8-2018

Developing Optimization Techniques for Logistical Tendering Using Reverse Combinatorial Auctions

Jennifer Kiser
East Tennessee State University

Developing Optimization Techniques for Logistical Tendering Using Reverse Combinatorial Auctions

---

A thesis

presented to

the faculty of the Department of Mathematics

East Tennessee State University

In partial fulfillment

of the requirements for the degree

Master of Science in Mathematical Sciences

---

by

Jennifer Kiser

August 2018

Michele Joyner, Ph.D., Chair

Jeff Knisley, Ph.D.

Robert Price, Ph.D.

Keywords: operations research, optimization, stochastic programming, logistical tender

ABSTRACT

Developing Optimization Techniques for Logistical Tendering Using Reverse Combinatorial Auctions

by

Jennifer Kiser

In business-to-business logistical sourcing events, companies regularly use a bidding process known as tendering in the procurement of transportation services from third-party providers. Usually in the form of an auction involving a single buyer and one or more sellers, the buyer must make decisions regarding with which suppliers to partner and how to distribute the transportation lanes and volume among its suppliers; this is equivalent to solving the optimization problem commonly referred to as the Winner Determination Problem. In order to take into account the complexities inherent to the procurement problem, such as considering a suppliers network, economies of scope, and the inclusion of business rules and preferences on the behalf of the buyer, we present the development of a mixed-integer linear program to model the reverse combinatorial auction for logistical tenders.

ACKNOWLEDGMENTS

I would like to first and foremost thank my Lord and Savior for guiding my steps, giving me faith, and providing me with wisdom and understanding each and every day. I would like to thank my husband, Derek, and my father, Floyd, for their support and guidance throughout this process. I would especially like to acknowledge and thank my thesis advisor, Dr. Michele Joyner, for her patience and advice and my thesis committee members, Dr. Jeff Knisley and Dr. Robert Price, for their input and assistance. Lastly I would like to mention Dr. Baptiste Lebreton for proposing the ideas behind the thesis work, for being available to answer many questions, for providing thoughtful feedback, and for providing the data set used in the main results of the work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 MOTIVATION

Companies regularly use tendering in the procurement of transportation services from third-party providers, usually in the form of an auction. Common types of auctions include forward auctions, reverse auctions, combinatorial auctions, and conditional (also referred to as expressive) auctions [24]. A traditional (forward) auction, simply referred to as an auction, is from the perspective of the seller where the highest bidder is the winner, while a reverse auction is from the perspective of the buyer where the lowest bidder is the winner [13]. In each of these types of auctions, bids are limited to single items. Combinatorial auctions allow bids to be placed on bundles, packages, or sets of items [30]. Similarly, reverse combinatorial auctions allow bids to be placed on one or more items and are from the perspective of the buyer.

The company that purchases the services is referred to as the buyer or shipper, and the corresponding company that provides the desired services is referred to as the supplier or carrier. Partnerships, and ultimately contracts, between the buyer and supplier are established through a bidding process, otherwise referred to as a tender. Buyers forecast expected demand on individual lanes, where a lane is defined to be a shipping route consisting of an origin and a destination. The buyer bids out the lanes for which it requires service, along with the estimated volume of the lane. The shipper responds with a quoted price for which it will service either an individual lane or a set of lanes in the form of a bid. By placing a bid, the carrier agrees to service that lane at the quoted price and enters into a contract with the buyer. Once the buyer receives bids from all potential carriers, the buyer evaluates all of the bids on each lane and chooses the winner, usually awarding an individual lane or a set of lanes to the lowest bidder.

Often the buyer will assign carriers to lanes based on other characteristics and constraints in addition to cost. Optimization problem solved by the buyer is an convenient and practical approach. The buyer matches carriers to the lanes it is best able to serve. This in turn reduces costs across the supply chain [7]. The optimization problem consists of minimizing total transportation cost subject to one or more constraints. For instance, the constraints may ensure that each lane is awarded to at most one carrier, the volume awarded on each lane does not exceed the expected demand, or the number of lanes awarded to a given carrier does not exceed the volume the carrier is willing to supply [4, 19, 33].

Operations research is a branch of mathematics that studies constrained optimization. A buyer can define an optimization problem in the form of a cost function subject to constraints. This allows the buyer to consider several factors in addition to the price of a bid when determining how to assign carriers to lanes. Each of the constraints to which the cost function is subject, along with any additional constraints that represent the buyer's business rules and/or preferences, are formalized mathematically. The cost function, which represents the total cost of its transportation services in the form of a sum of submitted bids, is minimized subject to all of the business constraints. The mathematical techniques used in operations research to solve the optimization problem provide the buyer with the optimal allocation.

The goal of this work is to present a comprehensive and cohesive outline of not only the formulation, but also the implementation of an optimization problem which to use in commercial logistical tendering. We proceed by presenting an introduction to tendering optimization and outline several aspects of the supply chain in chapter 2. We also include specific instances for which a business may find the use of optimization techniques both desirable and effective during the logistics procurement process. We proceed to present an overview of linear programming along with the combinatorial auction optimization models in chapter 3, while chapter 4 consists of an introduction to stochastic programming and the stochastic linear programming models. All of the methods and implementation details of the models are found in chapter 5, the results are presented in chapter 6, and lastly, our future work and conclusions can be found in chapter 7.

2 TENDERING OPTIMIZATION

In order to formalize the outlined optimization problem, it is important to outline several aspects of the supply chain pertaining to transportation procurement. This chapter consists of delineating logistics and sourcing as it relates to the supply chain, the tendering methodology pertaining to a firm's transportation needs, and how logistics pertains to economies of scope in sections 2.1, 2.2, and 2.3, respectively. The resulting allocation problem is described in section 2.4, followed by detailed examples in section 2.5 as further motivation for the need to apply operations research techniques to formally define the tendering optimization problem. Lastly, we conclude the chapter with a discussion on the role of electronic auctions in the procurement of logistical services and its use in solving the outlined allocation problem.

2.1 Introduction to Logistics and Sourcing

Logistics is regarded as the portion of the supply chain that deals with the organization and management of the transportation of goods and services [5]. The most common modes of transportation services utilized by businesses include air, freight or rail, water, road via truckload (TL) or less-than-truckload (LTL), pipeline, and intermodal [35]. Intermodal transportation services consist of any combination of two or more of the aforementioned services, such as the use of truckload or less-than truckload in conjunction with freight services where the trucking service acts as an intermediary in order to transport goods, perhaps from a warehouse or distribution center to a loading dock.

Because each transportation mode consists of unique trade-offs between cost and lead time, individual business needs must be considered when making logistical decisions, such as choosing a mode of transportation. A company may employ different modes of transportation for different products depending on the size and value of the product it needs shipped. The design and implementation of a company's supply chain further impacts the decisions involved in choosing transportation services. Depending on the use and proximity of warehouses and distribution centers, certain modes of transportation may be more cost efficient, maintain adequate inventory levels, and produce desired customer responsiveness levels [5].

In addition to choosing an adequate mode of transportation, a company must decide to either manage their own transportation services in-house or to outsource these services. Businesses that invest in establishing and managing their own transportation needs are commonly referred to as 4PL, while those businesses that procure transportation services from a third party logistics provider are referred to as 3PL [35]. In these types of business-to-business (B2B) transactions, the business that is purchasing or sourcing the transportation services is commonly referred to as the shipper or buyer, and the 3PLs that provide transportation services are commonly referred to as the carriers or suppliers.

A third party provider should provide a company with the benefits of combining lower costs with higher quality service than what the company could provide on its own. Additional benefits gained by outsourcing logistical services include being able to focus on the business's main strengths and core competencies, improvements in customer service, and decreases in labor problems due to the transfer of responsibility to the third party provider [10]. However, involving a third party also brings risk associated with it, such as shifting power from one enterprise to another or underestimating the costs (either monetary, time, or reputation costs) incurred in organizing and managing numerous outside firms [5]. The decision to outsource services through a 3PL should not solely be based on an increase in savings across the supply chain but should also take into account the risks involved when a third party entity is involved.

In todays global market, businesses often make the decision to procure or obtain transportation services from a third party logistics provider. In the United States alone, logistics and transportation spending in 2015 made up 8 percent of the country's annual gross domestic product (GDP) by totaling $1.48 trillion [36].

2.2 The Procurement and Tendering Process

In general when a firm makes the decision to outsource its transportation services, the company is faced with numerous decisions. The buyer must decide with which supplier to partner and how to allocate the needed transportation routes and volume among its suppliers. Procurement through tendering can be viewed as a three step process: bid preparation, bid execution, and bid analysis and assignment [4].

The first step in the procurement process is a planning phase – deciding the specific details of the buyer's demand, what needs to be tendered, and which suppliers to contact. When preparing the tender, the shipper first defines the main objective or goal of the tender. Most commonly the supplier will outsource transportation services with the goal of reducing costs or increasing revenue. Other common goals may involve sustainability, consolidating business, reducing lead times, etc. Defining the buyer's goals in the beginning will guide the tender and aid in decision making throughout the rest of the bid preparation stage.

Once the main objective of the tender is established, the buyer then decides the specifications of the lots to be bid out. For each lot (i.e., lane) in the tender, the point of origin, the destination, and the estimated volume on the lane must be specified. Other properties such as average payload, expected number of annual loads, and the current cost of servicing that lane may be included to encourage more robust bids from suppliers. In addition to determining the lanes to be sourced, the buyer must decide if a lane should be awarded to a single source or if more than one source will be allowed to service a lane. For example, a firm that has regular shipments from Los Angeles to Miami must decide if it will bid out the lanes between Los Angeles and Miami as numerous individual lanes, allow the demand to be divided into a portion of smaller lanes, or tender the entire set of lanes together to be awarded to a single supplier. These decisions directly affect the end results of the allocation based on the suppliers' reaction to the tender based on the bids received.

Another consideration to address while preparing the tender involves supplier eligibility: which suppliers will be considered in the tender process.

Supplier eligibility is another consideration the buyer will need to address during the preparation phase of the tender. Specifically, the buyer determines which suppliers will be considered in the tender process A buyer may allow any carrier to participate in the tender through an open market setting or establish a proprietary setting in which suppliers are put through a qualification process and then chosen to participate in the tender based on their capabilities [1]. The requirements for a supplier to be considered for qualification should be clearly outlined and coincide with the buyer's main objectives of the tender. Only those carriers that aid in achieving the goals of the buyer should be considered eligible for participation. The basis of the supplier's eligibility may include prior business relations, standards such as sustainability efforts, reputation and/or risk involvement in dealing with the prospective supplier, customer satisfaction rates, available resources, the ability to meet demand, etc.

Lastly, the tender preparation phase outlines how a supplier should submit their bids, the decision criteria the shipper uses to evaluate and select winning bids, and the time frame suppliers have to submit bids. How a supplier submits bids on a lane is determined by the type of auction used. The traditional auction format allows for bids to be submitted on each lane independent of all other lanes, while a combinatorial auction allows carriers to submit bids on packages or sets of lanes. Once all of these details have been established by the buyer, the next step is to execute the tender. In this step the tender is formally issued to the potential suppliers. After inviting carriers to the tender, the buyer awaits responses in the form of bids from the suppliers.

After the launch of the project and all bidding is complete, the buyer evaluates the bids and gives feedback to the suppliers. The approach a buyer uses to analyze bids varies depending on (1) the type of auction issued in the tender, (2) the inclusion of the business constraints and/or preferences, and (3) the buyer's main objective and goal(s) of the tender. The buyer determines the final lot allocation after analyzing the supplier's data under different scenarios. Finally, the buyer awards its business to the selected suppliers and informs all participants of the outcome of the tender.

2.3 Economies of Scope

Transportation providers face economies of scope due to synergies or a dependence between inbound and outbound travel costs. Carriers typically ship packages on a lane consisting of an origin and a destination. However, a carrier's revenue is highly dependent on the costs incurred from traveling not only to the destination, but also on the return trip. This creates an interdependence between the two adjacent lanes: the cost of a delivery from the origin to the destination also depends on either or possibly both the cost of the return trip to the origin or the cost of traveling to the next pickup location. For example, a carrier that delivers a product from New York to Chicago still incurs travel costs on the return trip from Chicago to New York. Economies of scope are achieved if the carrier can obtain a delivery (or partial delivery) on the return trip and thus increases its revenue. Factors in the carrier's network such as commuting without a load (i.e., deadheading), dead times while loading or unloading, and variability in lead times all have an additional effect on the cost of a shipment [4].

Shippers frequently issue a tender in order to outsource the needed transportation services to the lowest bidder. However, due to the inherently complicated nature of logistics, the process of sourcing transportation services via an auction introduces numerous complexities as previously outlined. In a single-lane, noncombinatorial auction format that awards suppliers based solely on the lowest bid, these complexities are not taken into account. Hence economies of scope are most likely not going to be achieved by the carrier in that scenario. Rather than awarding carriers based solely on lowest cost, the problem of how to assign carriers to the lanes that achieve economies of scope and ultimately minimize cost across the supply chain must be addressed. Based on the supplier's transportation network and cost structure, which is usually unavailable to the buyer during the auction, the buyer is left to solve the resulting allocation problem consisting of matching those carriers to the demand it can best handle.

2.4 The Winner Determination Problem

For this research, we focus on 3PL businesses in which B2B outsourcing occurs through the use of online combinatorial auctions. We are interested in outlining the process for which companies (referred to as buyers) acquire goods and/or services through an auction for which suppliers place bids on lanes, commonly referred to as lots. This method of selecting suppliers is commonly referred to as a tender [2]. In logistical services, the lot of the tender is comprised of the point of origin and the destination. For instance, routes from Chicago to Dallas, Paris to New York, or Los Angeles to Tokyo are examples of domestic and international lanes.

Beginning in the early 1990's, companies began using auctions in the procurement of transportation services. Sears Logistical Services (SLS) was one of the first corporations to implement an auction in order to consolidate their logistical services and reduce costs: in 1993, through the use of a combined-value auction, SLS reduced its transportation costs from $190 million to $165 million per year on a total of 854 lanes [19]. Other major corporations to carry out the procurement of transportation through electronic exchanges include Walmart, Home Depot Inc., Kmart Corporation, Staples Inc., Compaq Computer Company, and The Limited Inc, which claimed to have saved \$1.24 million in its shipping costs in 2001 by implementing an online combinatorial auction [6]. However, procurement auctions are not restricted to only the purchasing of transportation. In the early 2000's CombineNet hosted over 400 sourcing events for the procurement of a vast number of goods and services, such as transportation, direct and indirect materials, packaging, chemicals, healthcare, and telecommunications, and claims savings of \$4.4 billion to its customers between 2001 and 2006 [31].

Traditionally, the supplier that provides the "best" bid on a lane or set of lanes is chosen as the winner for that lot, where the "best" bid is equivalent to the lowest monetary bid placed on a given lane. However, with the use of online auctions in conjunction with an increase in the globalization of trade, the lowest bid may not necessarily result in the "best" or optimal allocation. Choosing the lowest bid on a given lane may have considerable long-term effects on the buyer's entire supply chain, relations between the supplier and buyer, and the supplier's ability to fulfill its obligations to its customers. Such characteristics, along with the inclusion of the buyer's business rules and preferences, must be taken into account in addition to the monetary value of each supplier's bid.

