Copyright

by

Kanthimathi Sathasivan

2011

**The Dissertation Committee for Kanthimathi Sathasivan Certifies that this is the approved version of the following dissertation:**

# Optimizing Cross-dock Operations under Uncertainty

**Committee:**

S. Travis Waller, Supervisor

Chandra R. Bhat

Randy B. Machemehl

David P. Morton

Zhanmin Zhang

# Optimizing Cross-dock Operations under Uncertainty

**by**

**Kanthimathi Sathasivan, B.Sc.; M.S.; M.A.; M.S.E.**

**Dissertation**

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

**Doctor of Philosophy**

**The University of Texas at Austin**

**December 2011**

To my family

## Acknowledgements

I am extremely grateful to my advisor Dr. S. Travis Waller for giving me the opportunity to pursue my doctoral degree under his guidance. The tremendous support and encouragement he showed at every step of this journey has made a big difference in my academic life. I would like to express my gratitude to Dr. Chandra R. Bhat for his moral support and technical advice. I am indebted to Dr. Randy B. Machemehl for his continuous guidance and motivation. I have been fortunate to work with Dr. David P. Morton for the past eight and a half years in both my masters and doctoral programs. I sincerely thank him for his valuable advice, patience and the time he spent enlightening me in this process. I also want to thank Dr. Zhanmin Zhang for his kindness and support.

I would like to thank Mr. Carlos Garza and Mr. Jon Pulido at H-E-B Head Quarters at San Antonio for their time and permission for observation of their cross dock facility and the data provided. I would like to express my thanks to Ms. Linda Bauer at the Goodwill Industries of Central Texas for sharing their operations with me.

I sincerely appreciate and thank Dr. ManWo Ng for the valuable discussions and encouragement on this research. I thank Dr. Avinash Unnikrishnan for helping me start my research. I am grateful to all the members of Dr. Waller's TeQson lab for sharing their knowledge, information, and friendship. In particular, I thank Jen, Natalia, Roshan, Nezam, Lauren, David Fajardo, David Suescun, Nan and Ti. In addition, I would like to thank Libbie Toler for her administrative help and emotional support. My thanks go to

Lisa Macias, Mandy Barrera, Annette Perrone, Lisa Smith, Lisa Cramer and Kathy Rose for providing administrative help.

I thank my parents and my sister for all their support during this period. Finally, I have no words to thank my husband, daughter and son for all they have done during the last eight years, from dropping off my assignments to picking up library books to driving me to school on days with sleepless nights. Without their encouragement, understanding, and intellectual inputs none of this would be possible.

Kanthimathi Sathasivan

*The University of Texas at Austin*

*December 2011*

## Optimizing Cross-dock Operations under Uncertainty

Publication No._____________

Kanthimathi Sathasivan, Ph.D.

The University of Texas at Austin, 2011

Supervisor: S. Travis Waller

Cross-docking is an important transportation logistics strategy in supply chain management which reduces transportation costs, inventory holding costs, order-picking costs and response time. Careful planning is needed for successful cross-dock operations. Uncertainty in cross-dock problems is inevitable and needs to be addressed. Almost all research in the cross-dock area assumes determinism. This dissertation considers uncertainty in cross-dock problems and optimizes these problems under uncertainty.

We consider uncertainty in processing times, using scenario-based and protection-based robust approaches. Using a heuristic method, we find a lower and upper bound and combine that with a meta-heuristic method to solve the problem. Also, we consider problems in two different industries (Goodwill and H-E-B) and address the uncertainties that happen frequently in their operations.

The scenario-based robust optimization model for the unloading problem using a min max objective is presented with examples. A surrogate heuristic procedure is used to

find a robust solution. Next, a two-space genetic algorithm, a meta-heuristic procedure, is applied to the unloading problem using the bounds obtained by the heuristic procedure. The results are closer to the optimal solution than those obtained by the two-space genetic algorithm without bounds. When compared with the regular genetic algorithm with bounds, the two-space algorithm performs well.

The protection-based approach considers a limit on the number of coefficients allowed to change with data uncertainty, protecting against the degree of conservatism. The management of trucks and reduction of overtime pay in the cross-dock operations of Goodwill is addressed through two models and uncertainty is applied to those models. A combined cross-dock operations model together with demand is formulated and the uncertainties are discussed for H-E-B operations. This dissertation does not address the recycling operation within the cross-dock of Goodwill, or the uncertainty in H-E-B data.

# Table of Contents

## List of Tables

## List of Figures

# Chapter 1: Introduction

Cross-docking moves products from manufacturers to retailers and customers with little to no storage. Raw materials and supplies for manufacturers are transported through cross-docks as well. Inbound trucks originate from different locations carrying products to be sent to various destinations. When inbound trucks arrive at a cross-dock, the products are unloaded from the inbound trucks, sorted, consolidated, and reloaded in the outbound trucks for delivery, typically within twenty-four hours.

## 1.1 Motivation

Cross-docking is an important transportation logistics strategy in supply-chain management for reducing transportation costs, holding costs of the inventory, order-picking costs, and response time to customer demand. A traditional distribution center holds goods first and then picks up the order and delivers goods to meet customer demands later; it is clear that cross-docking brings a significant overall reduction of inventory-holding and order-picking cost. In an era of global competition, this advantage brings increased attention to cross-docking as huge volumes of goods are transported to destinations all over the world.

Cross-docking was first applied successfully at the Chicago Area Consolidation Hub of UPS in which there were 122 receiving docks and 1050 shipping docks. It took an average of only 15 minutes on the conveyors to move a package for a distance of one mile during start-up days (Forger, 1995). During the last decade of the twentieth century, several manufacturers and retailers such as Home Foods of Milton, Pennsylvania, Office

Depot, Wal-Mart, and Toyota, have had tremendous success using cross-docking strategy (Apte and Viswanathan, 2000). Increases in major retail store chains and delivery of goods to consumers directly has allowed the growth of cross-docking from small facilities to very large facilities. There are more than 10,000 cross-docks in the United States and Canada (Bartholdi and Gue, 2004). The cross-docking industry is growing and businesses want to cut costs and serve customers faster and better.

According to The Commodity Flow Survey (CFS), conducted every five years as part of the Census Bureau's Economic Census, the 100,000 business establishments covered by the CFS shipped commodities worth about $11.7 trillion, weighing 12.5 billion tons and generating 3.3 trillion ton-miles in 2007 (U.S. Department of Transportation, 2010). The CFS summarizes and highlights freight shipments for each of the 50 states and the District of Columbia based on the businesses covered. The report concluded trucking continued to dominate the nation's movement of freight, accounting for 71% of the value ($8.3 trillion), 70% of weight (8.8 billion tons), and 39% of the ton-miles (1.3 trillion ton-miles). Texas and California dominated the origin and destination of freight by value and weight. Considering the ton-miles, these states also dominated in destination. However, goods originating from the state of Wyoming generated the most ton-miles with more than double the ton-miles generated by goods originating from Texas, which was second.

The above survey gives a glimpse of how much is transported within the country. One can imagine proportionately the amount of raw materials, parts, and supplies for manufacturing and the delivery of finished products transported within each nation and between countries all over the world. As competition increases globally, the efficiency and the robustness of these operations will be measured by the consumers and retailers, impacting future decisions. The success of cross-docking depends on information flow and proper communication with other members of the supply chain. At the same time, the characteristics of operational management within cross-dock facilities contribute equally to the success of cross-dock operations.

Uncertainty is always a reality. Certain things that are possible in theory may not be possible in practice. Whenever machines and humans are involved, unexpected events are prone to occur. There may not be sufficient workforce, machines may break down, the information flow may be delayed, or the supplies or parts may not arrive on time. The number of products expected or the types of products needed may not be available. Also, during transportation, delays may occur due to congestion or weather-related incidents. While solving cross-dock problems there is a need to address these uncertainties.

### 1.2 CROSS-DOCKING IN SUPPLY CHAIN AND LOGISTICS NETWORK

In order to understand cross-dock problems and to model them correctly, it is imperative to understand the overall functions of cross-dock facilities as well as the planning and information needed at every level. In other words, we need to study and understand cross-docking at macro and micro level. We start by looking at the supply chain and logistics network and then move to the cross-dock. A simple supply chain includes suppliers, manufacturers, distribution centers, and customers (Figure 1.1). Cross-dock facilities come under distribution centers and are the transshipment nodes in the supply chain network.

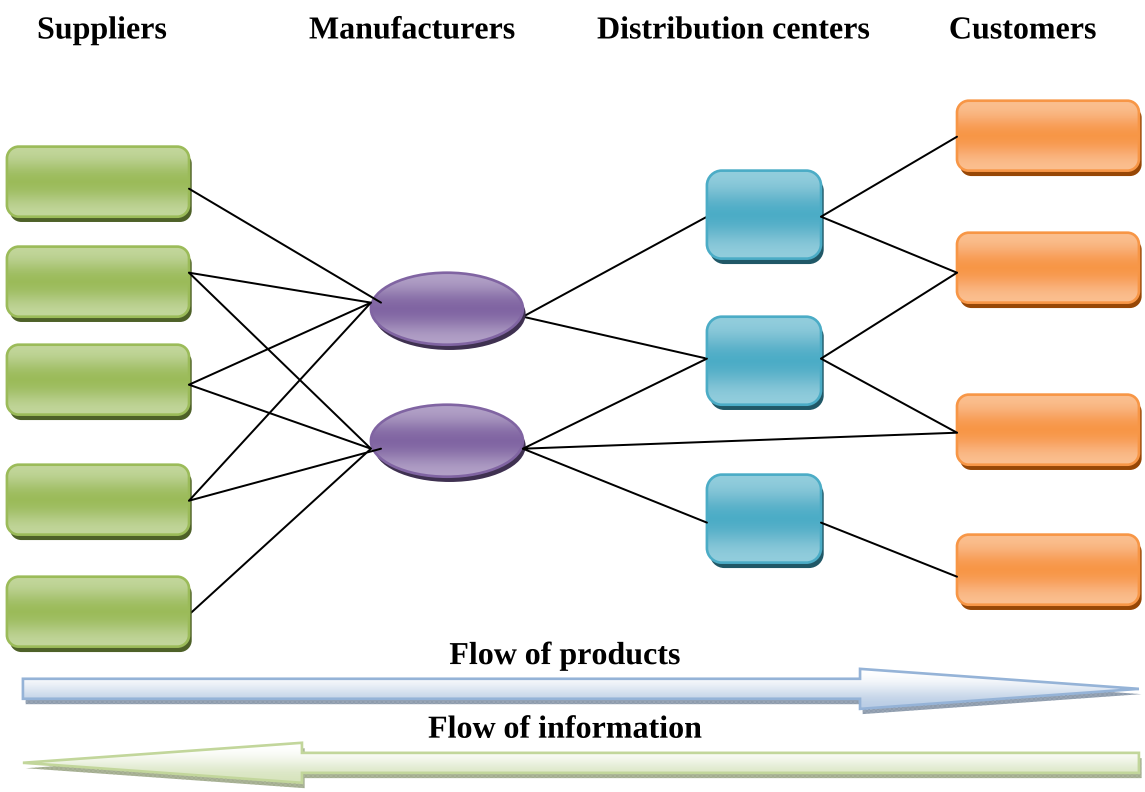


**Figure 1.1:** Supply chain network

Vogt (2010) defines a “cross-dock based supply chain” as “one in which the integrated supply chain includes a cross-dock facility and where the facilities and capabilities shared by members of the chain exist for the benefit of the chain as a whole, rather than of one downstream customer or another.” Cross-docks exist in several types of supply chains and are very valuable by acting as distribution centers for sending both parts for manufacturing and finished products for customers. The supply chain relationships, proper processes, and systems must be in place for successful operation of cross-docks on a large scale. The nine success factors for a supply chain containing a cross-dock as described by Vogt (2010) are appropriate products; reliable efficient suppliers; expert and reliable supply chain service providers; process improvement and problem solving capability; uniquely skilled management and staff; well-chosen computer systems; work balancing and minimization; efficient physical facility design and layout; and understanding how cross-dock based supply chains work. These factors are considered in solving different types of cross-dock problems.

The five functions of logistics network are order processing, inventory, transportation, warehousing - material handling - packaging and facility network and all of which are interrelated (see Figure 1.2). Order processing requires fast, accurate, timely information flow about customer purchase behavior. A minimum inventory must be maintained in order to satisfy desired customer service. Transportation moves inventory geographically to desired locations. Warehousing, material handling, and packaging are important operations of logistics. Facility networks consist of retail stores, warehouses, cross-dock operations, and manufacturing plants. Logistics makes individual businesses more efficient through control and by planning both internal and external operations.

Logistics minimize cost in each of the five functions individually where the cost is different for each function or collectively where the cost is the same for all five functions. On the other hand, in supply chains, total cost is minimized by considering the supply chain as a whole. One link may incur more cost in order to decrease the total cost.

Thus cross-docking plays important roles in both supply chain networks and logistics networks.

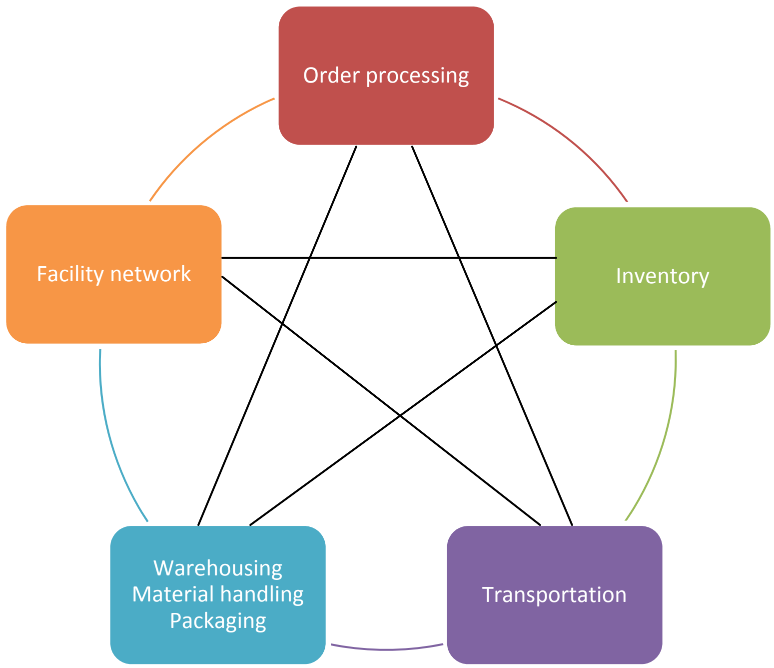


**Figure 1.2:** The five functions of logistics network

### 1.3 CROSS-DOCK FACILITY TYPES AND OPERATIONS

Cross docking receives products and moves them through distribution centers as quickly as possible without storing them. Ballou (1999) explains the network structure for warehousing. In a *consolidation warehouse* several less-than-full truck load (LTL) shipments of raw materials are consolidated into full truck load (FTL) shipments. In a *break-bulk warehouse* FTL shipments are consolidated into LTL shipments to send to customers or retailers, and in a *mixed warehouse* FTL shipments of each product is consolidated into multi-product FTL shipments.

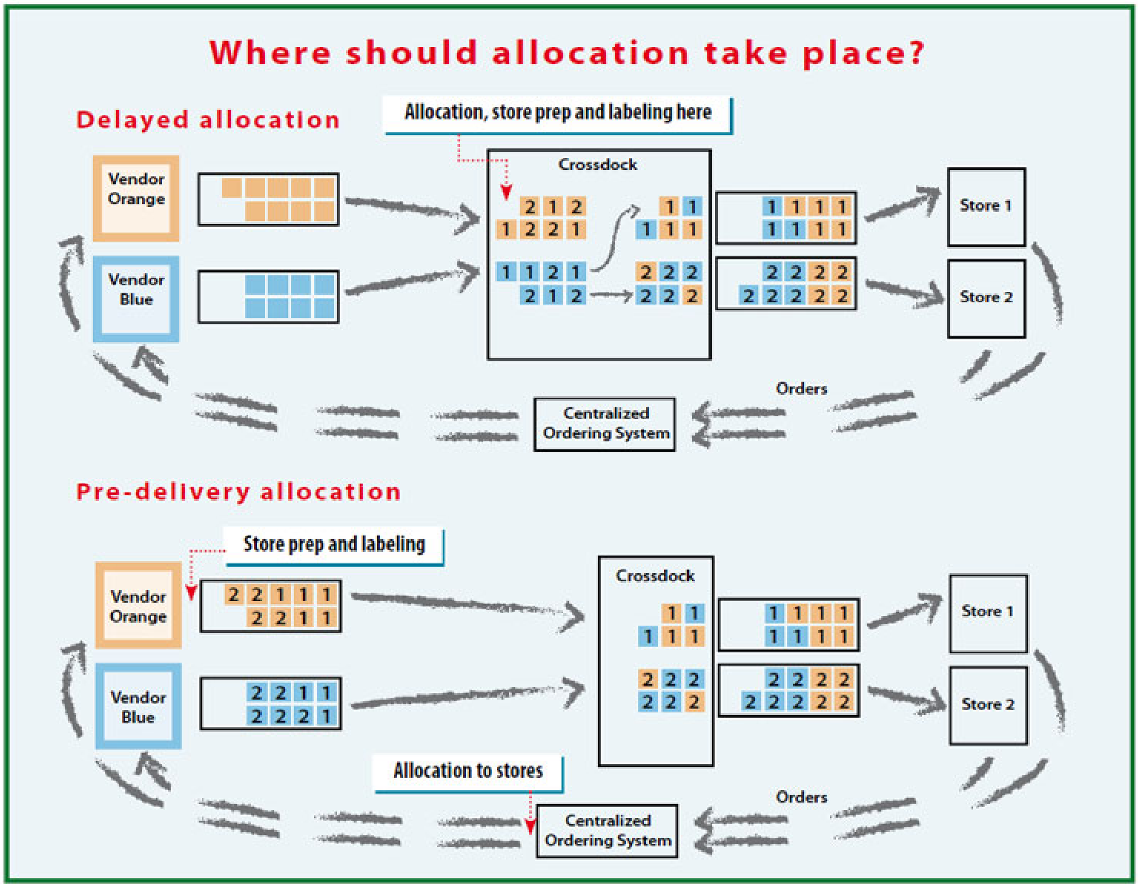


**Figure 1.3:** Pre-C and Post-C operations with two vendors and two stores (Gue, 2007)

Two additional typical operations in cross-dock are pre-distribution cross-docking operations (Pre-C) and post-distribution (Post-C) cross-docking operations (Bartholdi et al., 2001, Yan and Tang, 2009). In Pre-C operations the manufacturer prepares products for distribution. Bar codes are attached and the products are labeled, priced, and directly loaded and delivered to each store through the cross-dock. In Pre-C operations the manufacturer is required to know the demand quantity of every store to label the products accordingly. In Post-C operations the distribution preparation work and the operations cost for distribution preparation work are transferred from the manufacturing base to the cross-dock, closer to customers (for example see Figure 1.3).

Cross-dock can also be divided into single-stage cross-dock, two-stage cross-dock, and free-stage cross-dock based on the method used for staging freight (Bartholdi et al., 2001). A single stage cross-dock uses the method of placing inbound pallets on staging lanes at the receiving doors if the final destination is not known and at the shipping doors if the destination is known. A two-stage cross-dock has a center aisle for workers to sort and repack the pallets (see Figure 1.4). In a free-staging cross-dock there is a free staging area near receiving and/or shipping doors. In addition to these staging areas, cross-docks may also use a separate area to sort and repack the inbound pallets before transferring them to shipping doors (Napolitano 2000).

In a typical cross-dock the products are brought in by the inbound trucks as pallets, boxes, cartons or packages (Apte and Viswanathan (2000). These items can be either Pre-C or Post-C, and they are unloaded at the receiving door (or receiving dock, or inbound door). If they are Pre-C, the items are scanned, verified, and sorted by destination and sent to the appropriate shipping doors (or shipping docks, or outbound doors). If they are Post-C, the items are sorted, repacked, priced, and labeled for desired destinations and then sent to the shipping doors.

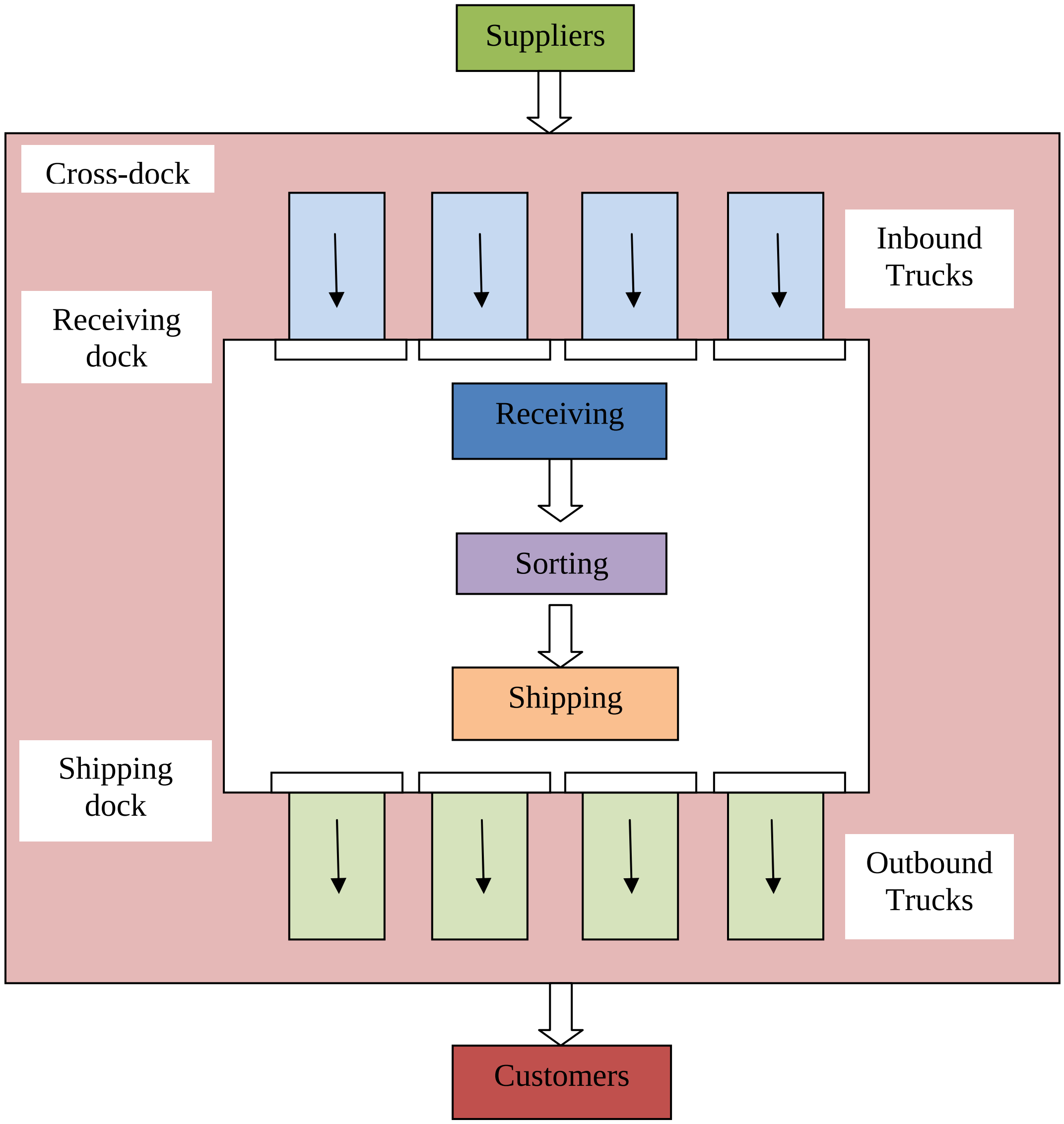


**Figure 1.4:** Flow in a typical cross-dock facility (adapted from Yu and Egbelu, 2008)

The above mentioned jobs are handled by workers or machines depending on the sophistication of the cross-dock facility. Conveyors or forklifts are commonly used to sort and move products in several cross-dock facilities. The number of doors used for receiving and shipping varies from facility to facility. Some products are stored temporarily until the appropriate outbound truck arrives at the shipping door. Perishable items and products that require refrigeration are moved quickly. The products are loaded into outbound trucks at the shipping docks. Once the outbound truck is loaded with appropriate products for a destination, the outbound truck leaves the dock. Information technology is used for the flow of information in cross-docks. For high volumes it is imperative to use technology, but for low volumes and Pre-C products minimal manual supervision may be enough.

