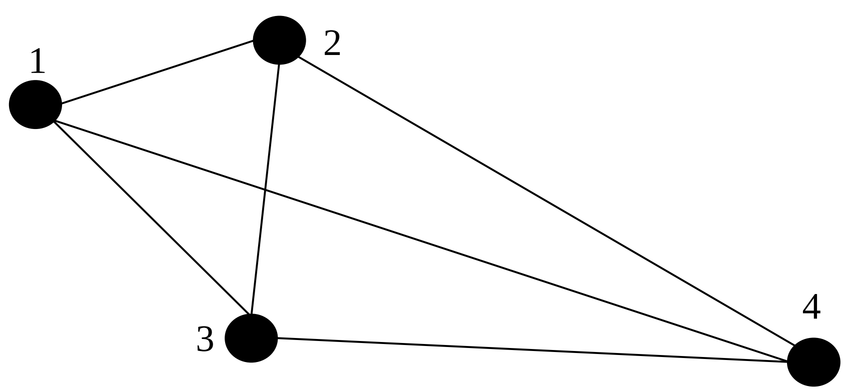

Figure 2: A starting graph with four projects and all possible shipping routes (K_4).

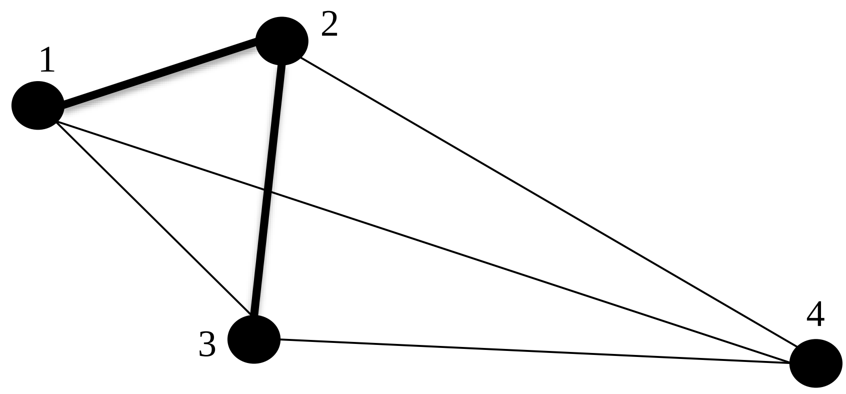

Figure 3: Frequent shipping routes of a network determined by solving random test problems (bold).

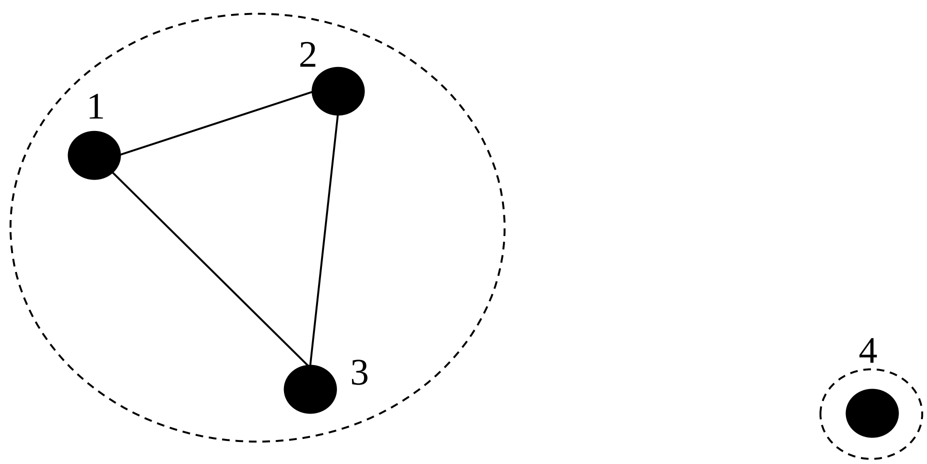

Figure 4: Clusters determined by the shipping frequency.

Table 2: Scenarios for the length of the project.

	Scenario (ω)						
	1	2	3	4	5	6	7
					Beyond schedule		
Description	Early	On-time					
Project Length (T^{ω} , in years)	4	5	6	7	8	9	>9
Probability (p_{ω})	0.05	0.25	0.30	0.20	0.10	0.07	0.03

Table 3: Purchase price (USD) of an excavator with age and cumulative utilization level (i,j) in time periods 1 and 4.

	j					
i	1	2	3	4	5	6
1	220000,240399	208000,227287	196000,214174	184000,201061	172000,187949	160000,174836
2	173000,206874	163400,195411	153800,183948	144200,172485	134600,161023	125000,149560
3	170000,203596	160400,192133	150800,180670	141200,169207	131600,157745	122000,146282
4	167000,200317	157400,188855	147800,177392	138200,165929	128600,154467	119000,143003
5	164000,197039	154400,185577	144800,174114	135200,162651	125600,151188	116000,139725
6	161000,193761	151400,182299	141800,170836	132200,159372	122600,147910	113000,136447

Table 4: Maintenance cost (USD) of an excavator with age and cumulative utilization level (i,j) in time periods 1 and 4.

	j					
i	1	2	3	4	5	6
1	10000,10927	21485,23478	47369,51762	89166,97434	147967,161688	224638,245468
2	10500,11473	21985,24024	47869,52308	89666,97981	148467,162234	225138,246014
3	11000,12019	22485,24571	48369,52854	90166,98527	148967,162781	225638,246561
4	11500,12566	22985,25117	48869,53401	90666,99073	149467,163327	226138,247107
5	12000,13112	23485,25663	49369,53947	91166,99620	149967,163873	226638,247653
6	12500,13659	23985,26210	49869,54493	91666,100166	150467,164420	227138,248200

Table 5: The required number of excavators in each time period under different scenarios.

Scenario (ω)	Time Period (Year)										Total
	1	2	3	4	5	6	7	8	9	10	
1	4	6	6	4							20
2	4	6	5	3	2						20
3	4	5	4	3	2	2					20
4	4	4	3	3	2	2	2				20
5	3	3	4	3	2	2	2	1			20
6	2	3	3	3	2	2	2	2	1		20
7	1	2	3	3	3	2	2	2	1	1	20

Table 6: Optimal solution to the recourse problem.

	Time Period (Year)										
	1	2	3	4	5	6	7	8	9	10	11
Demand	4	6	6	4							
Schedule for Scenario 1	P(2,1)3 P(3,1)1 O(2,1)3 O(3,1)1	P(2,1)3 I(3,2)1 O(2,1)3 O(3,2)2 O(4,2)1	S(5,3)1 O(3,2)3 O(4,2)1 O(4,3)2	S(5,4)2 O(4,3)3 O(5,3)1	S(5,4)3 S(6,4)1						
Demand	4	6	5	3	2						
Schedule for Scenario 2	P(2,1)3 P(3,1)1 O(2,1)3 O(3,1)1	P(2,1)2 O(2,1)2 O(3,2)3 O(4,2)1	P(2,1)1 I(3,2)1 S(5,3)1 O(2,1)1 O(3,2)1 O(4,3)3	S(5,4)3 O(3,2)1 O(4,2)1 O(4,3)1	S(5,4)1 O(4,3)1 O(5,3)1	S(5,4)1 S(6,4)1					
Demand	4	5	4	3	2	2					
Schedule for Scenario 3	P(2,1)3 P(3,1)1 O(2,1)3 O(3,1)1	P(2,1)1 O(2,1)1 O(3,2)3 O(4,2)1	S(5,3)1 O(3,2)1 O(4,3)3	P(2,1)2 S(5,4)3 O(2,1)2 O(4,3)1	S(5,4)1 O(3,2)2	O(4,3)2	S(5,4)2				
Demand	4	4	3	3	2	2	2				
Schedule for Scenario 4	P(2,1)3 P(3,1)1 O(2,1)3 O(3,1)1	O(3,2)3 O(4,2)1	S(5,3)1 O(4,3)3	P(2,1)3 S(5,4)3 O(2,1)3	I(3,2)1 O(3,2)2	I(4,3)1 O(4,2)1 O(4,3)1	S(5,4)1 O(5,3)2	S(6,4)2			
Demand	3	3	4	3	2	2	2	1			
Schedule for Scenario 5	P(2,1)3 P(3,1)1 O(2,1)3 O(3,1)1	I(3,2)1 O(3,2)2 O(4,2)1	O(4,2)1 O(4,3)2 O(5,3)1	P(2,1)2 S(5,4)2 S(6,4)1 O(2,1)2 O(5,3)1	S(6,4)1 O(3,2)2	P(2,1)1 I(4,3)1 O(2,1)1 O(4,3)1	S(5,4)1 O(3,2)1 O(5,3)1	S(6,4)1 O(4,3)1	S(5,4)1		

Table 6: Optimal solution to the recourse problem (Continued).

	Time Period (Year)										
Demand	2	3	3	3	2	2	2	2	1		
	P(2,1)3	I(3,2)1	P(2,1)1	P(2,1)1	S(6,4)1	P(2,1)1	P(2,1)1	O(3,2)1	S(5,4)1	S(5,4)1	
	P(3,1)1	O(3,2)2	S(4,3)1	S(5,4)1	O(3,2)1	S(5,4)1	S(5,4)1	O(4,3)1	O(4,3)1		
Schedule	O(2,1)3	O(4,2)1	S(5,3)1	O(2,1)1	O(4,3)1	O(2,1)1	O(2,1)1				
for	O(3,1)1		O(2,1)1	O(3,2)1		O(4,3)1	O(3,2)1				
Scenario 6			O(4,2)1	O(5,3)1							
			O(4,3)1								
Demand	1	2	3	3	3	2	2	2	1	1	
	P(2,1)3	I(3,2)2	S(5,3)1	P(2,1)1	P(2,1)2	I(3,2)1	S(5,4)1	P(2,1)1	S(6,4)1	O(4,3)1	S(5,4)1
Schedule	P(3,1)1	O(3,2)1	O(4,2)2	S(5,4)1	S(6,4)2	O(3,2)1	O(4,2)1	S(5,4)1	O(3,2)1		
for	O(2,1)3	O(4,2)1	O(4,3)1	O(2,1)1	O(2,1)2	O(4,3)1	O(4,3)1	O(2,1)1			
Scenario 7	O(3,1)1			O(5,3)2	O(3,2)1			O(5,3)1			

Table 7: Optimal solution for the expected value problem.

	Time Period (Year)										
	1	2	3	4	5	6	7	8	9	10	11
Expected Demand	3.67	4.67	4.05	3.05	1.93	1.40	0.80	0.30	0.10	0.03	0.00
Schedule	P(2,1)4 O(2,1)4	P(2,1)2 I(3,2)1 O(2,1)2 O(3,2)3	I(4,3)1 O(3,2)2 O(4,2)1 O(4,3)2	S(5,4)2 O(4,3)2 O(5,3)2	S(5,4)2 S(6,4)2 R 2	R 2	R 1	R 1	R 1	R 1	

Table 8: Machine requirements at each city in each year.

Project	Time Period (Year)										Total
	1	2	3	4	5	6	7	8	9	10	
San Antonio, TX	1	1	3	5	5	3	2				20
Bozeman, MT	3	4	4	5	4	4	2				26
Los Angeles, CA	5	4	4	3	2	1					19
Orlando, FL	1	1	4	4	5	4	3	2			24
Chicago, IL	2	4	4	4	4	1					19
San Francisco, CA	4	5	5	6	7						27

Table 9: Distance matrix between cities.

Project Location	Project Location					
	San Antonio, TX	Bozeman, MT	Los Angeles, CA	Orlando, FL	Chicago, IL	San Francisco, CA
San Antonio, TX	0	1613	1350	1160	1200	1730
Bozeman, MT	1613	0	1100	2397	1388	1000
Los Angeles, CA	1350	1100	0	2500	2000	382
Orlando, FL	1160	2397	2500	0	1124	2813
Chicago, IL	1200	1388	2000	1124	0	2132
San Francisco, CA	1730	1000	382	2813	2132	0

Table 10: Results for the numerical example of six cities.