The establishment of the winner determination problem is due to the complexities that stem from tenders performed on an electronic exchange. The winner determination problem equates to obtaining an optimal solution on all sets of lanes. This results in an optimal allocation of total demand among the chosen suppliers [6]. Such an allocation is considered optimal because, although it may not return the lowest bid on each individual lane, it will bring additional value to the entire supply chain. The value that an optimal allocation brings may differ on a case by case basis depending on the buyer's desired end result, such as minimizing transportation cost, increasing the number of sustainable carriers, increasing the number of minority owned carriers, or decreasing the lead time on shipments to its customers. In each of these scenarios, choosing the lowest bid on each individual lane will most likely not result in achieving the firm's main sourcing objective; whereas the buyer is able to achieve its main objective by obtaining an optimal allocation that takes into account the buyer's business rules and preferences. Thus solving the resulting winner determination problem, otherwise known as the allocation problem, is the main goal of the tender.

2.5 Allocation Examples

Next we present several examples to illustrate the resulting allocation problem in order to showcase its complexity. Consider the following scenario: suppose that a manufacturing company (referred to as the Buyer) needs to purchase freight services from its main facility in Boston to five regional distribution centers located throughout the country. The five lanes the buyer needs serviced all have the same origin with destinations in Los Angeles, Phoenix, Chicago, Jacksonville, and New York City, designated LA, PHO, CHI, JAX, and NYC, respectively.

The Buyer has implemented an auction for the procurement of its freight services involving five suppliers: Supplier A, B, C, D, and E, respectively. Table 1 contains the spreadsheet representation of data the Buyer compiled at the end of the bidding process. Each row in the table represents one bid, which consist of the lane being bid on, the bid amount, and the seller that placed the bid. The Buyer then determines the winner of each lane based on the lowest bid on a lane-by-lane basis without taking any constraints or preferences into consideration. Based solely on the lowest bid, the winning carrier for lane LA is Supplier C at a cost of $100, lane CHI is Supplier A at $95, lane PHO is Supplier D at $300, lane NYC is Supplier E at $75, and lane JAX is Supplier A at $180. The cost of transportation services on these five lanes totals to $750.

Table 1: Allocation main example: suppliers' bids sorted by lane and lowest bid amount.

Lane	**Bid Amount ($)**	**Placed By**
LA	100	Supplier C
LA	125	Supplier B
LA	160	Supplier A
LA	210	Supplier D
CHI	95	Supplier A
CHI	125	Supplier C
PHO	300	Supplier D
PHO	375	Supplier B
PHO	510	Supplier A
NYC	75	Supplier E
NYC	85	Supplier D
NYC	90	Supplier B
NYC	120	Supplier A
JAX	180	Supplier A

Next we present several scenarios that introduce specific business rules, constraints, and or preferences placed on the auction by the buyer using bids presented in Table 1, unless otherwise noted.

Scenario 1: Non-price Attributes. In addition to the price of a shipper's bid, the buyer wants to take into account one or more non-price attributes, such as a supplier's reputation or preference to an incumbent. The buyer has worked with supplier A and supplier B in the past and has established good working relationships with each of them. In light of this, the Buyer would like to continue working with both of these shippers, so the Buyer decides to award at least one lane to supplier A and supplier B. Alternatively, the Buyer could have chosen to award a certain % of lanes, in this case 20%, to its incumbent suppliers.

The traditional method outlined above based exclusively on the lowest bid does not award any business to supplier B; however, the Buyer could make a quick comparison and award lane NYC to supplier B instead of supplier E. This results in an increase in the Buyer's transportation cost by $15, resulting in a total cost of $765, but the benefits of working with and continuing to build relationships with its incumbent suppliers outweighs the price increase in the end. Table 2 indicates the winning bids for this scenario in bold text, while the bid placed by supplier E that was dropped is distinguished in italicized text for a convenient comparison between the bids.

Table 2: Inclusion of non-price attributes allocation example: suppliers' bids in descending order by lane and lowest bid amount.

Lane	Bid ($)	Placed By	Lane	Bid ($)	Placed By
LA	**100**	**Supplier C**	PHO	375	Supplier B
LA	125	Supplier B	PHO	510	Supplier A
LA	160	Supplier A	NYC	*75*	*Supplier E*
LA	210	Supplier D	NYC	85	Supplier D
CHI	**95**	**Supplier A**	NYC	**90**	**Supplier B**
CHI	125	Supplier C	NYC	120	Supplier A
PHO	**300**	**Supplier D**	JAX	**180**	**Supplier A**

Scenario 2: Minimizing Risk. The Buyer may be in a position such that it

wants to minimize risk and the costs associated with organizing and collaborating with several suppliers. The Buyer may decide to limit the number of different suppliers it will award business. Notice that the Buyer's original allocation allowed it to maintain contracts with four suppliers: supplier A, C, D, and E. If the Buyer desires to partner with no more than three carriers, the traditional method of choosing suppliers would not incorporate this preference. Because of the small number of bids received in the tender, the Buyer could easily notice that awarding the Los Angeles lane to supplier D instead of supplier E would fulfill its preference and only increase its transportation cost by $10. Table 3 presents only the bids that require consideration in order to reduce the number of suppliers in the allocation from four to three with the winning bids in bold and the dropped bid in italics. It is important to note that in a tender consisting of only 14 bids it is not difficult to make this comparison, but doing so for a tender with hundreds or possibly thousands of bids would not be feasible or desirable.

Table 3: Minimization of risk allocation example: suppliers' bids in descending order by lane and lowest bid amount.

Lane	**Bid ($)**	**Placed By**	**Lane**	**Bid ($)**	**Placed By**
LA	**100**	**Supplier C**	PHO	375	Supplier B
LA	125	Supplier B	NYC	*75*	*Supplier E*
PHO	**300**	**Supplier D**	NYC	**85**	**Supplier D**

Scenario 3: Conditional Bids. Conditional bids are often used in a tender to allow the shipper to partition lanes into a desirable package of lanes for which it could service at a lower cost than any subset of the individual lanes. These savings are

passed along to the buyer resulting in lower costs across the supply chain. Conditional bids are often in the form of XOR statements: conditions on a bid are specified by the supplier using if, else, or if-else statements [19]. Bids of this form typically allow a carrier to benefit from economies of scope by taking advantage of other lanes it services within its network. We present three scenarios that a carrier may take advantage of conditional bids to reduce it's cost by placing more competitive bids.

Based on the supplier's cost structure, a supplier may be able to provide transportation services on a set (or package) of lanes at a lower cost than if each lane was considered individually. For instance, supplier C could reduce it's internal costs if it serviced both lanes to Los Angeles (LA) and Chicago (CHI), perhaps by forming a closed loop based using other services it provides. If the Buyer allows bidders to place conditional bids, supplier C could place a more competitive bid such that it would service LA and CHI for \$175 if awarded both lanes. The resulting bid could be viewed as reducing its bid on CHI by \$50 (from \$125 to \$75) under the condition that supplier C would be awarded LA and CHI, which would be a win-win scenario since the Buyer reduces its procurement costs by \$20 and supplier C improves its underlying cost structure. Table 4 presents the bids from the current allocation without the inclusion of conditional bids versus the alternative allocation that takes into account the use of conditional bids.

Alternatively, a carrier's limited resources may restrict the total amount of additional volume under new contracts it is willing to undertake. However, because the carrier does not know which lanes it may be awarded, the carrier does not want to limit the bids it places and risk being awarded less business than desired. Based

Table 4: Conditional bids scenario: current allocation versus alternative allocation containing supplier C's packaged bid set.

Current Allocation				Alternative Allocation		
Lane	**Bid ($)**	**Placed By**		**Lane**	**Bid ($)**	**Placed By**
LA	**100**	**Supplier C**		LA	**100**	**Supplier C**
LA	125	Supplier B		LA	125	Supplier B
CHI	**95**	**Supplier A**		CHI	95	Supplier A
CHI	125	Supplier C		CHI	**75**	**Supplier C**

on supplier A's current volume across its network, it possesses enough resources to service either the lane to CHI or JAX but not both lanes. Given the current bids in Table 1, supplier A would be awarded both of these lanes without the ability to follow through on both contracts, while the use of conditional bids would not lead to problems associated with this situation such as delayed shipments, negative effects on the buyer-shipper relationship, and higher transportation costs. In this circumstance the Buyer would need to award lane CHI to supplier C, resulting in an increase of its costs by $30 while avoiding the previously outlined difficulties.

Finally, a carrier may wish to be more competitive in its bids by offering a discount for its services if award a minimum volume. Rather than placing a conditional bid on specific lanes to take advantage of economies of scope, this tactic allows a carrier to incorporate economies of scale by increasing its overall production. For instance, supplier D might offer a discount of 20% if awarded two or more lanes. The Buyer could easily reduce each of the bids placed by supplier D by 20% and compare the current bids with this set of new bids presented in Table 5. A quick assessment shows that the Buyer would not be able to lower its transportation costs using an allocation that incorporates supplier D's conditional bid. In the end, the Buyer would benefit from its original allocation chosen from Table 1.

Table 5: Conditional bids containing supplier D's reduced bids (sorted by lane and lowest bid amount).

Lane	Bid Amount ($)	Placed By
LA	100	Supplier C
LA	125	Supplier B
LA	160	Supplier A
LA	210	Supplier D
CHI	95	Supplier A
CHI	125	Supplier C
PHO	300	Supplier D
PHO	375	Supplier B
PHO	510	Supplier A
NYC	75	Supplier D
NYC	85	Supplier E
NYC	90	Supplier B
NYC	120	Supplier A
JAX	180	Supplier A

With only five lanes and five suppliers participating in the tender, the Buyer could manage all of the constraints and conditional bids outlined without much difficulty. Each alternate allocation could be produced and compared using spreadsheets or tables and the allocation resulting in the lowest costs that follows the desired constraints would be chosen. Because of the small number of constraints in our example, it would not be difficult for the Buyer to determine the optimal allocation; however, in the worst case scenario the Buyer is at risk of receiving up to $(2^5 - 1)$ or 31 different conditional bids [6]. While sorting through all of these bids would not be impossible, analyzing all of them simultaneously results in an allocation that would be more difficult and tedious to obtain.

In light of this, how much more difficult would it be for a company to determine an allocation for a tender consisting of hundreds of participants and thousands of bids? A problem of that magnitude would be nearly impossible to solve using only spreadsheets and tables. Businesses rely heavily on software that is able to handle and analyze a large number of supplier-lane combinations – possibly an exponential number of combinations. This further motivates us and leads us to the need for software which implements operations research in the form of optimization techniques. One type of software that has these and other capabilities, such as streamlining the entire tendering process from beginning to end, is the electronic auction.

2.6 Electronic Auctions

Traditionally, the procurement of logistical services involved reverse auctions that use either a request for quote (RFQ) or request for proposal (RFP) for which the buyer issues a request for services to potential suppliers, the suppliers submit either quotes or proposals, respectively, for the desired services, and the carriers are selected based only on lowest cost [4, 6, 24]. As discussed previously, determining transportation services based on the cost of an individual lane may not result in the optimal allocation because of the intertwined cost adjacent lanes have on one another throughout the network. This interdependence in a carrier's network is a problem not addressed by the traditional procurement method of a reverse auction.

Auctions performed using online software implement powerful mathematical optimization solvers and algorithms in order to determine the optimal allocation of carriers' bids. In addition to providing optimal solutions to the winner determination problem, these tools have the ability to include side constraints, business rules, and buyer preferences into the final allocation [31]. The inclusion of these features puts both the buyer and shipper at an advantage by utilizing cost saving techniques not offered through traditional procurement methods.

Obtaining an optimal solution that accounts for business constraints results in the optimal assignment of carriers to the lanes for which they are able to best serve. Awarding business to carriers in this fashion has a positive impact on the entire supply chain. If the buyer reduces the costs of its suppliers, the buyer benefits when these savings are passed on in the form of lower transportation costs. Pairing carriers with lanes optimally potentially provides the buyer with other benefits in the form of increased customer satisfaction, improvements in carrier efficiency, shorter and/or more reliable lead times, or better relations with its suppliers. These benefits may further directly or indirectly reduce costs across the supply chain.

Electronic auctions offer many other time and cost saving benefits to both the supplier and shipper. Honeywell, a United States based technology company with approximate sales of $39 billion in 2016, claims that the use of electronic sourcing tools has reduced the time it takes for suppliers to submit bids from up to three weeks to less than 72 hours [1, 11]. Because the auction is performed through an online platform, bids are received simultaneously in real-time, allowing for improved communication and synchronous negotiations with suppliers.

Online platforms offer the extra advantage of being able to better handle the complexities of combinatorial auctions through sophisticated optimization algorithms.

Determining an allocation on a small set of lanes is trivial and may be accomplished using spreadsheets; however, combinatorial auctions may result in an exponential number of possible combinations of bids. Such a problem is categorized as NP-complete: in the worst case scenario the buyer is not able to obtain an exact solution to the winner determination problem in polynomial time [26]. In these situations an approximation may be used.

Even though the use of online software in the procurement of logistical service has numerous benefits, namely monetary savings for the buyer, there are still some issues to address. In particular, software offered through a third-party provider can be viewed as a "black box" because of the lack of information regarding the optimizer and/or algorithms the software uses to obtain its solutions. While some companies may release older versions of their search algorithms, it is in the best interest of the company that the ideas and techniques behind their algorithms are kept proprietary. From the buyer's perspective, the buyer inputs information regarding the lots, the suppliers' corresponding bids, and business constraints. The software outputs the optimal allocation of lots. One of the key interests in this research is to investigate how a buyer can be assured an optimal allocation has been determined if little to nothing is known about the optimization techniques used in determining the given allocation.

3 COMBINATORIAL AUCTION OPTIMIZATION MODELS

In the following chapters we formalize the winner determination problem both as a mixed-integer linear program and as a stochastic linear program. Our goal is to solve the allocation problem by modeling the combinatorial auction as an optimization problem. Since we are interested in the auction from the buyer's perspective, the goal of the linear program is to assign carriers to lanes in such a fashion that the collection of accepted bids results in the minimal cost to the buyer. Solving this optimization problem accomplishes the buyer's goal of minimizing its total logistical costs by determining an optimal allocation of suppliers across all lanes. Using this optimization program, we then incorporate stochastic programming into the model in order to investigate how the allocation is affected by uncertain changes in future demand levels.

We proceed by first presenting a brief overview of linear programming in section 3.1, then proceed to present the reverse combinatorial auction (RCA) base optimization model, which contains only the most basic constraints that ensure an optimal allocation. We then build upon the base model in later sections by introducing various constraints that the buyer may wish to use in the tender process. Each constraint represents one or more of the buyer's business rules. Therefore any number of the constraints may be simultaneously incorporated into the optimization model to model different business scenarios.