Different assumptions can be made to solve problems involving cross-dock operations. These operations should be done in such a way that the customers are served without delay while minimizing the cost. Apte and Viswanathan (2000) stated “the benefits of cross docking can only be achieved by: (1) effective handling of physical flow of goods; (2) effective deployment of advanced information technology to manage the flow of information; (3) effective use of full truck load (FTL) shipments; and (4) effective use of proper planning and management tools.” However, even with careful planning, there are always uncertainties that occur in cross-dock operations.

### 1.4 UNCERTAINTY IN CROSS-DOCK OPERATIONS

In addition to demand uncertainty from the consumer end and supply uncertainty from the manufacturing end (which are not in the cross-dock area), unexpected events happen in cross-docking facilities as well. Unloading inbound trucks, handling materials inside the cross-dock and loading outbound trucks are operations prone to uncertainty. Several scenarios can be considered depending on the available work force, the skills of the workers, the number of operating machines, the amount of products to be handled, and the types of products to be handled. Transportation delays have a major impact on the cross-dock operations. An outbound truck needs to wait until the arrival of the inbound truck that has a pallet that needs to be shipped through that outbound truck. A delay in the arrival time of inbound trucks results in a delay on the departure time of the outbound trucks. In this dissertation, we address some of these uncertainties in unloading, transportation and scheduling in cross-dock problems.

### 1.5 DISSERTATION OUTLINE

In chapter 1, an introduction to cross-docking and the general motivation for this research are covered along with some background information on cross-docking and its operations. The rest of this dissertation is organized as follows: Chapter 2 consists of the literature review on different types of cross-docking problems. It is divided into sections of various aspects of cross-docking. Chapter 3 describes uncertainty in unloading operations with a simple scenario-based problem description, formulation, examples, and a heuristic method that gives bounds to the solution together with experiments and results. Chapter 4 explores a solution approach using a meta-heuristic method combined with the heuristic method described in chapter 4. In chapter 5, uncertainty is controlled by using a protection-based robust approach and the results are discussed. Chapter 6 and chapter 7 include cross-dock operations and uncertainties that arise in two different Texas firms, Goodwill Industries of Central Texas and H-E-B respectively. Problem formulations with and without considering uncertainties and solution methods are discussed together with results. Finally, in chapter 8, conclusions and future research directions are presented.

# Chapter 2: Literature Review

Cross-docking has recently found renewed interest as a logistic strategy to reduce storage, handling, and transportation costs (Boysen and Fliedner, 2010). Before 2000, little research had been conducted in cross-dock operations, and only a handful of research articles are available for that period of time. A wide variety of articles were published in the literature after 2000. The literature review is divided into several topics within cross-docking.

## 2.1 Cross-dock Layout, Shape and Door Assignment

A good layout to the cross-dock facility contributes significantly to the efficiency of operations inside the cross-dock. The research in this area mainly addresses the determination of the best layout in cross-docking. The size of the cross-dock is determined in terms of the number of doors. The doors are spaced as close as possible while accounting for safety in backing trailers to the opening. Normally the doors are equally spaced with a 12-foot offset. At the LTL carrier Yellow Transportation, the average cross-dock hub—not end-of-line satellite terminal—has about 180 doors and ranges in size from 63 to 300 doors. The percentage of receiving doors ranges between 21% and 67% (Trussell, 2001).

Heragu, et al. (2005) developed a linear programming model and a heuristic algorithm to jointly determine the functional area sizes in warehouse (storage, forward and cross docking area) and the product allocation so that the total material handling cost is minimized. Vis and Roodbergen (2008) proposed a cross-docking layout model as a minimum cost flow problem to determine the temporary storage location for incoming unit loads in order to minimize the travel distances of forklift trucks with these unit loads. To solve the model they proposed a row-based storage assignment algorithm.

Bartholdi and Gue (2004) considered the various shapes of cross-docks and found that the best shape is determined by the freight flow pattern and the number of facilities. The best shapes are I for small to medium dock sizes, T for dock sizes between 150 and 200 and X for larger than 200 doors. Hauser and Chung (2006) considered the cross-docking in a manufacturing industry context (Toyota Motor Manufacturing plant, USA). They proposed a Genetic Algorithm to rearrange the layout to minimize the labor workload and lead time. The study found that a V layout performs better than the current layout shapes of I and T.

In addition, the dock door assignment is considered as part of the design of the cross-dock layout and operations. In this proposal, the door assignment is discussed with layout. The dock door assignment determines the short- to mid-term planning of the cross-dock operations and affects the material flow inside the cross-dock. Good assignment of dock doors will make operations more efficient. Gue (1999) developed a material flow model to minimize the flow inside the cross-dock. Using ‘look ahead scheduling’ to determine the dock door assignment resulted in a layout that reduced the cost significantly compared to the model using ‘first-come-first-served policy.’ Bartholdi and Gue (2000) also proposed a layout design model to improve the layout through proper assignment of the dock doors and minimizing the travel time, material handling, and congestion. This model reduces the labor load and labor cost through efficient layout of cross docking, and it is solved by Simulated Annealing.

Yanchang and Min (2009) considered three cross-dock designs—namely rectangular with conveyor, rectangular without conveyor, and cross shape with conveyor—and compared their performances. They proposed a model that assigns trucks to dock doors to minimize the total distance of indoor freights and used a Genetic Algorithm to conclude that rectangular shaped hubs with conveyors is the most efficient design.

Tsui and Chang (1990) proposed a model to determine the assignment of receiving doors and shipping doors that minimizes the travel distance of the forklifts. A microcomputer-supported tool based on bilinear programming was used to solve the model. Tsui and Chang (1992) extended this work by proposing a solution for the bilinear programming model using a Branch and Bound algorithm. Oh, et al. (2006) formulated a non-linear mathematical model to assign the destinations to the shipping doors in a cross-docking system of mail distribution centers, minimizing the travel distance of the pallets in the center and solving the problem using two different methods: three-phase heuristic procedure and Genetic Algorithm.

Lim, et al. (2006a) considered the capacity of cross-dock and time window constraints in the truck dock assignment problem. The model assigns the truck to a dock door between its time windows, minimizing the total shipping distance of transferring cargo from the inbound to the outbound dock and it was solved by Tabu Search and

Genetic Algorithm. The objective is modified to minimize the total cost that consists of operation cost and penalty cost of unfulfilled demand in Lim, et al. (2006b), who used Genetic Algorithm to solve the model. Miao, et al. (2009), continued this work and they used Tabu search combined with Genetic Algorithm to solve the problem.

Bozer and Carlo (2008) considered the assignment of trailers to dock doors minimizing the material handling workload in rectangular cross-docks using Simulated Annealing. Ko, et al. (2008) minimized both the number of workers engaged in loading operations and the imbalance ratio among the workers in assigning destinations to shipping dock doors by using Genetic Algorithm with a line-balancing heuristic.

### 2.2 CROSS-DOCK NETWORK DESIGN

Cross-dock network design problems determine the number and the location of cross- docks and the number of vehicles in the network. Donaldson et al. (1998) considered a model to determine the number of vehicles, their routes, and the flow for first class mail through the United Stated Postal Services that minimizes transportation costs. Ratliff et al. (1998) proposed a load-driven network system design to determine the number and location of the mixing centers in the automotive delivery system in a railroad network context minimizing the total delay, which consists of transportation delay and loading delay of the above models use relaxation and branch and bound.

Syarif et al. (2002) proposed a spanning tree-based Genetic Algorithm to determine the number of plants and distribution centers that should be opened in order to minimize the cost. Jayaraman and Ross (2003) suggested a two-phase model that determines the Production Logistic Outbound and Transportation (PLOT) design to minimize the total cost. The model determines which cross-docks and warehouses are opened and their product allocation using a meta-heuristic procedure based on Simulated Annealing. The model in Ross and Jayaraman (2007) is similar to the work in Jayaraman and Ross (2003), but to solve the model, they use a better heuristic method, GABU-SA (combination of Genetic Algorithm-Tabu Search-Simulated Annealing) and RESCALE-SA.

Sung and Song (2003) presented a model that determines the location of cross-docks and allocating vehicles for the associated direct services in the context of service networks to minimize the costs of locating cross-docks and allocating vehicles using a Tabu Search algorithm. Sung and Yang (2008) proposed a more efficient branch-and-price algorithm as an exact algorithm to solve this model. Gümüs and Bookbinder (2004) formulated a model to determine the cross-docking network design to minimize the cost. Chen et al. (2006) proposed a cross-docking network design model that considers delivery and pick up time windows, warehouse capacities and inventory handling cost and minimizes the transportation and inventory cost. They used Tabu Search and Simulated Annealing in solving the model.

Bachlaus et al. (2008) formulated a multi-objective optimization problem to minimize the fixed and variable costs and to maximize the plant flexibility and volume flexibility and used Hybrid Taguchi-Particle Swarm Optimization (HTPSO) to design a network consisting of suppliers, plants, distribution centers, cross docks, and customer zones. Kreng and Chen (2008) presented a model that determines a production-distribution strategy to decide whether to use cross-dock or traditional distribution center.

### 2.3 Cross-dock Operations

Cross-dock facilities should be operated efficiently in order to save costs and deliver goods without any delay. Such facilities are required to make operational decisions involving short term (daily, weekly) planning. The dock door assignment problem comes under this category as well and is addressed in section 2.1 together with layout and design. Truck or trailer scheduling determines the sequences of inbound and outbound trucks. The daily operation of cross-dock facilities and the smoothness of operations inside the cross-dock facilities are managed with good scheduling. Poor truck scheduling can lead to congestion and poor product flow. Due to long processing times or long makespan the cost increases and the objective of saving money is lost. Scheduling problems often use the objective of minimizing the makespan in order to minimize the cost.

Yu (2002) identified thirty-two models that could be considered based on the operating conditions of the facility and strategies employed. He studied three different models to schedule both inbound and outbound trucks to minimize the makespan. He assumed that the receiving and shipping docks are separate and the items contained in an inbound truck and the items needed for an outbound truck are known in advance for all three models. His solution methods included complete enumeration, Branch and Bound, and several heuristic algorithms.

The truck scheduling problem has traditionally been formulated as a machine scheduling problem (Pinedo, 2002). For example, Li et al. (2004) formulated the cross-docking problem as a machine scheduling problem where the inbound and outbound trucks are modeled as jobs and workers are seen as machines. The objective is to minimize the total penalty of earliness and tardiness in incoming and outgoing containers. To solve the resulting NP-hard scheduling problem, they proposed two heuristic solution strategies, Squeaky Wheel Optimization embedded in a Genetic Algorithm and Linear Programming within a Genetic Algorithm. Recently, Alvarez-Perez et al. (2009) re-considered the same scheduling problem as in Li et al. (2004), but proposed a more efficient alternative meta-heuristic solution approach called Reactive GRASP and Tabu Search (RGTS).

McWilliams et al. (2005) proposed a model that determines the schedule of trailers in the unloading dock to minimize the time span of transfer operation. The model used a simulation-based scheduling algorithm using a Genetic Algorithm (GA). Ley and Elfayoumy (2007) suggested a scheduling model of inbound and outbound trucks to minimize the distances between them and used a Genetic Algorithm (GA) to solve the problem.

Yu and Egbelu (2008) developed a scheduling model for a single inbound dock and a single outbound dock that can determine the truck schedule and product allocation simultaneously. They assume a temporary storage buffer to hold items at the shipping dock and minimize the total operation time. Arabani et al. (2009) followed this work by applying simulated annealing algorithm to minimize the makespan. Vahdani and Zandieh (2010) continued the work of Yu and Egbelu (2008) by proposing five meta-heuristic methods to solve and improve the solution: Genetic Algorithm (GA), Tabu Search (TS), Simulated Annealing (SA), Electromagnetism-like Algorithm (EMA) and Variable Neighborhood Search (VNS). Soltani and Sadjadi (2010) propose a hybrid Simulated Annealing method and a hybrid Variable Neighborhood Method for scheduling the trucks in cross-dock to minimize the flow time.

Song and Chen (2007) studied the model that consists of multiple inbound vehicles and one outbound vehicle to minimize the makespan and used heuristic methods based on Johnson's rule. Chen and Lee (2009) developed a model with one inbound and one outbound trailer related to the model of Song and Chen (2007) to minimize the makespan. They modeled the problem as a two-machine flow shop problem and used Branch and Bound algorithm to solve the model. This work was extended by modifying the model in which at least one stage has a parallel machine as a hybrid cross-docking scheduling problem (Chen and Song, 2009). This model allows more than one inbound and outbound trailer, and it is solved by a heuristic method based on Johnson's rule. Chen et al. (2009) studied a two-stage scheduling problem and introduced some special polynomially solvable cases. They proposed several heuristic algorithms and approximation ratio analyses to address the scheduling problems.

Shakeri et al. (2008) developed a generic model that combined the truck scheduling problem with the dock door assignment problem. Wei et al. (2009) proposed the shortest remaining production time rule (SRT) for dispatching the truck by using a system assisted by radio frequency identification (RFID) technology. Maknoon and Baptiste (2009) proposed a truck scheduling model limited to a single incoming and outgoing door for the inbound and outbound semitrailer to minimize the moving path for products using dynamic programming and a heuristic approach.

McWilliams (2009) proposed a dynamic load-balancing algorithm to solve the parcel hub scheduling problem (McWilliams et al., 2005) formulating it as linear binary model. Boysen et al. (2010) considered the truck scheduling problem with one inbound door and one outbound door to minimize the makespan and used a decomposition approach. Boysen (2010) formulated a truck scheduling model for the food industry where storage is forbidden and a strict cooling procedure is followed. The products in the receiving dock are immediately loaded to the outbound truck and shipped to the customer. This model minimizes the processing time, flow time, and tardiness of outbound trucks using dynamic programming and Simulated Annealing. Recently, Zhang et al. (2010) considered alternative objectives and developed a multi-objective optimization model for the truck scheduling problem and proposed a restriction-approximation approach to solve the problem.

The transshipment model deals with the optimal amount of products to be sent between optimal locations at the optimal time through optimal routes. The flow allocation of products inside the cross-dock from inbound trucks to outbound trucks to satisfy customer demand can be modeled as a transshipment problem. It answers questions such as how much to ship, at what times, on which routes and between which locations (Lim et al., 2005). They consider the capacity of the cross-docks, the inventory and time window constraint to extend the traditional transshipment model to study various models that combine multiple or single delivery, multiple or single shipping and fixed or flexible scheduling. Miao et al. (2008) considered the transshipment problem with fixed scheduling and strict time windows for shipping and delivery to minimize inventory holding costs and shipping costs. Miao et al. (2009) formulated a model for single shipping and delivery with a fixed schedule to minimize the holding penalty cost, the transportation cost and inventory holding cost. They used Genetic Algorithm-based methods to solve both of the above models.

Larbi et al. (2007) considered a transshipment operation scheduling model inside the cross-dock to minimize truck replacement cost and inventory holding cost. In their graph- based model they assume one inbound and one outbound door and propose using the shortest path method. They extend the work as a dynamic programming model and solve it using heuristic methods by considering multiple inbound and outbound doors (Larbi et al., 2009). In addition they studied transshipment operations in cross-docks under the availability of full information, partial information, and no information and provide a different solution method for each case (Labri et al., 2010).

Unnikrishnan et al. (2009) proposed a two-stage stochastic formulation to determine the optimal storage capacity for transshipment nodes in a shipper carrier network under stochastic demand. They used several solution methods, such as L-shaped method, regularized decomposition, and introduce a capacity-shifting heuristic to solve the problem. Comparing pre-distribution cross-docking (Pre-C) and post-distribution cross-docking (Post-C) models, Pre-C has lower operations cost at the cross-dock but higher transshipment costs for higher demand uncertainty, while Post-C provides higher operations costs at the cross-dock but lower transshipments costs (Yan and Tang, 2009, Tang and Yan, 2010). Not much research has been done in product allocation and vehicle routing in cross-dock operations. Lee et al. (2006) and Wen et al. (2009) considered the vehicle routing problem, and Li et al. (2008, 2009) investigated the product allocation problem in cross docking.

### 2.4 Survey and Classification

Boysen and Fliedner (2010) review the literature using seven categories ordered from strategic to operational. They introduce a classification of deterministic truck scheduling using a tuple-notation which is commonly followed in machine scheduling and queueing models. They use the classification based on door environment, operational characteristics, and objectives denoted by $\alpha$, $\beta$ and $\gamma$ respectively and together as the tuple $[\alpha | \beta | \gamma]$. With the help of this classification, existing literature is reviewed and future research needs are identified. Agustina et al. (2010) reviews the mathematical models for cross-dock problems using three categories: operational level, tactical level, and strategic level.

Mula et al. (2010) considered supply chain production planning models that considers transport as a product distribution resource and those that center on tactical and/or operational decision levels and their possible combination with aspects of a strategic nature. They propose a taxonomy framework based on the following elements: supply chain structure, decision level, modeling approach, purpose, shared information, limitations, novelty, and application. They accomplish that by expanding the taxonomy proposed by Huang et al. (2003) which consists of four classification criteria: supply chain structure, decision level, modeling approach, and shared information. They provide the classified review in table format.

Almost all the research in cross-dock problems considers a deterministic approach. In this dissertation we propose a stochastic approach by introducing uncertainty in cross-dock problems. As a first step, in the next chapter we consider a simple problem where the inbound trucks are unloaded or the inbound trucks are processed. We consider uncertainty in the time to unload. In addition, there is a weight assigned to each truck indicating the priority of the truck to be unloaded. We model this problem using two different robust approaches and discuss the solutions together with a meta-heuristic solution approach in the next three chapters. In two chapters following that we consider uncertainties in the cross-dock operations of two different Texas industries.

# Chapter 3: Scenario-Based Robust Unloading Problem

The unloading schedule of inbound trucks plays an important role in the smooth operation of cross docks. Careful planning is needed to unload trucks in the inbound dock to minimize the waiting time of trucks in the outbound dock. When there is uncertainty in unloading times it adds more pressure on the careful scheduling of these trucks to unload. The product types of the truck (for example: perishable products, products for disaster relief, products with delivery time window), the destination of the products, the transportation route all play a role in making the decision. We introduce weights to the trucks to capture these different factors.

Though uncertainty has been used in other areas of transportation research (e.g. Ukkusuri et al., 2007; Khoury and Hobeika, 2007; Duthie et al., 2011; Szeto et al., 2011; Ng and Waller, 2011; Ng et al., 2011) to the best of our knowledge, there is no study that incorporates some form of uncertainty into the truck scheduling problem. As noted in Boysen et al (2010), handling times of trucks are merely estimated average times that are bound to have heavy inaccuracies. It is possible to either overestimate or underestimate the unloading times which will influence the optimal solution to a great extent. In other words, truck unloading schemes obtained based on the assumption of determinism might prove suboptimal in the real-world. More specifically, we are interested in the construction of robust (to be defined in Section 3.2) schedules. To this end, we examine a basic version of the truck scheduling problem and examine the impact of uncertainty on the truck schedules.

## 3.1 PROBLEM DESCRIPTION AND MODEL FORMULATION

As the main contribution of the current research is the introduction of uncertainty in the cross-dock problem, we consider the special case in which we are interested in unloading *n* inbound trucks as fast as possible, in the presence of stochastic unloading times; in upcoming chapters we address the situation in which we also consider the outbound operations. Another contribution is that we include a weight for each truck to denote its importance in the order of unloading. As will be seen below, the uncertainty in unloading times together with the inclusion of weights will lead to a substantially difficult problem. The unloading time for truck *j* is denoted by $p_{j,} j = 1,2,\ldots,n$.

Furthermore, we assume that

1. All trucks are available for unloading at time 0.
2. No two trucks can be unloaded at the same time (e.g. because of the limited availability of workers and/or equipment).
3. Trucks are fully unloaded before other trucks can be processed.
4. Each truck *j* has a weight $w_j$ associated with it, denoting the importance of processing truck *j* relative to the other trucks waiting on the dock to be processed (e.g. trucks with perishable goods might have a higher weight). The weights could also indicate holding costs.

The goal is to determine the optimal schedule in which to unload the trucks. More specifically, the objective is to minimize the total (sum of) weighted completion times (*TWC*), which can serve as a service measure of the cross-dock facility. When the unloading times are deterministic, the above (single scenario) truck unloading problem can be modeled as a machine scheduling problem, in which the trucks are jobs and workers the machines. In standard machine scheduling terminology (Pinedo, 2002), one can write $1 \| \sum w_j C_j$.

We define $x_{jk} = 1$ if truck *j* precedes truck *k* in the unloading schedule and 0 otherwise. Here, the values $x_{jj} = 0$ for all *j*. The completion time of truck *j* is then equal to $\sum_{k=1}^{n} p_k x_{kj} + p_j$ and the weighted completion time of truck *j* is $w_j \left( \sum_{k=1}^{n} p_k x_{kj} + p_j \right)$. The integer programming formulation of (*TWC*) is

$$z = \min \sum_{j=1}^{n} \sum_{k=1}^{n} w_j p_k x_{kj} + \sum_{j=1}^{n} w_j p_j$$

subject to

$$x_{kj} + x_{jk} = 1 \qquad \text{for } j, k = 1, \ldots, n, \ j \neq k,$$

$$x_{kj} + x_{lk} + x_{jl} \geq 1 \qquad \text{for } j, k, l = 1, \ldots, n, \ j \neq k, \ j \neq l, k \neq l,$$

$$x_{jk} \in \{0,1\} \qquad \text{for } j, k = 1, \ldots, n,$$

$$x_{jj} = 0 \qquad \text{for } j = 1, \ldots, n.$$

The constraints are explained in the robust version below. The Weighted Shortest Processing Time first (WSPT) rule provides an optimal sequence for (*TWC*) e.g. see Pinedo (2002). According to this rule, trucks are processed in the decreasing order of their weighted processing times, $w_j$ / $p_j$. In this paper, we allow for uncertainties in the unloading times $p_j$. Depending on the uncertainties we will have several scenarios. This leads to a scenario-based robust scheduling problem.