	Time Period (years)									
	1	2	3	4	5	6	7	8	9	10
Demand	1	1	3	5	5	3	2			
Schedule	P(1,1)1	P(2,1)1	P(2,1)1	P(2,1)2	O(3,2)2	O(4,3)2	O(5,4)2	S(6,5)2		
for San	O(1,1)1	O(2,1)1	O(2,1)1	O(2,1)2	O(4,3)1	O(5,4)1	S(6,5)1			
Antonio,		I(2,2)1	O(3,2)2	O(3,2)1	O(5,4)2	S(6,5)2				
TX				O(4,3)2						
Demand	3	4	4	5	4	4	2			
	P(1,1)1	P(2,1)1	O(3,2)1	P(2,1)2	O(3,2)2	O(4,3)2	O(5,4)2	S(6,5) 2		
Schedule	O(1,1)1	O(2,1)1	O(3,3)1	O(2,1)2	O(5,4)2	O(5,4)2	S(6,5)2			
for	P(2,1)2	O(2,2)1	O(4,3)2	O(4,3)1	S(6,5)2	S(6,5)2				
Bozeman,	O(2,1)2	O(3,2)2		I(4,4)1						
MT				O(5,4)2						
Demand	5	4	4	3	2	1				
Schedule	P(1,1)2	O(2,2)1	O(3,3)1	I(4,4)1	O(4,3)1	O(5,4)1	S(6,5)1			
for Los	O(1,1)2	O(3,2)3	O(4,3)3	O(5,4)3	O(5,4)1	S(6,5)1				
Angeles,	P(2,1)3				S(6,5)3					
CA	O(2,1)3									
Demand	1	1	4	4	5	4	3	2		
	P(2,1)1	P(2,1)1	P(2,1)2	O(3,2)2	P(2,1)2	O(3,2)2	O(4,3)2	O(5,4)2	S(6,5)2	
Schedule	O(2,1)1	O(2,1)1	O(2,1)2	O(4,3)1	O(2,1)2	O(5,4)2	O(5,4)1	S(6,5)1		
for		I(3,2)1	O(3,2)1	O(5,3)1	O(4,3)2	S(6,5)1	S(6,5)2			
Orlando,			O(4,2)1		O(5,4)1					
FL					S(6,4,)1					
Demand	2	4	4	4	4	1				
Schedule	P(1,1)1	P(2,1)2	O(3,2)2	P(2,1)1	O(3,2)1	O(4,3)1				
for	O(1,1)1	O(2,1)2	O(3,3)1	O(2,1)1	O(5,4)3	S(6,5)3				
Chicago,	P(2,1)1	O(2,2)1	O(4,3)1	O(4,3)2	S(6,5)1					
IL	O(2,1)1	O(3,2)1		I(4,4)1						
				O(5,4)1						
Demand	4	5	5	6	7					
Schedule	P(1,1)4	O(2,2)5	P(2,1)3	O(3,2)3	O(4,3)2	S(6,5)5				
for San	O(1,1)4		O(2,1)3	O(4,3)3	O(5,4)5					
Francisco,			I(3,3)3	I(4,4)2						
CA			O(3,3)2							
Shipping Routes		T(2,2) 1 from (3 $\rightarrow$ 6)			T(4,3)1 from (6 $\rightarrow$ 3)	T(5,4)2 from (6 $\rightarrow$ 2)	T(5,4)1 from (5 $\rightarrow$ 4)			

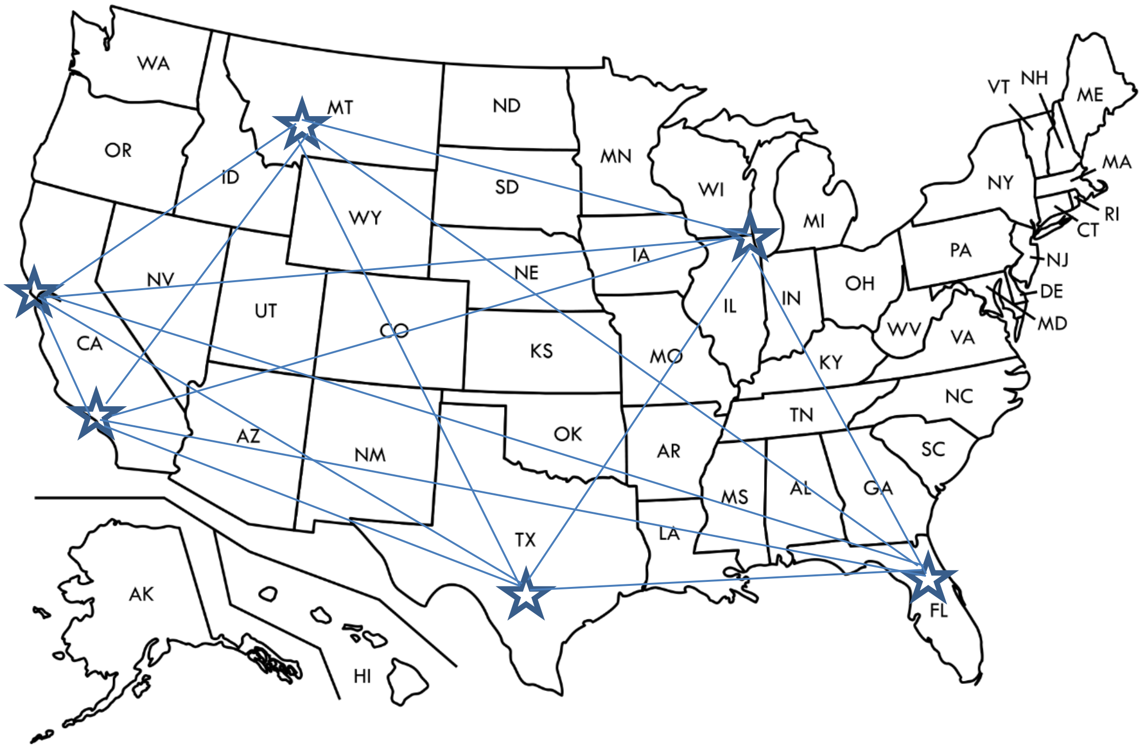

Figure 5: All possible shipping routes between the 6 cities.

Figure 6: Visual representation of the shipping solutions (decisions).

Table 11: The operations each machine has

Operation	Machine Type		
	1	2	3
1	1	0	1
2	0	1	1

Table 12: Holding, Renting, and Purchase costs for various machine types.

Cost	Machine Type		
	1	2	3
Holding	\$5000	\$5000	\$10,000
Renting	\$90,000	\$90,000	\$120,000
Purchase	\$230,000	\$220,000	\$240,000

Table 13: Operations and Maintenance cost for each machine performing a specific operation.

Operation	Machine Type		
	1	2	3
1	\$11,000	/	\$13,000
2	/	\$11,000	\$8800

Table 14: Demand and Optimal Schedule for each operation in each time period.

	Time Period (years)								
	1	2	3	4	5	6	7	8	9
Demand $o = 1$	1	1	3	5	5	3	2	1	
Demand $o = 2$	2	3	3	3	3	1			
Demand Total	3	4	6	8	8	4	2	1	
Optimal Schedule	P(1,1,2)2 P(2,1,1)1 O(2,2)2 O(1,1)1	P(2,1,1)1 P(2,1,3)1 O(3,2)1 O(1,1)1 O(2,2)2 I(3,2,1)1	P(2,1,1)1 O(1,1)1 P(2,1,3)1 O(3,2)1 O(1,1)1 O(3,2)1 O(2,2)1 O(1,1)1 I(3,3,2)1	P(2,1,1)2 O(1,1)1 O(1,1)2 O(3,2)1 O(1,1)1 O(2,2)1 O(3,2)1 O(1,1)1 I(4,4,2)1	O(1,1)1 O(1,1)2 O(3,2)1 O(1,1)1 O(2,2)2 O(3,1)1 S(6,5,3)1	O(1,1)2 O(1,1)1 O(3,2)1 S(6,5,1)1 S(6,5,3)1	O(1,1)2 S(6,5,1)1 S(6,5,3)1	S(6,5,1)2	

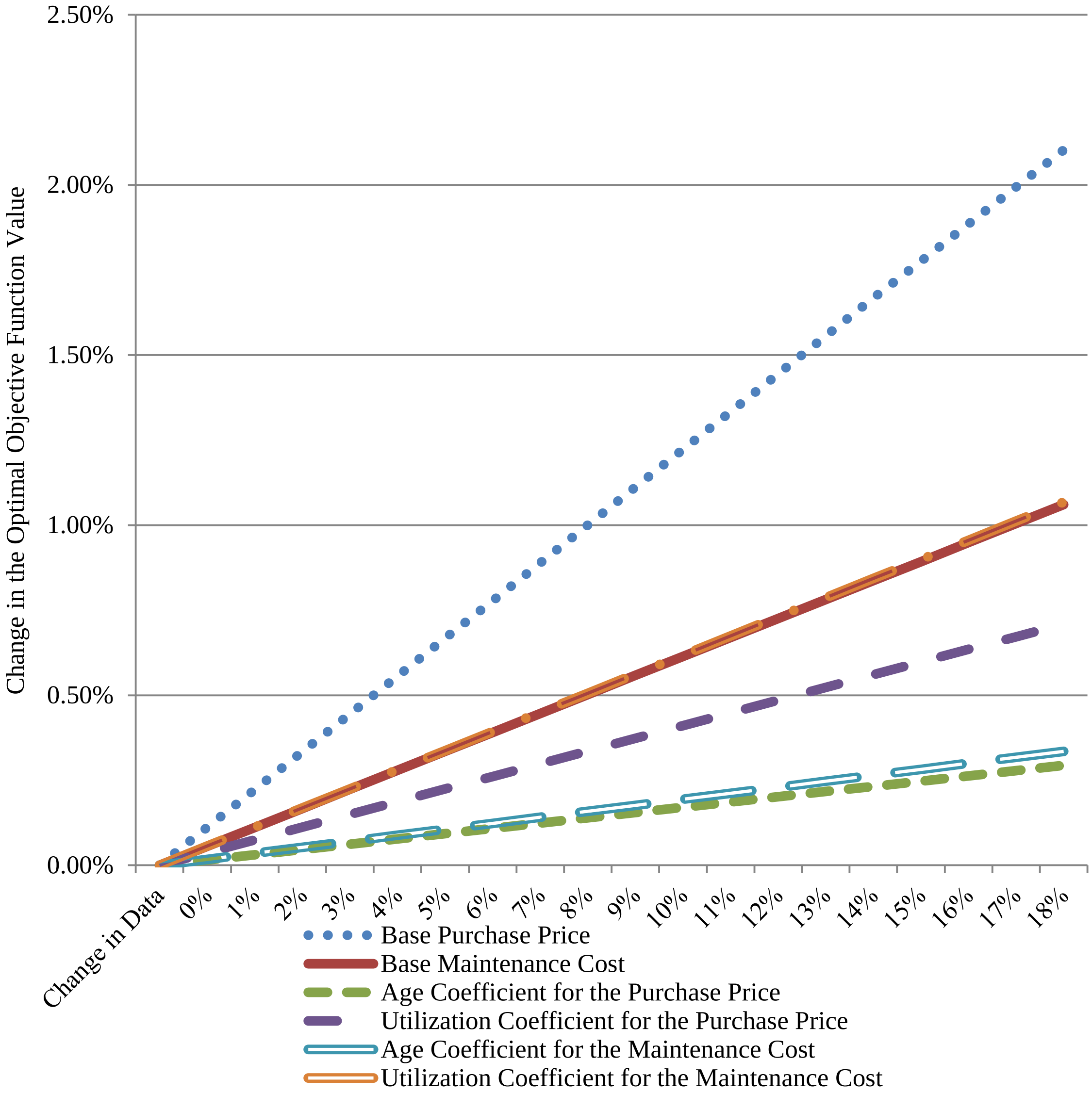

Figure 7: The effects of changing the base prices, age coefficients, and utilization coefficients on the objective function.