After establishing the base model in section 3.2.1, we present an example to illustrate its formulation and implementation as a stand alone model in section 3.2.2. Then we proceed to introduce of a new business rule corresponding to the number of distinct carriers in the program and its corresponding constraint(s) as a linear program in section 3.3.1. This section is organized such that the carrier constraint model is presented as an extension of the base model and is followed with an example to illustrate the development of the model.

3.1 Introduction to Linear Programming

Optimization problems involving the maximization or minimization of a linear function subject to one or more linear constraints are classified as linear programs [21]. The linear function that is being either maximized or minimized is referred to as the objective function, and each of the constraints may be in the form of an equality or inequality [8]. Given an objective function and a set of constraints, we can write any linear program in standard form given by

$$\textit{minimize}\ c_1x_1 + x_2x_2 + \cdots + c_nx_n$$

$$\textit{subject to}\ a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i,\ i = 1 \ldots m$$

$$x_1 \geq 0, x_2 \geq 0, \ldots, x_n \geq 0$$

where the goal of the program is to find values for the variables $x_1, \ldots, x_n$ that minimize the objective function. Such a solution that minimizes the objective while satisfying all of the constraints is said to be a feasible solution [16]. Typically this is presented in a more compact form using vector notation given by

$$\textit{minimize } \mathbf{c}^T\mathbf{x} \tag{1}$$

$$\textit{subject to } A\mathbf{x} \leq \mathbf{b} \tag{2}$$

$$\mathbf{x} \geq 0 \tag{3}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the variable that minimizes the objective function, $\mathbf{c}^T \in \mathbb{R}^n$ and $A \in \mathbb{R}^{m \times n}$ are the coefficients of the objective and the constraints, respectively, and $\mathbf{b} \in \mathbb{R}^m$ is the right hand side of the constraint inequality [27]. In a concrete linear programming model, the values of $\mathbf{c}^T$ and A will be defined explicitly, and in an abstract model, the values of the coefficients will be defined implicitly and then provided by a data set at run time.

Linear programming is used extensively in the field known as operations research, which can be traced back to the first world war when it was used to improve military operations [28]. Operations research uses optimization techniques for decision-making in business and other industries involving highly complex problems. In addition to linear programming problems that involve a linear objective function and linear constraints, mixed-integer programming problems may also be found under the umbrella of operations research. These types of problems have the additional requirement that one or more of the variables are restricted to taking on integer values [30].

Numerous real world problems can be modeled as either linear programming problems or mixed-integer programming problems. Examples of these types of problems include the assignment or allocation problem of assigning objects to tasks, the maximum flow problem of determining which route across a network maximizes flow, the minimum spanning tree problem finds the connected sub-tree with the shortest distance across the network, and the shortest route problem determines the shortest distance between two nodes within a network [25].

3.2 Reverse Combinatorial Auction: Base Optimization Model

In this research, we have developed a linear program that models the highly complex problem of procurement through outsourcing. Specifically, our work is focused on business-to-business sourcing events used in transportation procurement from the perspective of the business that is purchasing the logistical services. While one of our goals is to develop a linear program that incorporates numerous business rules and constraints, the base optimization model includes complexities that require the use of optimization techniques to obtain a feasible solution to the problem. Even with the inclusion of only the basic business rules and constraints, some of the complexities the buyer may face could include the large number of suppliers, bids, and lanes, organizing an extensive national or global network, how the demand across lanes is spread out over a period of time, and the potential for bids placed on a bundle or package of lanes [34]. Therefore we first present the base optimization model as a stand alone model that is able to handle these complexities.

3.2.1 Deterministic RCA Base Model

As is evident from the examples presented in section 2.5, the sourcing event is modeled as a reverse combinatorial auction from the perspective of the buyer. Since the buyer desires to procure its needed services at the lowest possible cost, we employ a minimization optimization problem in the base model in which the objective function minimizes the cost of the set of accepted bids, which represent the purchasing of transportation services from third-party providers. Minimizing the objective function in this way ensures that any feasible solution of the model will produce the minimal costs to the buyer subject to the given set of constraints. The overall cost to the buyer is represented in the objective function as the sum of the winning bids.

Although the base model does not incorporate any business constraints, the linear program does include constraints to ensure a feasible solution is obtained for the buyer. A feasible solution must provide a set of winning bids that meet the buyer's total demand across all lanes in the network. Moreover, each bid must be accepted in an all-or-nothing fashion, and each bid cannot be awarded more than one time. For a given bid, either the entire bundle of items need be assigned to a single carrier or none of the lots are awarded to the carrier. This ensures that no partial bids are awarded. All of these considerations will be represented in the linear program as constraints for which any feasible solution satisfies.

The first and most basic combinatorial auction optimization problem we present is a linear problem consisting of an objective function and three constraints adapted from [3, 18]. We let the sets $LANES$ and $BIDS$ represent the set of all items in the auction and the set of all received bids, respectively. The indexing over these respective lanes is given by $l = 1, \ldots, m$, for $l \in LANES$, which represents item l in the auction, and $b = 1, \ldots, n$, for $b \in BIDS$, which represents bid b out of the set of all bids. Note that there are a total of m bids and n items. Each bid $b \in BIDS$ is comprised of the set $\{v_b, S_b\}$ where v_b represents the bid value or price of bid b and

S_b is the subset of items contained corresponding to a bundle/package of lanes. We assume that $S_b \neq \emptyset \; \forall \; b \in BIDS$. We represent the total demand over all lanes with D. If each lane occurs with multiplicity 1 and the total demand is distributed evenly across all lanes, then $D = m$ [30].

Referring to the procurement example outlined in section 2.5, Table 1 consisting of five lanes (LA, CHI, PHO, NYC, and JAX) with five bidders A, B, C, D, and E, and 14 bids; we may denote the lanes and bids using the set notation outlined above as $LANES = \{1, 2, 3, 4, 5\}$ and $BIDS = \{1, 2, ..., 14\}$, respectively. Then our indexing sets are $l = 1, \ldots, 5$ and $b = 1, \ldots, 14$, respectively. We also note that using this notation we have bid $b = 1$ is comprised of the set $\{100, \{1\}\}$, where the bid amount is $v_1 = 100$ and the package of lanes is the singleton set $S_1 = \{1\}$.

In standard form the program is given by

$$minimize \sum_{b=1}^{n} x_b v_b \tag{4}$$

$$subject\ to \sum_{b=1}^{n} x_b |S_b| = D \tag{5}$$

$$\sum_{b=1}^{n} \delta_{lb} x_b = d_l \quad \forall \; l = 1, \ldots, m \tag{6}$$

$$x_b \in \{0, 1\}. \tag{7}$$

The constraint given in Equation (6) refers to an incidence matrix for bids, δ where the l^{th} row corresponds to the l^{th} item and the b^{th} column, $\boldsymbol{\delta}_b$, corresponds to the b^{th} bid. Each element of the matrix δ is defined as

$$\delta_{lb} = \begin{cases} 1 & \text{if } \; l \in S_b \\ 0 & \text{if } \; l \notin S_b \end{cases}.$$

For our example, the incidence matrix for bids, is given by

$$\delta = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

where $\delta_{1,1} = 1$ represents the first lane, $l = 1$, being in the bid package of the first bid, $b = 1$, and $\delta_{2,1} = 0$ shows that the second lane, $l = 2$, is not part of the bid package for the first bid, $b = 1$. Summing over each row of the δ matrix produces the total number of bids placed on the corresponding lane, while summing over each column gives the total number of lanes in the corresponding bid package. Note that the sum of each column being equal to one shows that each bid in the program consists of only a single lane rather than bid packages of more than one lane.

Lastly, the decision variable corresponding the the b^{th} bid is given by x_b where

$$x_b = \begin{cases} 1 & \text{if bid } b \text{ is accepted} \\ 0 & \text{otherwise} \end{cases}.$$

All model variables and parameters are summarized in Table 6.

Table 6: Parameters and variables for the deterministic RCA Base Model

m:	Total number of unique lanes in the network
n:	Total number of active bids in the tender
l:	Indexing parameter over the set of all lanes, $l = 1, \ldots, m$
b:	Indexing parameter over the set of all bids, $b = 1, \ldots, n$
v_b:	Value of bid b
S_b:	Bundle of lanes in bid b
x_b:	Decision variable for bid b; 1 if bid b is accepted, 0 otherwise
δ_{lb}:	Variable for lane l; 1 if lane $l \in S_b$, 0 otherwise
D:	Total demand across all lanes in the network
d_l	Total demand on lane l

3.2.2 Base Model Example

We extend the procurement example presented in section 2.5 to include five new bid packages that consist of more than one lane to illustrate the Reverse Combinatorial Auction (RCA) model. Table 7 presents the extended data set consisting of the 19 total bids.

Table 7: Bid packages to illustrate the reverse combinatorial auction base model.

Lane(s)	**Bid Amount ($)**	**Placed By**
LA	100	Supplier C
LA	125	Supplier B
LA	160	Supplier A
LA	210	Supplier D
CHI	95	Supplier A
CHI	125	Supplier C
PHO	300	Supplier D
PHO	375	Supplier B
PHO	510	Supplier A
NYC	75	Supplier D
NYC	85	Supplier E
NYC	90	Supplier B
NYC	120	Supplier A
JAX	180	Supplier A
LA, JAX	300	Supplier A
LA, CHI, NYC	350	Supplier C
CHI, NYC	190	Supplier D
CHI, PHO, NYC, JAX	690	Supplier E
LA, CHI, PHO	450	Supplier B

The linear program still consists of five lanes (LA, CHI, PHO, NYC, and JAX) with five bidders A, B, C, D, and E, with a total of 19 bids. Denoting the lanes and bids using our set notation gives the sets $LANES = \{1, 2, 3, 4, 5\}$ and $BIDS = \{1, 2, ..., 19\}$, respectively. Then our indexing sets are $l = 1, \ldots, 5$ and $b = 1, \ldots, 19$, respectively. Bids one through 14, as presented in the previous section, are denoted in the same fashion; bid $b = 1$ is comprised of the set $\{100, \{1\}\}$, where the bid amount is $v_1 = 100$ and the package of lanes is the singleton set $S_1 = \{1\}$. Note that bid $b = 15$ consisting of two lanes is denoted by the set $\{300, \{1, 5\}\}$, where the bid amount is $v_{15} = 300$ and the package of lanes is the set $S_{15} = \{1, 5\}$ referring to lanes LA and JAX, respectively.

In this example, the incidence matrix for bids δ, delta, for the 19 bids is given by

$$\delta = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

In contrast to the δ matrix in section 3.2.1, we note that summing over columns 15 through 19 indicates the cardinality of the corresponding bid package, where $|S_{15}| = 2$, $|S_{16}| = 3$, $|S_{17}| = 2$, $|S_{18}| = 4$, and $|S_{19}| = 3$, are all greater than one.

Lastly, the linear program given by

$$\begin{aligned} \textit{minimize} \ & \sum_{b=1}^{n} x_b v_b \\ \textit{subject to} \ & \sum_{b=1}^{n} x_b |S_b| = D \\ & \sum_{b=1}^{n} \delta_{lb} x_b = d_l \quad \forall\, l = 1, \ldots, m \\ & x_b \in \{0, 1\} \end{aligned}$$

consists of 19 decision variables $\mathbf{x} = [x_1, x_2, \ldots, x_{19}]$. Solving the reverse combinatorial auction linear program results in each of the decision variables taking on the binary values of either zero or one, where $x_i = 1$ refers to the acceptance of bid $b = i$

and $x_i = 0$ results in the rejection of bid $b = i$.

3.2.3 Base Model Example: Implementation and Results

Before proceeding to the formulation and implementation of the carrier constraint model, we wish to present the results of the example data set when implemented through computer simulation. While we will discuss the details of the methods and implementation of the model(s) in chapter 5, the goal of this section is to fully complete the presentation of the example data using the base model here. Moreover, this will allow us to focus on solely the results of the larger, real world data set provided by Eastman Chemical Company in chapter 6, thus minimizing any confusion between the results of the two different data sets.

For implementation of the base model example, the Python file and the corresponding data file were executed using GLPK as the default solver. The computer code contained in the Python file used in the implementation of the base model example data set may be found in Appendix A. We refer the reader to chapter 5 for an in-depth discussion of the program's implementation and methods.

The program consisted of one objective, seven constraints, and 20 variables, for which the objective was minimized. Upon termination, one optimal solution was found with an objective value of \$705 and the resulting decision variable values were returned as $x_{10} = x_{14} = x_{19} = 1$ and $x_i = 0$, for $i \neq 10, 14, 19$. In other words, the 10^{th}, 14^{th}, and 19^{th} bids were accepted with all other bids being rejected. Table 8 represents the information pertaining to the winning bids.

Table 8: Winning bid packages upon implementation of the base model example via Pyomo.

Lane(s)	**Bid Amount ($)**	**Placed By**
NYC	75	Supplier D
JAX	180	Supplier A
LA, CHI, PHO	450	Supplier B

3.3 Reverse Combinatorial Auction: Carrier Constraints Model

3.3.1 Deterministic Carrier Constraint Model

Using the reverse combinatorial auction base optimization model given in Equations 4 through 7, we wish to incorporate one or more constraints in order to determine a minimum and/or a maximum number of different carriers in the final allocation. We begin by first formulating the carrier constraint for a maximum number of carriers as follows.

First, we introduce a new indexing set, denoted $CARRIERS$, which represents the set of all carriers in the linear program where $c = 1, \ldots, r$ indexes over the set and the total number of participants in the tender is equal to r. Two new parameters, denoted z_c and M_c are also introduced into the base optimization model to track which carriers are assigned bids and will be in the final allocation. The variable z_c tracks whether each individual carrier c will appear in the final allocation, where

$$z_c = \begin{cases} 1, & \text{if carrier } c \text{ has at least one accepted bid} \\ 0, & \text{otherwise} \end{cases},$$

and M_c denotes the total number of bids placed by carrier c [3].

For illustrative purposes, suppose carrier r places two bids, i.e., $M_c = 2$, and let

x_1 and x_2 be the corresponding decision variables, respectively. Then if either bid or both bids from carrier r are accepted, i.e., $x_1 = 1$ and/or $x_2 = 1$, then carrier c has at least one accepted bid and $z_r = 1$. Alternatively, we can write that if $x_1 + x_2 \geq 1$, then $z_c = 1$. On the other hand, if both bids are rejected we have that $x_1 + x_2 = 0$, and so $z_c = 0$. This allows us to denote z_c for this example as

$$z_c = \begin{cases} 1, & \text{if } \ x_1 + x_2 \geq 1 \\ 0, & \text{otherwise} \end{cases}.$$

We further note that because z_c is binary, we also have the inequality $x_1 + x_2 \leq M_c z_c$. Recall, z_c can only equal 0 if x_1 and $x_2 = 0$; therefore, if x_1 and $x_2 = 0$, $z_c = 0$ and hence $x_1 + x_2 = 0 \leq M_c z_c = 0$. In the case that only one $x_i = 1$, $i = 1$ or 2, then the inequality still holds as $z_c = 1$ in this case, and $x_1 + x_2 = 1 \leq M_c z_c = 2(1) = 2$. If both x_1 and x_2 equal 1, then $x_1 + x_2 = 2$ and $z_c = 1$ which implies $x_1 + x_2 = 2 \leq M_c z_c = 2$.