The uncertainties for general robust optimization problems can be described by the scenario set $S$ (Kouvelis and Yu, 1997). Each scenario $s \in S$ may be realized with a positive, but perhaps unknown, probability. The cost of making a decision $x$ under $s \in S$ is $f^s(x)$. The feasible region under scenario $s \in S$ is $X_s$. No restrictions are placed on the scenario set $S$. In the case where unloading times are given by intervals, i.e., interval $[l_i, u_i]$ to specify the range of processing times of job $i$, there are an infinite number of scenarios. If we consider a finite number of scenarios, the feasible region can be defined as $X = \bigcap_{s \in S} X_s$. For the robust single machine scheduling problem (*RSS*), the interval processing times (Daniels and Kouvelis, 1995) and the discrete scenario (Yang and Yu, 2002) were considered and the correlations among processing times of different jobs can be fully captured through finite scenarios while it cannot be sufficiently addressed through intervals.

In our problem, we decided to avoid using interval unloading times since adding weights complicates the problem and with the interval unloading times it is not possible to address the problem with finite number of scenarios. The unloading time vector for scenario $s \in S$ is $p^s = (p_1^s, p_2^s ..., p_n^s)$, which is entirely different from the deterministic problem where each truck has a single unloading time. Now the constraint sets for all scenarios are the same. i.e., $X_s = X,\ s \in S$ with $X$ containing the assignment constraints and integer requirements on all the variables. We can formulate the robust total weighted completion time problem (*RTWC)* as a min-max problem and take three robust measures. Here the objective is to minimize the maximum flow time considering all the scenarios. The objective function for scenario s is

$$f^s(x) = \sum_{j=1}^{n}\sum_{k=1}^{n} w_j p_k^s x_{kj} + \sum_{j=1}^{n} w_j p_j^s$$

We can define the Robust total weighted completion time *(RTWC)$_R$* problem where $R \in \{a, d, r\}$ as

$$z_R = \min\ y \tag{3.1}$$

subject to

$$y_R^s \geq \sum_{j=1}^{n}\sum_{k=1}^{n} w_j p_k^s x_{kj} + \sum_{j=1}^{n} w_j p_j^s \qquad \text{for } s \in S, \tag{3.2}$$

$$x_{kj} + x_{jk} = 1 \qquad \text{for } j, k = 1,\ldots,n,\ j \neq k, \tag{3.3}$$

$$x_{kj} + x_{lk} + x_{jl} \geq 1 \qquad \text{for } j, k, l = 1,\ldots,n,\ j \neq k,\ j \neq l, k \neq l, \tag{3.4}$$

$$x_{jk} \in \{0,1\} \qquad \text{for } j, k = 1,\ldots,n, \tag{3.5}$$

$$x_{jj} = 0 \qquad \text{for } j = 1,\ldots,n, \tag{3.6}$$

where

$$y_R^s(x) = \begin{cases} y & R = a, \\ y + z^s & R = d, \\ z^s(1+y) & R = r. \end{cases} \tag{3.7}$$

and $z^s$ is the optimal objective value of *(TWC)* with unloading time vector $p^s$. That is, $z^s = \min_{x \in X} f^s(x)$ is the optimal objective value under a single scenario $s \in S$. Equation (3.1) gives the objective function of the robust formulation. It minimizes the maximum weighted completion time over all scenarios. Constraints (3.2) picks the maximum weighted completion time for each scenario *s*. Constraints (3.3) make sure either truck *k* precedes truck *j* or truck *j* precedes truck *k* and not both (only one precedence is true for any pair). Constraints (3.4) ensure that between any three trucks *j*, *k*, *l*, at least one of the following is true: Either truck *k* precedes truck *j* or truck *j* precedes truck *i* or truck *i* precedes truck *k* (at least one is true for any three trucks). Note that constraints (3.3) and (3.4) together ensure transitivity property (if truck *k* precedes truck *j* and truck *j* precedes truck *l*, then truck *k* precedes truck *l*) and the value of constraints (3.4) will not exceed 2. Constraints (3.5) and (3.6) are binary (non-negativity) constraints. The notation in constraints (3.2) is explained in (3.7) as three robust measures for the scenario-based optimization problem, indexed by "*a*" for absolute, indexed by "*d*" for deviation and indexed by "*r*" for relative deviation.

Using the robust formulation we obtain a schedule which will minimize the sum of completion times over all possible scenarios whereas if we follow the deterministic route when there is a change in unloading time (e.g. available employees, characteristics of the goods to be unloaded) the problem has a different optimal schedule under each scenario. Solving a large problem for each scenario could be expensive and timeconsuming. By introducing a stochastic approach with all possible scenarios, a single schedule could improve the efficiency of the cross-dock.

**Proposition 3.1:** *RTWC* is NP-complete for all three robust measures.

**Proof:** Yang and Yu (2002) proved that the robust single machine scheduling problem (*RSS*) is NP-complete for all three robust measures even in the case of two scenarios. According to the complexity hierarchy, the (*RSS*) problem reduces to (*RTWC*) problem with unit weight. This proves that the robust problem (*RTWC*) is NP-hard in the ordinary sense or NP-complete for all three robust measures. Q.E.D.

Also, the problem of finding the deviation and the problem of finding the relative deviation measure can be transformed into the problem of finding the absolute measure (Yang and Yu, 2002). So we only consider the problem of finding the absolute measure.

### 3.2 SOLUTION ALGORITHM: SURROGATE HEURISTIC FOR THE *RTWC* PROBLEM

We follow the notation used by Yang and Yu (2002) in the surrogate of the $(RSS)_a$ problem ((*RSS*) problem with absolute measure) using SPT rule (see section 3.4 for example) and modify the heuristic to solve (*RTWC*) problem. Now for the (*RTWC*) problem, we have the following. For a feasible schedule $\sigma$,

$z_a(\sigma) = \max_{s \in S} f^s(\sigma) = \max_{s \in S} \sum_{j=1}^{n} \sum_{k=1}^{n} w_j p_k^s x_{kj} + \sum_{j=1}^{n} w_j p_j^s$ is the objective function of the $(RTWC)_a$ problem ((*RTWC*) problem with *absolute* measure). We take $z_{SU}(\sigma) = \frac{1}{|S|} \sum_{s \in S} f^s(\sigma)$, the average total weighted completion time over all the scenarios, as a new objective function. The minimization problem with the new objective function and the same set of constraints as those in *(RTWC)$_a$* is called the surrogate relaxation of *(RTWC)$_a$*. If $\sigma_a$ and $\sigma_{SU}$ are the optimal solutions of *(RTWC)$_a$* and its surrogate relaxation, respectively, we can see that the following set of inequalities holds for $(RTWC)_a$ problem:

$$z_{SU}(\sigma_{SU}) \leq z_{SU}(\sigma_a) \leq z_a(\sigma_a) \leq z_a(\sigma_{SU})$$

We have $z_{SU}(\sigma_{SU}) \leq z_{SU}(\sigma_a)$, since $\sigma_{SU}$ is the optimal solution to the surrogate problem and $\sigma_a$ is a feasible solution to the surrogate problem. The surrogate problem is a relaxation of $(RTWC)_a$ and that gives the inequality $z_{SU}(\sigma_a) \leq z_a(\sigma_a)$. Since $\sigma_a$ is the optimal solution to $(RTWC)_a$ and $\sigma_{SU}$ is a feasible solution to $(RTWC)_a$ we have $z_a(\sigma_a) \leq z_a(\sigma_{SU})$.

Thus for the optimal value $z_a(\sigma_a)$ of the $(RTWC)_a$ problem, there is a lower-bound $z_{SU}(\sigma_{SU})$, and an upper-bound $z_a(\sigma_{SU})$ which is associated with a feasible solution.

Let $\theta = \dfrac{\max_{s \in S} f^s(\sigma_{SU})}{\min_{s \in S} f^s(\sigma_{SU})}$. Considering the upper bound, Yu and Yang (1998) show that $\dfrac{z_a(\sigma_{SU})}{z_a(\sigma_a)} \leq \dfrac{\theta |S|}{\theta + |S| - 1}$ for the (*RSS*) problem, that is, the ratio is bounded from above. This ratio is also bounded from below by 1. Next, we derive a complementary ratio relationship using the lower bound.

**Proposition 3.2:** $\frac{z_{SU}(\sigma_{SU})}{z_a(\sigma_a)} \geq \frac{1}{\theta|S|}$

**Proof:** $z_{SU}(\sigma_{SU}) = \frac{1}{|S|}\sum_{s\in S} f^s(\sigma_{SU})$

$$= \frac{1}{|S|}\sum_{s\in S} f^s(\sigma_{SU}) + \min_{s\in S} f^s(\sigma_{SU}) - \min_{s\in S} f^s(\sigma_{SU})$$

$$= \frac{1}{|S|}\sum_{s\in S} f^s(\sigma_{SU}) + \min_{s\in S} f^s(\sigma_{SU}) \times \frac{\max_{s\in S} f^s(\sigma_{SU})}{\max_{s\in S} f^s(\sigma_{SU})} - \min_{s\in S} f^s(\sigma_{SU})$$

$$= \frac{1}{|S|}\sum_{s\in S} f^s(\sigma_{SU}) + \frac{1}{\theta}\max_{s\in S} f^s(\sigma_{SU}) - \min_{s\in S} f^s(\sigma_{SU})$$

$$\geq \frac{1}{|S|}\sum_{s\in S} f^s(\sigma_{SU}) + \frac{1}{\theta} z_a(\sigma_{SU}) - \frac{1}{|S|}\sum_{s\in S} f^s(\sigma_{SU})$$

$$\geq \frac{1}{|S|}\sum_{s\in S} f^s(\sigma_{SU}) + \frac{1}{\theta} z_a(\sigma_{SU}) - |S|\frac{1}{|S|}\sum_{s\in S} f^s(\sigma_{SU})$$

$$= (1-|S|)z_{SU}(\sigma_{SU}) + \frac{1}{\theta} z_a(\sigma_{SU})$$

Now, $z_{SU}(\sigma_{SU}) \geq (1-|S|)z_{SU}(\sigma_{SU}) + \frac{1}{\theta} z_a(\sigma_{SU})$

$$\Rightarrow |S|z_{SU}(\sigma_{SU}) \geq \frac{1}{\theta} z_a(\sigma_{SU})$$

$$\Rightarrow z_{SU}(\sigma_{SU}) \geq \frac{1}{\theta|S|} z_a(\sigma_{SU})$$

Also, $z_a(\sigma_a) \leq z_a(\sigma_{SU})$

$$\Rightarrow \frac{z_{SU}(\sigma_{SU})}{z_a(\sigma_a)} \geq \frac{z_{SU}(\sigma_{SU})}{z_a(\sigma_{SU})} \geq \frac{1}{\theta|S|}$$ **Q.E.D**.

Thus, the ratio is bounded from below. This ratio is also bounded from above by 1. The above mentioned ratios give the worst case bounds in comparing $z_a(\sigma_a)$, the objective value to the *(RTWC)ₐ* problem with its surrogate value, $z_a(\sigma_{SU})$, and the objective value to the surrogate relaxation problem, $z_{SU}(\sigma_{SU})$. As one of these ratios gets closer to 1, the surrogate value will be closer to the absolute objective value. For example, if $\theta = 1$, then for any number of scenarios, the surrogate solution is the same as absolute solution since we have

$$1 \leq \frac{z_a(\sigma_{SU})}{z_a(\sigma_a)} \leq 1 \Rightarrow \frac{z_a(\sigma_{SU})}{z_a(\sigma_a)} = 1 \Rightarrow z_a(\sigma_{SU}) = z_a(\sigma_a).$$

The *(RTWC)ₐ* problem is hard to solve whereas the surrogate relaxation is converted to a single scenario problem *(TWC)*. We can apply the WSPT rule in the surrogate relaxation problem to get its optimal solution, $\sigma_{SU}$. Then we take that solution as an approximate optimal solution for the original problem. To get the weighted unloading times of surrogate problem for *n* trucks with |*S*| scenarios it will take $O(|S|n)$ time. After that, to apply the WSPT rule we need to sort the trucks according to their weighted unloading time. The sorting can be done in $O(n \log n)$ time. Thus the computational complexity of the surrogate heuristic is $O(n \log n + |S|n)$.

## 3.3 NUMERICAL EXAMPLES OF *RSS* AND *RTWC* PROBLEM AND SURROGATE HEURISTICS

### 3.3.1 Example: Optimal and Surrogate solution to *RSS* problem

The problem of finding a robust schedule for unloading trucks in a cross-dock facility under a set of scenarios can be described as finding a sequence which accommodates the uncertainties and at the same time produces an optimal schedule. A complete enumeration for this problem is to take each possible schedule (permutation), find the sum of completion times under each of the scenarios, and take the maximum sum of completion times among all of the scenarios. Then we pick the schedule with the minimum among these maximum completion times. Let us look at a simple example with three trucks and two scenarios for *RSS* problem which is a special case of *RTWC* problem with unit weight. The unloading times under scenarios 1 and 2 are

$$p_1^1 = 8 \qquad p_1^2 = 7$$
$$p_2^1 = 5 \qquad p_2^2 = 4$$
$$p_3^1 = 6 \qquad p_3^2 = 8$$

There are six different schedules:

$$(1,2,3) \qquad f = \max\{24+10+6, 21+8+8\} = \max\{40,37\} = 40$$
$$(1,3,2) \qquad f = \max\{24+12+5, 21+16+4\} = \max\{41,41\} = 41$$
$$(2,1,3) \qquad f = \max\{15+16+6, 12+14+8\} = \max\{37,34\} = 37$$
$$(2,3,1) \qquad f = \max\{15+12+8, 12+16+7\} = \max\{35,35\} = 35$$
$$(3,1,2) \qquad f = \max\{18+16+5, 24+14+4\} = \max\{39,42\} = 42$$
$$(3,2,1) \qquad f = \max\{18+10+8, 24+8+7\} = \max\{36,39\} = 39$$

The objective value is $\min\{40,41,37,35,42,39\} = 35$ and the optimal schedule is (2,3,1).

Let us look at the solution for surrogate heuristic. For this instance, we have

$$\overline{p_1} = \frac{15}{2} \qquad \overline{p_2} = \frac{9}{2} \qquad \overline{p_3} = 7$$

We have $\overline{p_2} < \overline{p_3} < \overline{p_1}$. Using the SPT rule the schedule is (2,3,1).

$$z_{SU}(\sigma_{SU}) = 3 \cdot \frac{9}{2} + 2 \cdot 7 + 1 \cdot \frac{15}{2} = 35$$

Here, we get the same optimal schedule as in the complete enumeration.

### 3.3.2 Example: Optimal and Surrogate solution to *RTWC* problem

Consider the above instance of example 3.3.1. Now we will add weights to the trucks.

$$\begin{array}{lll} p_1^1 = 8 & p_1^2 = 7 & w_1 = 5 \\ p_2^1 = 5 & p_2^2 = 4 & w_2 = 6 \\ p_3^1 = 6 & p_3^2 = 8 & w_3 = 4 \end{array}$$

Let us look at the complete enumeration for this problem. There are six different schedules:

$$\begin{array}{ll} (1,2,3) & f = \max\{40+78+76, 35+66+76\} = \max\{194,177\} = 194 \\ (1,3,2) & f = \max\{40+56+114, 35+60+114\} = \max\{210,209\} = 210 \\ (2,1,3) & f = \max\{30+65+76, 24+55+76\} = \max\{171,155\} = 171 \\ (2,3,1) & f = \max\{30+44+95, 24+48+95\} = \max\{169,167\} = 169 \\ (3,1,2) & f = \max\{24+70+114, 32+75+114\} = \max\{208,221\} = 221 \\ (3,2,1) & f = \max\{24+66+95, 32+72+95\} = \max\{185,199\} = 199 \end{array}$$

The objective value is min{194,210,171,169,221,199} = 169 = $z_a(\sigma_a)$ and the optimal schedule $\sigma_a$ is (2,3,1). In this problem, for surrogate heuristic, we need to calculate the average weighted unloading time for each job under these scenarios. We have the average unloading times,

$$\overline{p}_1 = \frac{15}{2} = 7.5 \qquad \overline{p}_2 = \frac{9}{2} = 4.5 \qquad \overline{p}_3 = \frac{14}{2} = 7.$$

The weighted average unloading time is $\frac{w_i}{\overline{p}_i}$.

$$\frac{w_1}{\overline{p}_1} = \frac{10}{15} = \frac{2}{3} \qquad \frac{w_2}{\overline{p}_2} = \frac{12}{9} = \frac{4}{3} \qquad \frac{w_3}{\overline{p}_3} = \frac{4}{7}$$

The schedule $\sigma_{SU}$ is (2,1,3) under WSPT rule.

$$z_{SU}(\sigma_{SU}) = 6(4.5) + 5(4.5 + 7.5) + 4 \cdot (4.5 + 7.5 + 7) = 163.$$

Here we have a different schedule, but the objective value is close to $z_a(\sigma_a)$ = 169. The objective value of the surrogate problem is a lower bound and the same solution applied to the original problem gives an upper bound. We see that $z_a(\sigma_{SU}) = 171$ and $z_{SU}(\sigma_a) = 168$.

We have $z_{SU}(\sigma_{SU}) \leq z_{SU}(\sigma_a) \leq z_a(\sigma_a) \leq z_a(\sigma_{SU})$.

### 3.4 COMPUTATIONAL RESULTS

The robust formulation and the surrogate heuristic with WSPT rule are used for the experiments. For the robust problem *RTWC*, the surrogate heuristic is coded in C++. Machine generated random numbers (0,100] for unloading times and random weights (0,10] are used in the experiments. The same program also uses the schedule obtained from the surrogate heuristic in the original problem under each scenario to get the maximum total weighted completion time among all scenarios. This is the upper bound and this is taken as the approximate solution to the problem.

In prior research, for the (*RSS*) problem, the actual solution to the problem was not compared with any of the heuristic solutions (e.g. Yang and Yu, 2002). We decided to solve each instance using the formulation and get the actual solution. This decision restricted us to try the problem sizes we have below. Also, most of the research involving inbound and outbound trucks in cross-dock problems considers a maximum of 20 trucks (e.g. Yu and Egbelu, 2008, Zhang et al., 2010). We think it is more important to compare the heuristic solution with the actual solution than to increase the size of the problem. Hence, the same unloading times and weights are used with the integer programming formulation in GAMS/CPLEX 11.2.0 to get the actual solution, which is the absolute solution to the objective function. In addition to that, we explore the possibility of using the average of the lower bound and upper bound to guide us in the future instead of comparing the solution to the upper bound alone.

The deviation and percent deviation of the lower bound, the upper bound and the average from the absolute objective value is then calculated. The percent deviation is calculated by using the following formulas for the lower bound, the upper bound, and the average respectively:

(1) Percent deviation = ((Lower bound – Absolute objective)/Absolute objective) 100%,

(2) Percent deviation = ((Upper bound – Absolute objective)/Absolute objective) 100%,

(3) Percent deviation = ((Average – Absolute objective)/Absolute objective) 100%,

The following tables and figures show the computational results for different numbers of jobs and scenarios. For each size, the program was run six times using six different sets of unloading times and weights.

**Table 3.1:** Five trucks and two scenarios, $n$ = 5, $s$ = 2.

| $z_{SU}(\sigma_{SU})$ | $z_a(\sigma_{SU})$ | $z_a(\sigma_a)$ | (LB+UB)/2 | % deviation from Absolute | | |
|---|---|---|---|---|---|---|
| LB | UB | Absolute | Average | LB | UB | Average |
| 3139.5 | 3144 | 3144 | 3141.75 | -0.14 | 0 | -0.07 |
| 3051.5 | 3746 | 3746 | 3398.25 | -18.54 | 0 | -9.28 |
| 2647.5 | 3162 | 2904 | 2904.75 | - 8.83 | 8.88 | 0.03 |
| 3236 | 3723 | 3507 | 3479.5 | -7.73 | 6.16 | -0.78 |
| 4431 | 4672 | 4540 | 4551.5 | -2.4 | 2.91 | 0.25 |
| 3015 | 4099 | 3459 | 3557 | -12.84 | 18.5 | -2.83 |

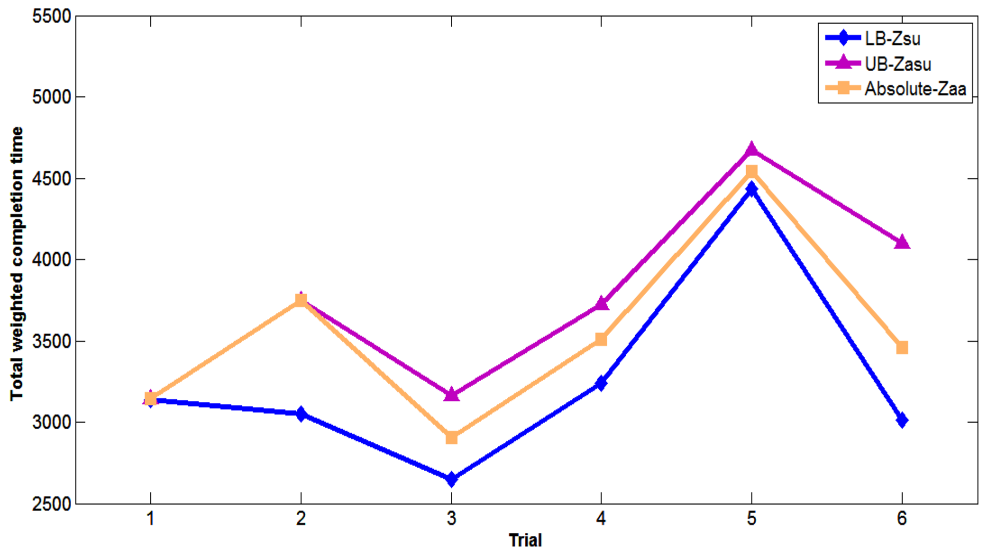


**Figure 3.1:** Comparison of lower bound (Zsu), upper bound (Zasu), and the absolute (Zaa) for five trucks and two scenarios, $n$ = 5, $s$ = 2.

We can see that (see Table 3.1 and Figure 3.1) two times the absolute objective is the same as the upper bound and four times it is close to the average. The upper bound deviates from the absolute objective by 0% to 18.5%.