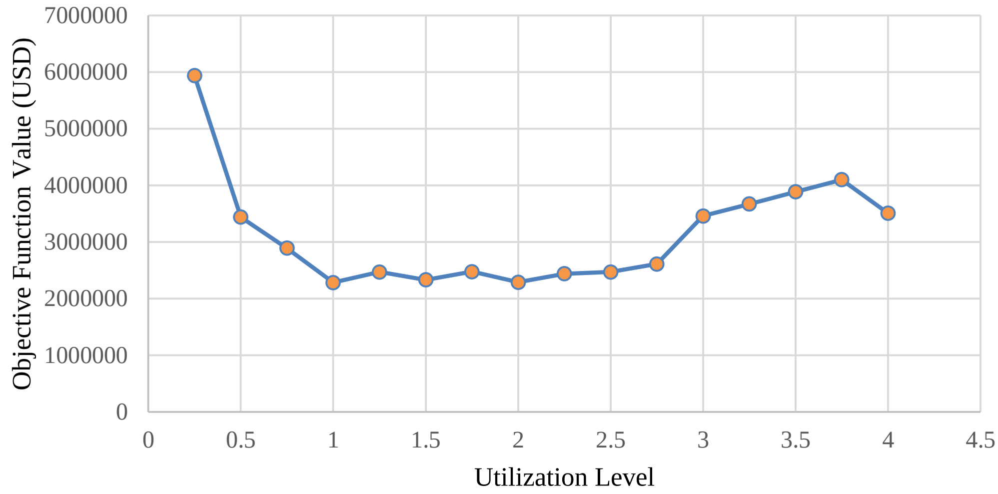

Figure 8: Optimal objective function values (OFVs) at varying utilization levels.

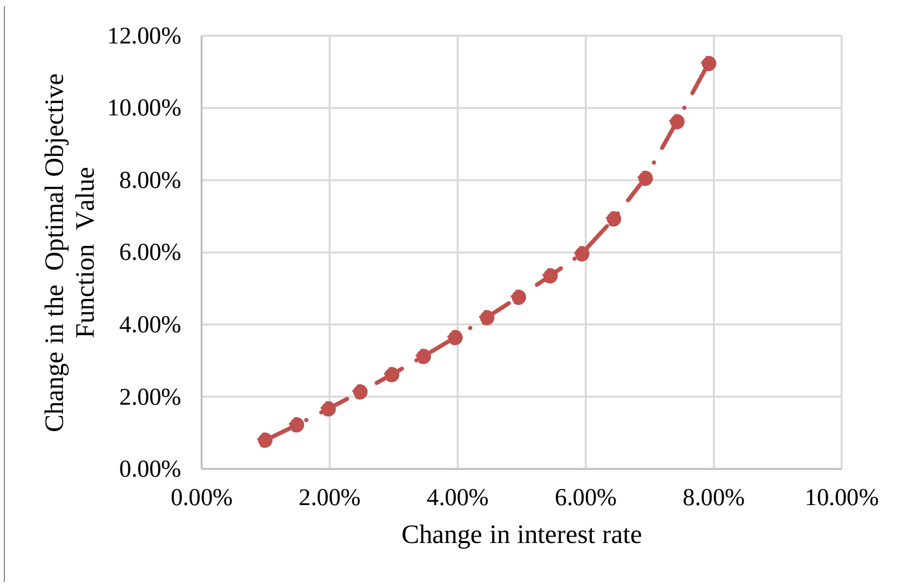

Figure 9: The sensitivity of the OFV to a percentage increase in assumed interest rate.

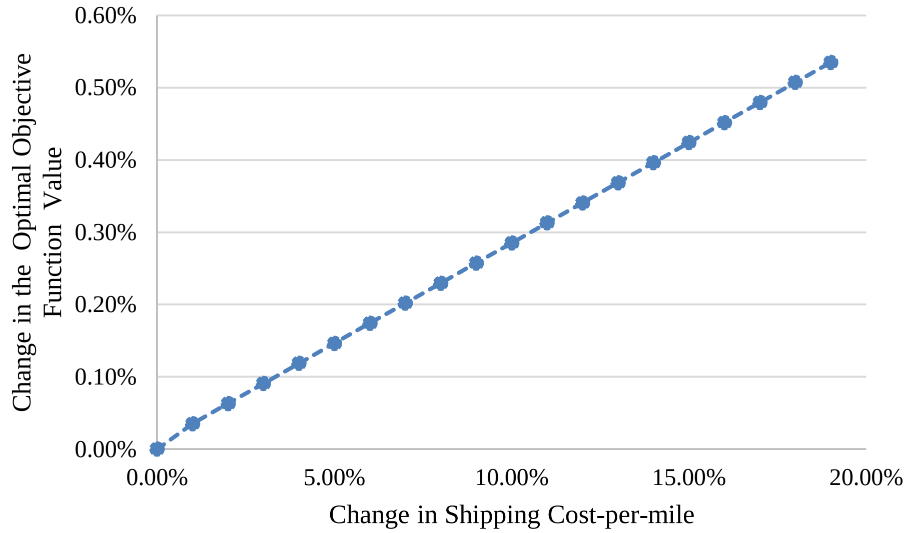

Figure 10: The sensitivity of the OFV to a percentage increase in the shipping cost per mile.

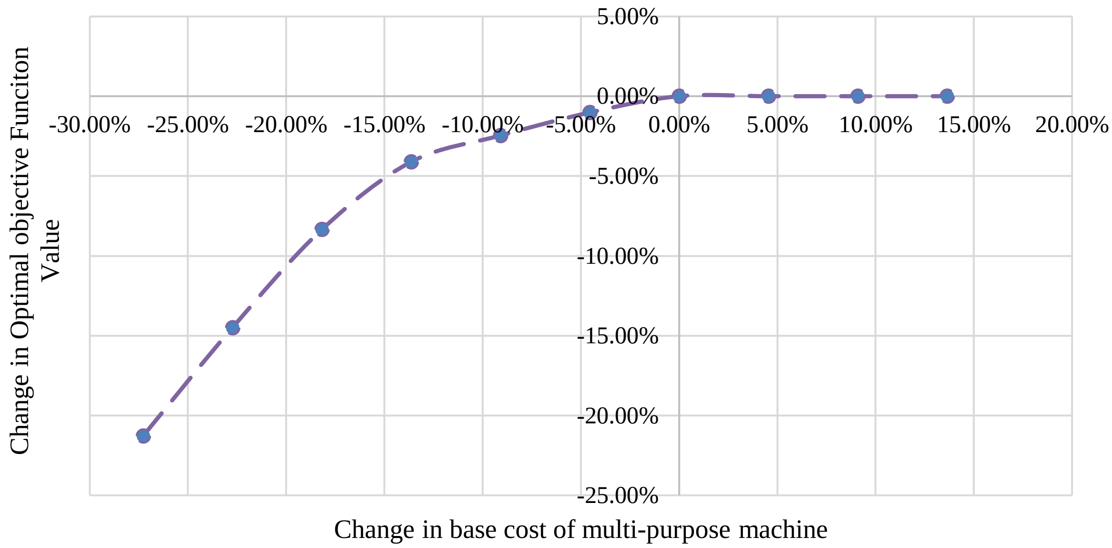

Figure 11: Sensitivity of the cost of the multi-purpose machine OFV.

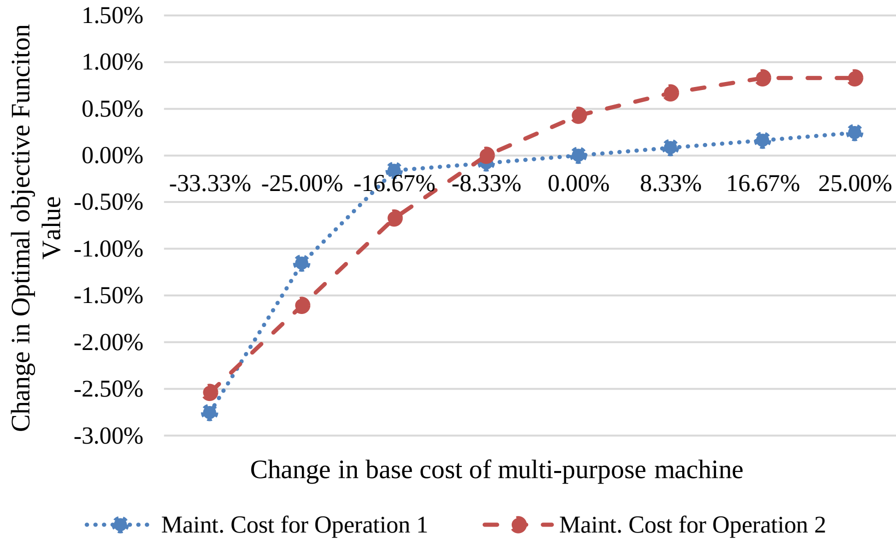

Figure 12: Sensitivity of the maintenance cost of the multi-purpose machine on the optimal OFV.

Table 15: The problem sizes solved for each presented parameter.

	Problem Size			
Number	*P*	*T*	*U*	*L*
1	5	6	72	72
2	5	8	72	72
3	5	12	72	72
4	5	24	72	72
5	5	36	72	72
6	10	6	72	72
7	10	8	72	72
8	10	12	72	72
9	10	36	72	72
10	15	6	72	72
11	15	8	72	72
12	15	12	72	72
13	15	24	72	72
14	15	36	72	72
15	25	6	72	72
16	25	8	72	72
17	25	12	72	72
18	25	24	72	72
19	25	36	72	72

Table 16: Computational results of three solution methods for each defined problem size.

Problem Size	Master Problem Solution Time (s)	Master Problem OFV	CDA Solution Time (s)	CDA OFV	No Shipping Solution Time (s)	No Shipping OFV	Gap 1 (%)	Gap 2 (%)
1	23.0	11487914.7	58.0	11491183.6	11.0	11495498.8	0.03%	0.04%
2	28.0	10761291.0	67.0	10766003.4	34.0	10766003.5	0.04%	0.00%
3	60.0	15717444.0	873.0	15746179.0	353.0	15755033.6	0.18%	0.06%
4	133.0	30162371.2	764.0	30161638.3	170.0	30193235.9	0.00%	0.10%
5	413	38915377.0	145.0	38996430.5	250.0	39021657.5	0.21%	0.06%
6	68.0	16897557.8	63.0	16906242.4	46.0	17128209.4	0.05%	1.31%
7	89.5	16646129.0	177.0	16646128.9	114.0	16946954.0	0.00%	1.81%
8	228.0	23135014.9	1584.0	23167955.5	84.0	23511251.4	0.14%	1.48%
9	N/A	N/A	397.0	62303571.3	N/A	N/A	N/A	N/A
10	150.0	25527551.1	248.0	25527915.7	89.0	26174613.9	0.00%	2.53%
11	214.22	26204783.2	80.3	26208024.3	66.0	27066686.7	0.01%	3.29%
12	349.0	36719271.6	319.3	36719271.6	128.0	37810463.9	0.24%	2.97%
13	N/A	N/A	594.0	71058577.8	N/A	N/A	N/A	N/A
14	N/A	N/A	1184.7	99288922.5	N/A	N/A	N/A	N/A

Table 16: Computational results of three solution methods for each defined problem size (Continued).

Problem Size	Master Problem Solution Time (s)	CDA OFV	No Shipping Solution Time (s)	OFV	Solution Time (s)	OFV	Gap 1 (%)	Gap 2 (%)
16	N/A	N/A	222.5	44030475.8	135.9	45266770	N/A	2.81%
17	N/A	N/A	746.8	61392672.4	N/A	N/A	N/A	N/A
18	N/A	N/A	1870.3	119007914.2	N/A	N/A	N/A	N/A
19	N/A	N/A	1478.1	261370244.6	N/A	N/A	N/A	N/A

Appendix B: Computer Codes

The presented codes were developed in Python.