We can further verify these inequalities by noticing that, without loss of generality, if $x_1 + x_2 = 1$, then the first inequality is satisfied regardless of whether $z_c = 0$ or $z_c = 1$. However, in this situation the second inequality becomes $1 \leq M_c z_c = 2z_c$, which implies that $z_c \geq 1/2$. Therefore z_c must be equal to 1. Alternatively, if $x_1 + x_2 = 0$ where both bids are rejected, then this forces z_c to be equal to zero [29].

We can generalize this for a carrier c placing $M_c = n$ total bids with corresponding decision variables $x_1, x_2, \ldots, x_n$, then

$$z_c = \begin{cases} 1, & \text{if } \ x_1 + x_2 + \cdots + x_n \geq 1 \\ 0, & \text{otherwise} \end{cases}.$$

This leads to

$$z_c \leq x_1 + x_2 + \cdots + x_n \leq M_c z_c. \tag{8}$$

Similar to the base model, we also introduce a new incidence matrix for carriers, denoted γ, where the c^{th} row, corresponds to the c^{th} carrier and the b^{th} column corresponds to the b^{th} bid. Each entry in this incidence matrix for lanes is defined as

$$\gamma_{cb} = \begin{cases} 1 & \text{if carrier } c \text{ placed bid } b \\ 0 & \text{otherwise} \end{cases}.$$

For our example in section 2.5, the incidence matrix for carriers, denoted γ, is given by

$$\gamma = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

where $\gamma_{3,1} = 1$ signifies that the third carrier, $c = 3$, placed the first bid, $b = 1$, corresponding to row 3, column 1. We note that summing over each row of γ produces the total number of bids placed by a carrier, i.e., $\sum_{b \in BIDS} \gamma_{cb} = M_c \; \forall c \in CARRIERS$.

This defines a new parameter in the optimization model that serves as a counter for the total number of carriers in the final allocation. Note that we now assume each bid is of the form $\{v_b, S_b, c_b\}$, which includes the parameter c_b that is used to track which bids belong to each carrier when setting up the incidence matrix for lanes γ.

When combined with the constraints outlined in the previous section, we obtain the following linear program:

$$minimize \sum_{b=1}^{n} x_b v_b \tag{9}$$

$$subject\ to \sum_{b=1}^{n} x_b |S_b| = D \tag{10}$$

$$\sum_{b=1}^{n} \delta_{lb} x_b = d_l \quad \forall\, l = 1, \ldots, m \tag{11}$$

$$\sum_{b=1}^{n} \gamma_{cb} x_b - M_c z_c \leq 0 \quad \forall\, c = 1, \ldots, r \tag{12}$$

$$z_c - \sum_{b=1}^{n} \gamma_{cb} x_b \leq 0 \quad \forall\, c = 1, \ldots, r \tag{13}$$

$$\sum_{c=1}^{r} z_c \geq minCarrier \tag{14}$$

$$\sum_{c=1}^{r} z_c \leq maxCarrier \tag{15}$$

$$x_b \in \{0, 1\} \quad \forall\, b = 1, \ldots, n \tag{16}$$

$$z_c \in \{0, 1\} \quad \forall\, c = 1, \ldots, r. \tag{17}$$

The model variables and parameters are summarized in Table 9.

In the carrier constraint model given by Equations (9) – (17), it can be observed that the constraint given by Equation (10) is the same as in the base model, which requires the total demand over all lanes to be met by the accepted bids. Likewise the constraint given by Equation (11) requires the total demand on each lane to be met by the accepted bids. Constraints given by Equations (12) and (13) use the inequality established in Equation 8, $z_c <= x_1 + ... + x_n <= M_c z_c$ for each carrier c. In other words, for each carrier c, z_c must satisfy this constraint if carrier c is present in a winning bid (indicated by the incidence matrix for lanes, γ). Next we

Table 9: Parameters and variables for the RCA Model with maximum carrier constraints.

m:	Total number of unique lanes in the network
n:	Total number of active bids in the tender
j:	Total number of unique carriers in the tender
l:	Indexing parameter over the set of all lanes, $l = 1, \ldots, m$
b:	Indexing parameter over the set of all bids, $b = 1, \ldots, n$
r:	Indexing parameter over the set of all carriers, $c = 1, \ldots, r$
v_b:	Value of bid b
S_b:	Bundle of lanes in bid b
c_b:	Carrier of bid b
x_b:	Decision variable for bid b; 1 if bid b is accepted, 0 otherwise
z_c:	Decision variable for carrier c; 1 if carrier r has at least one Accepted bid, 0 otherwise
δ_{lb}:	Variable for lane l; 1 if lane $l \in S_b$, 0 otherwise
γ_{cb}:	Variable for carrier c; 1 if carrier c placed bid b, 0 otherwise
M_c:	Total number of bids placed by carrier r
D:	Total demand across all lanes in the network
d_l:	Total demand on lane l
$maxCarrier$:	Maximum number of different carriers
$minCarrier$:	Minimum number of different carriers

let the predetermined maximum number of carriers the buyer wishes to have in the final allocation be denoted by $maxCarrier$, and the predetermined minimum number of carriers be denoted as $minCarrier$, where Equations (14) and (15) represent the constraints that requires the maximum and minimum number of carriers to be met by the total number of different carriers, respectively. Finally, the constraints represented by Equations (16) and (17) require the decision variables to only take on the values of either zero or one corresponding to a decision of inclusion for a value of one and exclusion for a value of zero.

If only one of the two carrier constraints needs to be implemented into the model rather than restricting both the minimum and maximum number of constraints, a simple modification is necessary. If a buyer requires a minimum number of carriers but not a maximum, then the buyer would set $maxCarrier = a$, where a is the total number of carriers in the program. On the other hand, if a buyer desires a maximum number of carriers but not a minimum, then let $minCarrier = 0$.

Thus we have outlined how to incorporate a constraint that allows for a restriction on the maximum or minimum number of carriers in the program. We proceed by presenting an example of the Deterministic Maximum Carrier Constraint Model in the next section.

3.3.2 Carrier Constraint Model Example

We again refer to the extension the procurement example in section 3.2.2 where the bid packages are given again in Table 10. All of the variable and parameter values presented in the base model example found in section 3.2.2 have not been changed, but now we consider the inclusion of the carrier constraints on the linear program, for which we must consider information regarding the carriers. Our new indexing set corresponding to the five suppliers A, B, C, D, and E, respectively, is given by $CARRIERS = \{1, 2, \ldots, 5\}$ with indexing set $c = 1, \ldots, 5$. Additionally, we denote the total number of bids placed by each carrier as the vector M_c, where we have $M_c = [6, 4, 3, 4, 2]$. In this example the incidence matrix γ is given by

$$\gamma = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Table 10: Bid packages used in the RCA Base Model example.

Lane(s)	Bid Amount ($)	Placed By
LA	100	Supplier C
LA	125	Supplier B
LA	160	Supplier A
LA	168	Supplier D
CHI	95	Supplier A
CHI	125	Supplier C
PHO	240	Supplier D
PHO	375	Supplier B
PHO	510	Supplier A
NYC	68	Supplier D
NYC	75	Supplier E
NYC	90	Supplier B
NYC	120	Supplier A
JAX	180	Supplier A
LA, JAX	300	Supplier A
LA, CHI, NYC	350	Supplier C
CHI, NYC	190	Supplier D
CHI, PHO, NYC, JAX	690	Supplier E
LA, CHI, PHO	450	Supplier B

where $\gamma_{i,j} = 1$ if carrier $c = i$ placed bid $b = j$, and $\gamma_{i,j} = 0$ if carrier $c = i$ did not place bid $b = j$. We can see that the sum of each row equals the corresponding value of M_c. For example, summing over the first row represents the six bids placed by carrier $c = 1$ and corresponds to $M_1 = 6$.

In addition to the decision variable x_i in the base model, the carrier constraint model also contains the decision variable z_c where a value of 1 corresponds to carrier c belonging to the set of carriers with at least one bid in the final allocation. Otherwise a value of 0 corresponds to carrier c does not have have any accepted bids and will be appear in the final allocation.

3.3.3 Carrier Constraint Model Example: Implementation and Results

Next we conclude the presentation of the carrier model example using the data presented in Table 10 by presenting the several different results of the program for various restrictions on the maximum and/or minimum number of carriers. We have included the Pyomo code from Python file in Appendix B for which the reader may reference.

First we note that for $maxCarrier = 3$, the program confirms the results presented in section 8 for which suppliers A, B, and D were chosen to participate in the final allocation for a total cost of \$705. This was established by the optimal solution by setting the decision variables $z_1 = z_2 = z_5 = 1$ and $z_3 = z_4 = 0$. Hence we may change the values of the maximum number of carriers and/or minimum number of carriers to observe changes in the optimal allocation.

Let $maxCarrier = 2$. Then the resulting linear program consists of one objective, 19 constraints, and 25 decision variables. Upon termination of the program, one solution was obtained that satisfied all of the constraints and minimized the objective. The resulting total cost of the allocation was found to be \$720 with $x_{12} = x_{14} = x_{19} = 1$ representing the winning bids and $x_i = 0$ for $i \neq 12, 14, 19$ representing the losing bids. Moreover, we have $z_1 = z_2 = 1$ for the two carriers that will be in the final allocation and $z_3 = z_4 = z_5 = 0$ for those carriers that will not be participating in the allocation. The information regarding the winning bids is presented in Table 11. We may conclude that for $maxCarrier = 2$, we have suppliers A and B as the only carriers, which has caused an increase in the tender cost due to the restriction imposed by the maximum number of carriers.

Table 11: Winning bids for the carrier constraint example with $maxCarrier = 2$.

Lane(s)	**Bid Amount ($)**	**Placed By**
NYC	90	Supplier B
JAX	180	Supplier A
LA, CHI, PHO	450	Supplier B

Suppose that we wish to allocate all of the volume to the same carrier by letting $maxCarrier = 1$. Then the resulting optimal solution allocates all five lanes to supplier A for a total cost of $1,025. Our optimization solver provides the solution given by $x_5 = x_9 = x_{13} = x_{15} = 1$ and $x_i \neq 1$ for the remaining decision variables corresponding to the winning and losing bids, and for the carriers, $z_1 = 1$ while $z_2 = z_3 = z_4 = z_5 = 0$. Hence bids 3, 5, 13, and 15 placed by supplier A allows for all of the demand to be met by a single carrier. We summarize the winning bids in Table 12.

Table 12: Winning bids for the carrier constraint example with $maxCarrier = 1$.

Lane(s)	**Bid Amount ($)**	**Placed By**
CHI	95	Supplier A
PHO	510	Supplier A
NYC	120	Supplier A
LA, JAX	300	Supplier A

Note in each of these previous instances, only the maximum carrier constraint is activated, which is achieved by setting $minCarrier = 0$.

Next let $minCarrier = 4$. Here we only want to activate the minimum number of carriers, so we may set $maxCarrier = 5$. The optimal solution returns decision variable values of $x_1 = x_5 = x_7 = x_{10} = x_{14} = 1$ and $z_1 = z_3 = z_4 = z_5 = 1$ with the remaining decision variables taking on the value of zero. The total allocation cost is \$760 when four different shippers are issued accepting bids. Table 13 summarizes this simulation's results.

Table 13: Winning bids for the carrier constraint example with $minCarrier = 4$.

Lane(s)	**Bid Amount (\$)**	**Placed By**
LA	100	Supplier C
CHI	95	Supplier A
PHO	300	Supplier D
NYC	85	Supplier E
JAX	180	Supplier A

4 STOCHASTIC PROGRAMMING

4.1 Overview of Stochastic Programming

The deterministic model bases its outcomes on data for which the demand values are estimated. However, in reality a business does not know with certainty that these predictions will be reliable. When dealing with uncertainty in one or more of the model parameters, such as demand on individual lanes, an alternate approach is to develop a stochastic optimization program. Stochastic programming implements uncertainty in parameter values by using the expected value(s) of the parameter in the objective function, and then either minimizes or maximizes over the expected value similarly to the deterministic model [17]. In regards to the implementation of stochastic programs for the use of transportation procurement, there are two common approaches: two-stage stochastic programming and simulation via K-scenarios [20].

Transportation procurement models in the form of the winner determination problem utilize unknown values of future demand levels across all lanes in a network. Although there is uncertainty in the needed shipment levels, a business must make decisions regarding which carriers to assign to each lane using an estimated value of demand. Only after the decision has been made and enforced and after a period of time has elapsed are the true demand values realized. Issues occur that may be very costly to the shipper when the actual and estimated demand values greatly differ [12].

Typically two outcomes occur when the estimated demand values do not match the actual demand. If the actual demand values are less than the predicted values, the carrier may lose revenue because of the lower volume that needs to be shipped. On the other hand, if the actual demand values exceed the predicted value, there may be an outcome such that the carrier is unable to ship the excess volume, and the shipper is faced with a new set of decisions regarding how to transport the excess demand, possibly at a higher cost by outsourcing to another third-party logistics provider [22]. In either case, once the actual volume on each lane is known, the shipper must make a recourse decision if the true demand differs from the estimated values.

One approach of dealing with uncertainty through stochastic programming is the two-stage stochastic linear program. In this optimization problem there are two decision variables referring to each of the two stages: the first set of decisions are made under uncertainty using the estimated demand values, while the second set of decisions, often referred to as recourse decisions, are made after the true demand values are known without the presence of uncertainty [22]. The classical two-stage stochastic linear program introduces uncertainty into the deterministic model as the expectation of the cost of the recourse action and is given in [15] by

$$\begin{aligned} &\textit{minimize } \mathbf{c}^T\mathbf{x} + \mathbb{E}_D[\min(\mathbf{q}^T\mathbf{y})] \\ &\textit{subject to } A\mathbf{x} = \mathbf{b} \\ &\qquad\qquad T\mathbf{x} + W\mathbf{y} \leq \mathbf{h} \\ &\qquad\qquad \mathbf{x} \geq 0, \quad \mathbf{y} \geq 0 \end{aligned}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the first-stage decision variable made before the actual demand values are known, and $\mathbf{y} \in \mathbb{R}^m$ is the second-stage decision variable based on the parameters, $\mathbf{q}, T, W, \mathbf{h}$, and the cost of the recourse action $\min(\mathbf{q}^T y)$, is subject to the constraint $T\mathbf{x} + W\mathbf{y} \leq \mathbf{h}$ that is activated when the predicted and realized demand values differ [32].

We can observe that the objective function of the classical two-stage stochastic linear program is comprised of a deterministic part, $\mathbf{c}^T\mathbf{x}$, and a stochastic part, $\mathbb{E}_D[\min(\mathbf{q}^T\mathbf{y})]$. In this generalized form, each of these parts represent two different decisions to be made: which bids to accept/reject and how much volume to allocate to those carriers with accepted bids. The coefficients of the objective function are given by $\mathbf{c}^T$ and $\mathbf{q}^T$, while the coefficients of the constraints for which the objective must satisfy are represented by A, $\mathbf{b}$, T, W, and $\mathbf{h}$. While there is no change in the definition of the first-stage decision variable, $\mathbf{x}$, which signifies those bids the carrier should accept corresponding to a forecasted demand value, the introduction of the second-stage decision variable, $\mathbf{y}$, allows the shipper to determine how much volume should be assigned to each carrier on a given lane using the actual demand values. Note that for the type of transportation procurement problem we are interested in formulating, we do not consider the amount of volume to allocate to each carrier, rather we only assume that the stochastic portion will include the total demand value as the random variable.