**Table 3.2:** Five trucks and four scenarios, n = 5, s = 4.

| $z_{SU}(\sigma_{SU})$ | $z_a(\sigma_{SU})$ | $z_a(\sigma_a)$ | (LB+UB)/2 | % deviation from Absolute | | |
|---|---|---|---|---|---|---|
| LB | UB | Absolute | Average | LB | UB | Average |
| 1733.5 | 2768 | 2765 | 2250.75 | -37.31 | 0.11 | -18.6 |
| 3154.75 | 3699 | 3699 | 3426.88 | -14.71 | 0 | -7.36 |
| 4075.25 | 5142 | 4605 | 4608.63 | -11.5 | 11.66 | 0.08 |
| 2301.25 | 3506 | 3361 | 2903.63 | -31.53 | 4.31 | -13.61 |
| 1889.75 | 3130 | 3078 | 2509.88 | -38.61 | 1.69 | -18.46 |
| 2238.5 | 3248 | 3248 | 2743.25 | -31.08 | 0 | -15.54 |

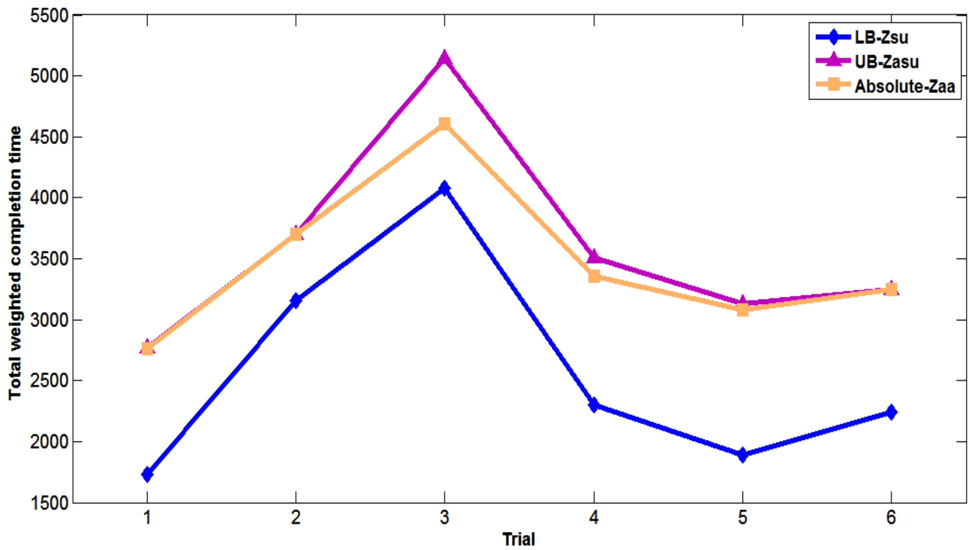


**Figure 3.2:** Comparison of lower bound (Zsu), upper bound (Zasu), and the absolute (Zaa) for five trucks and four scenarios, $n$ = 5, $s$ = 4.

Here, we see that (see Table 3.2, Figure 3.2) two times the absolute objective is the same as the upper bound and three times it is close to the upper bound and once to the average. The upper bound deviates from the absolute objective by 0% to 11.66%. In the case of five trucks when we increased the number of scenarios from two to five we observe that the upper bound is very close to the objective. Also, considering the upper bound or the average for five trucks, the deviation from the absolute objective is only by 0% to 4.31%.

**Table 3.3:** Ten trucks and three scenarios, $n = 10$, $s = 3$.

| $z_{SU}(\sigma_{SU})$ | $z_a(\sigma_{SU})$ | $z_a(\sigma_a)$ | (LB+UB)/2 | % deviation from Absolute | | |
|---|---|---|---|---|---|---|
| LB | UB | Absolute | Average | LB | UB | Average |
| 14158 | 16477 | 15185 | 15317.5 | -6.76 | 8.51 | 0.87 |
| 6701.33 | 7523 | 7301 | 7112.17 | -8.21 | 3.04 | -2.59 |
| 7110.67 | 7796 | 7649 | 7453.33 | -7.04 | 1.92 | -2.56 |
| 10647 | 12268 | 11260 | 11457.5 | -5.44 | 8.95 | 1.75 |
| 13841.7 | 14864 | 14776 | 14352.8 | -6.32 | 0.6 | -2.86 |
| 9793 | 11716 | 11438 | 10754.5 | -14.38 | 2.43 | -5.98 |

In this case (see Table 3.3, Figure 3.3) the split is even. The absolute objective is close to the upper bound three times, and it is close to the average three times. The upper bound deviates from the absolute objective by 0.6% to 8.95%. The deviation is within 9% in this case.

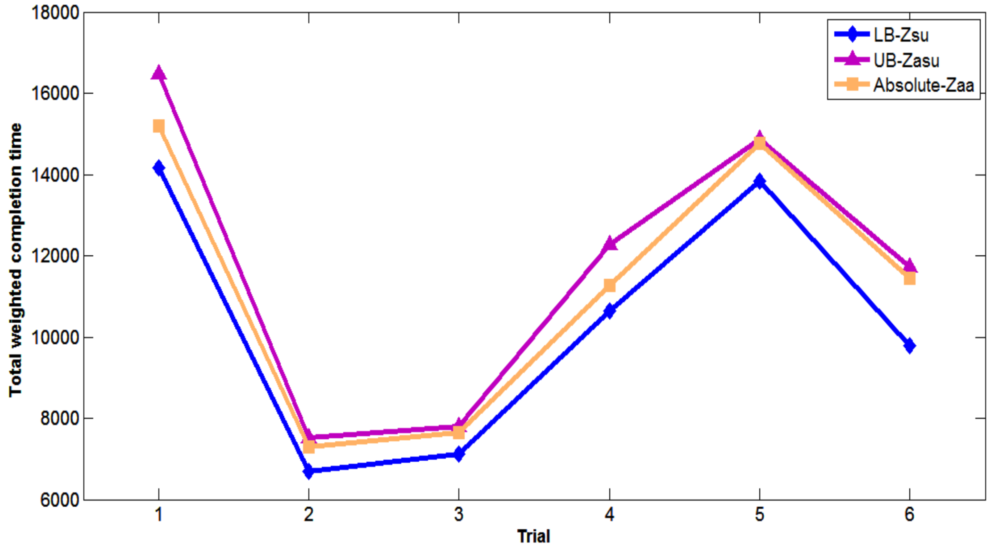


**Figure 3.3:** Comparison of lower bound (Zsu), upper bound (Zasu), and the absolute (Zaa) for ten trucks and three scenarios, $n$ = 10, $s$ = 3.

**Table 3.4:** Ten trucks and five scenarios, $n$ = 10, $s$ = 5.

| $z_{SU}(\sigma_{SU})$ | $z_a(\sigma_{SU})$ | $z_a(\sigma_a)$ | (LB+UB)/2 | % deviation from Absolute | | |
|---|---|---|---|---|---|---|
| LB | UB | Absolute | Average | LB | UB | Average |
| 11682.6 | 15451 | 15076 | 13566.8 | -22.51 | 2.49 | -10.01 |
| 8269.4 | 10833 | 10023 | 9731.2 | -17.5 | 8.08 | -2.91 |
| 11765.4 | 14537 | 13268 | 13151.2 | -11.33 | 9.56 | -0.88 |
| 10872.4 | 13481 | 12964 | 12176.7 | -16.13 | 3.99 | -6.07 |
| 9692.2 | 14873 | 13261 | 12282.6 | -26.91 | 12.16 | -7.38 |
| 9380.8 | 12050 | 11936 | 10715.4 | -21.41 | 0.96 | -10.23 |

When we increased the number of scenarios to five (see Table 3.4, Figure 3.4), the absolute objective is close to the upper bound three times and it is close to the average three times. The upper bound deviates from the absolute objective by 0.96% to 12.16%.

Considering the upper bound, or the average for ten trucks, the deviation from the absolute objective is only by 0.6% to 7.38%.

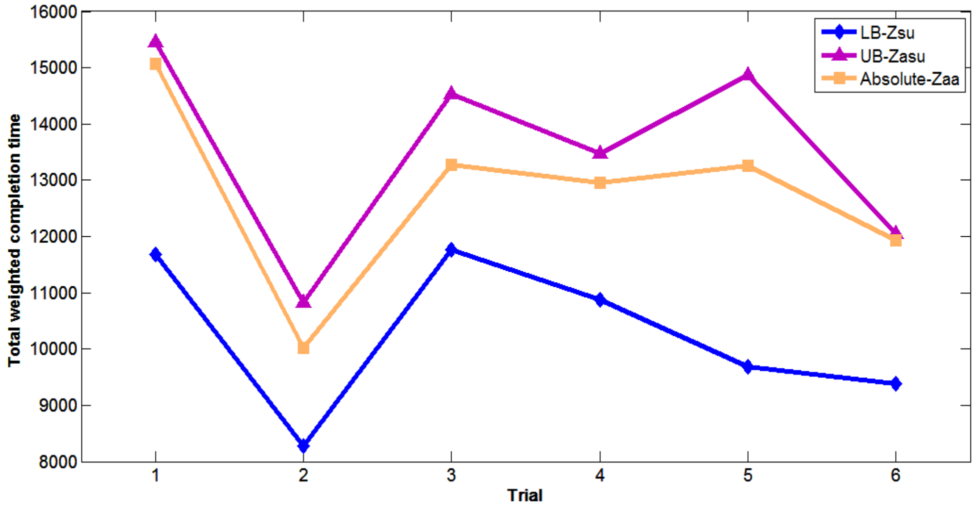


**Figure 3.4:** Comparison of lower bound (Zsu), upper bound (Zasu), and the absolute (Zaa) for ten trucks and five scenarios, $n$ = 10, $s$ = 5.

**Table 3.5:** Twenty trucks and five scenarios, $n$ = 20, $s$ = 5.

| $z_{SU}(\sigma_{SU})$ | $z_a(\sigma_{SU})$ | $z_a(\sigma_a)$ | (LB+UB)/2 | % deviation from Absolute | | |
|---|---|---|---|---|---|---|
| LB | UB | Absolute | Average | LB | UB | Average |
| 41218.8 | 52532 | 47471 | 46875.4 | -13.17 | 10.66 | -1.26 |
| 36923 | 41876 | 39894 | 39399.5 | -7.45 | 4.97 | -1.24 |
| 31165 | 35157 | 32465 | 33161 | -4.00 | 8.29 | 2.14 |
| 35121.6 | 38579 | 37367 | 36850.3 | -6.01 | 3.24 | -1.38 |
| 21458.8 | 27376 | 24665 | 24417.4 | -13.00 | 10.99 | -1.00 |
| 38325.4 | 46475 | 43204 | 42400.2 | -11.29 | 7.57 | -1.86 |

In the case of 20 trucks, the absolute objective is close to the average all six times (see Table 3.5). The upper bound deviates from the absolute objective by 3.24% to

10.99%. Considering the upper bound or the average for 20 trucks the deviation from the absolute objective is only by 1% to 2.14% which is very good considering five scenarios (see Figure 3.5).

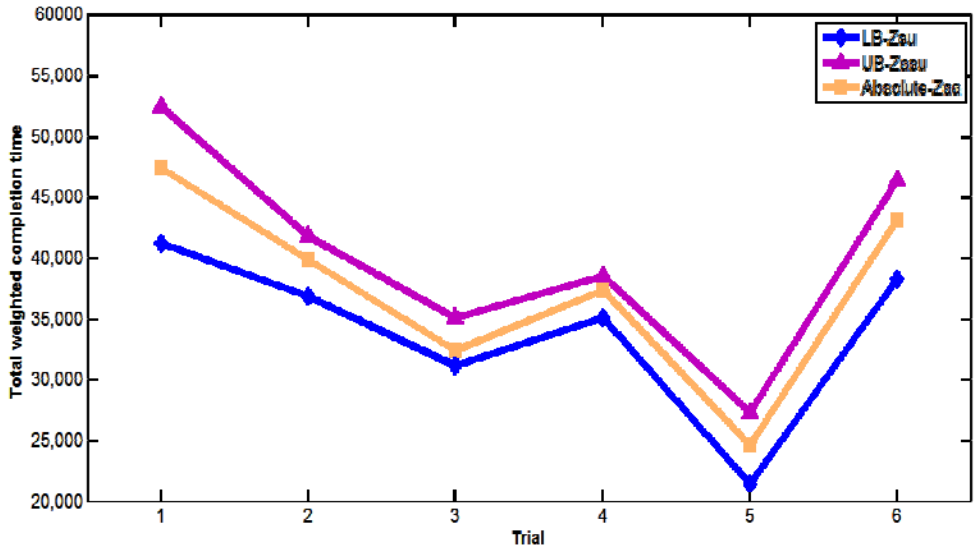


**Figure 3.5:** Comparison of lower bound (Zsu), upper bound (Zasu), and the absolute (Zaa) for twenty trucks and five scenarios, $n$ = 20, $s$ = 5.

From the above tables and figures we have summarized the number of times the absolute objective is close to the lower bound $N$(LB) and the number of times the absolute objective is close to the upper bound $N$(UB) among the six times the experiments are done for each size in Table 3.6. Here, $n$ denotes the number of jobs, $s$ denotes the number of scenarios, and $N$(AVE) denotes the number of times the absolute objective is closer to the average than the upper bound. From the table we observe that in most of the trials the absolute objective is closer to the upper bound than the lower bound. Another observation is the absolute objective value is relatively closer to the average. Considering the upper bound or the average for each case, whichever is closer to the absolute objective, we see that the values differ from the absolute objective value within 8%.

**Table 3.6:** Comparison of number of times close to the actual objective

| $n$ | 5 | 5 | 10 | 10 | 20 | Total |
|---|---|---|---|---|---|---|
| $s$ | 2 | 4 | 3 | 5 | 5 | |
| $N$(LB) | 3 | 1 | 2 | 0 | 1 | 7 |
| $N$(UB) | 3 | 5 | 4 | 6 | 5 | 23 |
| $N$(AVE) | 4 | 1 | 3 | 3 | 6 | 17 |

**Table 3.7:** Sum of completion times for ten trucks and five scenarios, $n$ = 10, $s$ = 5.

| $z_{SU}(\sigma_{SU})$ | $z_a(\sigma_{SU})$ | $z_a(\sigma_a)$ | (LB+UB)/2 | % deviation from Absolute | | |
|---|---|---|---|---|---|---|
| LB | UB | Absolute | Average | LB | UB | Average |
| 2601.8 | 3085 | 2996 | 2847.9 | -12.86 | 2.97 | -4.94 |
| 2264 | 2526 | 2400 | 2395 | -5.67 | 5.25 | -0.21 |
| 2567.8 | 3176 | 2867 | 2871.9 | -10.44 | 10.78 | 0.17 |
| 2530.8 | 3171 | 2842 | 2850.9 | -10.95 | 11.58 | 0.31 |
| 2602.4 | 3212 | 2777 | 3206 | -6.29 | 15.66 | 4.69 |
| 2568.2 | 3322 | 3206 | 2945.1 | -19.89 | 3.62 | -8.14 |

For comparison, an experiment on the sum of completion times in an *RSS* problem is done using C++ and GAMS/CPLEX 11.2.0. It is run with ten jobs and five scenarios (see Table 3.7). Here we can see that the absolute objective value is closer to the lower bound in three trials and to the upper bound in three trials. The absolute objective value is closer to the average than the upper bound in four trials.

Thus the surrogate heuristic works well to find the unloading schedule under uncertainty including weights. It is fast and easy to calculate since it uses WSPT rule.

The same robust formulation and methods can be applied to the whole process from unloading the inbound trucks to sorting, entering and loading in the outbound trucks by considering all of these together as processing time for each inbound truck and considering the weight only for the inbound truck.

In this chapter, we introduced uncertainty and weights to the unloading problem. We experimented with the surrogate procedure empirically and see that the surrogate solution gives an upper bound and a lower bound that is close to the actual optimal solution of the robust problem. In the next chapter, we use a meta-heuristic method to solve the scenario-based robust problem. We use a two-space genetic algorithm for min max problems. We try to use the solution from surrogate procedure as bounds and see the difference it makes in finding the solution.

# Chapter 4: Genetic Algorithm Approach

## 4.1 INTRODUCTION

In this chapter, we use a meta-heuristic procedure to solve the robust unloading problem defined in chapter 3. Some researchers have used heuristic methods (e.g. Lin et al., 2011; Peng et al., 2011) to solve transportation problems. Other solution methods to solve problems in transportation include the use of meta-heuristics such as genetic algorithm (for example, Ng et al., 2009; Karoonsoontawong and Lin, 2011), tabu search (Fan and Machemehl, 2008), and simulated annealing (Ng et al., 2010). The Genetic Algorithm (GA) is one of the most commonly used meta-heuristic techniques for discrete combinatorial optimization problems. As mentioned earlier in chapter 2, McWilliams et al. (2005), Ley and Elfayoumy (2007), Vahdani and Zandieh (2010), and Miao et al. (2009) use GA to solve cross-dock problems.

GA was introduced formally at the University of Michigan by John H. Holland (1975). GA is a programming technique that follows biological evolutionary principles as a strategy to solve problems. GA maintains a population of potential solutions to the given problem that can be quantitatively evaluated using a fitness function. The initial candidate solutions are normally generated at random. The fitness value of each candidate in this population is obtained and evaluated using the fitness function. Though some solutions may work, the aim of GA is to improve them. Most of the solutions in a randomly generated candidate set may not work and these will be deleted. However, a few of them may look promising toward solving the problem.

These promising individual candidates are maintained and allowed to reproduce. A new population is generated using crossover, mutation, and selection operations on the basis of these fitness values. The new population is evaluated in the following iteration of the algorithm. The random variations introduced into the population may have improved some individuals and again these more fit individuals are selected and copied over into the next generation. The expectation is that the average fitness of the population will increase each round, and very good solutions can be found by repeating this process hundreds or thousands of rounds.

### 4.2 TWO-SPACE GENETIC ALGORITHM FOR MIN MAX PROBLEMS

A simple genetic algorithm can be described by the following steps: Let $f$ denote the fitness function.

1. Create an initial generation $G(0)$. Let $t = 0$.

2. For each individual, $i \in G(t)$, evaluate its fitness value $f(i)$.

3. Create generation $G(t+1)$ by reproduction, crossover and mutation.

4. Let $t = t + 1$. Unless $t$ equals the maximum number of generations, return to Step 2.

Herrmann (1999) proposed a two-space genetic algorithm for solving min max problems. The two-space genetic algorithm maintains two populations where the first population represents solutions and the second scenarios. During each generation, an individual in one population is evaluated with respect to all of the individuals in the other population. This will lead the algorithm to converge to a robust solution and its worst-case scenario.

They describe the two-space algorithm in the context of scenario-based robust optimization problem as follows:

Let $X$ be the set of all solutions and let $S$ be the set of all possible scenarios. The performance of a solution $x \in X$ under scenario $s \in S$ is $F(x, s)$. The problem is to find the solution that has the best worst-case performance, which is the same as minimizing (over all solutions) the maximum (over all scenarios) performance:

$$\min_{x \in X} \max_{s \in S} F(x, s)$$

The two-space genetic algorithm maintains two distinct populations: $P_1$ has individuals that represent solutions in $X$, and $P_2$ has individuals that represent solutions in $S$. For a solution $x \in P_1$, the objective function $h(x)$ evaluates that solution's worst-case performance with respect to the second population:

$$h(x) = \max\{F(x, s) : s \in P_2\}$$

The algorithm penalizes large $h(x)$ and rewards small $h(x)$, so that solutions with better worst-case performances will survive. Similarly, for a scenario $s \in P_2$, the objective function $g(s)$ evaluates the best solution in the first population:

$$g(s) = \min\{F(x, s) : x \in P_1\}$$

The algorithm penalizes small $g(s)$ and rewards large $g(s)$, so scenarios with worse optimal solutions will survive.

The two-space genetic algorithm can be summarized as follows:

1. Create initial generations $P_1(0)$ and $P_2(0)$. Let $t = 0$.

2. For each individual $x \in P_1(t)$, evaluate $h(x) = \max\{F(x,s) : s \in P_2(t)\}$ .

3. For each individual $s \in P_2(t)$, evaluate $g(s) = \min\{F(x,s) : x \in P_1(t)\}$ .

4. Create generation $P_1(t+1)$ by reproduction, crossover and mutation.

5. Create generation $P_2(t+1)$ by reproduction, crossover and mutation.

6. Let *t* = *t* + 1. Unless *t* equals the maximum number of generations, return to Step 2.

If the set *S* is not large, then one could evaluate $\max_{s \in S} F(x,s)$ for each solution *x* that the search finds and simple GA can be used. On the other hand, if the set *S* is large, repeatedly searching *S* to determine $\max_{s \in S} F(x,s)$ will lead to excessive computational effort. However, by using the two-space genetic algorithm the computational effort can be considerably reduced. By searching the solution set and the scenario set simultaneously the expectation is that the chosen objective functions encourage the algorithm to converge to a robust solution. In addition, Herrmann (1999) provides the following explanation for convergence:

Suppose that there exists a solution $z \in X$ and a scenario $t \in S$ such that

$$F(z,t) = \min_{x \in X} F(x,t) = \max_{s \in S} F(z,s)$$

Consequently,

$$F(z,t) = \min_{x \in X} \max_{s \in S} F(x,s) = \max_{s \in S} \min_{x \in X} F(x,s)$$

If the initial populations are sufficiently large, then for all $x \in P_1$, $h(x)$ is approximately $\max_{s \in S} F(x,s)$. Likewise, for all $s \in P_2$, $g(s)$ is approximately $\min_{x \in X} F(x,s)$. Thus, the populations are likely to converge towards $z$ and $t$.

Also, consider any generation such that $z \in P_1$ and $t \in P_2$. Then, $h(z) = F(z, t)$ and $g(t) = F(z, t)$. For all other $x \in P_1$, $h(x) \geq F(x, t) \geq F(z, t) = h(z)$. Thus, $z$ is more likely to survive. Similarly, for all other $s \in P_2$, $g(s) \leq F(z, s) \leq F(z, t) = g(t)$. Thus, $t$ is more likely to survive. Consequently, in this case, the genetic algorithm will converge to $z$, the most robust solution, and $t$, that solution's worst-case scenario. Next we provide two examples to show that $\min_{x \in X} \max_{s \in S} F(x,s)$ is not necessarily the same as $\max_{s \in S} \min_{x \in X} F(x,s)$.

### 4.2.1 Examples of Different Cases

**Table 4.1:** Example for $F(z,t) = \min_{x \in X} \max_{s \in S} F(x,s) = \max_{s \in S} \min_{x \in X} F(x,s)$

| Solution | Scenario | | | $h(x_i)$ |
|---|---|---|---|---|
| | $s_1$ | $s_2$ | $s_3$ | |
| $x_1$ | 1 | 2 | 3 | 3 |
| $x_2$ | 8 | 9 | 4 | 9 |
| $x_3$ | 10 | 6 | 9 | 10 |
| $g(s_j)$ | 1 | 2 | 3 | |

From Table 4.1 we can see that $\min_{x \in X} \max_{s \in S} F(x,s) = 3$ from the last column and $\max_{s \in S} \min_{x \in X} F(x,s) = 3$ from the last row.