Base Model

```
from gurobipy import *
import matplotlib.pyplot as mpl
import numpy as np
import math as math
import xlsxwriter


workbook = xlsxwriter.Workbook('17B-Costs.xlsx')
worksheet = workbook.add_worksheet('costftn')
   # Add a bold format to use to highlight cells.
bold = workbook.add_format({'bold': 1})
underline = workbook.add_format({'underline': 1})
   # Add a number format for cells with money.
res_format = workbook.add_format({'num_format': '0.000000'})
bad_res_format = workbook.add_format({'num_format':'0.000000','color':
              '#FF0000','bold':1})
   # Add an Excel date format.
date_format = workbook.add_format({'num_format': 'mmmm d yyyy'})
   # coloring
blue_format = workbook.add_format({'color': '#0000FF'})
red_format = workbook.add_format({'color': '#FF0000'})
green_format = workbook.add_format({'color': '#008000'})



row = 1
```

```
L = 6
U = 6
#20000 hours is the life
# each year: 2000 hours of work
# UtilFactor range: [0.2,4]
W=1
T=[5,6,7,8,9,10,11]

c = {}
for i in range(L):
    for j in range(U):
        for t in range(max(T)):
            if i == 0:
                c[i,j,t] = int((220000 - 12000*j)*(1.03)**t)
            else:
                c[i,j,t] = int((c[0,j,t]*(.8) - 3000*i)*(1.03)**t)
            # print c[i,j,t] .577 - hartman

m = {}

for i in range(L):
    for j in range(U):
        for t in range(max(T)):
            m[i,j,t] = int((10000 + 500*i + 5000*j*((j+1)**1.2) - j**1.2)*(1.03)**t)

s = {}
for t in range(max(T)):
    for i in range(L):
```

```
        for j in range(U):
            s[i,j,t] = (.57)*(c[i,j,t])


# cap = {}
# for t in range(max(T)):
#     for w in range(W):
#         cap[t,w] = 2
#
# from mpl_toolkits.mplot3d import Axes3D
# fig = mpl.figure(figsize=(7,6))   # This creates a new figure object.
# ax = fig.add_subplot(111, projection='3d')   # This adds a subplot to the figure with 3D
                projection, and returns the axes object.
# x, y = np.linspace(0.0, L, L), np.linspace(0.0, U, U)   # Create the 2D space
# X, Y = np.meshgrid(x, y)  # Get the plaid version (the 'meshgrid' version, similar to
                Matlab's meshgrid function)
# z = (200000*(.577) - 1000*X - 5000*(Y**.8))*(1.03)**3
# csf = ax.contourf(X, Y, z, 15)
# cs = ax.contour(X, Y, z, 15, cmap=mpl.cm.Oranges_r)
# csl = ax.clabel(csf, fmt='%2.1f', colors='k', fontsize=14)
# cbar = mpl.colorbar(csf)
# mpl.show()
#
q = {}
for t in range(max(T)):
    q[t] = 90000*1.03**t


h = {}
for t in range(max(T)):
    h[t] = 5000*1.03**t
```

```
o = {}
for t in range(max(T)):
    o[t] = 50000*1.03**t

r = {}
r[ 0 , 0 ]= 4
r[ 0 , 1 ]= 6
r[ 0 , 2 ]= 6
r[ 0 , 3 ]= 4
r[ 0 , 4 ]= 0
r[ 0 , 5 ]= 0
r[ 0 , 6 ]= 0
r[ 0 , 7 ]= 0
r[ 0 , 8 ]= 0
r[ 0 , 9 ]= 0
r[ 0 , 10 ]= 0

r[ 1 , 0 ]= 4
r[ 1 , 1 ]= 6
r[ 1 , 2 ]= 5
r[ 1 , 3 ]= 3
r[ 1 , 4 ]= 2
r[ 1 , 5 ]= 0
r[ 1 , 6 ]= 0
r[ 1 , 7 ]= 0
r[ 1 , 8 ]= 0
r[ 1 , 9 ]= 0
r[ 1 , 10 ]= 0
```

r[2 , 0]= 4

r[2 , 1]= 5

r[2 , 2]= 4

r[2 , 3]= 3

r[2 , 4]= 2

r[2 , 5]= 2

r[2 , 6]= 0

r[2 , 7]= 0

r[2 , 8]= 0

r[2 , 9]= 0

r[2 , 10]= 0

r[3 , 0]= 4

r[3 , 1]= 4

r[3 , 2]= 3

r[3 , 3]= 3

r[3 , 4]= 2

r[3 , 5]= 2

r[3 , 6]= 2

r[3 , 7]= 0

r[3 , 8]= 0

r[3 , 9]= 0

r[3 , 10]= 0

r[4 , 0]= 3

r[4 , 1]= 3

r[4 , 2]= 4

r[4 , 3]= 3

r[4 , 4]= 2

r[4 , 5]= 2

r[4 , 6]= 2

r[4 , 7]= 1

r[4 , 8]= 0

r[4 , 9]= 0

r[4 , 10]= 0

r[5 , 0]= 2

r[5 , 1]= 3

r[5 , 2]= 3

r[5 , 3]= 3

r[5 , 4]= 2

r[5 , 5]= 2

r[5 , 6]= 2

r[5 , 7]= 2

r[5 , 8]= 1

r[5 , 9]= 0

r[5 , 10]= 0

r[6 , 0]= 1

r[6 , 1]= 2

r[6 , 2]= 3

r[6 , 3]= 3

r[6 , 4]= 3

r[6 , 5]= 2

r[6 , 6]= 2

r[6 , 7]= 2

r[6 , 8]= 1

```
r[ 6 , 9 ]= 1
r[ 6 , 10 ]= 0

# r[ 0 , 0 ]= 2
# r[ 0 , 1 ]= 2
# r[ 0 , 2 ]= 2
# r[ 0 , 3 ]= 2
# r[ 0 , 4 ]= 2
# r[ 0 , 5 ]= 2
# r[ 0 , 6 ]= 2
# r[ 0 , 7 ]= 2
# r[ 0 , 8 ]= 2
# r[ 0 , 9 ]= 2
# r[ 0 , 10 ]= 0

p = {}
p[0] = 0.05
p[1] = 0.25
p[2] = 0.30
p[3] = 0.20
p[4] = 0.10
p[5] = 0.07
p[6] = 0.03

# R={}
# for t in range(10):
#     R[t] = sum([float(p[w]*r[w,t]) for w in range(W)])

model = Model("17B")
```

```
x={}
for i in range(L):
    for j in range(U):
        x[i,j] = model.addVar(vtype='I',lb=0,ub=GRB.INFINITY)

y={}
for i in range(L):
    for j in range(U):
        for w in range(W):
            for t in range(max(T)):
                y[i,j,t,w] = model.addVar(vtype='I',lb=0,ub=GRB.INFINITY)

z = {}
for w in range(W):
    for t in range(max(T)):
        z[t,w] = model.addVar(vtype='I',lb=0, ub=GRB.INFINITY)
Z = model.addVar(vtype='I',lb=0, ub=GRB.INFINITY)

S={}
for i in range(L):
    for j in range(U):
        for w in range(W):
            for t in range(max(T)):
                S[i,j,t,w] = model.addVar(vtype='I',lb=0,ub=GRB.INFINITY)

I={}
for i in range(L):
    for j in range(U):
```

```
        for w in range(W):
            for t in range(max(T)):
                I[i,j,t,w] = model.addVar(vtype='I',lb=0,ub=GRB.INFINITY)

a={}
for i in range(L):
    for j in range(U):
        for w in range(W):
            for t in range(max(T)):
                a[i,j,t,w] = model.addVar(vtype='I',lb=0,ub=GRB.INFINITY)
# E={}
# for w in range(W):
#     for t in range(max(T)):
#         E[t,w] = model.addVar(vtype='I',lb=0,ub=GRB.INFINITY)

model.update()

# # model.addConstr(x[0,0]==3)
# # model.addConstr(x[1,8]==1)
# # model.addConstr(Z==1)
# # model.addConstr(quicksum(x[i,j] for i in range(L) for j in
                  range(youmotherfucker))==4)
#EEV Stochastic Programming
# cx1={}
# for i in range(L):
#     for j in range(U):
#         model.addConstr(x[1,0] == 4)
# cx2={}
# model.addConstr(quicksum(x[i,j] for i in range(L) for j in range(U)) == 4)
```

```
c2={}
for w in range(W):
   for t in range(max(T)):
      # c2[w,t]=model.addConstr(z[t,w] + quicksum(a[i,j,t,w] for i in range(L) for j in
              range(U)) >= r[w,t],name='c2_'+str(w)+"-"+str(t))
      # EV        ## model.addConstr(z[t,w]==1)
      c2[w,t]=model.addConstr(z[t,w] + quicksum(a[i,j,t,w] for i in range(L) for j in
              range(U)) >= quicksum(p[ww]*r[ww,t] for ww in
              range(7)),name='c2_'+str(w)+"-"+str(t))


c3={}
for w in range(W):
   for t in range(max(T)):
      for i in range(L):
         c3[w,t,i] = model.addConstr( a[i,U-1,t,w] == 0 ,
              name="c3_"+str(w)+"_"+str(t)+"_"+str(i))
c4={}
for w in range(W):
   for t in range(max(T)):
      for j in range(U):
         c4[w,t,j] = model.addConstr( a[L-1,j,t,w] == 0 )
c5={}
c52={}
for w in range(W):
   for i in range(L-1):
      for j in range(U-1):
         c5[w,i,j] = model.addConstr(a[i,j,0,w]+I[i,j,0,w]+S[i,j,0,w] == y[i,j,0,w])
```

```
            c52[w,i,j] = model.addConstr(y[i,j,0,w] == x[i,j])
c6={}
for w in range(W):
    for t in range(1,max(T)):
        for j in range(1,U):
            for i in range(1,L):
                c6[w,t,j,i] = model.addConstr(a[i,j,t,w]+I[i,j,t,w]+S[i,j,t,w] == y[i,j,t,w]+a[i-1,j-
                  1,t-1,w]+I[i-1,j,t-1,w])
c62={}
for w in range(W):
    for t in range(1,max(T)):
            for i in range(1,L):
                c62[w,t,0,i] = model.addConstr(a[i,0,t,w]+I[i,0,t,w]+S[i,0,t,w] ==
                  y[i,0,t,w]+I[i-1,0,t-1,w])
c7={}
for w in range(W):
    for t in range(1,max(T)):
        for j in range(U):
            c7[w,t,j] = model.addConstr(a[0,j,t,w]+I[0,j,t,w]+S[0,j,t,w] == y[0,j,t,w])
c8={}
for w in range(W):
    for t in range(1,max(T)):
        for i in range(L):
            c8[w,t,i] = model.addConstr(a[i,0,t,w]+I[i,0,t,w]+S[i,0,t,w] == y[i,0,t,w])

c9={}
for w in range(W):
    for t in range(1,max(T)):
        for j in range(1,U):
```

```
            c9[w,t,j] = model.addConstr( S[L-1,j,t,w] == y[L-1,j,t,w] + a[L-2,j-1,t-1,w] + I[L-
                2,j,t-1,w] )

c10={}
for w in range(W):
    for t in range(1,max(T)):
        for i in range(1,L):
            c10[w,t,i] = model.addConstr( S[i,U-1,t,w] == y[i,U-1,t,w] + a[i,U-2,t-1,w] )
c11={}
for w in range(W):
    for j in range(1,U):
        c11[w,j] = model.addConstr( S[L-1,j,0,w] == y[L-1,j,0,w] )
c12={}
for w in range(W):
    for i in range(1,L):
        c12[w,i] = model.addConstr( S[i,U-1,0,w] == y[i,U-
                1,0,w],name="c_12_"+str(w)+"_"+str(i) )
c13={}
c14={}
for i in range(L):
    for w in range(W):
        c13[i] = model.addConstr( S[i,0,0,w] == 0 )
for j in range(U):
    for w in range(W):
        c14[j] = model.addConstr( S[0,j,0,w] == 0 )
c15={}
for w in range(W):
    c15[w,t] = model.addConstr(z[0,w] == Z)
```

```
c16={}
for w in range(W):
    c16[w] = model.addConstr(quicksum(S[i,j,t,w] for t in range(T[w],max(T)) for i in
            range(L) for j in range(U))==0)

# c17={}
# for w in range(W):
#     for t in range(max(T)):
#         c17[t,w] = model.addConstr(E[t,w] >= quicksum(a[i,j,t,w] for i in range(L) for j in
            range(U)) - cap[t,w])

# Fix a solution in the model
# if isFixed==1:
#     for i in range(L):
#         for j in range(U):
#             model.addConstr(x[i,j]==xx[i,j])
#             for w in range(W):
#                 for t in range(max(T)):
#                     model.addConstr(y[i,j,t,w]==yy[i,j,t,w])
#                     model.addConstr(I[i,j,t,w]==II[i,j,t,w])
#                     model.addConstr(a[i,j,t,w]==aa[i,j,t,w])
#                     model.addConstr(S[i,j,t,w]==SS[i,j,t,w])
#     for t in range(max(T)):
#         for w in rangr(W):
#             model.addConstr(z[t,w]==zz[t,w])


model.update()
```

```
objective =  0 #Z*q+Z*o + quicksum((c[i,j,0]+o+m[i,j,0])*x[i,j] for i in range(L) for j in
            range(U))
for w in range(W):
   for t in range(max(T)):
      objective += (1/1.03)**t*float(p[w])*(q[t]*z[t,w] + o[t]*z[t,w] +
            quicksum(o[t]*a[i,j,t,w] + c[i,j,t]*y[i,j,t,w] + m[i,j,t]*a[i,j,t,w] + h[t]*I[i,j,t,w]
            - s[i,j,t]*S[i,j,t,w] for i in range(L) for j in range(U)))
      #float(p[w])*

model.setObjective(objective)
model.setParam("MIPGap",0.00)
model.modelSense = GRB.MINIMIZE
model.update()
model.optimize()
print("model status is:",model.status)
if model.status==GRB.OPTIMAL:
   print("Optimal", model.objVal)
   Jimmy = model.objVal
   xx={}
   yy={}
   SS={}
   II={}
   aa={}
   zz={}

   for i in range(L):
      for j in range(U):
         xx[i,j] = x[i,j].x
         if x[i,j].x>0:
```

```
            print("X",i,j,x[i,j].x)
for w in range(W):
    print("****SCENARIO ",w)
    for t in range(max(T)):
        zz[t,w] = z[t,w].x
        print("     ****TIME PERIOD ",t)
        expy = 0
        for ww in range(7):
            expy += p[ww]*r[ww,t]
        print("expected demand",expy)
        if z[t,w].x>0:
            print("R",z[t,w].x)
        for i in range(L):
            for j in range(U):
                yy[i,j,t,w] = y[i,j,t,w].x
                SS[i,j,t,w] = S[i,j,t,w].x
                II[i,j,t,w] = I[i,j,t,w].x
                aa[i,j,t,w] = a[i,j,t,w].x
                if y[i,j,t,w].x>0:
                    print("P(",i+1,",",j+1,")",y[i,j,t,w].x)
                if S[i,j,t,w].x>0:
                    print("S(",i+1,",",j+1,")",S[i,j,t,w].x)
                if I[i,j,t,w].x>0:
                    print("I(",i+1,",",j+1,")",I[i,j,t,w].x)
                if a[i,j,t,w].x>0:
                    print("O(",i+1,",",j+1,")",a[i,j,t,w].x)

# for w in range(W):
#     for t in range(max(T)):
```

```
    #         zz[t,w] = z[t,w].x
    #         if z[t,w].x>0:
    #             print("z",t,w,z[t,w].x)

    row=1

    worksheet.write('A1', 'i', bold)
    worksheet.write('B1', 'j', bold)
    worksheet.write('D1', 'Cost', bold)
    worksheet.write('E1', 'O&M', bold)
    for i in range(L):
        for j in range(U):
            for t in range(1):
                worksheet.write_number(row, 0, i)
                worksheet.write_number(row, 1, j)
                worksheet.write_number(row, 3, c[i,j,t])
                worksheet.write_number(row, 4, m[i,j,t])
                row += 1

    workbook.close()
    # return (xx,yy,SS,II,aa,zz,Jimmy)
else:
    model.computeIIS()
#   Print the names of all of the constraints in the IIS set.
    print("--------IIS CONSTRAINTS")
    for c in model.getConstrs():
        if c.IISConstr > 0:
            print(c.ConstrName)
#   Print the names of all of the variables in the IIS set.
```

```
    print("--------IIS VARIABLES")
    for v in model.getVars():
        if v.IISLB > 0 or v.IISUB > 0:
            print(v.VarName)
```