In the transportation procurement problem, we let the random variable in the stochastic program be the sum of the demand on the individual lanes represented by the overall demand,

$$D = \sum_{l \in LANES} d_l.$$

Hence the total demand as the random variable D_k is defined as the sum of the demand on the individual lanes for the k^{th} variable such that

$$D_k = \sum_{l \in LANES} d_{lk}.$$

If the shipper views the overall demand as a random variable with some known probability distribution, then the implementation of the stochastic program can be performed using a K-scenarios approach. Specifically, if our random variable can take on a finite number of different values, K, such that

$$D \in \{D_1, D_2, \ldots, D_K\}$$

with corresponding probabilities $p_1, p_2, \ldots, p_K$, where the probability of the k^{th} demand value is given by $Pr(k) = P(D = D_k) = p_k$, then the Law of Large Numbers allows us to assume a uniform distribution with a summation of the average cost of the recourse action in order to replace the expectation of the cost of the recourse action with a summation of the average cost of the recourse action [17, 32]. Thus we have that

$$\mathbb{E}_D[\min(\mathbf{q}^T\mathbf{y})] = \sum_{k=1}^{K} p_k \mathbf{q}^T \mathbf{y_k},$$

where each random demand value D_k corresponds to the decision variable $\mathbf{y_k}$ for the k^{th} variable where $\mathbf{y_k} = [y_{1k}, y_{2k}, \ldots, y_{mk}]$, with the objective coefficients $\mathbf{q}^T$ given by the bid values, $\mathbf{v}^T$. Essentially this allows us to reduce the two-stage stochastic linear program to

$$\begin{aligned} &\textit{minimize } \mathbf{c}^T\mathbf{x} + \sum_{k=1}^{K} p_k \mathbf{q}^T \mathbf{y_k} \\ &\textit{subject to } A\mathbf{x} = \mathbf{b} \\ &\qquad T\mathbf{x} + W\mathbf{y_k} \leq \mathbf{h_k} \quad k = 1, \ldots, K \\ &\qquad \mathbf{x} \geq 0, \quad \mathbf{y_k} \geq 0 \end{aligned}$$

where we utilize the weighted average in the objective function.

Furthermore, we can proceed using the sample average approximation where for a large enough value of K and if the set of random variables $D_k, k = 1, ..., K$ is independent and identically distributed across the scenarios, then the probability of each of our demand values is given by $p_k = 1/K$ for each $k = 1, \ldots, K$ [17]. We assume D_k is independent and identically distributed and that k is sufficiently large so that we may assume a uniform distribution. Hence we may make this substitution in our objective function to obtain

$$\begin{aligned} &\textit{minimize } \mathbf{c}^T\mathbf{x} + \sum_{k=1}^{K} \frac{1}{K} \mathbf{q}^T \mathbf{y_k} \\ &\textit{subject to } A\mathbf{x} = \mathbf{b} \\ &\qquad T\mathbf{x} + W\mathbf{y_k} \leq \mathbf{h_k} \quad k = 1, \ldots, K \\ &\qquad \mathbf{x} \geq 0, \quad \mathbf{y_k} \geq 0 \end{aligned}$$

for random variable of the continuous type.

Our main concern with creating a stochastic linear program for tending procurement is to not only analyze how changes in the predicted demand values affect the overall cost, but specifically consider how variation among those lanes with the greatest predicted demand values affects cost. We proceed in section 4.2 by formulating a stochastic linear program of the base model presented in sections 3.2.1 and 3.2.2 for a tendering auction. Much of the focus for the remaining paper will be implementation of the stochastic linear program base model; however, data containing combinatorial bids is not included in the implementation of the base model in this paper and is left as an area of future work. Section 4.3 contains the formulation of the stochastic carrier constraint model. The methods and implementation of the stochastic base model can be found in chapter 5.

4.2 Stochastic Linear Program: Base Model

Based on the deterministic linear program presented in section 3.2.1, we wish to introduce a way to model uncertainty in the estimated demand values. According to the methods previously outlined, we achieve this through the optimization of the expected value of demand in the form of a summation using a K-scenarios approach and the sample average approximation. Our assumptions under this model are that our choice of K is sufficiently large, we have a known probability distribution for the random variable representing the demand on a lane, d_l, and the set of random variables for total demand, D_k, is independent and identically distributed across all scenarios.

For the stochastic linear program, if we have that the coefficients of the objective function, $\mathbf{c}^T$ and $\mathbf{q}^T$, are assigned to be $\mathbf{c} = [0, 0, \ldots, 0]$ and

$$\mathbf{q} = \mathbf{v} = [v_1, v_2, \ldots, v_n],$$

respectively, and if the coefficients for the constraints are given by the matrices T and W such that T is defined to the zero matrix and

$$W = \begin{bmatrix} \mathbb{1}^T \\ \delta \end{bmatrix} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \delta_{11} & \delta_{12} & \dots & \delta_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{m1} & \delta_{m2} & \dots & \delta_{mn} \end{bmatrix}$$

where $\mathbb{1}$ is the ones vector and δ is the incidence matrix for lanes. Moreover, $\mathbf{h_k}$ is a vector of the right hand side of the constraints such that

$$\mathbf{h_k} = \begin{bmatrix} D_k \\ \mathbf{d_k} \end{bmatrix} = \begin{bmatrix} D_k \\ d_{1k} \\ d_{2k} \\ \vdots \\ d_{mk} \end{bmatrix}$$

where $\mathbf{h_k}$ will vary with the set of random variables, D_k. Then we have the following base model of the stochastic linear program given as

$$\begin{aligned} &minimize \sum_{k=1}^{K} \sum_{b=1}^{n} \left(y_{bk} v_b / K \right) \\ &subject\ to \sum_{b=1}^{n} y_{bk} = D_K \quad \forall\, k = 1, \dots, K \\ &\qquad\qquad \sum_{b=1}^{n} \delta_{lb} y_{bk} = d_{lk} \quad \forall\, l = 1, \dots, m, \quad \forall\, k = 1, \dots, K \\ &\qquad\qquad y_{bk} \in \{0, 1\} \end{aligned}$$

where K is the total number of scenarios, or samples of the random variable for the set of demand values D_k, in the program, and the decision variable is $\mathbf{y_k}$.

We now include double indexed variables and parameters to represent values corresponding to one sample (i.e., scenario) of the random variable. The parameter for the demand on each lane, d_{lk}, is not only indexed by the corresponding lane, l, but it is also associated with a specific scenario as shown by the index k. Moreover, our decision variable given by y_{bk} represents the decision to either reject or accept bid b for scenario k, where the random demand value of the lane corresponding to bid b may cause a bid to be accepted in some scenarios and rejected in others, thus changing the outcome of the linear program. The overall demand value, given by D_k, which is the sum of the individual demand across all lanes, will also vary as the demand values on each lane are sampled. The remaining parameters are left unchanged as their values will not differ from one sample of the random variable to another. A summary of the definitions of these values can be found in Table 14.

Table 14: Parameters and variables for the stochastic RCA Base Model

m:	Total number of unique lanes in the network
n:	Total number of active bids in the tender
K:	Total number of scenarios obtained by sampling the random variable
l:	Indexing parameter over the set of all lanes, $l = 1, \ldots, m$
b:	Indexing parameter over the set of all bids, $b = 1, \ldots, n$
k:	Indexing parameter over the set of all scenarios, $k = 1, \ldots, K$
v_b:	Value of bid b
y_{bk}:	Decision variable for bid b and scenario k; 1 if bid b for scenario k is accepted, 0 otherwise
δ_{lb}:	Variable for lane l; 1 if lane $l \in S_b$, 0 otherwise
D_k:	Total demand for scenario k across all lanes in the network
d_{lk}	Total demand on lane l for scenario k

4.3 Stochastic Linear Program: Carrier Constraint Model

We next present an extension of the base model of the stochastic linear program to include constraints for the maximum and/or minimum number of different carriers to appear in the program. Similarly to the methods used to include the constraints in the deterministic model by building on the base model, we introduce a second decision variable that will be doubly indexed according to the corresponding carrier, c, and the corresponding scenario, k such that both of the decision variables will depend on a sample of the random variable. All of the assumptions used in the base stochastic linear program also apply to the carrier constraint stochastic linear program.

The stochastic carrier constraint model is given by

$$\begin{aligned}
minimize \; & \sum_{k=1}^{K}\sum_{b=1}^{n}\big(y_{bk}v_b/K\big) \\
subject\ to \; & \sum_{b=1}^{n} y_{bk} = D_K \quad \forall\, k = 1,\dots,K \\
& \sum_{b=1}^{n} \delta_{lb} y_{bk} = d_{lk} \quad \forall\, l = 1,\dots,m, \;\; \forall\, k = 1,\dots,K \\
& \sum_{b=1}^{n} \gamma_{cb} y_{bk} - M_c z_{ck} \leq 0 \quad \forall\, c = 1,\dots,j, \quad \forall\, k = 1,\dots,K \\
& z_{ck} - \sum_{b=1}^{n} \gamma_{cb} y_{bk} \leq 0 \quad \forall\, c = 1,\dots,j, \quad \forall\, k = 1,\dots,K \\
& \sum_{c=1}^{j} z_{ck} \geq minCarrier \quad \forall\, k = 1,\dots,K \\
& \sum_{c=1}^{j} z_{ck} \leq maxCarrier \quad \forall\, k = 1,\dots,K \\
& y_{bk} \in \{0,1\} \\
& z_{ck} \in \{0,1\}
\end{aligned}$$

where K is again the total number of scenarios in the program. Similar to the deterministic carrier constraint model, we introduce the decision variable z_{ck} corresponding to carrier c for scenario k, which will appear in the final allocation for a value of 1 as determined by the optimal solution of the program. The formulation of the remaining constraints and parameters does not change from the deterministic model found in section 3.3.1. The variable and parameter definitions for the stochastic carrier constraint model can be found in Table 15.

Table 15: Parameters and variables for the stochastic RCA Carrier Constraint Model

m:	Total number of unique lanes in the network
n:	Total number of active bids in the tender
j:	Total number of unique carriers in the tender
K:	Total number of scenarios obtained by sampling the random variable
l:	Indexing parameter over the set of all lanes, $l = 1, \ldots, m$
b:	Indexing parameter over the set of all bids, $b = 1, \ldots, n$
r:	Indexing parameter over the set of all carriers, $c = 1, \ldots, r$
k:	Indexing parameter over the set of all scenarios, $k = 1, \ldots, K$
v_b:	Value of bid b
c_b:	Carrier of bid b
y_{bk}:	Decision variable for bid b and scenario k; 1 if bid b for scenario k is accepted, 0 otherwise
z_{ck}:	Decision variable for carrier c and scenario k; 1 if carrier c has at least one accepted bid, 0 otherwise
δ_{lb}:	Variable for lane l; 1 if lane $l \in S_b$, 0 otherwise
γ_{cb}:	Variable for carrier c; 1 if carrier c placed bid b, 0 otherwise
M_c:	Total number of bids placed by carrier c
D_k:	Total demand for scenario k across all lanes in the network
d_{lk}	Total demand on lane l for scenario k
$maxCarrier$:	Maximum number of different carriers
$minCarrier$:	Minimum number of different carriers

5 METHODS AND IMPLEMENTATION

5.1 Software and Packages

In this chapter we discuss the methods used to implement the models and discuss in detail how to solve reverse combinatorial auction for a set of data. Implementation of the deterministic and stochastic models was completed in the cloud through the collaborative software system CoCalc by SageMath Incorporated [14]. Optimization techniques primarily used the Anaconda Python 3 distribution and relied heavily on Pyomo: Python Optimization Modeling Objects, which is designed specifically for the use of modeling and analyzing complex optimization problems through the use of its built-in modeling objects and solvers [9].

Python was chosen as the main programming language not only because of its object-oriented abilities, but also because of the extensive number of libraries and software packages that are readily available through its open-source nature. Furthermore, because Pyomo supports the object-oriented structure and can be used in conjunction with Python and its myriad of packages and libraries, the entire modeling process was completed seamlessly within a single Jupyter notebook. In addition to the capabilities brought by the Pyomo software package, the usage of Python's data analysis library pandas allowed for straightforward methods to upload data used to instantiate the optimization model's sets and parameters by employing the library's built-in data structures [23]. Additionally, NumPy and SciPy packages were utilized for there scientific computing capabilities on the resulting optimization solution data. Lastly, the Matplotlib plotting library allowed for the use of high quality visualization and plotting tools.

We proceed to discuss the format of the data used and then outline the general steps of the computational modeling process. Note that slight modifications of the process are required depending on the model and the modeler's preferences, such as in regards to the input data format, inclusion of certain constraints, desired output format, data analysis, etc. The data used for this research was presented within an excel spreadsheet consisting of three worksheets for the lots, bids, and vendors data, respectively. All of the code implemented using a Jupyter notebook for the Eastman data set can be found in Appendices C and D.

5.2 Data and Preprocessing Steps

Before we proceed to the computational steps of the modeling process, we first discuss the characteristics of the data set and the prepocessing steps taken to ensure the data was in a compatible format. All of the example data provided throughout the chapter represents data used in a European bulk truck tender. While data was provided by Eastman Chemical Company for the use of the stochastic programming optimization model, we will also be referring to example data as all data from Eastman Chemical Company used in this thesis is confidential. Moreover, the data provided does not include bid packages, but each bid corresponds to a single lane. Appendix C contains the code for the deterministic models in section 3.2.1 for which the data consists of packaged bids; otherwise, it will be assumed that any data discussed or referred to in this and the following chapters will not consist of packaged bids. The implementation methods provided here will serve as guidelines for the use of packaged bids if the reader so desires.

Excel spreadsheets are used to organize the tender data and consist of three worksheets for information regarding the lots (i.e., the lanes), the submitted bids, and the carriers participating in the tender, where each row of the worksheet represents a single lane, a single bid, or a single vendor, respectively. Data pertaining to the lanes includes information on the lane identification number, the proportion of the minimum and maximum number of shares of the total volume to be allocated, the total volume represented as number of tanks, and the current rate for transport on a lane, if such information is available to the shipper. Data on the submitted bids include an identification number for each bid, which consists of the corresponding lane number and carrier name, the name of the carrier (i.e., the logistics service provider), and the bid amount for one unit of volume on the corresponding lane. The carrier worksheet consists of the shipper's name, the minimum and maximum number of tanks to be allocated to the carrier, and if the carrier is considered active, where the active variable might serve as a constraint to automatically include or exclude a vendor if desired.