**Table 4.2:** Example for $F(z,t) = \min_{x \in X} \max_{s \in S} F(x,s) \neq \max_{s \in S} \min_{x \in X} F(x,s)$

| Solution | Scenario $s_1$ | $s_2$ | $s_3$ | $h(x_i)$ |
|---|---|---|---|---|
| $x_1$ | 7 | 9 | 5 | 9 |
| $x_2$ | 4 | 7 | 6 | 7 |
| $x_3$ | 8 | 5 | 3 | 8 |
| $g(s_j)$ | 4 | 5 | 3 | |

From Table 4.2 we can see that $\min_{x \in X} \max_{s \in S} F(x,s) = 7$ from the last column and $\max_{s \in S} \min_{x \in X} F(x,s) = 5$ from the last row. We can see that they are not equal. The two spaces may not converge in this case. Thus we can have any one of these cases in the problems.

For the numerical experiments, Herrmann (1999) considered a parallel machine scheduling problem in which $F(z,t) = \min_{x \in X} \max_{s \in S} F(x,s) = \max_{s \in S} \min_{x \in X} F(x,s)$. They agree that the algorithm may need an adjustment when this condition is not true. Furthermore, they suggest that an appropriate objective function would maintain diversity in the population of scenarios and evaluate how often a scenario is a worst-case scenario. Another possibility is to compare a surviving individual's performance (with respect to the current population of scenarios) to its worst-case performance in the past using memory for more accurate evaluations.

### 4.3 Numerical Experiments and Results

For the numerical experiments we used the two-phase genetic algorithm in two different ways. In the first method, we did not use any bounds to evaluate the fitness of the candidate solutions. In the second method, we used the surrogate heuristic from chapter 3 to find the lower bound and the upper bound to the actual optimal solution of the robust problem. These bounds were then used to evaluate the fitness of the candidate solutions in each generation. In both methods the algorithm copied the best individual to the next generation following the elitist strategy.

The two-phase genetic algorithm is coded in Java. Machine-generated random numbers (0,100] for unloading times and random weights (0,10] are used in the experiments. First we find the solution to the surrogate problem using surrogate heuristic to obtain a lower bound to the solution. Then the same program uses the schedule obtained from the surrogate heuristic in the original problem under each scenario to get the maximum total weighted completion times among all scenarios, which is the upper bound. In both the methods, during each iteration a new random set of generations is created. Simultaneously, a subset of scenarios is chosen at random.

We use the objective function as the fitness function. Each generation is evaluated under all chosen scenarios and the maximum for each generation against the chosen scenarios is found. Then the generations are sorted by this fitness value. For the first method we select few most fit solutions which are the solutions with least maximum values. For the second method we select the generations with the fitness value within the interval between the lower bound and the upper bound. All others are deleted and a new set of generations is created by copying the fit solutions using elitist strategy and by creating random generations to complete the set. The most fit generations mate and their offspring replace the least fit generations. The reason why we decided to use the bounds in the fitness function can be described using the following example: We considered a problem with five trucks and three scenarios with random weights and random unloading times in the interval (1,10]. We used the two-space genetic algorithm and ran it for twenty iterations. The lower bound was 379 and the upper bound was 429.

**Table 4.3:** Results convincing to use bounds

| Iteration | Cost | Outside bounds |
|---|---|---|
| 1 | 552 | Yes |
| 2 | 412 | No |
| 3 | 408 | No |
| 4 | 418 | No |
| 5 | 525 | Yes |
| 6 | 424 | No |
| 7 | 465 | Yes |
| 8 | 444 | Yes |
| 9 | 531 | Yes |
| 10 | 416 | No |
| 11 | 465 | Yes |
| 12 | 398 | No |
| 13 | 545 | Yes |
| 14 | 540 | Yes |
| 15 | 522 | Yes |
| 16 | 485 | Yes |
| 17 | 418 | No |
| 18 | 425 | No |
| 19 | 406 | No |
| 20 | 469 | Yes |

In Table 4.3, more than 50% of the time the results were not between the lower bound and the upper bound. Thus when we take the average of the costs (total weighted completion time) from the twenty iterations it deviates away from the optimal solution. When we used the bounds with the fitness function it eliminated unfit functions and the costs are closer to the optimal solution.

The scenario subset is evaluated for each generation, and the minimum among generations is found. A few scenarios with maximums among these minimums are chosen for the first method, and the scenarios with minimums within the lower bound and upper bound are chosen for the second method. The rest of the scenarios are dropped. New scenarios are randomly selected to complete the set. The process is repeated for several rounds until the solution does not improve for a certain number of rounds.

Twenty-five instances with a different number of trucks and a different number of scenarios were tested. For each instance of the problem, the two-space genetic algorithm was run for twenty iterations. Each iteration ran for several rounds until the solution was the same for a certain number of (same as the number of trucks) rounds. The minimum solution, the average of the solutions, and the mode were analyzed. In the absence of the actual solution it is very hard to predict which one will be a better solution. So we decided to compare the average of the twenty iterations between the two methods together with the percent deviation from the optimal solution. Now we describe the parameters used in the algorithm in Table 4.4.

**Table 4.4:** Parameter description for two-space genetic algorithm

| | |
|---|---|
| Number of generations | 2 x the number of trucks |
| Number of scenarios chosen | 0.75 x the total number of scenarios |
| Number of generations kept | Number of trucks |
| Number of scenarios kept | 0.60 x number of scenarios chosen |
| Mutation Probability | 0.04 |

The same data is then used in GAMS/CPLEX 11.2.0 to find the actual optimal solution to the robust problem in order to compare the solution and the effectiveness of the algorithm. The results are summarized in Table 4.5 for every instance with lower bound (LB), upper bound (UB), solution of the two-space genetic algorithm without using the bounds (TSWOB), solution of the two-space genetic algorithm with bounds (TSWB), optimal solution and the deviation of each of these from the optimal solution.

**Table 4.5:** Comparison of two-space genetic algorithm with optimal solution

| Number of Trucks | Number of scenarios | Bounds | | TSGA | | Optimal solution | Deviation | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Lower (LB) | Upper (UB) | W/o Bounds (TSWOB) | W/ Bounds (TSWB) | | LB (DLB) | UB (DUB) | TSWOB (DTSWOB) | TSWB (DTSWB) |
| 5 | 3 | 1324 | 1619 | 1453 | 1560 | 1619 | -18.22 | 0.00 | -10.25 | -3.64 |
| 5 | 5 | 2206 | 3592 | 3626 | 3437 | 3592 | -38.59 | 0.00 | 0.95 | -4.32 |
| 10 | 5 | 7989 | 11206 | 12256 | 10494 | 10178 | -21.51 | 10.10 | 20.42 | 3.10 |
| 10 | 5 | 10176 | 13146 | 13549 | 12809 | 12344 | -17.56 | 6.50 | 9.76 | 3.77 |
| 10 | 5 | 9395 | 13523 | 12771 | 12058 | 11855 | -20.75 | 14.07 | 7.73 | 1.71 |
| 10 | 5 | 4750 | 7237 | 7860 | 7101 | 6854 | -30.70 | 5.59 | 14.68 | 3.60 |
| 10 | 5 | 14690 | 17235 | 18598 | 16918 | 16469 | -10.80 | 4.65 | 12.93 | 2.73 |
| 10 | 5 | 11837 | 16943 | 16357 | 15487 | 14993 | -21.05 | 13.01 | 9.10 | 3.29 |
| 10 | 8 | 9832 | 11556 | 12596 | 11274 | 10838 | -9.28 | 6.62 | 16.22 | 4.02 |
| 10 | 8 | 10369 | 14239 | 15092 | 13678 | 13227 | -21.61 | 7.65 | 14.10 | 3.41 |
| 10 | 8 | 11562 | 15875 | 15414 | 14468 | 13499 | -14.35 | 17.60 | 14.19 | 7.18 |
| 10 | 8 | 9130 | 10590 | 12237 | 10541 | 10493 | -12.99 | 0.92 | 16.62 | 0.46 |
| 10 | 8 | 9982 | 12725 | 13829 | 12420 | 11706 | -14.73 | 8.70 | 18.14 | 6.10 |
| 10 | 10 | 13785 | 17441 | 18513 | 17054 | 16441 | -16.15 | 6.08 | 12.60 | 3.73 |
| 10 | 10 | 16840 | 20129 | 19971 | 18991 | 18707 | -9.98 | 7.60 | 6.76 | 1.52 |
| 10 | 10 | 13117 | 18688 | 19469 | 17823 | 17005 | -22.86 | 9.90 | 14.49 | 4.81 |
| 10 | 10 | 9971 | 13864 | 13540 | 13113 | 12368 | -19.38 | 12.10 | 9.48 | 6.02 |
| 10 | 10 | 6708 | 9042 | 11211 | 9010 | 8834 | -24.07 | 2.35 | 26.91 | 1.99 |
| 10 | 10 | 12088 | 16718 | 16221 | 15360 | 14977 | -19.29 | 11.62 | 8.31 | 2.56 |
| 20 | 10 | 54493 | 67607 | 73318 | 66865 | 64273 | -15.22 | 5.19 | 14.07 | 4.03 |
| 20 | 10 | 34810 | 41111 | 53301 | 41106 | 40660 | -14.39 | 1.11 | 31.09 | 1.10 |
| 20 | 10 | 41215 | 53188 | 59475 | 52052 | 49015 | -15.91 | 8.51 | 21.34 | 6.20 |
| 20 | 10 | 49377 | 62914 | 71203 | 61190 | 58529 | -15.64 | 7.49 | 21.65 | 4.55 |
| 20 | 10 | 50601 | 66128 | 68210 | 65120 | 60625 | -16.53 | 9.08 | 12.51 | 7.41 |
| 20 | 10 | 41935 | 51056 | 63904 | 50430 | 49112 | -14.61 | 3.96 | 30.12 | 2.68 |

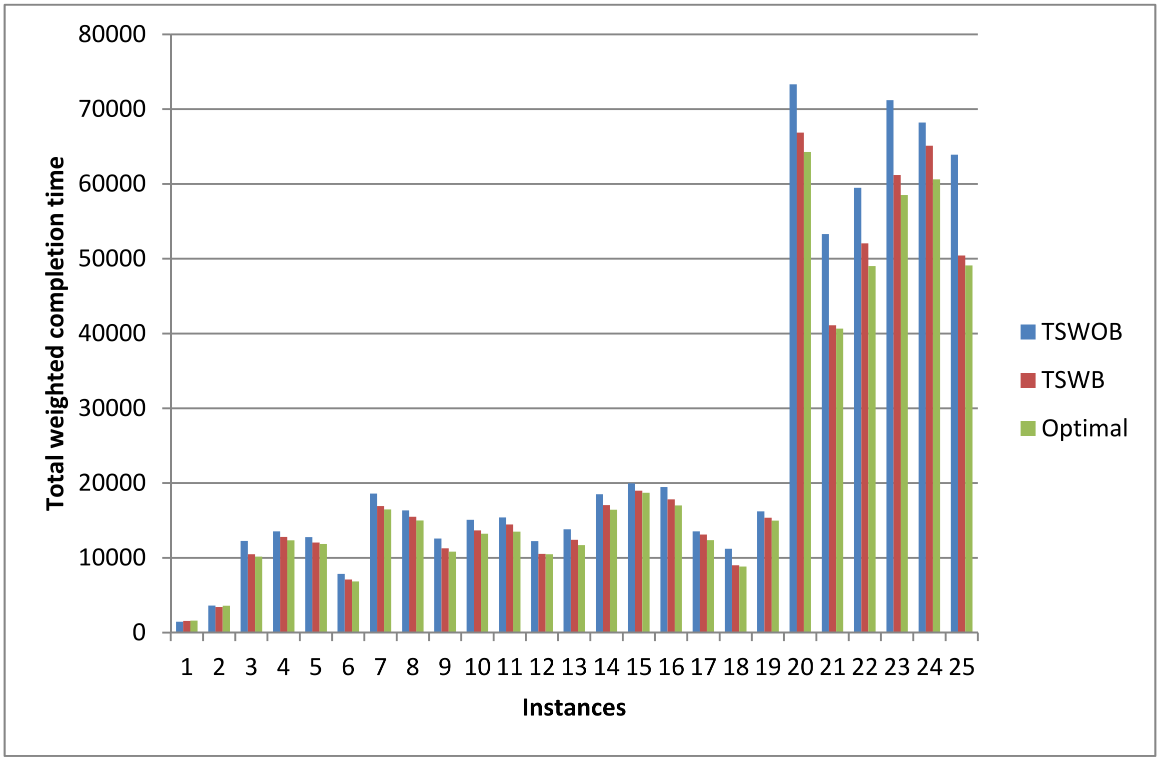


**Figure: 4.1:** Comparison of two-space genetic algorithm with optimal solution

From the above results it is very clear that the two-space genetic algorithm combined with upper and lower bounds performs much better than the one that does not use any bounds. Except for one case, in all instances TSWB lead to solutions closer to the optimal solution. Thus more than 95% of the times TSWB outperforms TSWOB.

### 4.3.1 Two-space genetic algorithm vs GA

Next we were curious about how well these results performed when compared to regular genetic algorithms. For that we ran 20 instances with different numbers of trucks and scenarios. We summarize the results in Table 4.6 and Figure 4.2. In these experiments, GA was run with and without bounds. The fitness function value was found for each generation for each scenario. In other words, 100% of the scenarios were used for each run and the fitness value was calculated for all the scenarios. The maximum solution under each scenario is found, and a few minimum solutions among those were kept. These best fit generations were copied to the next generation.

Each instance was run for twenty iterations. The solution here is the minimum among all the twenty iterations since we used all the scenarios. Also, the two-space genetic algorithm was run with and without bounds for the same instances. In order to see how well they perform we increased the threshold by selecting only 50% of the scenarios in the second population and 60% of the best fit scenarios were kept. For the two-space genetic algorithm we took the average of all twenty iterations as solution. In spite of that the two-space genetic algorithm with bounds performed very well and the solutions were very close to the optimal solution. The results are summarized in Table 4.6 and Figure 4.2.

The GA solutions with bound (GAWB) deviate from the optimal solution by 0.12% to 3.85%. The two-space genetic algorithm (TSWB) deviates from the optimal solution by 0.03% to 7.9%. It is clear from these experiments that TSWB outperforms GAWOB except for one instance. That is 95% of the times TSWB performs better than GAWOB. Also, we can see that almost half the time TSWB performs better than GAWB and the solution is very close to the optimal solution.

**Table 4.6:** Two-space genetic algorithm vs. GA

| Trucks | Scenarios | UB | LB | GAWB | GAWOB | TSWB | TSWOB | Optimal solution | DUB | DLB | DGAWB | DGAWO B | DTSWB | DTSWOB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 5 | 18729 | 12482 | 17434 | 17985 | 15878 | 15750 | 17248 | 8.59 | -27.63 | 1.08 | 4.27 | -7.94 | 8.69 |
| 10 | 5 | 7407 | 6450 | 7254 | 7818 | 7098 | 8088 | 7193 | 2.98 | -10.33 | 0.85 | 8.69 | -1.32 | -12.44 |
| 10 | 5 | 10799 | 9196 | 10636 | 11305 | 10622 | 12685 | 10426 | 3.58 | -11.80 | 2.01 | 8.43 | 1.88 | -21.67 |
| 10 | 5 | 9352 | 7765 | 9213 | 9475 | 9231 | 9616 | 9129 | 2.44 | -14.94 | 0.92 | 3.79 | 1.12 | -5.33 |
| 10 | 5 | 21402 | 15810 | 18953 | 19304 | 19353 | 18923 | 18717 | 14.35 | -15.53 | 1.26 | 3.14 | 3.40 | -1.10 |
| 10 | 10 | 17522 | 12120 | 16329 | 17492 | 16025 | 17295 | 16056 | 9.13 | -24.51 | 1.70 | 8.94 | -0.19 | -7.72 |
| 10 | 10 | 11693 | 9747 | 11263 | 12202 | 11484 | 12616 | 11209 | 4.32 | -13.04 | 0.48 | 8.86 | 2.45 | -12.55 |
| 10 | 10 | 14840 | 11599 | 14240 | 14970 | 14219 | 15573 | 14223 | 4.34 | -18.45 | 0.12 | 5.25 | -0.03 | -9.49 |
| 10 | 10 | 12894 | 8926 | 11563 | 11952 | 11464 | 12525 | 11177 | 15.36 | -20.14 | 3.45 | 6.93 | 2.57 | -12.06 |
| 10 | 10 | 14920 | 10667 | 14508 | 15040 | 14015 | 15403 | 14429 | 3.40 | -26.07 | 0.55 | 4.23 | -2.87 | -6.75 |
| 15 | 10 | 23057 | 17590 | 22579 | 26267 | 21980 | 26615 | 21778 | 5.87 | -19.23 | 3.68 | 20.61 | 0.93 | -22.21 |
| 15 | 10 | 31233 | 23901 | 30464 | 32946 | 30224 | 33980 | 29722 | 5.08 | -19.58 | 2.50 | 10.85 | 1.69 | -14.33 |
| 15 | 10 | 29187 | 21080 | 27634 | 29501 | 28010 | 30964 | 26711 | 9.27 | -21.08 | 3.46 | 10.45 | 4.86 | -15.92 |
| 15 | 10 | 33634 | 25545 | 32203 | 34699 | 32739 | 36127 | 31499 | 6.78 | -18.90 | 2.23 | 10.16 | 3.94 | -14.69 |
| 15 | 10 | 36655 | 27338 | 34993 | 36774 | 34648 | 37570 | 33856 | 8.27 | -19.25 | 3.36 | 8.62 | 2.34 | -10.97 |
| 20 | 10 | 65131 | 49874 | 62751 | 67437 | 61073 | 66972 | 60617 | 7.45 | -17.72 | 3.52 | 11.25 | 0.75 | -10.48 |
| 20 | 10 | 53993 | 42157 | 51716 | 58539 | 52377 | 58682 | 49798 | 8.42 | -15.34 | 3.85 | 17.55 | 5.18 | -17.84 |
| 20 | 10 | 47845 | 35261 | 46978 | 53979 | 46158 | 44712 | 46386 | 3.15 | -23.98 | 1.28 | 16.37 | -0.49 | 3.61 |
| 20 | 10 | 49868 | 40312 | 48436 | 60865 | 49199 | 60413 | 47737 | 4.46 | -15.55 | 1.46 | 27.50 | 3.06 | -26.55 |
| 20 | 10 | 57651 | 42823 | 53520 | 64274 | 54457 | 63463 | 51577 | 11.78 | -16.97 | 3.77 | 24.62 | 5.58 | -23.05 |

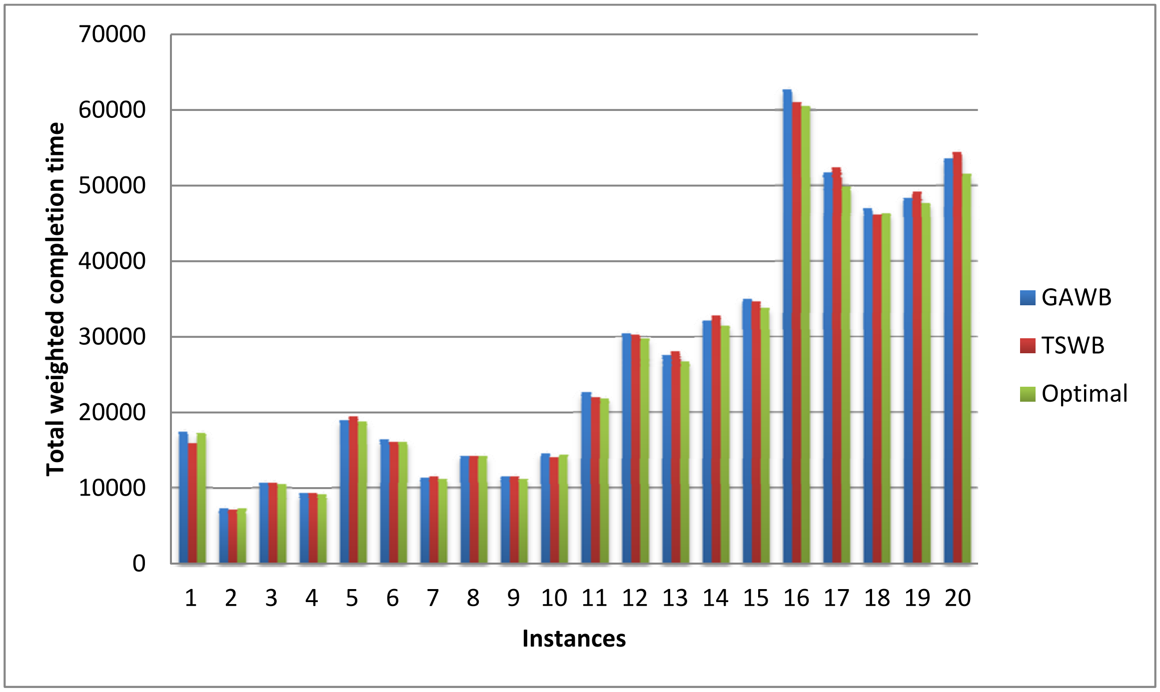


**Figure 4.2:** Two-space algorithm vs. genetic algorithm

From all the above experiments it is very clear that both GA and the two-space genetic algorithm coupled with lower bound and upper bound from surrogate heuristic explained in chapter 3 performs very well and the solution is close to the optimal solution. In the next chapter we will see an alternative robust approach.

# Chapter 5: An Alternative Protection Based Robust Approach

## 5.1 INTRODUCTION

In the previous two chapters, we looked at the scenario-based robust optimization for the unloading problem. In that approach, the solution minimizes the worst case performance under all its scenarios. The size of the resulting optimization model increases drastically as a function of the number of scenarios which in turn poses substantial computational challenges. We consider another robust approach in this chapter in which the solution may be suboptimal but the degree of conservatism can be adjusted to data uncertainty. Soyster (1973) came up with an approach and a robust formulation that is linear under the model of data uncertainty U. Ben-Tal and Nemirovski (2000) provided another model that is less conservative under data uncertainty U than Soyster's model. Bertsimas and Sim (2003, 2004) proposed their new approach with a nonlinear programming formulation by introducing a parameter, $\Gamma_i$, for each $i$ to control the number of coefficient changes. Then they show that it can be solved as a linear optimization problem. For clarity and readability we provide the different models discussed in the literature that leads to the robust model we used which is based on the model of Bertsimas and Sim.

## 5.2 LINEAR OPTIMIZATION AND DATA UNCERTAINTY

Consider the following nominal linear optimization problem:

$$\begin{aligned} \text{maximize} \quad & c'x \\ \text{subject to} \quad & Ax \le b \end{aligned} \qquad (5.1)$$

$$l \leq x \leq u.$$

In the above formulation, assume that data uncertainty only affects the elements in matrix *A* and that without loss of generality, the coefficients *c*' of the objective function are not subject to uncertainty, since they can use the objective, maximize *z*, add the constraint $z - c'x \leq 0,$ and thus include this constraint into $Ax \leq b$.