Multiple Operation Types Model

```
import numpy as np
import scipy.stats as stats
import math as mt
import random as random
import openpyxl
from random import randint
import os

M = 3
O = 2
L = 6
U = 6

#22500 hours is the life
# each year: 1500 hours of work
# T=[4,8,10]
TT = 10
for SENSE in
        [200000,220000,230000,240000,250000,300000,350000,400000,1000000]:
        #[1000, 10000,20000,170000,35000,50000,62000,75000, 88000, 100000]
    print("SENSE
        IS:____________________________________________________________
        _",SENSE)
```

```
c = {}
interest = 1.03
for i in range(L):
    for j in range(U):
        for t in range(TT):
            for k in range(M):
                if i == 0:
                    c[i,j,t,0] = int((220000 -12000*mt.ceil(j/12))*(interest)**t) #12000
              DIVIDE BY TIME CHANGE mt.ceil(j/12)
                    c[i,j,t,1] = int((220000 -12000*mt.ceil(j/12))*(interest)**t) #12000
              DIVIDE BY TIME CHANGE mt.ceil(j/12)
                    c[i,j,t,2] = int((SENSE -12000*mt.ceil(j/12))*(interest)**t) #12000
              DIVIDE BY TIME CHANGE mt.ceil(j/12)
                else:
                    c[i,j,t,0] = int((.8*c[0,j,t,0] - 3000*mt.ceil(i/12))*(interest)**t) #3000
                    c[i,j,t,1] = int((.8*c[0,j,t,0] - 3000*mt.ceil(i/12))*(interest)**t) #3000
                    c[i,j,t,2] = int((.8*c[0,j,t,0] - 3000*mt.ceil(i/12))*(interest)**t) #3000


ex = {}
for k in range(M):
    for o in range(O):
        if k == 0 and o == 0:
            ex[k,o] = 3
        elif k == 0 and 0 == 1:
            ex[k,o] = 3
        elif k == 1 and 0 == 0:
            ex[k,o] = 3
        elif k == 1 and 0 == 1:
```

```
            ex[k,o] = 3
        elif k == 2 and 0 == 0:
            ex[k,o] = 3
        else:
            ex[k,o] = 0

q = {}
for k in range(M):
    for t in range(TT):
        if k == 0:
            q[k,t] = 100000000 #100000*(1.03)**t
        else:
            q[k,t] = 100000000 #75000*(1.03)**t

h = {}
for k in range(M):
    for t in range(TT):
        if k == 0:
            h[k,t] = 1000*(interest)**t
        elif k == 1:
            h[k,t] = 1000*(interest)**t
        else:
            h[k,t] = 1000*(interest)**t

om = {}
for k in range(M):
    for t in range(TT):
        for o in range(O):
            if k == 0 and o == 0:
```

```
                om[k,t,o] = 5
            elif k == 1 and 0 == 1:
                om[k,t,o] = 6
            elif k == 2 and 0 == 0:
                om[k,t,o] = 4
            elif k == 2 and 0 == 1:
                om[k,t,o] = 3
            else:
                om[k,t,o] = 0

m = {}
for i in range(L):
    for j in range(U):
        for t in range(TT):
            for k in range(M):
                for o in range(O):
                    if k == 0:
                        m[i,j,k,t,0] = int((10000 + 500*mt.ceil(i/12) +
              5000*mt.ceil(j/12)*((mt.ceil(j/12)+1)**1.2) -
              mt.ceil(j/12)**1.2)*(interest)**t)
                        m[i,j,k,t,1] = int((10000 + 500*mt.ceil(i/12) +
              5000*mt.ceil(j/12)*((mt.ceil(j/12)+1)**1.2) -
              mt.ceil(j/12)**1.2)*(interest)**t)
                    elif k == 1:
                        m[i,j,k,t,0] = int((10000 + 500*mt.ceil(i/12) +
              5000*mt.ceil(j/12)*((mt.ceil(j/12)+1)**1.2) -
              mt.ceil(j/12)**1.2)*(interest)**t)
```

```
                m[i,j,k,t,1] = int((10000 + 500*mt.ceil(i/12) +
            5000*mt.ceil(j/12)*((mt.ceil(j/12)+1)**1.2) -
            mt.ceil(j/12)**1.2)*(interest)**t)
              else:
                m[i,j,k,t,0] = int((10000 + 500*mt.ceil(i/12) +
            5000*mt.ceil(j/12)*((mt.ceil(j/12)+1)**1.2) -
            mt.ceil(j/12)**1.2)*(interest)**t)
                m[i,j,k,t,1] = int((10000 + 500*mt.ceil(i/12) +
            5000*mt.ceil(j/12)*((mt.ceil(j/12)+1)**1.2) -
            mt.ceil(j/12)**1.2)*(interest)**t)



s = {}
for t in range(TT):
    for i in range(L):
        for j in range(U):
            for k in range(M):
                s[i,j,t,0] = (.57)*(c[i,j,t,0])
                s[i,j,t,1] = (.57)*(c[i,j,t,1])
                s[i,j,t,2] = (.57)*(c[i,j,t,2])
                 #int((.75)*(c[i,j,k,t]) - 500*i - 1500*(j**.8))
                # elif t >= i and k == 1:
                #     s[i,j,k,t] = int((.75)*(c[i,j,k,t-i]) - 5000*i - 1500*(j**.8))
                # elif t < i and k== 0:
                #     s[i,j,k,t] = int((.75)*(c[i,j,k,t-i]) - 5000*i - 1500*(j**.8))
                # else:
                #     s[i,j,k,t] = int((.75)*(c[i,j,k,t-i]) - 5000*i - 1500*(j**.8))
```

```
r={}
wb = openpyxl.load_workbook('25ProjectsMM.xlsx')
type(wb)
wb.get_sheet_names()
sheet = wb.get_sheet_by_name('Projects1')
for t in range(TT):
    for o in range(O):
        r[t,o] = (sheet.cell(row=o+2,column=t+2).value)
        # r[w,t,p] =
            randint((sheet.cell(row=p+2,column=t+2).value),(sheet.cell(row=p+2,column
            =t+2).value)+2)
        # print("Demand is:  ",t,p,r[w,t,p])


u = {}
# for k in range(M):
#     for o in range(O):
#         u[k,o] = random.randint(0,1)
u[0,0] = 1
u[0,1] = 0

u[1,0] = 0
u[1,1] = 1

u[2,0] = 1
u[2,1] = 1
```

```
# pr = {}
# pr[0] = 1 #0.05
# pr[1] = 0.68
# pr[2] = 0.27
# for w in range(W):
#     p[w] = 0.01  #CHECK THIS

# R={}
# for t in range(TT):
#     for o in range(O):
#         R[t,o] = sum([int(pr[t,o])])




from gurobipy import *
model = Model("17B")

x={}
for i in range(L):
    for j in range(U):
        for k in range(M):
            x[i,j,k] = model.addVar(vtype='I',lb=0,ub=GRB.INFINITY)

y={}
for i in range(L):
    for j in range(U):
        for t in range(TT):
            for k in range(M):
                y[i,j,t,k] = model.addVar(vtype='I',lb=0,ub=GRB.INFINITY)
```

```
z = {}
for t in range(TT):
    for k in range(M):
        for o in range(O):
            z[t,k,o] = model.addVar(vtype='I',lb=0, ub=GRB.INFINITY)
Z = model.addVar(vtype='I',lb=0, ub=GRB.INFINITY)

S={}
for i in range(L):
    for j in range(U):
        for t in range(TT):
            for k in range(M):
                S[i,j,t,k] = model.addVar(vtype='I',lb=0,ub=GRB.INFINITY)

I={}
for i in range(L):
    for j in range(U):
        for t in range(TT):
            for k in range(M):
                I[i,j,t,k] = model.addVar(vtype='I',lb=0,ub=GRB.INFINITY)

a={}
for i in range(L):
    for j in range(U):
        for t in range(TT):
            for k in range(M):
                for o in range(O):
                    a[i,j,t,k,o] = model.addVar(vtype='I',lb=0,ub=GRB.INFINITY)
```

```
# O = {}
# for w in range(W):
#     for t in range(T[w]):
#         O[t,w] = model.addVar(vtype='I',lb=0, ub=GRB.INFINITY)
model.update()


# model.addConstr(x[0,0]==3)
# model.addConstr(x[1,8]==1)
# model.addConstr(Z==1)
# model.addConstr(quicksum(x[i,j] for i in range(L) for j in range(L))==4)
#

# model.addConstr(x[1,8,1] == 1)
# model.addConstr(x[5,6,1] == 2)
# model.addConstr(quicksum(x[i,j,k] for i in range(L) for j in range(L) for k in
           range(M)) == 3)
c1={}
for t in range(TT):
   for o in range(O):
      c1[t,o]=model.addConstr(quicksum(u[k,o]*(z[t,k,o] + quicksum(a[i,j,t,k,o] for i in
           range(L) for j in range(U))) for k in range(M)) >= r[t,o],name='c1_'+str(t)+"-
           "+str(o))
#         model.addConstr(z[t,w]==1)

c11={}
for t in range(TT):
```

```
    c11[t]=model.addConstr(quicksum(z[t,k,o] for k in range(M) for o in range(O)) +
             quicksum(a[i,j,t,k,o] for i in range(L) for j in range(U) for k in range(M) for o
             in range(O)) >= quicksum(r[t,o] for o in range(O)),name='c11_'+str(t))
#         model.addConstr(z[t,w]==1)

c12={}
for t in range(TT):
    for k in range(M):
        c12[t,k]=model.addConstr(quicksum(z[t,k,o] for o in range(O)) +
             quicksum(a[i,j,t,k,o] for i in range(L) for j in range(L) for o in range(O)) <=
             quicksum(u[k,o]*r[t,o] for o in range(O)),name='c12_'+str(t)+"_"+str(k))
#         # model.addConstr(z[t,w]==1)

c2={}
for t in range(TT):
    for i in range(L):
        for k in range(M):
            for o in range(O):
                c2[t,i,k,o] = model.addConstr(a[i,U-1,t,k,o] == 0)
c3={}
for t in range(TT):
    for j in range(U):
        for k in range(M):
            for o in range(O):
                c3[t,j,k] = model.addConstr(a[L-1,j,t,k,o] == 0)
c4={}
for t in range(1,TT):
    for j in range(1,U):
        for i in range(1,L):
```

```
            for k in range(M):
                c4[t,j,i,k] = model.addConstr(quicksum(a[i,j,t,k,o] for o in
              range(O))+I[i,j,t,k]+S[i,j,t,k] == y[i,j,t,k]+quicksum(a[i-1,j-1,t-1,k,o] for o in
              range(O))+I[i-1,j,t-1,k])
c5={}
c6={}
for i in range(L-1):
    for j in range(U-1):
        for k in range(M):
            for o in range(O):
                c5[i,j,k] = model.addConstr(quicksum(a[i,j,0,k,o] for o in
              range(O))+I[i,j,0,k]+S[i,j,0,k] == y[i,j,0,k])
                c6[i,j,k] = model.addConstr(y[i,j,0,k] == x[i,j,k])
c7={}
for t in range(1,TT):
    for j in range(U):
        for k in range(M):
            c7[t,j,k] = model.addConstr(quicksum(a[0,j,t,k,o] for o in
              range(O))+I[0,j,t,k]+S[0,j,t,k] == y[0,j,t,k])
c8={}
for t in range(1,TT):
    for i in range(L):
        for k in range(M):
            for o in range(O):
                c8[t,i,k] = model.addConstr(quicksum(a[i,0,t,k,o] for o in
              range(O))+I[i,0,t,k]+S[i,0,t,k] == y[i,0,t,k])

c9={}
for t in range(1,TT):
```

```
    for j in range(1,U):
        for k in range(M):
            for o in range(O):
                c9[t,j,k] = model.addConstr( S[L-1,j,t,k] == y[L-1,j,t,k] + quicksum(a[L-2,j-
              1,t-1,k,o] for o in range(O)) + I[L-2,j,t-1,k] )
c10={}
for t in range(1,TT):
    for i in range(1,L):
        for k in range(M):
            for o in range(O):
                c10[t,i,k] = model.addConstr( S[i,U-1,t,k] == y[i,U-1,t,k] + quicksum(a[i,U-
              2,t-1,k,o] for o in range(O)) )
c11={}
for j in range(1,U):
    for k in range(M):
        c11[j,k] = model.addConstr( S[L-1,j,0,k] == y[L-1,j,0,k] )
c12={}
for i in range(1,L):
    for k in range(M):
        c12[i,k] = model.addConstr( S[i,L-1,0,k] == y[i,L-
              1,0,k],name="c_12_"+str(i)+"_"+str(k))


c13={}
c14={}
for i in range(L):
    for k in range(M):
        c13[i,k] = model.addConstr( S[i,0,0,k] == 0 )
        c14[i,k] = model.addConstr( S[0,i,0,k] == 0 )
c15={}
```

```
for k in range(M):
    for o in range(O):
        c15[t,k] = model.addConstr(z[0,k,o] == Z)

# calg = {}
# for j in range(U):
#     for i in range(L-1):
#         for k in range(M):
#             for t in range(TT):
#                 model.addConstr(S[i,j,t,k] == 0)
model.update()




objective =0# Z*q+Z*o + quicksum((c[i,j]+o+m[i,j,t])*x[i,j] for i in range(L) for j in
            range(L))
for t in range(TT):
    for l in range(M):
        objective += quicksum(q[l,t]*z[t,l,o] for o in range(O)) +
            quicksum(c[i,j,t,l]*y[i,j,t,l] + quicksum((m[i,j,l,t,o]+ om[l,t,o] +
            ex[l,o])*(a[i,j,t,l,o]) for o in range(O)) + h[l,t]*I[i,j,t,l] - s[i,j,t,l]*S[i,j,t,l] for i
            in range(L) for j in range(U))

model.setObjective(objective)
model.setParam("MIPGap",0.00)
model.modelSense = GRB.MINIMIZE
model.update()
model.optimize()
```

```
print("model status is:",model.status)
if model.status==GRB.OPTIMAL:
    print("Optimal", model.objVal)
    for i in range(L):
        for j in range(U):
            for k in range(M):
                if x[i,j,k].x>0:
                    print("X",i+1,j+1,k+1,x[i,j,k].x)

    for t in range(TT):
        print("     ****TIME PERIOD ",t)
        for o in range(O):
            print("DEMAND of Operation",o+1,"is",r[t,o])
        for i in range(L):
            for j in range(U):
                for k in range(M):
                    if y[i,j,t,k].x>0:
                        print("P(",i+1,j+1,t+1,k+1,")",y[i,j,t,k].x)
                    if S[i,j,t,k].x>0:
                        print("S(",i+1,j+1,t+1,k+1,")",S[i,j,t,k].x)
                    if I[i,j,t,k].x>0:
                        print("I(",i+1,j+1,t+1,k+1,")",I[i,j,t,k].x)
                    for o in range(O):
                        if a[i,j,t,k,o].x>0:
                            print("O(",i+1,j+1,t+1,k+1,o+1,")",a[i,j,t,k,o].x)
#                          print "A_(",t,",",i,",",j,")^",w,"=",int(a[i,j,t,w].x)

    for t in range(TT):
        for k in range(M):
```

```
            if z[t,k,o].x>0:
                print("R(",t+1,k+1,o+1,")",z[t,k,o].x)
    else:
        model.computeIIS()
    #   Print the names of all of the constraints in the IIS set.
        print("--------IIS CONSTRAINTS")
        for c in model.getConstrs():
            if c.IISConstr > 0:
                print(c.ConstrName)
    #   Print the names of all of the variables in the IIS set.
        print("--------IIS VARIABLES")
        for v in model.getVars():
            if v.IISLB > 0 or v.IISUB > 0:
                print(v.VarName)
```