For our computational needs and due to the form of the data set received by our liason at Eastman Chemical Company, three modifications were applied to the data set in the preprocessing stage, which we will outline here. Firstly, because the data set contained duplicates bids that were able to be ignored, we removed all of those duplicated bids. This was completed manually in the preproccessing stage rather than in the Jupyter notebook to eliminate the need to execute unnecessary commands each time the linear program was processed. Next we removed any vendors from the data set that did not submit any bids in the tender. This was done because of errors it was causing in the linear program and had no effect on the final outcomes as those carriers would not have any bids to consider.

In the final preprocessing stage, we chose to include a dummy or ghost vendor in the data set with bids on every lane that were much higher than the actual submitted bids. These bids were created in order to represent those lanes that would not be chosen as being optimally serviced by a participating vendor in the tender. In this fashion, the shipper would have information regarding which lanes would not appear in the allocation so that they would be able to take further action to provide a logistical provider whether it be through a second tender or through individual negotiations with suppliers. The creation of a ghost vendor is the most important preprocessing step because of the implications that would arise if a lane was not allocated to any of the participating vendors during the initial tender.

5.3 Modeling Process

The start of each project within a Jupyter notebook on the CoCalc server begins with importing the main libraries previously mentioned, along with the entirety of the Pyomo module. Next we create an instance of a pyomo model for the implementation of the linear program. Here we have chosen to define it as a concrete model because, although the model will be implemented in pyomo in its abstract form, the data is to uploaded and fed into the model at the time of its creation inside of the Jupyter notebook.

The data file(s) were uploaded next, and a Pandas DataFrame was created for each

	Rate	Logistics Service Provider	LaneID	ModelRateCUR
0	1@Carrier1	1	1	160
1	2@Carrier1	1	2	95
2	3@Carrier1	1	3	510
3	4@Carrier1	1	4	200
4	5@Carrier1	1	5	180
5	1@Carrier2	2	1	125
6	2@Carrier2	2	2	70
7	3@Carrier2	2	3	375

Figure 1: The first eight rows of the bids DataFrame using an example data set.

worksheet in the excel file. Figure 1 illustrates a snippet of the DataFrame containing the submitted bids where each row represents a single bid and the columns represent the corresponding variable values of the bid: the bid's identification key, the name of carrier that placed the bid, the lane number, and the rate per unit volume at which the carrier is willing to service the lane. Note that the example data set is more generic than data provided for an actual tender; here the logistics service providers are simply numbered one through eight rather than given more realistic names.

Next a DataFrame was built for the program's parameter(s) δ and γ (only if the carrier constraint was being included in the linear program). Each of these DataFrames were then used to create dictionaries of the δ and γ values, respectively, which were then used to initialize the corresponding pyomo model parameters that were to be used in buidling the optimization program. Recall that the δ parameter gives information regarding which lanes are in a bid package to the linear program, while γ gives information regarding which carriers placed which bids. Both of these parameters were used in defining the constraints of the linear program.

In a similar fashion, the DataFrames containing the bids and the lanes were used to construct dictionaries of the bid values and the total volume on each lane, respectively.

Each of these dictionaries were then used to initialize two pyomo model parameters where the bidValue parameter contains the value of each bid and the demand parameter contains the total volume on each lane. Since each of these parameters is defined as a vector in the theoretical model, the size of each of the parameters was defined by a set object in pyomo where the size of each set is determined by the corresponding DataFrame's index values.

Additional parameters, such as the total demand, which is represented here as the total number of lanes in the program, and the maximum and/or minimum number of carriers, were defined as would be required for the program's constraints. If the carrier constraints were to be included in the linear program, an array containing the total number of bids placed by a each carrier, represented by M in the carrier constraint model, also had to be created. This array was then used to initialize the pyomo model parameter.

Here we can see the overall trend or steps needed thus far in developing the computational model(s) in pyomo out of the abstract model(s) presented previously. For each parameter in the abstract model, we first define the parameter in python using a suitable data structure, either an array for those parameters that are single indexed or a dictionary for the double indexed parameters. These python objects are then used to initialize the pyomo model object that represents the theoretical parameter. These parameters will then be used to build the objective function and constraints that make up the optimization model.

In addition to defining and initializing each of the parameters as a pyomo model parameter object, we also define the decision variable(s). The base model includes the decision variable, $\mathbf{x}$, for each bid in the program. The carrier constraint model includes not only the decision variable given by $\mathbf{x}$, but also includes a second decision variable, $\mathbf{z}$, for each carrier in the program. Unlike the parameter indices, the number of indices for the decision variables, $\mathbf{x}$ and $\mathbf{z}$, may be either single or double indexed if the deterministic or stochastic model, respectively.

Once all of the parameter and variable model objects are created in pyomo, the final step in establishing the pyomo model is to define the objective function and all of the constraints. In pyomo this is achieved by writing a function for each object that will return the expression given in the linear program. Then the pyomo model objects are created using the built-in Objective and Constraint functions.

After the entire linear program has been created in pyomo, the model is ran through one of pyomo's built-in solvers. For our research, we used the GNU Linear Programming Kit (GLPK) as the default solver.

6 RESULTS

We now proceed to present several results of the deterministic and stochastic models using the data set provided by Eastman Chemical Company. As mentioned in section 5.2, because the data is confidential, we will not discuss any specifics of the data. Any references to exact lanes, bids, and/or carriers in the data set will be done using an alias.

While the majority of the results presented here deal with the inclusion of variation in the base model through implementation of the stochastic base model, we first briefly discuss the results of the deterministic base model to use as a basis of comparison for the stochastic model results. We then present results that include variation on only the lane with the highest estimated demand value, followed by results that include variation on the two lanes with the highest demand values. In each of these two simulations, we will discuss how the total cost changes through the inclusion of random demand values, the distribution of the random demand values on those lanes with varying demand, and compare the variation on the allocation's total cost when one lane versus two lanes have random demand. Lastly, we do include on result of the deterministic carrier constraint model, but we leave the implementation of the stochastic carrier constraint model for an area of possible future work.

6.1 Deterministic Model

For confidentiality reasons, all results presented in the following sections will refer to the suppliers by an alias name. The original data set contains 234 lanes that must be allocated among 17 suppliers that placed bids on one or more of the lanes. This shippers will be referred to as Supplier 1, Supplier 2, etc. As discussed in section 5.2, one additional supplier was added to the data set, referred to as a dummy vendor, for the purpose of ensuring a feasible solution to the linear program would be found. We now proceed to present the results of the base model and then the carrier constraint model for the deterministic demand values.

6.1.1 Base Model

Upon obtaining the feasible solution to the deterministic base model linear program, the results show that the 234 lanes are optimally allocated among 15 of the 17 suppliers that participated in the tender. We found that the dummy vendor was not assigned any lanes, which is desirable and favorable for the buyer since this implies that all of the lanes will be serviced in the initial tender without the need for a second round of bidding. The total cost of the deterministic base model to service all 234 lanes is $8,241,155.

A summary of the results for each supplier may be found in Table 16, which contains the total number of lanes assigned to each supplier and the total cost to service all of the lanes assigned to the respective supplier. Note that the rate on the individual lanes varies and is left undisclosed. Furthermore, there are two carriers that do not appear in Table 16, Supplier 12 and Supplier 16, as a result of not procuring any lanes in the tender process.

Table 16: Summary of the results of the winning carriers for the deterministic base model.

Carrier	Lane Count	Total Volume	Total Rate
Supplier 1	7	109	$177,636
Supplier 2	14	65	$224,290
Supplier 3	48	990	$225,5901
Supplier 4	6	32	$50,387
Supplier 5	23	372	$698,971
Supplier 6	28	251	$617,639
Supplier 7	1	470	$1,681,190
Supplier 8	11	124	$335,531
Supplier 9	14	195	$426,551
Supplier 10	3	51	$140,050
Supplier 11	2	10	$14,325
Supplier 13	16	65	$121,685
Supplier 14	24	340	$552,031
Supplier 15	11	167	$426,549
Supplier 17	26	317	$518,419

6.1.2 Carrier Constraints Model

For the carrier constraint model using the deterministic demand values, our results consider only the activation of the maximum number of constraints and the change in the tender's total cost due to the inclusion of this restriction. It can be observed from Figure 2 that there is a negative association between the number of carriers in the final allocation with the tender's total cost. As the restriction on the maximum number of carriers is lessened, the cost of the tender decreases.

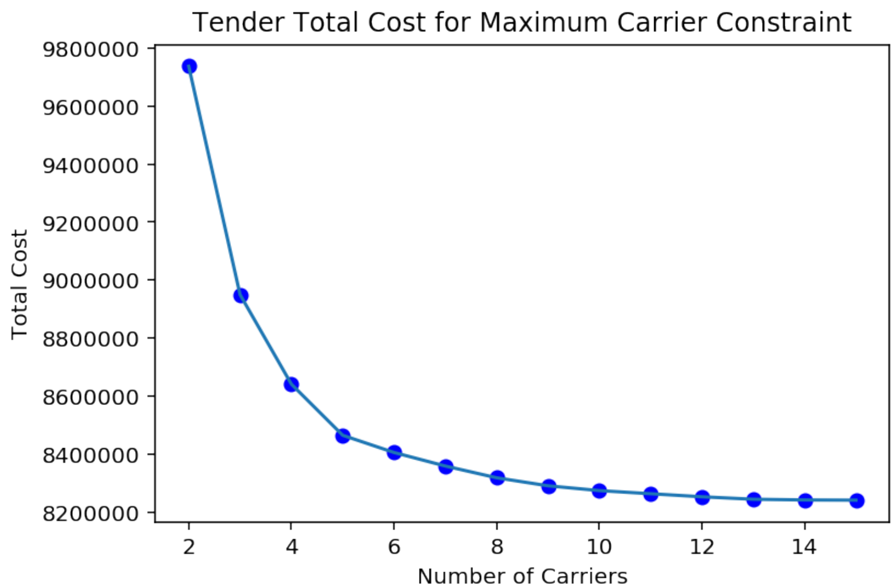

Figure 2: Maximum number of carriers versus total allocation cost.

6.2 Stochastic Base Model

For the stochastic base model, our primary goal is to introduce variation first on those lanes with the highest demand in order to observe how unknown changes in the demand on those lanes would affect the overall cost of the allocation. We focus on the lanes with the highest demand, because we expect to be able to attribute the greatest changes in the total cost to those lanes. Specifically, we are interested in the lanes with a total demand volume of 200 tanks or more. In the data set provided, there are only two lanes that have total demand over 200 for which we will refer to as lane 1 and lane 2, where lane 1 has a total demand of 470 tanks and lane 2 has a total demand of 345 tanks. Before presenting the results of the stochastic base model in sections 6.2.2 and 6.2.3, we first wish to discuss how the K values for demand were chosen using the triangular distribution.

6.2.1 Stochastic Results Using the Triangular Distribution

Results for the stochastic portion of the models is based on the triangular distribution for choosing the random demand values. A triangular distribution is defined by three values: $a =$ minimum value, $b =$ maximum value, and $c =$ the mode. A triangular distribution is denoted by $Tri(a, b, c)$ [37]. The probability density function for the triangular distribution is

$$f(x) = \begin{cases} \frac{2(x-a)}{(b-a)(c-a)} & \text{where } a \leq x \leq c \\ \\ \frac{2(b-x)}{(b-a)(c-a)} & \text{where } c < x \leq b \end{cases}$$

which implies that the cumulative density function is given by

$$F(x) = \begin{cases} \frac{(x-a)^2}{(b-a)(c-a)} & \text{where } a \leq x \leq c \\ \\ \frac{(b-x)^2}{(b-a)(c-a)} & \text{where } c < x \leq b \end{cases}.$$

The mean of a triangular distribution in terms of its parameters is given by

$$\frac{a+b+c}{3}.$$

represents the mean of the triangular distribution [37]. Implementation of the triangular distribution is achieved using the distribution's built-in function available through Python's NumPy library. We refer the reader to Appendix D for the code used in the implementation of the stochastic portion of the work.

For the stochastic model, we focus on randomizing only a subset of the lanes at a time. We begin by randomizing the demand on the lane with the highest estimated demand value first. After obtaining the results for the scenarios involving only a single lane having random demand, we proceed to also randomize the demand on the lane with the next highest estimated demand value. Proceeding in this manner, by increasing the number of lanes with stochastic demand one at a time, we desire to observe the behavior of the tender's total cost, along with changes in the variation of the cost across all simulations.

Results for each simulation are achieved as follows. We first choose the lane(s) that will have random demand values as explained previously and designate the total number of K-scenarios. Once the lane(s) are chosen to randomize, we fix the demand values for the K-scenarios on the remaining lanes using the lane's respective deterministic demand value. On the lanes that have been chosen to have random demand values, a new demand value is obtained from the triangular distribution for each K-scenario using the lane's estimated demand value as the mode. The minimum and maximum parameters for the triangular distribution are obtained by varying the mode by a predetermined percentage that depends on the lane's estimated demand.

For the lanes with more than 20 units volume, we decrease the deterministic demand by 10% to obtain the triangular distribution's minimum, and we increase the deterministic demand by 10% to obtain the triangular distribution's maximum. Likewise for lanes having between 5 and 19 units of volume, we decrease the deterministic demand by 5% and increase the deterministic demand by 5% to obtain the triangular distribution's minimum and maximum parameter values, respectively. For lanes with less than 5 units of volume, we decrease the deterministic demand by 1% and increase the deterministic demand by 1% to obtain the triangular distribution's minimum and maximum parameter values, respectively. For example, if we let μ be the expected (i.e., deterministic) volume for lane i and assume that the estimated demand is greater than 20 units of volume, then the triangular distribution used to obtain the random demand values is defined as $Tri(0.9\mu, \mu, 1.1\mu)$.

6.2.2 Variation on lane 1 only.

We begin by varying the total demand on lane 1 for a total of $K = 1000$ simulations. In choosing the $1,000$ demand values for lane 1, we fixed the demand on every other lane, and drew each random demand value for lane 1 from a triangular distribution as described in section 6.2.1. The expected demand volume on lane 1 in the Eastman data set is equal to $\mu = 470$ units of volume. Thus the triangular distribution for lane 1 is given by

$$Tri(0.9\mu, \mu, 1.1\mu) = Tri(423, 470, 517)$$

where the minimum volume for lane 1 is 423 units and the maximum is 517 units.

The resulting average cost over all K-scenarios when only lane 1 is varied totals \$8,242,313 as compared to \$8,241,155 when no variation is assumed. Additional summary statistics of the $K = 1000$ simulations may be found in Table 17. We can observe that with the introduction of random demand values on lane 1, not only is there an increase in the mean, but the median total cost is also larger than the total cost for the deterministic model. However, we find that the resulting increase in the average cost of the allocation is 0.014%, which is less than a one percent increase from the deterministic model.

Table 17: Summary statistics for total cost with variation on one lane only.

Statistic	**Allocation Cost**
Minimum	\$8,080,190
First quartile	\$8,198,231
Median	\$8,244,732
Third quartile	\$8,288,550
Maximum	\$8,398,543

Figure 3 shows the histogram of the random demand values for lane 1 with $K = 1000$ scenarios with an overlay of the triangular distribution's probability density function to confirm the random values fit the distribution.