**Model of Data Uncertainty U:** Consider a particular row *i* of the matrix *A*. Let $J_i$ be the set of coefficients in row *i* that is subject to uncertainty. Each entry $a_{ij}$, $j \in J_i$, is modeled as a symmetric and bounded random variable $\tilde{a}_{ij}$, $j \in J_i$ that takes values in $[a_{ij} - \hat{a}_{ij}, a_{ij} + \hat{a}_{ij}]$. Associated with the uncertain data $\tilde{a}_{ij}$, the random variable $\eta_{ij} = (\tilde{a}_{ij} - a_{ij}) / \hat{a}_{ij}$, which obeys an unknown but symmetric distribution, takes values in $[-1,1]$.

### 5.2.1 The Robust Formulation of Soyster

Soyster (1973) considers column-wise uncertainty. Under the model of data uncertainty U, the robust formulation of (5.1) is as follows:

$$\begin{aligned}
\text{maximize} \quad & c'x \\
\text{subject to} \quad & \sum_j a_{ij} x_j + \sum_{j \in J_i} \hat{a}_{ij} y_j \leq b_i && \forall i \\
& -y_j \leq x_j \leq y_j && \forall j \qquad (5.2) \\
& l \leq x \leq u \\
& y \geq 0.
\end{aligned}$$

Let $x^*$ be the optimal solution of Formulation (5.2). At optimality, clearly, $y_j = |x_j^*|$, and thus $\sum_j a_{ij} x_j^* + \sum_{j \in J_i} \hat{a}_{ij} |x_j^*| \le b_i \quad \forall i.$

Bertsimas and Sims (2004) show that for every possible realization $\tilde{a}_{ij}$ of the uncertain data, the solution remains feasible; that is, the solution is "robust." From data uncertainty U, since $\eta_{ij} = (\tilde{a}_{ij} - a_{ij}) / \hat{a}_{ij}$, we have $\tilde{a}_{ij} = a_{ij} + \eta_{ij} \hat{a}_{ij}$,

$$\sum_j \tilde{a}_{ij} x_j^* = \sum_j a_{ij} x_j^* + \sum_{j \in J_i} \eta_{ij} \hat{a}_{ij} x_j^*$$

$$\le \sum_j a_{ij} x_j^* + \sum_{j \in J_i} \hat{a}_{ij} |x_j^*| \le b_i, \ \forall i, \text{ since } \eta_{ij} \in [-1,1].$$

For every $i$th constraint, the term $\sum_{j \in J_i} \hat{a}_{ij} |x_j^*|$ gives the necessary protection of the constraint by maintaining a gap between $\sum_j a_{ij} x_j^*$ and $b_i$. Also, model (5.2) is linear.

### 5.2.2 The Robust Formulation of Ben-Tal and Nemirovski

The robust solution of Soyster's method has an objective function value much worse than the objective function value of the solution of the nominal linear optimization problem. The method gives the highest protection but it is the most conservative. Ben-Tal and Nemirovski (2000) propose the following robust problem to address this issue.

$$\begin{aligned}
\text{maximize} \quad & c'x \\
\text{subject to} \quad & \sum_j a_{ij} x_j + \sum_{j \in J_i} \hat{a}_{ij} y_{ij} + \Omega_i \sqrt{\sum_{j \in J_i} \hat{a}_{ij}^2 z_{ij}^2} \le b_i && \forall i \\
& -y_{ij} \le x_j - z_{ij} \le y_{ij} && \forall i, j \in J_i \qquad (5.3)
\end{aligned}$$

$$l \le x \le u$$

$$y \ge 0.$$

Under the model of data uncertainty U, the probability that the *i*th constraint is violated is at most $\exp(-\Omega_i^2/2)$. Robust Model (5.3) is less conservative than Model (5.2) as every feasible solution of the latter problem is a feasible solution to the former problem. Since model (5.3) is nonlinear, it is not very attractive.

**5.3 THE ROBUST APPROACH BY BERTSIMAS AND SIM**

The authors Bertsimas and Sim (2004) propose a robust formulation that is linear. In this section we present their approach, bounds, theorems and propositions developed by them relevant to the problem. Consider the *i*th constraint of the nominal problem, $a_i' x \le b_i$. Let $J_i$ be the set of coefficients $a_{ij}$, $j \in J_i$ that are subject to parameter uncertainty, i.e. $\tilde{a}_{ij}$, ; $j \in J_i$ takes values according to a symmetric distribution with a mean equal to the nominal value $a_{ij}$ in the interval $[a_{ij} - \hat{a}_{ij}, a_{ij} + \hat{a}_{ij}]$. For every *i*, they introduce a parameter $\Gamma_i$, not necessarily an integer, that takes the values in the interval $[0, |J_i|]$. The role of the parameter $\Gamma_i$ is to adjust the robustness of the proposed method against the level of conservatism of the solution. The authors argue that intuitively it is unlikely that all of the $a_{ij}$, $j \in J_i$ will change. The goal is to be protected against all cases that up to $\lfloor \Gamma_i \rfloor$ of these coefficients are allowed to change and one coefficient $a_{ij}$ changes by $(\Gamma_i - \lfloor \Gamma_i \rfloor)\hat{a}_{ij}$. In other words, if nature behaves in a restrictive manner, then only a subset of the coefficients will change in order to adversely affect the solution. In that case, the robust solution will be feasible deterministically, and moreover, even if more than $\lfloor \Gamma_i \rfloor$ of the coefficients change, then the robust solution will be feasible with very high probability.

They consider the following (still nonlinear) formulation:

$$\begin{aligned}
\text{maximize} \quad & c'x \\
\text{subject to} \quad & \sum_j a_{ij} x_j + \max_{\{S_i \cup \{t_i\} | S_i \subseteq J_i, |S_i| = \lfloor \Gamma_i \rfloor, t_i \in J_i \setminus S_i\}} \left\{ \sum_{j \in S_i} \hat{a}_{ij} y_i + \left(\Gamma_i - \lfloor \Gamma_i \rfloor\right) \hat{a}_{it_i} y_{t_i} \right\} \le b_i \qquad \forall i \\
& -y_j \le x_j \le y_j \qquad \forall j \\
& l \le x \le u \qquad (5.4) \\
& y \ge 0.
\end{aligned}$$

If $\Gamma_i$ is chosen as an integer the $i$th constraint is protected by $\beta_i(x, \Gamma_i) = \max_{\{S_i | S_i \subseteq J_i, |S_i| = \Gamma_i\}} \left\{ \sum_{j \in S_i} \hat{a}_{ij} |x_j| \right\}$.

When $\Gamma_i = 0$, $\beta_i(x, \Gamma_i) = 0$, and the constraints are equivalent to that of the nominal problem. If $\Gamma_i = |J_i|$, Soyster's method can be used. Therefore, by varying $\Gamma_i \in [0, |J_i|]$, we have the flexibility of adjusting the robustness of the method against the level of conservatism of the solution. The authors prove the following proposition in order to reformulate Model (5.4) as a linear optimization model.

**Proposition 5.3.1**: Given a vector $x^*$ the protection function of the $i$th constraint,

$$\beta_i(x^*,\Gamma_i) = \max_{\{S_i \cup \{t_i\} | S_i \subseteq J_i, |S_i| = \lfloor \Gamma_i \rfloor, t_i \in J_i \setminus S_i\}} \left\{ \sum_{j \in S_i} \hat{a}_{ij} |x_j^*| + (\Gamma_i - \lfloor \Gamma_i \rfloor) \hat{a}_{it_i} |x_j^*| \right\}, \tag{5.5}$$

equals the objective function of the following linear optimization problem:

$$\begin{aligned} \beta_i(x^*,\Gamma_i) = \text{ maximize} \quad & \sum_{j \in J_i} \hat{a}_{ij} |x_j^*| z_{ij} \\ \text{subject to} \quad & \sum_{j \in J_i} z_{ij} \le \Gamma_i \\ & 0 \le z_{ij} \le 1 \qquad \forall j \in J_i. \end{aligned} \tag{5.6}$$

**Proof:** Clearly the optimal solution value of Problem (6) consists of $\lfloor \Gamma_i \rfloor$ variables at 1 and one variable at $\Gamma_i - \lfloor \Gamma_i \rfloor$. This is equivalent to the selection of subset $\{S_i \cup \{t_i\} | S_i \subseteq J_i, |S_i| = \lfloor \Gamma_i \rfloor, t_i \in J_i \setminus S_i\}$ with corresponding cost function $\left\{ \sum_{j \in S_i} \hat{a}_{ij} |x_j^*| + (\Gamma_i - \lfloor \Gamma_i \rfloor) \hat{a}_{it_i} |x_j^*| \right\}$.

The authors reformulate Model (5.4) as a linear optimization model in the following theorem.

**Theorem 5.3.1:** Model (5.4) has an equivalent linear formulation as follows:

$$\begin{aligned} \text{maximize} \quad & c'x \\ \text{subject to} \quad & \sum_j a_{ij} x_j + z_i \Gamma_i + \sum_{j \in J_i} p_{ij} \le b_i \quad && \forall i \\ & z_i + p_{ij} \ge \hat{a}_{ij} y_j && \forall i, j \in J_i \\ & -y_j \le x_j \le y_j && \forall j \end{aligned} \tag{5.7}$$

$$l_j \le x_j \le u_j \qquad \forall j$$

$$p_{ij} \ge 0 \qquad \forall i, j \in J_i$$

$$y_j \ge 0 \qquad \forall j$$

$$z_i \ge 0 \qquad \forall i.$$

**Proof:** First consider the dual of Problem (5.6):

$$\begin{aligned} \text{minimize} \quad & \sum_{j \in J_i} p_{ij} + \Gamma_i z_i \\ \text{subject to} \quad & z_i + p_{ij} \ge \hat{a}_{ij} \left| x_j^* \right| \qquad \forall i, j \in J_i \\ & p_{ij} \ge 0 \qquad \forall j \in J_i \\ & z_i \ge 0 \qquad \forall i. \end{aligned} \tag{5.8}$$

By strong duality, since Problem (5.6) is feasible and bounded for all $\Gamma_i \in [0, |J_i|]$, then the dual problem (5.8) is also feasible and bounded and their objective values coincide. Using Proposition 5.3.1, $\beta_i(x^*, \Gamma_i)$ is equal to the objective function value of Problem (5.8). Substituting this in Problem (5.4), they obtain that Problem (5.4) is equivalent to the linear optimization problem.

### 5.3.1 Probability Bounds of Constraint Violation

The parameter $\Gamma_i$ controls the tradeoff between the probability of violation and the effect to the objective function of the nominal problem, which is what the authors call *the price of robustness.* The authors discuss the probability that the $i$th constraint is violated first, and then they find bounds for that probability.

**Proposition 5.3.2:** Let $x^*$ be an optimal solution of Problem (7). Let $S_i{}^*$ and $t_i{}^*$ be the set and the index respectively that achieve the maximum for $\beta_i(x^*, \Gamma_i)$ in Equation (5.5). Suppose that the data in matrix $A$ are subjected to the model of data uncertainty U.

(a) The probability that the $i$th constraint is violated satisfies:

$$\Pr\left(\sum_j \tilde{a}_{ij} x_j^* > b_i\right) \leq \Pr\left(\sum_{j \in J_i} \gamma_{ij} \eta_{ij} \geq \Gamma_i\right), \text{ where} \tag{5.9}$$

$$\gamma_{ij} = \begin{cases} 1, & \text{if } j \in S_i^* \\ \dfrac{\hat{a}_{ij}\left|x_j^*\right|}{\hat{a}_{ir^*}\left|x_{r^*}^*\right|}, & \text{if } j \in J_i \setminus S_i^* \end{cases} \quad \text{and } r^* = \underset{r \in S_i^* \cup \{t_i^*\}}{\arg\min} \hat{a}_{ir}\left|x_r^*\right|.$$

(b) The quantities $\gamma_{ij}$ satisfy $\gamma_{ij} \leq 1$ for all $j \in J_i \setminus S_i^*$.

After the proof of this theorem, they are naturally led to bind the probability $\Pr\left(\sum_{j \in J_i} \gamma_{ij} \eta_{ij} \geq \Gamma_i\right)$. Their next theorem provides a bound that is independent of the solution $x^*$.

**Theorem 5.3.2:** If $\eta_{ij}, j \in J_i$ are independent and symmetrically distributed random variables in $[-1,1]$, then

$$\Pr\left(\sum_{j \in J_i} \gamma_{ij} \eta_{ij} \geq \Gamma_i\right) \leq \exp\left(-\frac{\Gamma_i^2}{2\left|J_i\right|}\right). \tag{5.10}$$

In their next theorem they give different bounds for the probability $\Pr\left(\sum_{j \in J_i} \gamma_{ij} \eta_{ij} \geq \Gamma_i\right)$.

**Theorem 5.3.3:**

(a) If $\eta_{ij}, j \in J_i$ are independent and symmetrically distributed random variables in $[-1,1]$, then

$$\Pr\left(\sum_{j \in J_i} \gamma_{ij}\eta_{ij} \geq \Gamma_i\right) \leq B(n, \Gamma_i), \qquad (5.11)$$

where

$$B(n,\Gamma_i) = \frac{1}{2^n}\left\{(1-\mu)\sum_{l=\lfloor \nu \rfloor}^{n}\binom{n}{l} + \mu \sum_{l=\lfloor \nu \rfloor+1}^{n}\binom{n}{l}\right\}$$

$$= \frac{1}{2^n}\left\{(1-\mu)\binom{n}{\lfloor \nu \rfloor} + \mu \sum_{l=\lfloor \nu \rfloor+1}^{n}\binom{n}{l}\right\} \qquad (5.12)$$

where $n = |J_i|$, $\nu = (\Gamma_i + n)/2$, and $\mu = \nu - \lfloor \nu \rfloor$.

(b) The bound (5.11) is tight for $\eta_{ij}$ having a discrete probability distribution $\Pr(\eta_{ij} = 1) = 1/2$ and $\Pr(\eta_{ij} = -1) = 1/2$, $\gamma_{ij} = 1$, an integral value of $\Gamma_i \geq 1$, and $\Gamma_i + n$ being even.

(c) The bound (5.11) satisfies

$$B(n,\Gamma_i) \leq (1-\mu)C(n, \lfloor \nu \rfloor) + \sum_{l=\lfloor \nu \rfloor+1}^{n} C(n,l) \qquad (5.13)$$

where

$$C(n,l)=\begin{cases}\dfrac{1}{2^n}, & \text{if } l=0, \text{ or } l=n.\\ \dfrac{1}{\sqrt{2\pi}}\sqrt{\dfrac{n}{(n-l)l}}\cdot\exp\left(n\ln\left(\dfrac{n}{2(n-l)}\right)+l\ln\left(\dfrac{n-l}{l}\right)\right), & o.w.\end{cases} \quad (5.14)$$

(d) For $\Gamma_i = \theta\sqrt{n}$,

$\lim_{n\to\infty} B(n,\Gamma_i) = 1-\Phi(\theta)$,

where

$$\Phi(\theta)=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\theta}\exp\left(-\frac{y^2}{2}\right)dy$$

is the cumulative distribution function of a standard normal. By using De Moivre-Laplace approximation of the binomial distribution they conclude that

$$B(n,\Gamma_i)\approx 1-\Phi\left(\frac{\Gamma_i-1}{\sqrt{n}}\right), \quad (5.15)$$

even if $\Gamma_i$ does not scale as $\theta\sqrt{n}$.

While bound (5.11) is the best possible bound, it poses computational difficulties in evaluating the combinations of functions for large *n*. In comparing the four bounds, (5.10), (5.11), (5.13), and (5.15), according to their calculations they conclude that Bounds (5.11) and (5.13) dominate Bound (5.10).

## 5.4 ROBUST DISCRETE OPTIMIZATION

When both the cost coefficients and the data in the constraints of an integer programming problem are subject to uncertainty, the Bertsimas and Sims (2003) propose a robust integer programming problem which allows controlling the degree of conservatism of the solution in terms of probabilistic bounds on constraint violation.

**Uncertainty for matrix *A***: Let $N = \{1,2,...,n\}$. Each entry $a_{ij}$, $j \in N$ is modeled as independent, symmetric, and bounded random variable with an unknown distribution. $\tilde{a}_{ij}$, $j \in N$ that takes values in the interval $[a_{ij} - \hat{a}_{ij}, a_{ij} + \hat{a}_{ij}]$.

**Uncertainty for cost vector *c***: Each entry $c_j$, $j \in N$ takes values in the interval $[c_j, c_j + d_j]$, where $d_j$ represents the deviation from the nominal cost coefficient, $c_j$.

### 5.4.1 Robust Formulation for Discrete Optimization Problems

Let *c, l, u* be *n*-vectors, let *A* be an *m*x*n* matrix, and *b* be an *m*-vector. Consider the following nominal mixed integer programming (MIP) on a set of *n* variables, the first *k* of which are integral:

$$
\begin{aligned}
\text{minimize} \quad & c'x \\
\text{subject to} \quad & Ax \le b \\
& l \le x \le u \\
& x_i \in Z, \qquad i = 1,\dots,k.
\end{aligned}
\tag{5.18}
$$

Assume without loss of generality that data uncertainty affects only the elements of the matrix *A* and *c*, but not the vector *b,* since in this case we can introduce a new variable $x_{n+1}$ and write $Ax - bx_{n+1} \le 0$, $l \le x \le u$, $1 \le x_{n+1} \le 1$, thus augmenting *A* to include *b*. In addition to what was in the linear programming model, here each entry $c_j$

takes values in $[c_j, c_j + d_j]$ where $d_j$ represents the deviation from the nominal cost coefficient $c_j$, and we allow the possibility that $\hat{a}_{ij} = 0$, or $d_j = 0$.

**5.4.2 Robust MIP formulation**

For every $i$, they introduce a parameter $\Gamma_i$, $i$ = 0,1,…,$m$ that takes values in the interval $[0, |J_i|]$, where $J_i = \{ j \mid \hat{a}_{ij} > 0 \}$. $\Gamma_0$ is assumed to be an integer while $\Gamma_i$, $i$ = 1,…,$m$ are not necessarily integers. The parameter $\Gamma_0$ controls the level of robustness in the objective. The objective is to find an optimal solution that optimizes against all scenarios under which a number $\Gamma_0$ of the cost coefficients can vary in such a way as to maximally influence the objective. Let $J_0 = \{ j \mid d_j > 0 \}$. If $\Gamma_0 = 0$, completely ignore the influence of the cost deviations, while if $\Gamma_0 = |J_0|$, consider all possible cost deviations, which is indeed most conservative. In general, a high value of $\Gamma_0$ increases the level of robustness at the expense of higher nominal cost. The proposed robust counterpart of problem (5.18) is as follows:

$$\text{minimize} \qquad c'x \; + \max_{\{S_0 \mid S_0 \subseteq J_0, |S_0| \le \Gamma_0\}} \left\{ \sum_{j \in S_0} d_j |x_j| \right\}$$

$$\text{subject to} \qquad \sum_j a_{ij} x_j + \max_{\{S_i \cup \{t_i\} \mid S_i \subseteq J_i, |S_i| \le \lfloor \Gamma_i \rfloor, t_i \in J_i \setminus S_i\}} \left\{ \sum_{j \in S_i} \hat{a}_{ij} |x_j| + (\Gamma_i - \lfloor \Gamma_i \rfloor) \hat{a}_{it_i} x_{t_i} \right\} \le b_i \quad \forall i$$

$$l \le x \le u \tag{5.19}$$

$$x_i \in Z, \qquad i = 1,\ldots,k.$$

They next show that the robust approach in linear optimization extends to discrete optimization and discuss the bounds as in the case of linear robust optimization.

**Theorem 5.4.1:** Problem (5.19) has an equivalent MIP formulation as follows:

$$
\begin{array}{lll}
\text{minimize} & c'x + z_0\Gamma_0 + \sum_{j \in J_0} p_{0j} & \\
\text{subject to} & \sum_j a_{ij}x_j + z_i\Gamma_i + \sum_{j \in J_i} p_{ij} \le b_i & \forall i \\
& z_0 + p_{0j} \ge d_j y_j & \forall j \in J_0 \\
& z_i + p_{ij} \ge \hat{a}_{ij} y_j & \forall i \ne 0, j \in J_i \\
& -y_j \le x_j \le y_j & \forall j \\
& l_j \le x_j \le u_j & \forall j \\
& p_{ij} \ge 0 & \forall i, j \in J_i \\
& y_j \ge 0 & \forall j \\
& z_i \ge 0 & \forall i\,. \\
& x_i \in Z & i = 1,\ldots,k.
\end{array}
\qquad (5.20)
$$

### 5.4.3 Robust Combinatorial Optimization

Combinatorial optimization is an important class of discrete optimization whose decision variables are binary; that is $x \in X \subseteq \{0,1\}^n$. The nominal combinatorial optimization problem considered is

minimize $c'x$

subject to $x \in X$. (5.21)

The class of problems considered are the ones where each entry $c_j$, $j \in N = \{1,2,...,n\}$ takes values in the interval $[c_j, c_j + d_j]$, where $d_j \geq 0$, $j \in N$ while the set $X$ is fixed. The objective here is to find a solution $x \in X$ that minimizes the maximum cost $c'x$ such that, at most, $\Gamma$ of the coefficients $\tilde{c}_j$ are allowed to change:

$$Z^* = \min c'x + \max_{\{S \mid S \subseteq N, |S| \leq \Gamma\}} \left\{ \sum_{j \in S} d_j x_j \right\} \tag{5.22}$$

subject to $x \in X$.

Without loss of generality assume that the indices are ordered such that $d_1 \geq d_2 \geq ... \geq d_n$. Also, define $d_{n+1} = 0$ for notational convenience. Examples of such problems include the shortest path, the minimum spanning tree, the minimum assignment, the traveling salesman, the vehicle routing, and matroid intersection problems. For example, data uncertainty in the context of the vehicle routing problem captures the variability of travel times in some of the links of the network.

In the context of scenario-based uncertainty as in chapter 3, finding an optimally robust solution for a problem with $s$ scenarios involves solving the problem:

minimize $\max_{s \in S}\{c'_1 x, c'_2 x, ..., c'_s x\}$

subject to $x \in X$.

Finding the robust solution became NPhard even for the problems that are polynomially solvable in the scenario-based approach. Here, the robust counterpart of a polynomially solvable combinatorial optimization problem is also polynomially solvable.

### 5.4.4 Algorithm for Robust Combinatorial Optimization

The authors show that we can solve Problem (5.20) by solving at most $n$+1 nominal problems,

min $f'_i x$ subject to $x \in X$ for $i$ = 1,…,$n$+1.

**Theorem 5.4.2:** Problem (5.20) can be solved by solving the $n$+1 nominal problems:

$$Z^* = \min_{l=1,..,n+1} G^l, \tag{5.23}$$

where for $l$ = 1,…,$n$+1:

$$G^l = \Gamma d_l + \min\left( c'x + \sum (d_j - d_l) x_j \right) \tag{5.24}$$

subject to $x \in X$.