Multiple Projects

```
import numpy as np
import scipy.stats as stats
import math as mt
import random as random
import openpyxl
from random import randint
import os

#PROJECTS / CLUSTERS OF PROJECTS
P = [0,1,2,3,4,5,6,7,8,9]
#[0,16,7,10,8,17,2]#(CLUSTER1)
#[1,3,4,5,6,9,11,12,13,15,18,19,20,21,22] (CLUSTER2)
#[23,24,14] (CLUSTER3)
```

```
#[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24] (ALL PROJECTS)

FREQ = 1
freq = {}
for p in P:
    for p2 in P:
        freq[p,p2] = 0
for f in range(FREQ):
    W = 1
    pr = {}
    for w in range(W):
        pr[w]= 1/W
        # print("prrr",pr[w])

    def readData(weekNumber):
        fileName=str(weekNumber)+"/w"+str(weekNumber)+"-p.txt"
        cwd = os.getcwd()

        clusters={}
        with open(cwd+"/DATA/"+fileName) as f:
            for line in f:
                if "set J := " in line:
                    pointer = 8
                    i=0
                    while pointer+2<len(line):
                        clusters[i] = line[pointer+1:pointer+4]
                        i += 1
                        pointer += 6
                    C = i #or: len(consultants)
```

```
#                print(len(line))
#                print(line)
#                print(clusters)
            # print(C)
            break

#DISTANCE MATRIX
R = {}
for j in range(C):
    for c in range(C):
        R[j,c] = 0
fileName6=str(weekNumber)+"/w"+str(weekNumber)+"-r.txt"
with open(cwd+"/DATA/"+fileName6) as f:
    for line in f:
        if "#" not in line and "param" not in line:
            if ";" in line:
                break
            else:
                cluster1 = line[:3]
#                print "cons: ",consultant
                line = line[line.index("\t")+1:]
                cluster2 = line[:3]
#                print "clust: ",cluster
                line = line[line.index("\t")+1:]
                j = [k for k in range(C) if clusters[k]==cluster1][0]
                c = [l for l in range(C) if clusters[l]==cluster2][0]
                # print("R: ",line)
                R[j,c] = float(line)*10
return(clusters,R)
```

```
[clusters,R] = readData(15)

# print(R)
T = 30
r={}
wb = openpyxl.load_workbook('25Projects.xlsx')
type(wb)
wb.get_sheet_names()
sheet = wb.get_sheet_by_name('Projects1')
for w in range(W):
    for t in range(T):
        for p in P:
            r[w,t,p] = (sheet.cell(row=p+2,column=t+2).value)
            # r[w,t,p] =
              randint((sheet.cell(row=p+2,column=t+2).value),(sheet.cell(row=p+2,column
              =t+2).value)+2)
            # print("Demand is:  ",t,p,r[w,t,p])



L = 18
U = 18
interest = 1.01

c = {}
for i in range(L):
    for j in range(U):
        for t in range(T):
            if i == 0:
                c[i,j,t] = int((220000 - 4000*j)*(interest)**t) #12000
```

```
            else:
                c[i,j,t] = int((.8*c[0,j,t] - 1000*i)*(interest)**t) #3000
            # print c[i,j,t] .577 - hartman

m = {}
for i in range(L):
    for j in range(U):
        for t in range(T):
            m[i,j,t] = int((10000 + 167*i + 1667*j*((j+1)**1.2) - j**1.2)*(interest)**t)
              #500 and #5000

s = {}
for t in range(T):
    for i in range(L):
        for j in range(U):
                s[i,j,t] = (.57)*(c[i,j,t])

q = {}
for t in range(T):
    q[t] = 90000*interest**t

h = {}
for t in range(T):
    h[t] = 5000*interest**t

o = {}
for t in range(T):
    o[t] = 50000*interest**t
```

```
# for black in range(HORSE):

from gurobipy import *
model = Model("17C")

x={}
for i in range(L):
    for j in range(U):
        for p in P:
            x[i,j,p] = model.addVar(vtype='I',lb=0,ub=GRB.INFINITY)

y={}
for i in range(L):
    for j in range(U):
        for w in range(W):
            for t in range(T):
                for p in P:
                    y[i,j,t,w,p] = model.addVar(vtype='I',lb=0,ub=GRB.INFINITY)

# z = {}
# for w in range(W):
#     for t in range(T):
#         for p in P:
#             z[t,w,p] = model.addVar(vtype='I',lb=0, ub=0) #GRB.INFINITY)
# Z = model.addVar(vtype='I',lb=0, ub=GRB.INFINITY)

S={}
for i in range(L):
```

```
    for j in range(U):
        for w in range(W):
            for t in range(T):
                for p in P:
                    S[i,j,t,w,p] = model.addVar(vtype='I',lb=0,ub=GRB.INFINITY)

I={}
for i in range(L):
    for j in range(U):
        for w in range(W):
            for t in range(T):
                for p in P:
                    I[i,j,t,w,p] = model.addVar(vtype='I',lb=0,ub=GRB.INFINITY)

a={}
for i in range(L):
    for j in range(U):
        for w in range(W):
            for t in range(T):
                for p in P:
                    a[i,j,t,w,p] = model.addVar(vtype='I',lb=0,ub=GRB.INFINITY)

TC={}
for i in range(L):
    for j in range(U):
        for w in range(W):
            for t in range(T):
                for g in P:
                    for b in P:
```

```
            TC[i,j,t,w,g,b] = model.addVar(vtype='I',lb=0,ub=GRB.INFINITY)

model.update()



c1={}
for w in range(W):
    for t in range(T):
        for p in P:
            c1[w,t,p]=model.addConstr(quicksum(a[i,j,t,w,p] for i in range(L) for j in
              range(U)) >= r[w,t,p],name='c1_'+str(w)+"-"+str(t)+"-"+str(p))
# quicksum(r[w,t,p] for p in P)
#         model.addConstr(z[t,w]==1) z[t,w,p]+



c2={}
for w in range(W):
    for t in range(T):
        for i in range(L):
            for p in P:
                c2[w,t,i] = model.addConstr( a[i,U-1,t,w,p] == 0 ,
              name="c2_"+str(w)+"_"+str(t)+"_"+str(i))
c3={}
for w in range(W):
    for t in range(T):
        for j in range(U):
            for p in P:
                c3[w,t,j] = model.addConstr( a[L-1,j,t,w,p] == 0 )
```

```
#WITH SHIPPING CONSTRAINT
c4={}
for w in range(W):
    for t in range(1,T):
        for j in range(1,U):
            for i in range(1,L):
                for p in P:
                    c4[w,t,j,i,p] = model.addConstr(a[i,j,t,w,p]+I[i,j,t,w,p]+S[i,j,t,w,p] ==
              quicksum(TC[i,j,t,w,g,p] for g in P if g != p) - quicksum(TC[i,j,t,w,p,b] for b
              in P if b != p) + y[i,j,t,w,p]+a[i-1,j-1,t-1,w,p]+I[i-1,j,t-1,w,p])

#TURN OFF SHIPPING CONSTRAINT
# c4={}
# for w in range(W):
#     for t in range(1,T):
#         for j in range(1,U):
#             for i in range(1,L):
#                 for p in P:
#                     c4[w,t,j,i,p] = model.addConstr(a[i,j,t,w,p]+I[i,j,t,w,p]+S[i,j,t,w,p] ==
              y[i,j,t,w,p]+a[i-1,j-1,t-1,w,p]+I[i-1,j,t-1,w,p])

c5={}
c6={}
for w in range(W):
    for i in range(L-1):
        for j in range(U-1):
            for p in P:
```

```
            c5[w,i,j,p] = model.addConstr(a[i,j,0,w,p]+I[i,j,0,w,p]+S[i,j,0,w,p] ==
           y[i,j,0,w,p])
            c6[w,i,j,p] = model.addConstr(y[i,j,0,w,p] == x[i,j,p])
c7={}
for w in range(W):
    for t in range(1,T):
        for j in range(U):
            for p in P:
                c7[w,t,j,p] = model.addConstr(a[0,j,t,w,p]+I[0,j,t,w,p]+S[0,j,t,w,p] ==
               y[0,j,t,w,p])
c8={}
for w in range(W):
    for t in range(1,T):
        for i in range(L):
            for p in P:
                c8[w,t,i,p] = model.addConstr(a[i,0,t,w,p]+I[i,0,t,w,p]+S[i,0,t,w,p] ==
               y[i,0,t,w,p])

c9={}
for w in range(W):
    for t in range(1,T):
        for j in range(1,U):
            for p in P:
                c9[w,t,j,p] = model.addConstr( S[L-1,j,t,w,p] == y[L-1,j,t,w,p] + a[L-2,j-1,t-
               1,w,p] + I[L-2,j,t-1,w,p] )
c10={}
for w in range(W):
    for t in range(1,T):
        for i in range(1,L):
```

```
            for p in P:
                c10[w,t,i,p] = model.addConstr( S[i,U-1,t,w,p] == y[i,U-1,t,w,p] + a[i,U-2,t-
              1,w,p] )
c11={}
for w in range(W):
    for j in range(1,U):
        for p in P:
            c11[w,j,p] = model.addConstr( S[L-1,j,0,w,p] == y[L-1,j,0,w,p] )
c12={}
for w in range(W):
    for i in range(1,L):
        for p in P:
            c12[w,i,p] = model.addConstr( S[i,U-1,0,w,p] == y[i,U-
              1,0,w,p],name="c_12_"+str(w)+"_"+str(i))

c13={}
c14={}
for i in range(L):
    for w in range(W):
        for p in P:
            c13[i] = model.addConstr( S[i,0,0,w,p] == 0 )
for j in range(U):
    for w in range(W):
        for p in P:
            c14[j] = model.addConstr( S[0,j,0,w,p] == 0 )
# c15={}
# for w in range(W):
#     for p in P:
#         c15[w,t,p] = model.addConstr(z[0,w,p] == Z)
```

```
# model.update()




objective =0# Z*q+Z*o + quicksum((c[i,j]+o+m[i,j,t])*x[i,j] for i in range(L) for j in
            range(L))
for w in range(W):
    for t in range(T):
        for l in P:
            objective += (1/interest)**t*quicksum(o[t]*a[i,j,t,w,l] + c[i,j,t]*y[i,j,t,w,l] +
              m[i,j,t]*(a[i,j,t,w,l]) + h[t]*I[i,j,t,w,l] - s[i,j,t]*S[i,j,t,w,l] for i in range(L) for j
              in range(U))
for i in range(L):
    for j in range(U):
        for t in range(T):
            for w in range(W):
                objective += quicksum(R[g,b]*TC[i,j,t,w,g,b] for g in P for b in P)
            #((o[t]+q[t])*z[t,w,l] float(pr[w])
model.setObjective(objective)
model.setParam("MIPGap",0.00)
model.modelSense = GRB.MINIMIZE
model.update()
model.optimize()
print("model status is:",model.status)
if model.status==GRB.OPTIMAL:
    print("Optimal", model.objVal)
    Jimmy = model.objVal
    xx={}
```

```
yy={}
SS={}
II={}
aa={}
zz={}

for i in range(L):
    for j in range(U):
        for p in P:
            xx[i,j,p] = x[i,j,p].x
            if x[i,j,p].x>0:
                print("X",i+1,j+1,p+1,x[i,j,p].x)
# for w in range(W):
#     print("****SCENARIO ",w+1)
#     for p in P:
#         print("****PROJECT ",p+1)
#         for t in range(T):
#             # zz[t,w,p] = z[t,w,p].x
#             print("     ****TIME PERIOD ",t+1)
#             # # expy = 0
#             # # for ww in range(W):
#             # #     expy += p[ww]*r[ww,t]
#             # # print("expected demand",expy)
#             # if z[t,w,p].x>0:
#             #     print("R",z[t,w,p].x)
#             print("DEMAND:",r[w,t,p])
#             for i in range(L):
#                 for j in range(U):
#                     yy[i,j,t,w,p] = y[i,j,t,w,p].x
```

```
    #                   SS[i,j,t,w,p] = S[i,j,t,w,p].x
    #                   II[i,j,t,w,p] = I[i,j,t,w,p].x
    #                   aa[i,j,t,w,p] = a[i,j,t,w,p].x
    #                   if y[i,j,t,w,p].x>0:
    #                       print("P(",i+1,",",j+1,",",p+1,")",y[i,j,t,w,p].x)
    #                   if S[i,j,t,w,p].x>0:
    #                       print("S(",i+1,",",j+1,",",p+1,")",S[i,j,t,w,p].x)
    #                   if I[i,j,t,w,p].x>0:
    #                       print("I(",i+1,",",j+1,",",p+1,")",I[i,j,t,w,p].x)
    #                   if a[i,j,t,w,p].x>0:
    #                       print("O(",i+1,",",j+1,",",p+1,")",a[i,j,t,w,p].x)
    #
    for b in P:
        print("Project",b+1)
        for w in range(W):
            for i in range(L):
                for j in range(U):
                    for g in P:
                        for t in range(T):
                            if TC[i,j,t,w,g,b].x>0:
                                print("Receive(",i+1,",",j+1,",",g+1,",",b+1,")",TC[i,j,t,w,g,b].x)
                                freq[g,b] += 1

#

else:
    model.computeIIS()
#   Print the names of all of the constraints in the IIS set.
    print("--------IIS CONSTRAINTS")
```

```
        for c in model.getConstrs():
            if c.IISConstr > 0:
                print(c.ConstrName)
    #   Print the names of all of the variables in the IIS set.
        print("--------IIS VARIABLES")
        for v in model.getVars():
            if v.IISLB > 0 or v.IISUB > 0:
                print(v.VarName)
for g in P:
    for b in P:
        if freq[g,b]>1:
            print("Frequent Shipping from",g+1,"to",b+1,":",freq[g,b],"times")
```