Moreover, Figure 4 shows the histogram of the total cost of each scenario when variation in the demand value of lane 1 is introduced with $K = 1000$ scenarios. An overlay of the triangular distribution's probability density function is included to reveal that the total cost also follows the distribution. Note the total cost is in U.S. dollar amounts.

6.2.3 Variation on lanes 1 and 2.

Next we vary the total demand on lanes 1 and 2 simultaneously for $K = 1024$ scenarios. The dual variation was found by choosing a perfect square for the value of K where the square root of K is used for the number of random demand values on lane 1, i.e., we select $\sqrt{1024} = 32$ demand values for lane 1. Then for each of those values, we fix the demand on lane 1 and then vary the demand on lane 2 a total of 32 times. This is repeated for each random value of lane 1, which allows us to find a total of $1,024$ different combinations (or scenarios) of demand values with both lanes

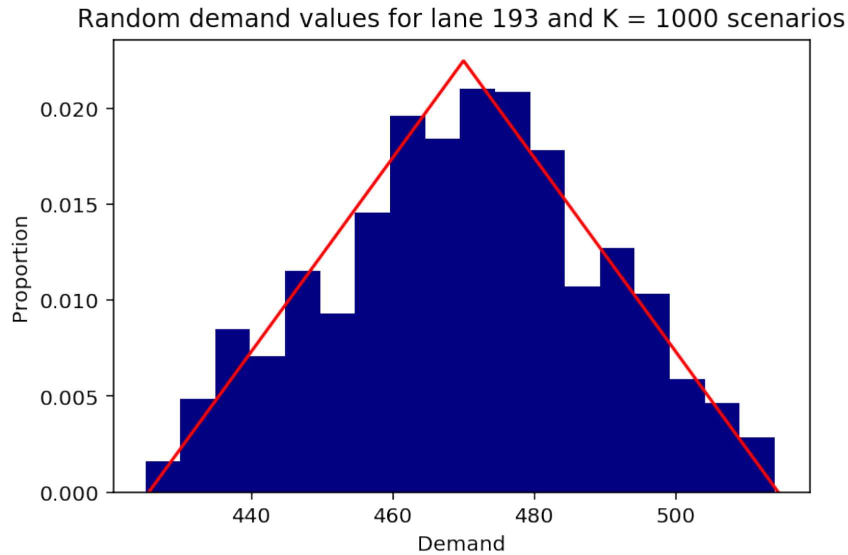

Figure 3: Random demand values for one lane with an overlay of the triangular distribution and $K = 1000$ total scenarios.

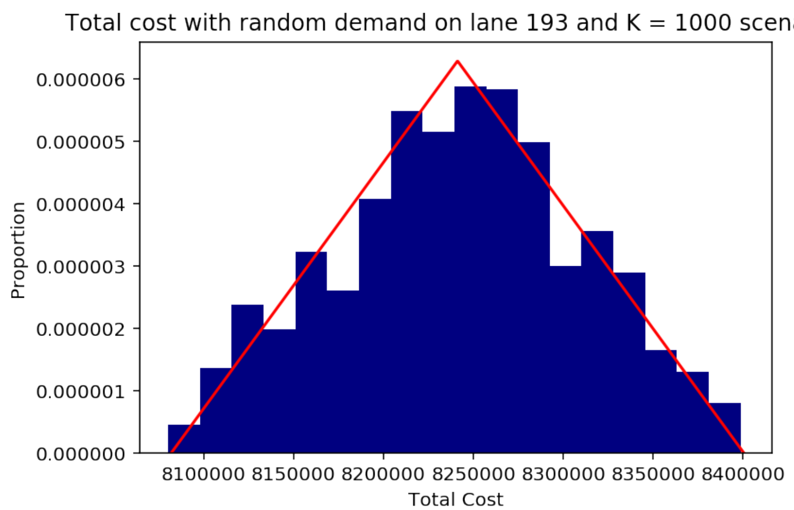

Figure 4: Total cost of each simulation with random demand values for variation on one lane with an overlay of the triangular distribution and $K = 1000$ total scenarios.

1 and 2 having random demand.

Our simulations show that the average cost of the allocation over all K-scenarios when lanes 1 and 2 are varied in this manner totals \$8,238,348 as compared with an average cost of \$8,242,313 when variation is only on lane 1 and a total cost of \$8,241,155 when no variation is assumed. The summary statistics for variation on both lanes is presented in Table 18, along with the summary statistics for variation on only lane 1. We can observe that the both the average total cost and the median total cost have both increased by varying the demand on lane 2 in addition to varying the demand on lane 1.

Table 18: Summary statistics for the total cost with variation on one lane only compared with variation on two lanes.

Statistic	**Allocation Cost Lane 1**	**Allocation Cost Lanes 1 and 2**
Minimum	\$8,080,190	\$8,063,386
First quartile	\$8,198,231	\$8,186,220
Median	\$8,244,732	\$8,239,756
Third quartile	\$8,288,550	\$8,290,183
Maximum	\$8,398,543	\$8,422,191

Figure 5 shows the histogram of the total cost of each scenario when variation in the demand value of lanes 1 and 2 is introduced with $K = 1024$ scenarios. An overlay of the triangular distribution's probability density function is included to reveal that the total cost also follows the distribution. Note the total cost is in U.S. dollar amounts.

We also notice that there is an increase in variation in the total cost when two lanes take on random demand values compared with only one lane. Figure 6 shows

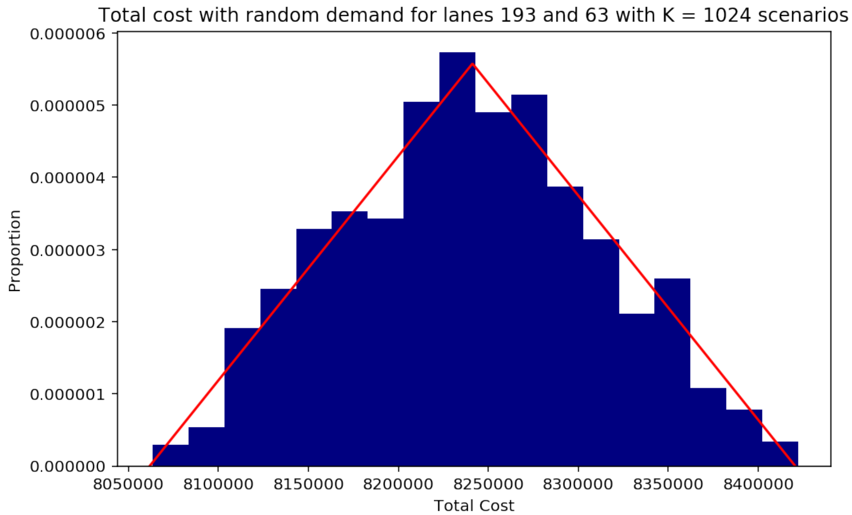

Figure 5: Total cost of each simulation with random demand values for two lanes with an overlay of the triangular distribution and $K = 1024$ total scenarios.

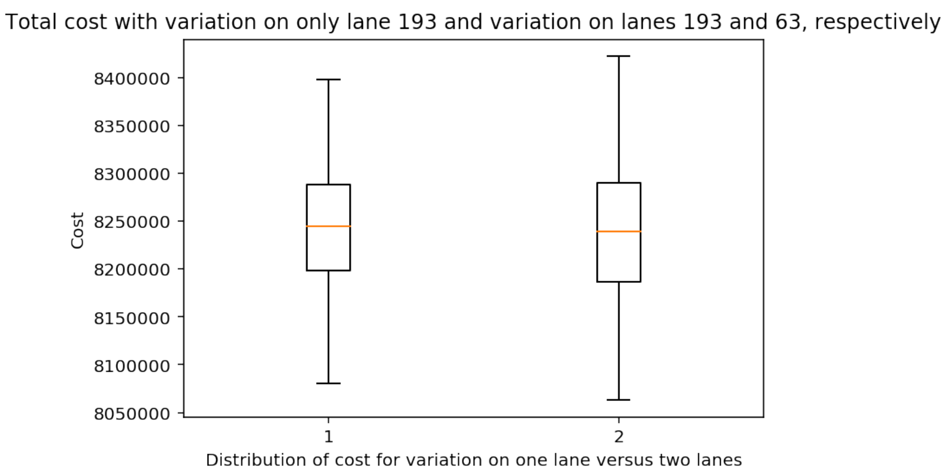

Figure 6: Side by side boxplot of the variation in total cost when only one lane has random demand versus two having random demand.

the side-by-side boxplots of the distribution of the total cost when only lane 1 has random demand versus lanes 1 and 2 having random demand values.

While the median of the two distributions does not differ significantly, we observe that the Buyer may be faced with a resulting allocation that would have a higher cost upon the actualized demand values, while on the other hand, the resulting allocation may have a lower cost upon the actualized demand values. This illuminates the element of risk involved in using estimated demand values in the tender process. We expect to see the variation in the total cost continue to increase as the number of lanes with random demand values increases.

7 CONCLUSION AND FUTURE WORK

In conclusion, in this thesis, we developed a mixed-integer constraint model for the procurement of logistical services through a reverse combinatorial auction. We have outlined the development of a base model that does not account for additional business rules in the form of constraints in the mixed-integer linear program. Additionally, we have developed a linear program that does consider restrictions placed on the maximum and minimum number of carriers in the program. A deterministic model has been presented for both the base model and the carrier constraint model that allows the buyer to use estimated demand values. Additionally, we have shown the development and use of a stochastic model for which certain lanes have random demand values drawn from a triangular distribution through stochastic programming techniques.

Implementation of all of the models we have presented was completed using the free CoCalc software available online. After completing any preprocessing steps for the data set to remove duplicates and ensure the data is error free and in a compatible format, all of the coding was written in Python for which numerous libraries were used including NumPy and SciPy, and the linear programming portion of the computational program was completed in the Python package Pyomo. The Python and Pyomo code may be located in the appendix.

Upon implementation of the stochastic models using the data set provided by Eastman Chemical Company, our results indicate that when variation is assumed on a lane's demand, the average cost of the tender will increase. We observe that when compared to the deterministic cost of $8,241,155 for the allocation, an average increase of \$1,158 occurs when variation for the lane with the highest demand is introduced. Over 1,000 simulations with random demand on one lane, we observe the tender's average total cost to be \$8,242,313 as compared to \$8,241,155 when no variation is assume. This upward trend in the allocation's cost continues when a second lane's demand is also randomized. Varying two lanes results in an average cost of \$8,238,348 as compared with \$8,242,313 when variation is only on lane 1 and \$8,241,155 when no variation is assumed. Hence the tender's average cost increases by \$2,807 from the deterministic model. Moreover, we expect that this trend will be seen as the number of lanes with random demand continues to increase.

While this work outlines the process taken to develop and implement a linear program representation of a reverse combinatorial model, there are numerous avenues of future work to be explored. In particular, increasing the number of lanes with random demand for the stochastic model to verify the behavior of the tender's cost would be the next steps to complete. Additional areas of future work include implementing the carrier constraint stochastic model, assuming combinations of bids using the Eastman data, and the implementation of the stochastic model with combination bids. Furthermore, there are numerous business rules that might be considered and combined with the base model in the form of additional constraints. Lastly, incorporating more business rules in the more of linear constraints is another area of future work. The models could include such rules as restricting the number of total units allocated, or allocating a certain percentage of the total volume, to a given carrier, restricting the total volume allocated to any carrier, incorporating incentives or priority to certain carriers, automatically rejecting or including certain carriers from the final allocation, limiting the number of different carriers shipping from the same location, and limiting the number of carriers shipping to the same region.

Transportation procurement benefits from the implementation of optimization techniques because of the complex aspects involved in determining how to assign carriers to lanes over large networks. Economies of scope, inter-dependencies across lanes, and back haul costs must be considered by the shipper when placing bids on either individual or sets of lanes. The reverse combinatorial auction allows the buyer to receive more robust and competitive bids that will ultimately reduce costs across the entire supply chain if the problem of assigning carriers to lanes is completed so that an optimal outcome will be produced. We have shown one method of achieving this optimal solution using techniques from operations research through deterministic and stochastic versions of a mixed-integer linear program.

BIBLIOGRAPHY

[1] Atkinson, W. (2001). How e-logistics changes shipper-carrier relationships. *Logistics Management & Distribution Report*, 10-10.

[2] Bergman, M. A., & Lundberg, S. (2013). *Tender evaluation and supplier selection methods in public procurement*. Journal of Purchasing and Supply Management, 19(2), 73-83.

[3] Caplice, C. G. (1996). *An optimization based bidding process: A new framework for shipper-carrier relationships* (Doctoral dissertation, Massachusetts Institute of Technology).

[4] Caplice, C., & Sheffi, Y. (2003). Optimizationbased procurement for transportation services. *Journal of Business Logistics*, 24(2), 109-128.

[5] Chopra, S., & Meindl, P. (2004). *Supply chain management: Strategy, planning, and operation*. Upper Saddle River, N.J: Prentice Hall.

[6] Elmaghraby, W., & Keskinocak, P. (2004). Combinatorial auctions in procurement. *The Practice of Supply Chain Management: Where Theory and Application Converge*, 245-258.

[7] Ergun, O., Kuyzu, G., & Savelsbergh, M. (2007). *Bid price optimization for simultaneous truckload transportation procurement auctions.* Technical report on Stewart School of Industrial Engineering, Georgia Institute of Technology, Atlanta.

[8] Ferguson, T. S. (2000). *Linear programming: A concise introduction.* Retrieved from: http://www.math.ucla.edu/~tom/LP.pdf.

[9] Hart, W. E. Pyomo: Python optimization modeling objects.

[10] Helms, M. M., & Dileepan, P. (2005, January). Transportation issues for supply chain management. In *Business Forum* (Vol. 27, No. 1, p. 8). California State University, Los Angeles, School of Business and Economics.

[11] Honeywell International Inc. (2017). *Honeywell 2016 annual report.* Retrieved from https://www.honeywell.com.

[12] Hrabec, D., Popela, P., Roupec, J., Mazal, J., & Stodola, P. (2015). Two-stage stochastic programming for transportation network design problem. In *Mendel 2015* (pp. 17-25). Springer, Cham.

[13] Jap, S. D. (2002). *Online reverse auctions: Issues, themes, and prospects for the future.* Journal of the Academy of Marketing Science, 30(4), 506-525.

[14] Jeng, J., Schilly, H., Snyder, H., & Stein, W. (2018). CoCalc by SageMath, Incorporated. [Computer Software]. Available from https://cocalc.com/.

[15] Kall, P., Wallace, S. W., & Kall, P. (1994). *Stochastic programming* (p. 307). Chichester: Wiley.

[16] Knisley, J. (2017). *Setting up Linear Programs.* Personal Collection of J. Knisley, East Tennessee State University, Johnson City, TN.

[17] Knisley, J. (2017). *Stochastic Programming with Pyomo*. Personal Collection of J. Knisley, East Tennessee State University, Johnson City, TN.