This theorem leads to the following algorithm.

**Algorithm A**

1. For $l$ = 1,…,$n$+1 solve the $n$+1 nominal problems Equations. (5.24):

   $$G^l = \Gamma d_l + \min_{x \in X}\left( c'x + \sum (d_j - d_l) x_j \right),$$

   and let $x^l$ be an optimal solution of the corresponding problem.

2. Let $l^* = \arg\min_{l=1,...,n+1} G^l$.

3. $Z^* = G^{l^*}; x^* = x^{l^*}$.

### 5.5 ROBUST UNLOADING PROBLEM

In section 5.3 and 5.4 we looked at the approach by Bertsimas and Sim (2004) and the theorems, bounds, and algorithm developed by them. In this section we use their approach to formulate the robust counterpart of the unloading problem from chapter 3. Then we solve the problem and use simulation to draw conclusions on the effectiveness of the method for this problem. The integer programming formulation of (*TWC*) is:

$$z = \min \sum_{j=1}^{n}\sum_{k=1}^{n} w_j p_k x_{kj} + \sum_{j=1}^{n} w_j p_j$$

subject to

$$x_{kj} + x_{jk} = 1 \qquad \text{for } j,k = 1,\dots,n,\ j \neq k,$$

$$x_{kj} + x_{lk} + x_{jl} \geq 1 \qquad \text{for } j,k,l = 1,\dots,n,\ j \neq k,\ j \neq l, k \neq l,$$

$$x_{jk} \in \{0,1\} \qquad \text{for } j,k = 1,\dots,n,$$

$$x_{jj} = 0 \qquad \text{for } j = 1,\dots,n.$$

In this problem, we assume that the coefficients of the objective function change. The weights $w_j,\ j \in N = \{1,2,\dots,n\}$ remain unchanged whereas the processing time changes. The processing times $p_j,\ j \in N = \{1,2,\dots,n\}$ take values in the interval $[p_j, p_j + d_j]$, where $d_j \geq 0,\ \ j \in N$ while the constraint set $X$ is fixed. Under the approach of Bertsimas and Sims the robust counterpart of this formulation is subject to the same constraints:

$$Z^* = \min \sum_{j=1}^{n}\sum_{k=1}^{n} w_j p_k x_{kj} + \sum_{j=1}^{n} w_j p_j + \max_{\{S|S\subseteq N, |S|\le\Gamma\}} \left\{ \sum_{j=1}^{n}\sum_{k\in S} w_j d_k x_{kj} + \sum_{j\in S} w_j d_j \right\}$$

This can be written as the following using theorem 5.4.1.

$$\min \sum_{j=1}^{n}\sum_{k=1}^{n} w_j p_k x_{kj} + \sum_{j=1}^{n} w_j p_j + Z\Gamma + \sum_{j=1}^{n}\sum_{k\in S} q_{kj}$$

subject to

$$Z + \sum_{k\in S} q_{kj} \ge \sum_{k\in S} w_j d_k x_{kj} + w_j d_j \qquad \text{for } j = 1,\dots,n,$$

$$x_{kj} + x_{jk} = 1 \qquad \text{for } j,k = 1,\dots,n,\ j \ne k,$$

$$x_{kj} + x_{lk} + x_{jl} \ge 1 \qquad \text{for } j,k,l = 1,\dots,n,\ j \ne k,\ j \ne l, k \ne l,$$

$$Z \ge 0$$

$$q_{kj} \ge 0 \qquad \text{for } j,k = 1,\dots,n,$$

$$x_{jk} \in \{0,1\} \qquad \text{for } j,k = 1,\dots,n,$$

$$x_{jj} = 0 \qquad \text{for } j = 1,\dots,n.$$

### 5.5.1 Computational Results

Numerical experiments conducted using GAMS/CPLEX 11.2.0. The weights $w_j$ and the processing times $p_j$ were randomly generated from uniform distribution in the interval (0,10] and [20,100] respectively. The increases $d_j$ in processing times were randomly generated uniform random numbers in the interval [10,100]. We tested the problem for 20 trucks and 30 trucks. That is for $|N| = 20$ and for $|N| = 30.$ We tried to run the experiment for 40 trucks, but limitations on the version of the program used prohibited the large problem instance from running. In our experiments, $\Gamma$ takes integral values from [0,*n*]. We ran the experiments for every $\Gamma$. First we ran the mixed integer programming model and then the algorithm. It did not take a long time in either method to run for one value of $\Gamma$. But it took about 30 to 45 minutes to run for all $\Gamma$ especially with 30 trucks. We can see that the objective values of the robust problem model are controlled by the values of $\Gamma$.

To analyze the robustness of the solution we follow the same method as in Bertsimas and Sim (2003). Let $x(\Gamma)$ be the optimal solution to the robust model under the parameter $\Gamma$ and $\overline{Z}(\Gamma)$ be the nominal total weighted completion time $z$ in the absence of any increase in processing times with the same solution $x(\Gamma)$. After that, the distribution of the objective is simulated by changing the processing times at random. In this simulation each processing time $p_j$ changes to $p_j + d_j$ with probability $\rho$. Under this simulation the standard deviation $\sigma(\Gamma)$ for each $\Gamma$ is calculated. 20000 simulation instances were created for each $\Gamma$ to find the standard deviation.

Table 5.1 shows the results for 20 trucks. Here the second column shows the nominal solution $\overline{Z}(\Gamma)$, and the third column shows the robust solution $Z^*(\Gamma)$. From Table 5.1 and Figure 5.1 we can see that the robust solution increases as $\Gamma$ increases, and there is an overall increase in the nominal solution. The last four columns in Table 5.1

show the standard deviation when the processing time $p_j$ changes to $p_j + d_j$ with probability $\rho$. We considered four different probabilities, 0.1, 0.2, 0.3 and 0.5.

**Table 5.1:** Robust solution, nominal solution and standard deviation for twenty trucks

| $\Gamma$ | $\overline{Z}(\Gamma)$ (nominal) | $Z^*(\Gamma)$ (robust) | $\sigma(\Gamma)$ if $\rho = 0.1$ | $\sigma(\Gamma)$ if $\rho = 0.2$ | $\sigma(\Gamma)$ if $\rho = 0.3$ | $\sigma(\Gamma)$ if $\rho = 0.5$ |
|---|---|---|---|---|---|---|
| 0 | 35954 | 35954 | 4989.2 | 4592.9 | 11901.6 | 18734.9 |
| 1 | 36227 | 38825 | 4750.6 | 4437.7 | 11438.9 | 17924.9 |
| 2 | 36455 | 41221 | 4778.5 | 4439.9 | 11440.9 | 17945.4 |
| 3 | 36455 | 43576 | 4719.6 | 4416.0 | 11358.7 | 17912.2 |
| 4 | 36612 | 45769 | 4788.2 | 4457.2 | 11474.6 | 17970.4 |
| 5 | 36563 | 47896 | 4763.9 | 4475.0 | 11419.3 | 17920.0 |
| 6 | 36627 | 50074 | 4777.8 | 4461.6 | 11468.3 | 18026.3 |
| 7 | 37304 | 52089 | 4741.9 | 4472.4 | 11350.0 | 17781.4 |
| 8 | 36809 | 54058 | 4808.0 | 4441.9 | 11393.6 | 17870.3 |
| 9 | 36971 | 55834 | 4813.8 | 4517.2 | 11456.5 | 17954.5 |
| 10 | 36928 | 57554 | 4704.1 | 4424.1 | 11307.8 | 17801.3 |
| 11 | 36665 | 59237 | 4809.1 | 4507.3 | 11480.8 | 17950.8 |
| 12 | 36796 | 60711 | 4658.2 | 4352.4 | 11151.1 | 17407.0 |
| 13 | 36716 | 62005 | 4625.2 | 4305.1 | 11067.9 | 17367.9 |
| 14 | 36717 | 63342 | 4576.5 | 4285.1 | 11044.7 | 17272.5 |
| 15 | 36629 | 64529 | 4543.8 | 4240.3 | 10891.5 | 17187.5 |
| 16 | 36665 | 65672 | 4592.1 | 4248.2 | 10940.7 | 17176.0 |
| 17 | 36622 | 66728 | 4532.9 | 4235.9 | 10890.9 | 17177.2 |
| 18 | 36622 | 67730 | 4551.8 | 4270.3 | 10976.8 | 17167.6 |
| 19 | 36827 | 68412 | 4482.9 | 4115.9 | 10767.5 | 16904.8 |
| 20 | 37431 | 68797 | 4373.3 | 3901.3 | 10457.9 | 16439.8 |

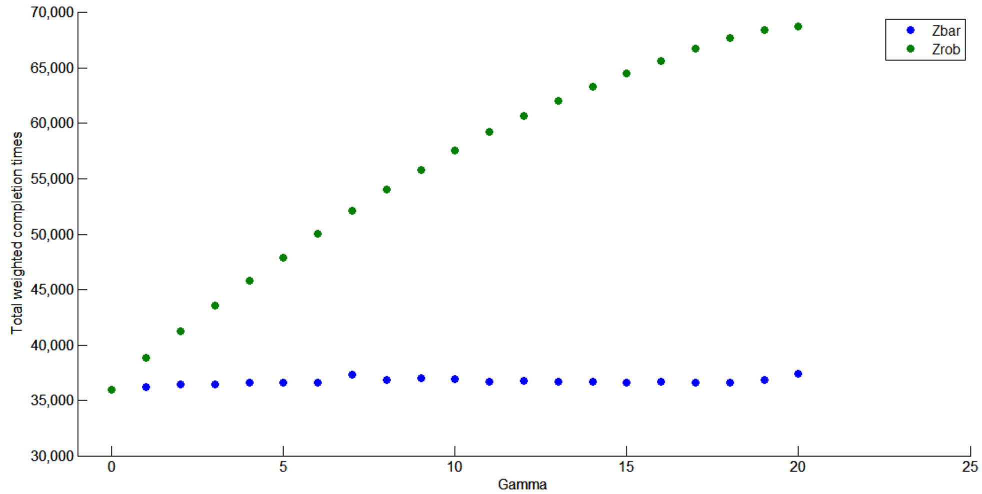


**Figure 5.1:** The nominal solution Zbar and the robust solution Zrob for twenty trucks

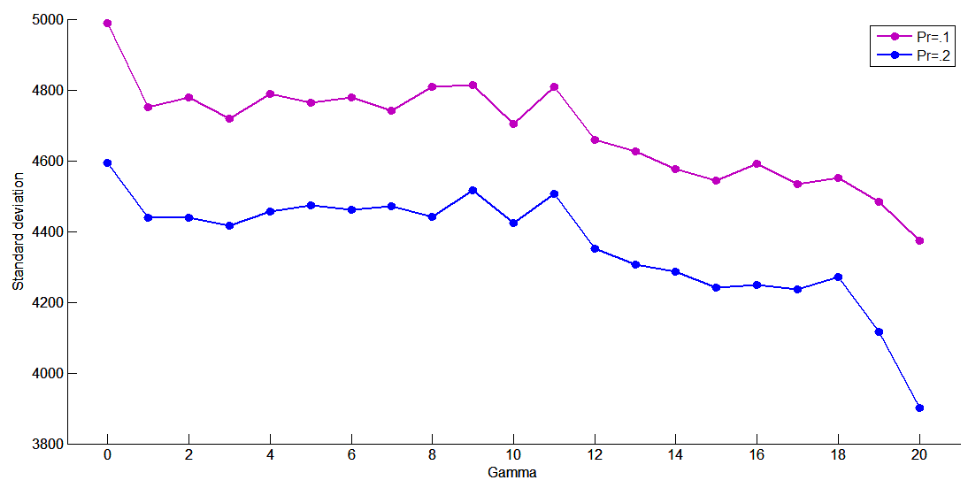


**Figure 5.2:** Standard deviation for twenty trucks when $\rho = 0.1$ and $\rho = 0.2$

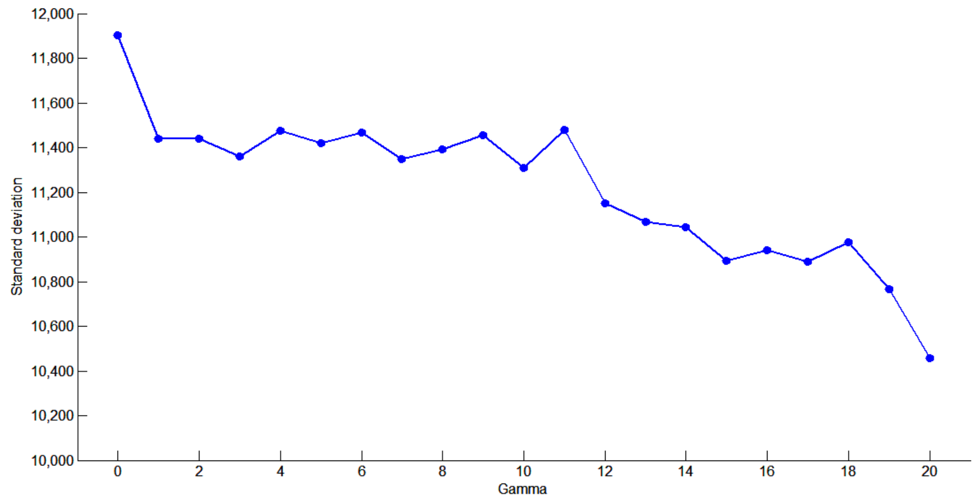


**Figure 5.3:** Standard deviation for twenty trucks when $\rho$ = 0.3

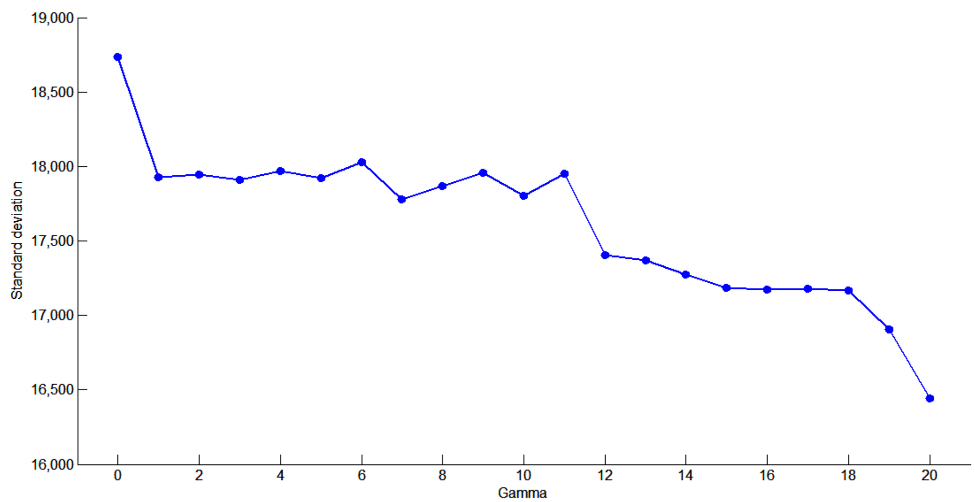


**Figure 5.4:** Standard deviation for twenty trucks when $\rho$ = 0.5

Figure 5.2 shows the standard deviation for $\rho = 0.1$ and $\rho = 0.2$. Figures 5.3 and 5.4 respectively show the standard deviation for $\rho = 0.3$ and $\rho = 0.5$. Overall as $\Gamma$ increases, the standard deviation decreases which suggests the robustness of the solution increases as $\Gamma$ increases. Now, we will look at the results for thirty trucks.

**Table 5.2:** Robust solution, nominal solution and standard deviation for thirty trucks

| $\Gamma$ | $\overline{Z}(\Gamma)$ (nominal) | $Z^*(\Gamma)$ (robust) | $\sigma(\Gamma)$ if $\rho = 0.2$ |
|---|---|---|---|
| 0 | 72276 | 72276 | 7872.0 |
| 1 | 72704 | 76340 | 7761.7 |
| 2 | 73396 | 80140 | 7611.3 |
| 3 | 73459 | 83347 | 7677.8 |
| 4 | 73491 | 86619 | 7621.1 |
| 5 | 73700 | 89864 | 7508.7 |
| 6 | 73871 | 92765 | 7595.2 |
| 7 | 73893 | 95804 | 7607.1 |
| 8 | 74085 | 98658 | 7541.6 |
| 9 | 74586 | 101865 | 7497.5 |
| 10 | 74594 | 105185 | 6823.7 |
| 11 | 74407 | 107628 | 6697.3 |
| 12 | 73613 | 109190 | 7548.7 |
| 13 | 73657 | 111838 | 7562.4 |
| 14 | 73831 | 114322 | 7547.4 |
| 15 | 74330 | 117864 | 6962.7 |
| 16 | 73460 | 120030 | 7721.3 |
| 17 | 74161 | 121264 | 7365.4 |
| 18 | 73777 | 123352 | 7320.9 |
| 19 | 73777 | 125440 | 7402.5 |
| 20 | 73931 | 128176 | 7305.1 |
| 21 | 74699 | 130535 | 7008.4 |
| 22 | 74206 | 131317 | 7199.8 |
| 23 | 74312 | 132770 | 7242.5 |
| 24 | 74374 | 134577 | 7187.6 |
| 25 | 74374 | 135843 | 7238.1 |
| 26 | 75178 | 135977 | 6115.0 |
| 27 | 75608 | 136679 | 5994.3 |
| 28 | 75214 | 137292 | 6080.6 |
| 29 | 75214 | 137796 | 6091.7 |
| 30 | 75401 | 137858 | 6019.9 |

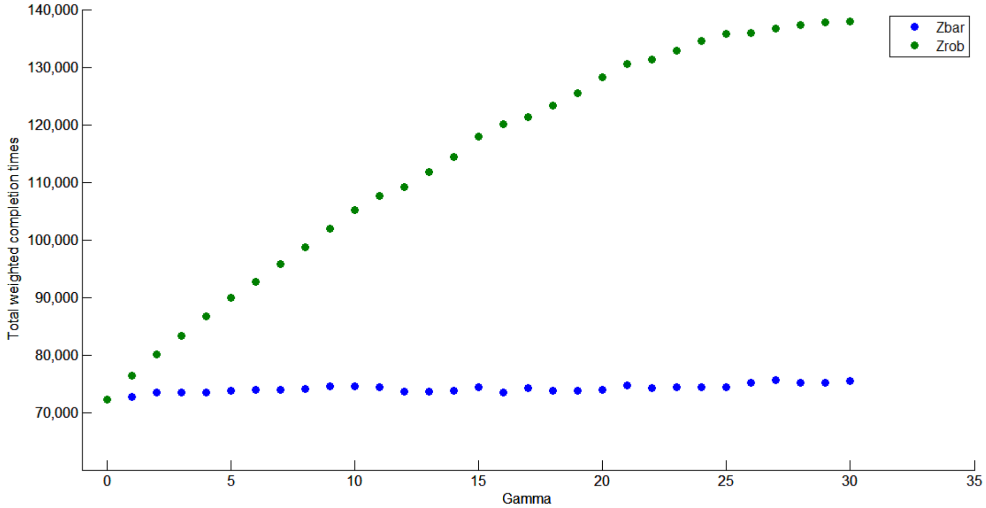


**Figure 5.5:** The nominal solution Zbar and the robust solution Zrob for thirty trucks

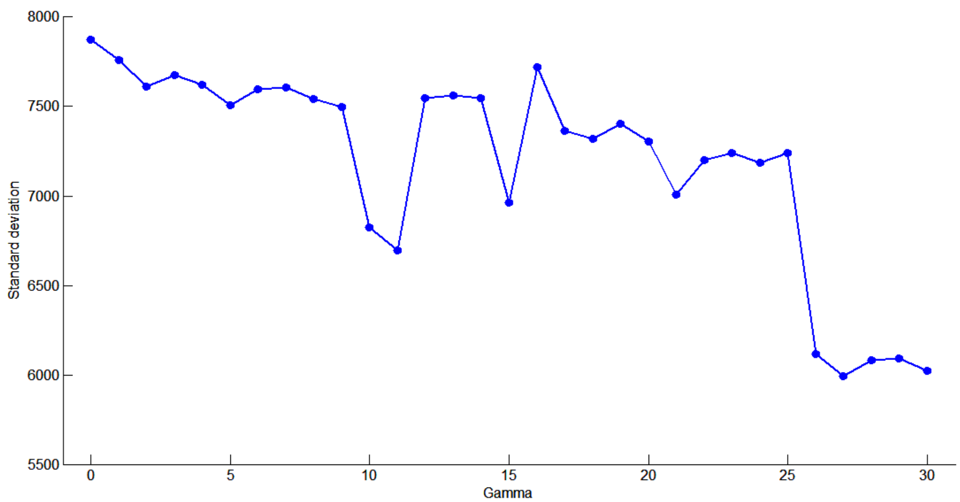


**Figure 5.6:** Standard deviation for thirty trucks when $\rho = 0.2$

The results of robust solutions and nominal solutions for thirty trucks are in Table 5.2 and Figure 5.5. The standard deviation for probability 0.2 is given in Figure 5.6. We see similar results in the case of thirty trucks also. Then we tried for forty trucks and we could not run it due to the limitations of the version of the program used. Next, we wanted to see how these results compare with the case when all trucks have unit weight. We use the same parameters for twenty trucks. The results are summarized in Table 5.3 and Figure 5.7 gives the robust and nominal solutions. Figure 5.8 shows the standard deviation for probability 0.2.