Optimal Utilization

```
import mySolverTP as ms
import xlsxwriter


workbook = xlsxwriter.Workbook('17B-utilFactor.xlsx')
worksheet = workbook.add_worksheet('sensAn')
    # Add a bold format to use to highlight cells.
bold = workbook.add_format({'bold': 1})
underline = workbook.add_format({'underline': 1})
    # Add a number format for cells with money.
res_format = workbook.add_format({'num_format': '0.000000'})
bad_res_format = workbook.add_format({'num_format':'0.000000','color':
                '#FF0000','bold':1})
    # Add an Excel date format.
date_format = workbook.add_format({'num_format': 'mmmm d yyyy'})
    # coloring
```

```
blue_format = workbook.add_format({'color': '#0000FF'})
red_format = workbook.add_format({'color': '#FF0000'})
green_format = workbook.add_format({'color': '#008000'})

row=1

P1 = 220000
P2 = 10000
P3 = 3000
P4 = 12000
P5 = 500
P6 = 5000
UtilFactor=1;

[x,y,S,I,a,z,OFV] = ms.mySolver(P1,P2,P3,P4,P5,P6,UtilFactor,0,0,0,0,0,0,0)

worksheet.write('A1', 'UtilFactor', bold)
worksheet.write('B1', 'OFV', bold)

for UtilFactor in [0.25,0.5,0.75,1,1.25,1.5,1.75,2,2.25,2.5,2.75,3,3.25,3.5,3.75,4]:
    [xx,yy,SS,II,aa,zz,OFVV] = ms.mySolver(P1,P2,P3,P4,P5,P6,UtilFactor,0,x,y,S,I,a,z)
    worksheet.write_number(row, 1, OFVV)
    worksheet.write_number(row, 0, UtilFactor)
    row += 1

workbook.close()

from gurobipy import *
```

```
def
        mySolver(purchasePrice,maintBaseCost,purchaseAge,purchaseUtil,maintAge,
        maintUtil,UtilFactor,isFixed,xx,yy,SS,II,aa,zz):

    import matplotlib.pyplot as mpl
    import numpy as np
    import math as math




    row = 1

    L = 20
    U = int(math.ceil(10/UtilFactor))
    #20000 hours is the life
    # each year: 2000 hours of work
    # UtilFactor range: [0.2,4]
    W=7
    T=[4,5,6,7,8,9,10]

    c = {}
    for i in range(L):
        for j in range(U):
            for t in range(max(T)):
                if i == 0:
                    c[i,j,t] = purchasePrice*(1.03)**t
                else:
                    c[i,j,t] = int((c[0,j,t]*(.577) - purchaseAge*i -
                  purchaseUtil*(j**(.8*UtilFactor)))*(1.03)**t)
```

```
        # print c[i,j,t]

m = {}

for i in range(L):
    for j in range(U):
        for t in range(max(T)):
            m[i,j,t] = int((maintBaseCost + maintAge*i +
              maintUtil*UtilFactor*j*((j+1)**1.2) - j**1.2)*(1.02)**t)

s = {}
for t in range(max(T)):
    for i in range(L):
        for j in range(U):
                s[i,j,t] = (.75)*(c[i,j,t])

cap = {}
for t in range(max(T)):
    for w in range(W):
        cap[t,w] = 2

# from mpl_toolkits.mplot3d import Axes3D
# fig = mpl.figure(figsize=(7,6))   # This creates a new figure object.
# ax = fig.add_subplot(111, projection='3d')   # This adds a subplot to the figure with
              3D projection, and returns the axes object.
# x, y = np.linspace(0.0, L, L), np.linspace(0.0, U, U)   # Create the 2D space
# X, Y = np.meshgrid(x, y)  # Get the plaid version (the 'meshgrid' version, similar to
              Matlab's meshgrid function)
# z = (200000*(.577) - 5000*X - 5000*(Y**.8))*(1.03)**3
```

```
# csf = ax.contourf(X, Y, z, 15)
# cs = ax.contour(X, Y, z, 15, cmap=mpl.cm.Oranges_r)
# csl = ax.clabel(csf, fmt='%2.1f', colors='k', fontsize=14)
# cbar = mpl.colorbar(csf)
# mpl.show()
#
q = 100000
h = 5000
o = 50000

r = {}
# r[ 0 , 0 ]= 4
# r[ 0 , 1 ]= 6
# r[ 0 , 2 ]= 6
# r[ 0 , 3 ]= 4
# r[ 0 , 4 ]= 0
# r[ 0 , 5 ]= 0
# r[ 0 , 6 ]= 0
# r[ 0 , 7 ]= 0
# r[ 0 , 8 ]= 0
# r[ 0 , 9 ]= 0
# r[ 1 , 0 ]= 4
# r[ 1 , 1 ]= 6
# r[ 1 , 2 ]= 5
# r[ 1 , 3 ]= 3
# r[ 1 , 4 ]= 2
# r[ 1 , 5 ]= 0
# r[ 1 , 6 ]= 0
# r[ 1 , 7 ]= 0
```