[18] Knisley, J. (2017). *Weak and Strong Duality*. Personal Collection of J. Knisley, East Tennessee State University, Johnson City, TN.

[19] Ledyard, J. O., Olson, M., Porter, D., Swanson, J. A., & Torma, D. P. (2002). The first use of a combined-value auction for transportation services. *Interfaces*, 32(5), 4-12.

[20] Liu, C., Fan, Y., & Ordez, F. (2009). A two-stage stochastic programming model for transportation network protection. *Computers & Operations Research*, 36(5), 1582-1590.

[21] Luenberger, D. G., & Ye, Y. (1984). *Linear and nonlinear programming* (Vol. 2). Reading, MA: Addison-wesley.

[22] Ma, Z., Kwon, R. H., & Lee, C. G. (2010). A stochastic programming winner determination model for truckload procurement under shipment uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, 46(1), 49-60.

[23] McKinney, W. (2012). *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. " O'Reilly Media, Inc.".

[24] Narahari, Y., & Dayama, P. (2005). *Combinatorial auctions for electronic business*. Sadhana, 30(2 & 3), 179-211.

[25] NCSS Statistical Software (2018). *Operations Research in NCSS*. Retrieved from: `https://www.ncss.com/software/ncss/operations-research-in-ncss`.

[26] Nisan, N. (2000, October). Bidding and allocation in combinatorial auctions. In *Proceedings of the 2nd ACM conference on Electronic commerce* (pp. 1-12). ACM.

[27] Oliver, M. (2012). *Practical guide to the simplex method of linear programming.*

[28] The Operational Research Society (2018). *History of O.R.* Retrieved from: `https://www.ncss.com/software/ncss/operations-research-in-ncss`.

[29] Orlin, J. (2013) *IP Reference guide for integer programming formulations.* [PDF document] Retrieved from `https://ocw.mit.edu/courses/sloan-school-of-management/15-053-optimization-methods-in-management-science-spring-2013/lecture-notes/MIT15\_053S13\_iprefguide.pdf`.

[30] Orlin, J., & Nasrabadi, E. (2013). *15.053 Optimization Methods in Management Science*. Massachusetts Institute of Technology: MIT OpenCourseWare. Retrieved from: `https://ocw.mit.edu.License:CreativeCommonsBY-NC-SA`.

[31] Sandholm, T. (2007). Expressive commerce and its application to sourcing: How we conducted $35 billion of generalized combinatorial auctions. *AI Magazine*, 28(3), 45.

[32] Shapiro, A., & Philpott, A. (2007). A tutorial on stochastic programming. *Manuscript*. Retrieved from: `https://www2.isye.gatech.edu/~ashapiro/publications.html`.

[33] Sheffi, Y. (2004). Combinatorial auctions in the procurement of transportation services. *Interfaces*, 34(4), 245-252.

[34] Peter, S. (2011) *Sourcing optimisation – extracting value from complexity*. Spend Matters UK/Europe. Retrieved from: `http://spendmatters.com/research-library/papers/sourcing-optimisation-extracting-value-from-complexity`.

[35] Sollish, F., & Semanik, J. (2007). *The Procurement and supply manager's desk reference*. Hoboken, NJ: Wiley

[36] U.S. Department of Commerce/International Trade Administration. (2017). *Logistics and Transportation Spotlight: The Logistics and Transportation Industry in the United States*. Washington, DC: Author.

[37] Weisstein, E. W. (2018). "Triangular Distribution." From MathWorld–A Wolfram Web Resource. Retrieved from: `http://mathworld.wolfram.com/TriangularDistribution.html`

APPENDICES

A Base Model Example Code

```
from pyomo.environ import *

model = AbstractModel()

model.numBids = Param(within=NonNegativeIntegers)

model.numItems = Param(within=NonNegativeIntegers)

model.BIDS = RangeSet(1, model.numBids)

model.LANES = RangeSet(1, model.numItems)

model.bidValue = Param(model.BIDS)

model.demand = Param(model.LANES)

model.delta = Param(model.LANES, model.BIDS, initialize=0)

model.cardinality = Param(model.BIDS)

model.x = Var(model.BIDS, domain = Binary)

def obj_expression(model):

    return summation(model.bidValue, model.x)

model.OBJ = Objective(rule=obj_expression, sense=minimize)

def constraint_rule(model, l):

    return sum(model.delta[l,b]*model.x[b]

            for b in model.BIDS) <= model.demand[l]

model.xConstraint = Constraint(model.LANES, rule=constraint_rule)
```

```
def demand_constraint_rule(model):
    return sum(model.x[b]*model.cardinality[b]
            for b in model.BIDS) >= model.numItems
model.demandConstraint = Constraint(rule=demand_constraint_rule)
```

B Carrier Constraint Model Example Code

```
from pyomo.environ import *

model = AbstractModel()

model.numBids = Param(within=NonNegativeIntegers)

model.numItems = Param(within=NonNegativeIntegers)

model.numCarriers = Param(within=NonNegativeIntegers)

model.BIDS = RangeSet(1, model.numBids)

model.LANES = RangeSet(1, model.numItems)

model.CARRIERS = RangeSet(1, model.numCarriers)

model.bidValue = Param(model.BIDS)

model.demand = Param(model.LANES)

model.cardinality = Param(model.BIDS)

model.M = Param(model.CARRIERS)

model.delta = Param(model.LANES, model.BIDS, initialize=0)

model.gamma = Param(model.CARRIERS, model.BIDS, initialize=0)

model.maxCarriers = Param()

model.minCarriers = Param()

model.x = Var(model.BIDS, domain = Binary)

model.z = Var(model.CARRIERS, domain = Binary)

def obj_expression(model):
```

```
    return summation(model.bidValue, model.x)
model.OBJ = Objective(rule=obj_expression, sense=minimize)
def constraint_rule(model, l):
    return sum(model.delta[l,b]*model.x[b] for b in model.BIDS)
        <= model.demand[l]
model.xConstraint = Constraint(model.LANES, rule=constraint_rule)
def demand_constraint_rule(model):
    return sum(model.x[b]*model.cardinality[b] for b in model.BIDS)
        >= model.numItems
model.demandConstraint = Constraint(rule=demand_constraint_rule)
def constraint2_rule(model, k):
    return sum(model.gamma[k,b]*model.x[b] for b in model.BIDS) -
        model.M[k]*model.z[k] <= 0
model.upperBoundConstraint = Constraint(model.CARRIERS,
    rule=constraint2_rule)
def constraint3_rule(model, k):
    return model.z[k] - sum(model.gamma[k,b]*model.x[b] for b
        in model.BIDS) <= 0
model.lowerBoundConstraint = Constraint(model.CARRIERS,
    rule=constraint3_rule)
```

```
def constraint4_rule(model):
    return sum(model.z[i] for i in model.CARRIERS) <=
        model.maxCarriers
model.zConstraint = Constraint(rule=constraint4_rule)
def constraint5_rule(model):
    return sum(model.z[i] for i in model.CARRIERS) >=
        model.minCarriers
model.zConstraint2 = Constraint(rule=constraint5_rule)
```

C Deterministic Portion of the Program Code via Jupyter Notebook

```
#Import of main libraries
%matplotlib inline
from matplotlib import pyplot as plt
import numpy as np
from __future__ import division, print_function
from pandas import read_excel
from pandas import DataFrame
from pandas import ExcelWriter
from pandas import ExcelFile
#Import of the pyomo module
from pyomo.environ import *
#Creation of a Concrete Model
model = ConcreteModel()
BidsDf = read_excel('TenderDataUpdatedCarriersRemoved.xlsx',
    sheet_name='Bids')
LanesDf = read_excel('TenderDataUpdatedCarriersRemoved.xlsx',
    sheet_name='Lots')
#Create a data from for delta
deltaDf = DataFrame(np.zeros((len(LanesDf.index),
```

```
    len(BidsDf.index))))
for bid in BidsDf.index:
    temp = BidsDf.loc[bid,'LaneID'][4:]
    deltaDf.at[float(temp) - 1, bid] = 1
#Number of lanes in the program
model.numItems = len(LanesDf.index)
## Define sets
model.BIDS = Set(initialize = BidsDf.index.values)
model.LANES = Set(initialize = LanesDf.index.values)
# Create a dictionary of the bid values
bidValues = dict()
for bid in BidsDf.index:
    bidValues[bid] = BidsDf.loc[bid, 'ModelRateCUR']*
        LanesDf.loc[int(BidsDf.loc[bid,'LaneID'][4:])-1,
        'Total Volume (# of tank)']
#Initialize bidValue parameter with the value of each bid
model.bidValue = Param(model.BIDS, initialize = bidValues,
    doc='Value of each bid in the program')
#Create a dictionary of the total volume on each lane
demandValues = dict()
```

```
for lane in LanesDf.index:
    demandValues[lane] = 1
#Initialize demand parameter with the total volume on each lane
model.demand = Param(model.LANES, initialize = demandValues,
    doc='Total demand on each lane')
#Create a dictionary of the delta values
delta = dict()
for lane in LanesDf.index:
    for bid in BidsDf.index:
        delta[(lane,bid)] = deltaDf.loc[lane,bid]
#Initialize the delta parameter
model.delta = Param(model.LANES, model.BIDS, initialize=delta,
    doc='delta gives information regarding which lanes
    are in a bid package')
#Define the decision variable
model.x = Var(model.BIDS, domain = Binary, doc='Decision variable
    for each bid in the program')
#Objective minimizes the sum of x_b * v_b over all bids
def obj_expression(model):
    return sum(model.bidValue[i]*model.x[i] for i in model.BIDS)
```

```
model.OBJ = Objective(rule=obj_expression, sense=minimize,
    doc='Objective function definition')
#Define constraints
def demand_constraint_rule(model):
    return sum(model.x[b] for b in model.BIDS) >= model.numItems
model.demandConstraint = Constraint(rule=demand_constraint_rule)
def constraint_rule(model, l):
    return sum(model.delta[l,b]*model.x[b] for b in model.BIDS)
        >= 1 #model.demand[l] = 1 for all l
model.xConstraint = Constraint(model.LANES, rule=constraint_rule)
#Display of the output in order to retrieve and use in python
def pyomo_postprocess(options=None, instance=None, results=None):
    model.x.display()
#Run the model
from pyomo.opt import SolverFactory
import pyomo.environ
opt = SolverFactory("glpk")
results = opt.solve(model)
#Create a dataframe consisting of the winning bids
winningBids = []
```

```
index = 0
bidNum = 0
for p in range(2433):
    if model.x[p].value > 0:
        winningBids.append(bidNum)
    bidNum += 1
    index += 1
winningBidsDf = BidsDf.iloc[winningBids]
```

D Stochastic Portion of the Program Code via Jupyter Notebook

D.1 Single Lane Randomization

```
#Begin randomizing the demand on the lane with highest
    volume only (lane 193 in the data file)
#Number scenarios
K = 1000
#Create a dictionary of the total volume on each lane
    and an array of the random volume for lane 193
demandArray = []
demandValues = dict()
for lane in LanesDf.index:
    for scenario in range(0,K):
        average = LanesDf.loc[lane, 'Total Volume (# of tank)']
        if average >= 400:
            demandValues[lane, scenario] =
                round(np.random.triangular(0.9*average,
                average, 1.1*average),0)
            demandArray.append(demandValues[lane, scenario])
        else:
            demandValues[lane, scenario] = average
```

```
#Create an array of the total cost for each k-scenario
costPerScenario = []
cost = 0
for scenario in range(0,K):
    cost = 0
    for bid in winningBidsDf.index:
        cost += winningBidsDf.loc[bid, 'ModelRateCUR']*
            demandValues[int(winningBidsDf.loc[bid,
            'LaneID'][4:])-1, scenario]
    costPerScenario.append(cost)
```

D.2 Two Lane Randomization

```
#Vary the demand on two lanes with highest volume
#Number scenarios
K = 32
#Create a dictionary of the total volume on each lane
demandValues2 = dict()
lane193DemandArray = []
lane63DemandArray = []
for lane in LanesDf.index:
    for scenario1 in range(0,K):
```

```
        for scenario2 in range(0,K):
            average = LanesDf.loc[lane, 'Total Volume
                (# of tank)']
            if average == 470:
                demandValues2[lane, scenario1, scenario2] =
                    round(np.random.triangular(0.9*average,
                    average, 1.1*average),0)
                lane193DemandArray.append(demandValues2[lane,
                scenario1, scenario2])
            elif average == 345:
                demandValues2[lane, scenario1, scenario2] =
                    round(np.random.triangular(0.9*average,
                    average, 1.1*average),0)
                lane63DemandArray.append(demandValues2[lane,
                    scenario1, scenario2])
            else:
                demandValues2[lane, scenario1, scenario2] =
                    average
#Create an array of the total cost for each k-scenario
costPerScenario2 = [] #cost per scenario lanes 193 and 63
```

```
    are varied
costOnlyVaryLane193 = [] #cost per scenario only lane 193
    is varied
cost1 = 0
cost2 = 0
for scenario1 in range(0,K):
    for scenario2 in range(0,K):
        cost1 = 0
        cost2 = 0
        for bid in winningBidsDf.index:
            cost1 += winningBidsDf.loc[bid,
                'ModelRateCUR']*demandValues2[int(
                winningBidsDf.loc[bid,'LaneID'][4:])-1,
                scenario1, scenario2]
            if int(winningBidsDf.loc[bid,'LaneID'][4:]) == 63:
                cost2 += winningBidsDf.loc[bid,'ModelRateCUR']*345
            else:
                cost2 += winningBidsDf.loc[bid,'ModelRateCUR']*
                    demandValues2[int(winningBidsDf.loc[bid,
                    'LaneID'][4:])-1, scenario1, scenario2]
```

```
costPerScenario2.append(cost1)
costOnlyVaryLane193.append(cost2)
```

VITA

JENNIFER KISER

Education:	A.S. Mathematics, Northeast State Community College, Blountville, Tennessee 2012
	B.S. Mathematics and Statistics, East Tennessee State University, Johnson City, Tennessee 2016
	M.S. Mathematical Sciences, East Tennessee State University, Johnson City, Tennessee 2018
Professional Experience:	Instructor, Upward Bound at ETSU, Johnson City, Tennessee, 2017–2018
	Graduate Assistant, Department of Mathematics & Statistics, East Tennessee State University, Johnson City, Tennessee, 2016–2018
Publications:	Eskandar, George; **Houser, Jennifer**; Prochaska, Ellen; Wojtkiewicz, Jessica; Lebair, Teresa; E Peercy, Bradford E.; Watts, Margaret; and Sherman, Arthur (2016) "Investigating How Calcium Diffusion Affects Metabolic Oscillations and Synchronization of Pancreatic Beta Cells," *Spora – A Journal of Biomathematics*: Vol. 2: Iss. 1, Article 1.
	Ahmed, Md Salman; **Houser, Jennifer**; Hoque, Mohammad A.; Raju, Rezaul; and Pfeiffer, Phil (2017). "Reducing Inter-Process Communication Overhead in Parallel Sparse Matrix-Matrix Multiplication," *International Journal of Grid and High Performance Computing (IJGHPC)*, 9(3), 46-59.