r[1 , 8]= 0

r[1 , 9]= 0

r[2 , 0]= 4

r[2 , 1]= 5

r[2 , 2]= 4

r[2 , 3]= 3

r[2 , 4]= 2

r[2 , 5]= 2

r[2 , 6]= 0

r[2 , 7]= 0

r[2 , 8]= 0

r[2 , 9]= 0

r[3 , 0]= 4

r[3 , 1]= 4

r[3 , 2]= 3

r[3 , 3]= 3

r[3 , 4]= 2

r[3 , 5]= 2

r[3 , 6]= 2

r[3 , 7]= 0

r[3 , 8]= 0

r[3 , 9]= 0

r[4 , 0]= 3

r[4 , 1]= 3

r[4 , 2]= 4

r[4 , 3]= 3

r[4 , 4]= 2

r[4 , 5]= 2

r[4 , 6]= 2

```
# r[ 4 , 7 ]= 1
# r[ 4 , 8 ]= 0
# r[ 4 , 9 ]= 0
# r[ 5 , 0 ]= 2
# r[ 5 , 1 ]= 3
# r[ 5 , 2 ]= 3
# r[ 5 , 3 ]= 3
# r[ 5 , 4 ]= 2
# r[ 5 , 5 ]= 2
# r[ 5 , 6 ]= 2
# r[ 5 , 7 ]= 2
# r[ 5 , 8 ]= 1
# r[ 5 , 9 ]= 0
# r[ 6 , 0 ]= 1
# r[ 6 , 1 ]= 2
# r[ 6 , 2 ]= 3
# r[ 6 , 3 ]= 3
# r[ 6 , 4 ]= 3
# r[ 6 , 5 ]= 2
# r[ 6 , 6 ]= 2
# r[ 6 , 7 ]= 2
# r[ 6 , 8 ]= 1
# r[ 6 , 9 ]= 1

p = {}
p[0] = 0.05
p[1] = 0.25
p[2] = 0.30
p[3] = 0.20
```

```
p[4] = 0.10
p[5] = 0.07
p[6] = 0.03

# R={}
# for t in range(10):
#     R[t] = sum([float(p[w]*r[w,t]) for w in range(W)])

model = Model("17B")

x={}
for i in range(L):
    for j in range(U):
        x[i,j] = model.addVar(vtype='I',lb=0,ub=GRB.INFINITY)

y={}
for i in range(L):
    for j in range(U):
        for w in range(W):
            for t in range(max(T)):
                y[i,j,t,w] = model.addVar(vtype='I',lb=0,ub=GRB.INFINITY)

z = {}
for w in range(W):
    for t in range(max(T)):
        z[t,w] = model.addVar(vtype='I',lb=0, ub=GRB.INFINITY)
Z = model.addVar(vtype='I',lb=0, ub=GRB.INFINITY)

S={}
```

```
for i in range(L):
    for j in range(U):
        for w in range(W):
            for t in range(max(T)):
                S[i,j,t,w] = model.addVar(vtype='I',lb=0,ub=GRB.INFINITY)

I={}
for i in range(L):
    for j in range(U):
        for w in range(W):
            for t in range(max(T)):
                I[i,j,t,w] = model.addVar(vtype='I',lb=0,ub=GRB.INFINITY)

a={}
for i in range(L):
    for j in range(U):
        for w in range(W):
            for t in range(max(T)):
                a[i,j,t,w] = model.addVar(vtype='I',lb=0,ub=GRB.INFINITY)
E={}
for w in range(W):
    for t in range(max(T)):
        E[t,w] = model.addVar(vtype='I',lb=0,ub=GRB.INFINITY)

model.update()

# # model.addConstr(x[0,0]==3)
# # model.addConstr(x[1,8]==1)
# # model.addConstr(Z==1)
```

```
# # model.addConstr(quicksum(x[i,j] for i in range(L) for j in
            range(youmotherfucker))==4)


c2={}
for w in range(W):
    for t in range(max(T)):
        c2[w,t]=model.addConstr(z[t,w] + quicksum(a[i,j,t,w] for i in range(L) for j in
            range(U)) >= int(math.ceil(r[w,t]/UtilFactor)),name='c2_'+str(w)+"-"+str(t))
        # model.addConstr(z[t,w]==1)
c3={}
for w in range(W):
    for t in range(max(T)):
        for i in range(L):
            c3[w,t,i] = model.addConstr( a[i,U-1,t,w] == 0 ,
            name="c3_"+str(w)+"_"+str(t)+"_"+str(i))
c4={}
for w in range(W):
    for t in range(max(T)):
        for j in range(U):
            c4[w,t,j] = model.addConstr( a[L-1,j,t,w] == 0 )
c5={}
c52={}
for w in range(W):
    for i in range(L-1):
        for j in range(U-1):
            c5[w,i,j] = model.addConstr(a[i,j,0,w]+I[i,j,0,w]+S[i,j,0,w] == y[i,j,0,w])
            c52[w,i,j] = model.addConstr(y[i,j,0,w] == x[i,j])
c6={}
for w in range(W):
```

```
    for t in range(1,max(T)):
        for j in range(1,U):
            for i in range(1,L):
                c6[w,t,j,i] = model.addConstr(a[i,j,t,w]+I[i,j,t,w]+S[i,j,t,w] == y[i,j,t,w]+a[i-
              1,j-1,t-1,w]+I[i-1,j,t-1,w])

c7={}
for w in range(W):
    for t in range(1,max(T)):
        for j in range(U):
            c7[w,t,j] = model.addConstr(a[0,j,t,w]+I[0,j,t,w]+S[0,j,t,w] == y[0,j,t,w])
c8={}
for w in range(W):
    for t in range(1,max(T)):
        for i in range(L):
            c8[w,t,i] = model.addConstr(a[i,0,t,w]+I[i,0,t,w]+S[i,0,t,w] == y[i,0,t,w])

c9={}
for w in range(W):
    for t in range(1,max(T)):
        for j in range(1,U):
            c9[w,t,j] = model.addConstr( S[L-1,j,t,w] == y[L-1,j,t,w] + a[L-2,j-1,t-1,w] +
              I[L-2,j,t-1,w] )

c10={}
for w in range(W):
    for t in range(1,max(T)):
        for i in range(1,L):
            c10[w,t,i] = model.addConstr( S[i,U-1,t,w] == y[i,U-1,t,w] + a[i,U-2,t-1,w] )
```

```
c11={}
for w in range(W):
    for j in range(1,U):
        c11[w,j] = model.addConstr( S[L-1,j,0,w] == y[L-1,j,0,w] )
c12={}
for w in range(W):
    for i in range(1,L):
        c12[w,i] = model.addConstr( S[i,U-1,0,w] == y[i,U-
            1,0,w],name="c_12_"+str(w)+"_"+str(i) )
c13={}
c14={}
for i in range(L):
    for w in range(W):
        c13[i] = model.addConstr( S[i,0,0,w] == 0 )
for j in range(U):
    for w in range(W):
        c14[j] = model.addConstr( S[0,j,0,w] == 0 )
c15={}
for w in range(W):
    c15[w,t] = model.addConstr(z[0,w] == Z)

c16={}
for w in range(W):
    c16[w] = model.addConstr(quicksum(S[i,j,t,w] for t in range(T[w],max(T)) for i in
            range(L) for j in range(U))==0)

c17={}
for w in range(W):
    for t in range(max(T)):
```

```
    c17[t,w] = model.addConstr(E[t,w] >= quicksum(a[i,j,t,w] for i in range(L) for j
       in range(U)) - cap[t,w])

# Fix a solution in the model
if isFixed==1:
  for i in range(L):
    for j in range(U):
      model.addConstr(x[i,j]==xx[i,j])
      for w in range(W):
        for t in range(max(T)):
          model.addConstr(y[i,j,t,w]==yy[i,j,t,w])
          model.addConstr(I[i,j,t,w]==II[i,j,t,w])
          model.addConstr(a[i,j,t,w]==aa[i,j,t,w])
          model.addConstr(S[i,j,t,w]==SS[i,j,t,w])
  for t in range(max(T)):
    for w in rangr(W):
      model.addConstr(z[t,w]==zz[t,w])


model.update()
#
#
#
#
objective =  0 #Z*q+Z*o + quicksum((c[i,j,0]+o+m[i,j,0])*x[i,j] for i in range(L) for j
       in range(U))
for w in range(W):
  for t in range(max(T)):
```

```
        objective += float(p[w])*(10000*E[t,w] + q*z[t,w] + o*UtilFactor*z[t,w] +
            quicksum(o*UtilFactor*a[i,j,t,w] + (20000 + c[i,j,t])*y[i,j,t,w] +
            m[i,j,t]*(a[i,j,t,w]) + h*(S[i,j,t,w]+I[i,j,t,w]) - s[i,j,t]*S[i,j,t,w]
            +10000*S[i,j,t,w] for i in range(L) for j in range(U)))

model.setObjective(objective)
model.setParam("MIPGap",0.00)
model.modelSense = GRB.MINIMIZE
model.update()
model.optimize()
print("model status is:",model.status)
if model.status==GRB.OPTIMAL:
    print("Optimal", model.objVal)
    Jimmy = model.objVal
    xx={}
    yy={}
    SS={}
    II={}
    aa={}
    zz={}

    for i in range(L):
        for j in range(U):
            xx[i,j] = x[i,j].x
            if x[i,j].x>0:
                print("X",i,j,x[i,j].x)
    for w in range(W):
        print("****SCENARIO ",w)
        for t in range(max(T)):
```

```
        print("    ****TIME PERIOD ",t)
        for i in range(L):
            for j in range(U):
                yy[i,j,t,w] = y[i,j,t,w].x
                SS[i,j,t,w] = S[i,j,t,w].x
                II[i,j,t,w] = I[i,j,t,w].x
                aa[i,j,t,w] = a[i,j,t,w].x
                if y[i,j,t,w].x>0:
                    print("y",i,j,t,w,y[i,j,t,w].x)
                if S[i,j,t,w].x>0:
                    print("S",i,j,t,w,S[i,j,t,w].x)
                if I[i,j,t,w].x>0:
                    print("I",i,j,t,w,I[i,j,t,w].x)
                if a[i,j,t,w].x>0:
                    print("a",i,j,t,w,a[i,j,t,w].x)

    for w in range(W):
        for t in range(max(T)):
            zz[t,w] = z[t,w].x
            if z[t,w].x>0:
                print("z",t,w,z[t,w].x)
    return (xx,yy,SS,II,aa,zz,Jimmy)
else:
    model.computeIIS()
#   Print the names of all of the constraints in the IIS set.
    print("--------IIS CONSTRAINTS")
    for c in model.getConstrs():
        if c.IISConstr > 0:
            print(c.ConstrName)
```

```
#   Print the names of all of the variables in the IIS set.
    print("--------IIS VARIABLES")
    for v in model.getVars():
        if v.IISLB > 0 or v.IISUB > 0:
            print(v.VarName)
```

VITA

Brett Allen Shields was born in Torrance, California to the parents of Brigitte and William Shields. He is the middle child of two other siblings: Katie and Nikki. He attended Pigeon Forge High School in Tennessee, where he was a two-time state finalist in wrestling. He subsequently attended Carson-Newman University to pursue his wrestling career in the NCAA. Brett finished his studies at King University, where he completed a Bachelor's of Science in Mathematics. He then continued his education in graduate school as a Graduate Teaching Associate at East Tennessee State University, where he earned his Master's of Science in Mathematical Science. After graduating from ETSU, Brett spent one year teaching secondary education mathematics before pursuing his PhD. He accepted a Graduate Research Assistantship in the spring of 2015 at the University of Tennessee Space Institute (UTSI), an offsite campus of the University of Tennessee Knoxville (UTK). Brett is currently a Tenure-Track Faculty member of Industrial Engineering at Francis Marion University (FMU).