**Table 5.3:** Robust solution, nominal solution and standard deviation for twenty trucks with unit weight

| $\Gamma$ | $\overline{Z}(\Gamma)$ (nominal) | $Z^*(\Gamma)$ (robust) | $\sigma(\Gamma)$ if $\rho = 0.2$ |
|---|---|---|---|
| | 9679 | 9679 | 1249.1 |
| 1 | 9679 | 10677 | 1260.9 |
| 2 | 9767 | 11697 | 1257.8 |
| 3 | 9754 | 12503 | 1249.7 |
| 4 | 9820 | 13344 | 1246.0 |
| 5 | 9844 | 14102 | 1238.8 |
| 6 | 9844 | 14812 | 1242.8 |
| 7 | 9882 | 15505 | 1234.5 |
| 8 | 10100 | 16265 | 1169.2 |
| 9 | 9882 | 16742 | 1246.0 |
| 10 | 9886 | 17310 | 1227.5 |
| 11 | 9909 | 17944 | 1247.6 |
| 12 | 9902 | 18317 | 1221.2 |
| 13 | 9902 | 18779 | 1229.7 |
| 14 | 9937 | 19253 | 1258.1 |
| 15 | 10006 | 19517 | 1235.8 |
| 16 | 10276 | 19421 | 1072.2 |
| 17 | 10230 | 19775 | 1118.3 |
| 18 | 10445 | 19711 | 1009.4 |
| 19 | 10503 | 19729 | 964.6 |
| 20 | 10511 | 19751 | 949.4 |

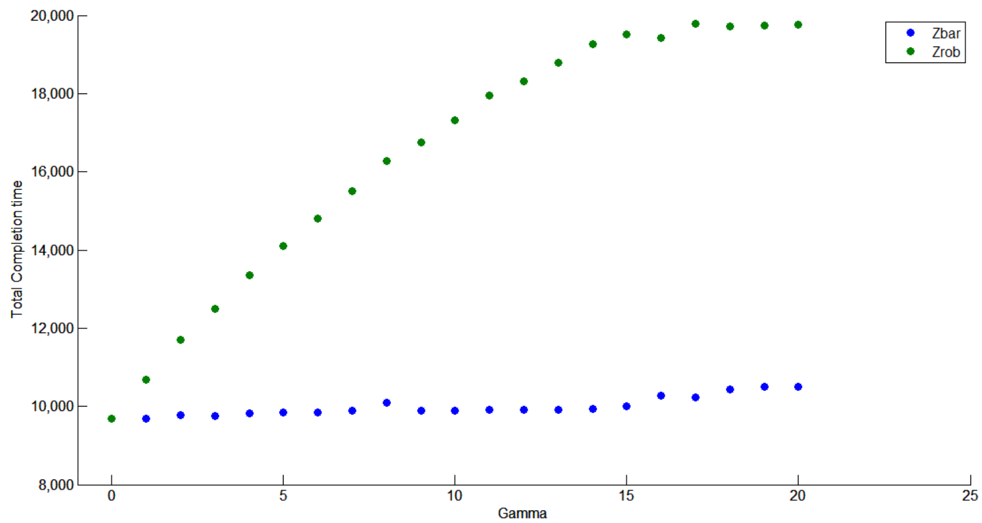


**Figure 5.7:** The nominal solution Zbar and the robust solution Zrob for twenty trucks with unit weights.

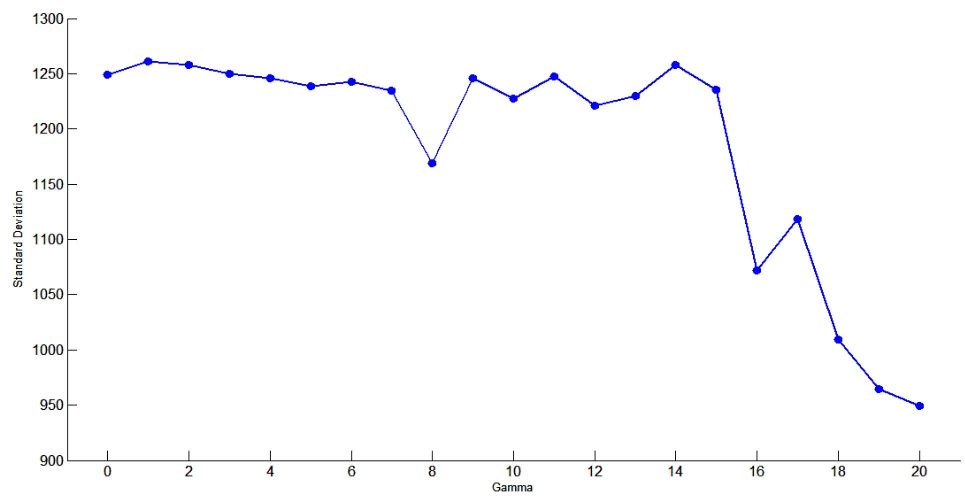


**Figure 5.8:** Standard deviation for twenty trucks with unit weight when $\rho = 0.2$

At the end, when $\Gamma$ gets larger and $\Gamma = 18$, there is a decrease in the robust value when the weights are one and then it goes up. The standard deviation behaves the same way as in the other cases. Overall, there is a decrease in the standard deviation indicating the robustness of the problem increases as $\Gamma$ increases.

From all the above experiments we see that the protection-based approach works well for this problem. Overall, the robust value increases with $\Gamma$. Although the increase on the nominal value and the decrease in standard deviation are not monotonic, the decrease in variation suggests that this approach helps in practice. This approach is less conservative compared to the scenario-based approach we used in chapters 3 and 4 where we optimized the worst case scenario. In the next two chapters we see the uncertainties in two different industries.

# Chapter 6: Cross-dock Operations at Goodwill Industries

## 6.1 INTRODUCTION

This introduction to Goodwill Industries is made possible by a meeting with the transportation coordinator and from their official website, www.austingoodwill.org. Goodwill Industries was founded by Rev. Edgar J. Helms in Boston in 1902. Goodwill Industries of Central Texas (GICT) was established in Austin in 1958. GICT is a private non-profit organization with mission statement "We provide job-related services and opportunities for people with barriers to employment" (austingoodwill.org). GICT served sixty-nine people during the first year of their operation and now they help thousands of people every year.

GICT have Adult Programs, Youth Programs, Commercial Services and Environmental Business Services (EBS). The Adult Program includes a job source program, community rehabilitation program, and other services for adults. The Youth Program helps at risk students to stay in school, earn a GED, or find a job. Their Commercial Services acquire contracts to serve local businesses with solutions in order to provide job opportunities for people with disabilities or other barriers. The EBS addresses electronic waste concerns. Their electronic and computer recycling program recycles 230 tons of electronic waste every month without sending any waste to landfills. Also, they recycle many other products by reusing them or reprocessing them to make new materials.

## 6.2 MOTIVATION

As a regular donor to GICT for many years, I was personally interested in their operations. The Technology warehouse was very close to my home and I saw their trucks operating whenever I passed the warehouse. A year ago, just as I started to work on cross-dock operations, I noticed there were no longer any trucks or activity. I was very curious and when I called them to see what happened, I found out that they closed that particular cross-dock. After some inquiry, I obtained an appointment to see their new cross-dock and talk to them about their cross-dock operations. As I was talking to them, I realized that they do their operations manually, and I decided to model the different problems and uncertainties that arise in their operations.

## 6.3 THE GICT NETWORK

GICT has 11 attended donation centers (ADC) and 23 stores around Central Texas. Two of these stores are computer stores. They cover areas up to Bastrop to the east, Fredricksburg to the west, Waco to the north, and San Marcos to the south. The ADCs are either small trailers or small bookstores. Also, they have a general resource center (GRC) which has a cross-dock, outlet store, and service center. In December 2010, they consolidated their Technology warehouse with six docks and Springdale warehouse with nine docks into a cross-dock with 32 docks and a general resource center. The GICT network has been constructed and the distances were calculated using the store locations and www.googlemaps.com. Figure 6.1 and Table 6.1 show the network and the distances between them together with travel time during nonpeak traffic hours respectively.

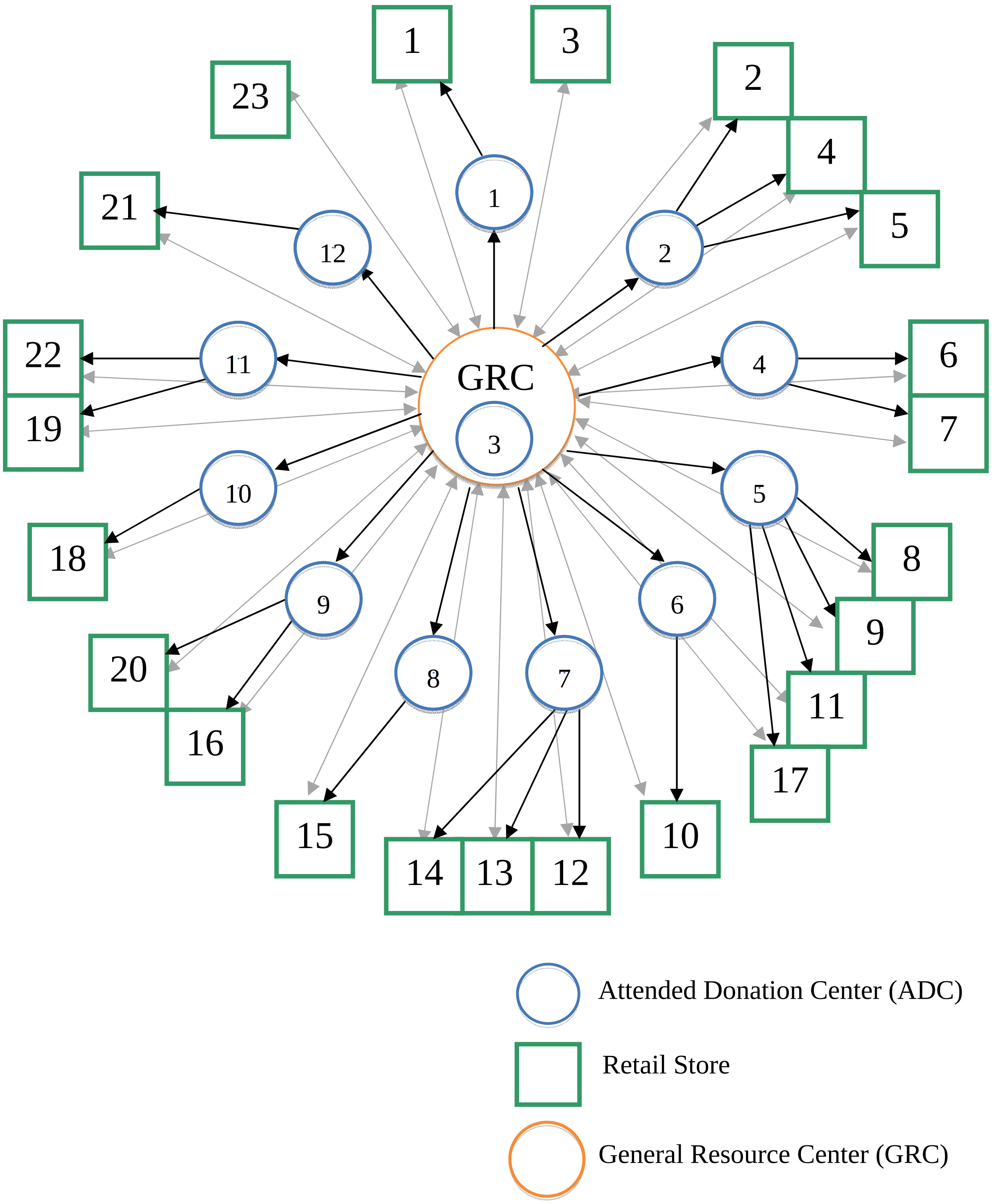


**Figure 6.1:** The GICT network

**Table 6.1:** Distance and travel time during nonpeak traffic times

| | GRC to store | | GRC to designated ADC for the store | | Designated ADC to store | |
|---|---|---|---|---|---|---|
| Stores | Distance in miles | Time in minutes | Distance in miles | Time in minutes | Distance in miles | Time in minutes |
| 1 | 24 | 30 | 21 | 30 | 9 | 17 |
| 2 | 16 | 19 | 16 | 23 | 4 | 10 |
| 3 | 13 | 17 | 0 | 0 | 13 | 17 |
| 4 | 11 | 21 | 16 | 23 | 5 | 10 |
| 5 | 6 | 9 | 16 | 23 | 15 | 19 |
| 6 | 7 | 12 | 8 | 17 | 7 | 15 |
| 7 | 7 | 13 | 8 | 17 | 5 | 9 |
| 8 | 12 | 21 | 13 | 24 | 5 | 9 |
| 9 | 11 | 18 | 13 | 24 | 15 | 23 |
| 10 | 19 | 27 | 20 | 27 | 3 | 8 |
| 11 | 26 | 34 | 8 | 17 | 34 | 46 |
| 12 | 28 | 33 | 28 | 36 | 3 | 6 |
| 13 | 31 | 35 | 28 | 36 | 7 | 13 |
| 14 | 31 | 37 | 28 | 36 | 5 | 10 |
| 15 | 36 | 45 | 38 | 47 | 18 | 28 |
| 16 | 34 | 40 | 24 | 34 | 20 | 24 |
| 17 | 78 | 90 | 8 | 17 | 74 | 91 |
| 18 | 23 | 33 | 24 | 33 | 3 | 6 |
| 19 | 10 | 15 | 17 | 22 | 8 | 11 |
| 20 | 26 | 34 | 24 | 34 | 8 | 16 |
| 21 | 29 | 37 | 22 | 32 | 14 | 19 |
| 22 | 10 | 17 | 17 | 22 | 14 | 19 |
| 23 | 107 | 121 | 0 | 0 | 107 | 121 |

## 6.4 Description of GICT Cross-dock and Operations

The donations are collected at the stores or at the ADCs. The donated goods from the ADCs directly go to the stores. The stores have an auction to sell the expensive jewelry, china, antiques, collectables, and cars once a week. The stores try to sort and sell the other donated items except the computers. The computers are brought to the GRC and sent to their computer stores. The unsold goods in the stores are transported to the GRC as well to make room for new items. These goods are unloaded at the cross-dock in the GRC. Gate numbers 20 to 32 are used to unload the inbound trucks. The inbound trucks are unloaded and sorted inside the cross-dock. Gates 18 and 19 are used for mail. Gate numbers 12 to 17 are used for outbound trucks. Gate numbers 12 to 17 are also used to reload the sorted clothes, metal, and household items, and gates 7 to 11 are used to unload the sorted items for whole sale in the outlet store. Gates 1 to 6 are part of retail store and they are not used. Figure 6.2 shows the layout of the GRC.

The trucks are normally unloaded in the order they come in. They can unload up to three trucks at a time. If any of the drivers want to leave, they come and help unload other trucks so that the time to start unloading their truck is decreased. Most Goodwill outbound trucks leave the GRC empty and either drive to the ADCs to pick up goods and deliver them to the stores, or they go to the stores directly to pick up the unsold goods. The only Goodwill outbound trucks that are not empty are the trucks that transport computers to their computer store. Other outbound trucks are mostly from the vendors who are buying the unsold sorted goods. There are four box trucks and five trucks with trailers. Each driver makes a maximum of three to five runs per day. Only three box trucks run per day since they use one truck for trash and it breaks down very often.

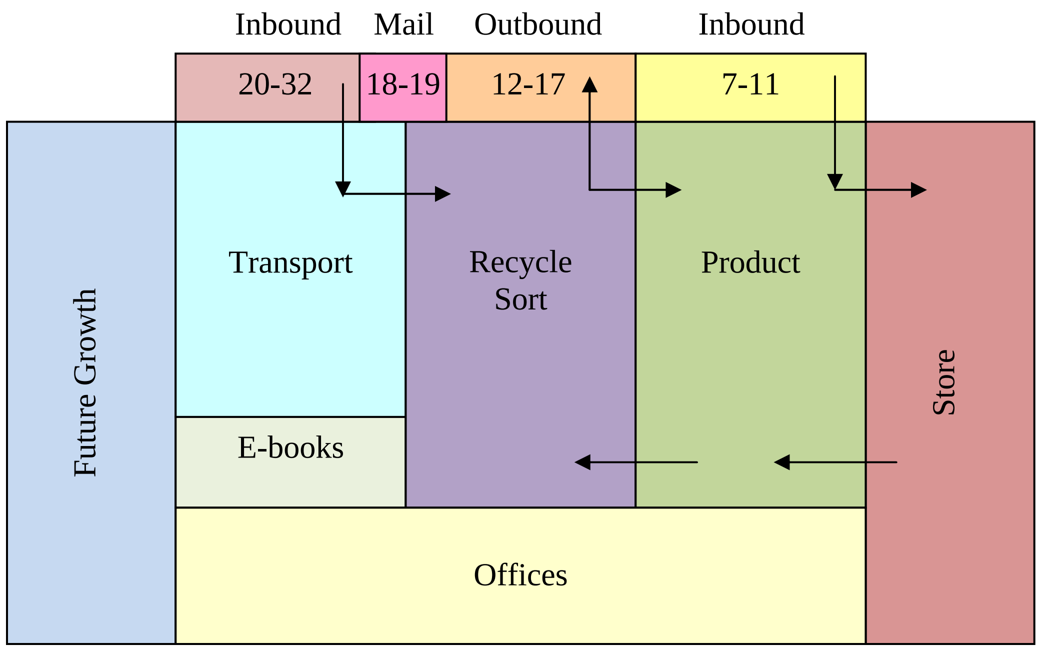


**Figure 6.2:** Layout of GRC and the gates

The trucks leave starting from 7:30 am and must be back in their center by 5:30 pm after their runs. The drivers need to be back in time to go home after work. Otherwise, Goodwill must pay overtime for their drivers. Paying overtime to the drivers is a major expense which they try to avoid. So the trips to the farthest stores must be planned in such a way that they will have enough time to load or unload and come back. The GRC has a GPS system to monitor the trucks, and they can track the exact location of each truck at any time. The trucks come back to the GRC with the unsold items and the items need to be recycled. They are unloaded at the inbound gates.

The unsold goods are sorted as computers, televisions, metal, shoes, and other items (clothes and household) at the GRC cross-dock in the recycle or sort area. Clothes and household items are put in large bins and loaded on the truck again to move to gates 7 to 11. They use five forklifts to move the items. The bins that contain sorted items are unloaded and sent through the door to their wholesale table in the outlet store. Each table stays there for an hour and half where customers pick up the items for $1.49 per pound. Then the rest of the items are moved (with the table) into the cross-dock again.

Clothes are separated and packed into large bales. Computers are sent to the computer stores where the hard drive is erased and the computers or their parts are sold or recycled in an environmentally safe way. Televisions, metal, shoes, bales of clothes, and other items are sold to different vendors and the vendors sell them in other countries. The items are picked up by outbound trucks from vendors to transport them. The GRC has a list of vendors handy, and they sell these goods to whichever vendor can pick them up the soonest to avoid storage cost. Finally, the bags of trash are compacted and disposed of.

The GRC operates seven days a week and ten hours a day. Each driver works ten hours a day for four days. To reduce their expenses they need to avoid paying overtime as far as possible. At the same time they have to serve all their stores. They have to make sure the stores have adequate inventory to sell and that the stores do not overflow with unsold goods. They need to sell the unsold items to vendors as soon as possible since they have no storage space available at the cross-dock. Since they have to sort the items at the cross-dock this can be considered a Post C operation.

### 6.4.1 Uncertainties in GICT Operations

There are several uncertainties in GICT operations. They don't know what products are on the truck until they unload it. On an average, it takes about twenty minutes to unload the truck. If there is a full truck load (FTL), then it takes about thirty to forty minutes to unload. If there is a television in the truck, then it takes more time to unload because additional care is needed to unload fragile items. Also, in summer when there are interns and youth working at the dock it takes more time to unload. The time to unload depends on the number of workers available at the dock as well.

When the trucks are on the road there may be traffic delays due to congestion or incidents. The trucks may be stopped for inspection; break-down and have to be towed; or be in the shop for maintenance. Drivers may not be available to work, and paying overtime to drivers is an expensive concern. Stores may delay the truck's departure if they are not ready by the time the truck arrives, or the truck may leave with less than truck load (LTL). Normally stores that are not very busy delay the trucks. Another major issue is the breakdown or in-house maintenance of balers, trash compactors, or forklifts. In addition, each location has uncertain peak and slow donation and sales periods. We try to address some of these uncertainties in the following sections.

## 6.5 Problem Description of Serving the Stores

First, we consider the problem of serving their stores. The store needs new inventory which comes through two different means: the donations dropped off at the stores directly or the donations dropped off at the ADCs. If a store needs inventory from

ADC, the inventory is picked up by a truck from the CRC and dropped off at the store. Since the trucks are dropping off goods at the stores, they can pick up unsold goods that need to be disposed off. Otherwise, the trucks will be traveling to GRC empty. Thus when a store requests for delivery, the disposal is included in that order. When a store needs a truck only for pick up, the truck from ADC goes empty to the store and returns with the unsold goods to dispose them off. The trucks must be back after all their trips by 5:30 pm. Since the traffic is at its peak towards that time, the farthest stores are served earlier in the day. Thus the time to serve each store is not altered by transportation delay. In addition to that, Goodwill will avoid having to pay overtime.

### 6.5.1 Description of Some Parameters and Objective

We let the parameter $p_s = 1,$ if store $s$ needs delivery from ADC (and to dispose of unsold goods) and $p_s = 0$ otherwise. If a store only requests for pickup of the unsold goods, an empty truck goes directly to the store. We let the parameter $q_s = 1,$ if store $s$ only needs pickup to dispose unsold inventory and $q_s = 0$ otherwise. If the two computer stores need inventory, they are picked up from the cross-dock at the GRC. Each store has a designated ADC to serve, and for the computer stores the GRC is considered in the place of an ADC in the formulation. Every truck serves one store during each trip and comes back to the GRC with unsold inventory. Thus each truck starts as an outbound truck at the GRC and ends as an inbound truck at the GRC during each trip. So we combine all the trucks into one set in the model.

We let $T_s$ be the total time needed to serve store *s*. We can also call $T_s$ as trip time for the truck serving store *s*. $T_s$ includes travel time from GRC to ADC, loading time at the ADC, travel time from the ADC to the store, unloading time at the store, loading time of unsold goods at the store, travel time from the store to the GRC, and unloading time of the inbound truck at the cross-dock. The total service time for a store is calculated using the values of $p_s$ and $q_s$. The objective here is to serve all the stores that request service and come back to GRC by 5:30 pm without incurring any overtime. This is captured by assigning a weight to the objective function and minimizing the weighted total service times. Here we assume the weight of each store to be the trip number of the truck serving that store in order to avoid long distances later in the day.

### 6.5.2 Mathematical Model

The GICT wants to avoid long distances during evening peak hours since they want all their trucks back at GRC before 5:30 pm so that they will not incur overtime pay. For that reason we define the objective to minimize the total weighted service time or trip time where the weight for each store is the trip number of the truck serving the store. In this model, we assume that all stores that request service can be served with no overtime. We call this model as Goodwill Weighted Service Time (*GWST*) problem. The mathematical formulation of (*GWST*) as an integer programming problem is as follows:

**Model Formulation:**

**Sets**:

$I$ set of trucks (inbound and outbound) indexed by $i$

| | |
|---|---|
| $A$ | set of ADCs |
| $S$ | set of stores to be served indexed by $s$ |
| $K$ | set of serving order indexed by $k$ |

**Parameters:**

| | |
|---|---|
| $\alpha$ | Fixed time to unload an inbound truck |
| $\beta$ | Fixed time to unload a truck at the store |
| $\gamma$ | Fixed time to load an outbound truck at the ADC |
| $\delta$ | Fixed time to load a truck at the store |
| $L$ | Limit on the arrival time of the inbound truck at the end of the day |
| $d_s$ | Time to travel from the GRC to store $s$ |
| $f_s$ | Time to travel from the GRC to the ADC designated to serve store $s$ |
| $g_s$ | Time to travel from the ADC designated to serve store $s$ to store $s$ |
| $u_i$ | Departure time of truck $i$ for its first trip from the GRC |
| $p_s$ | 1 if store $s$ needs delivery from the ADC (and pickup unsold inventory) and 0 otherwise |
| $q_s$ | 1 if store $s$ needs to dispose unsold inventory only and 0 otherwise |
| $T_s$ | Total trip time to serve store $s$ |

**Binary variable:**

| | |
|---|---|
| $x_{isk}$ | 1 if truck $i$ serves store $s$ on trip $k$ and 0 otherwise |

**Mathematical Model:**

$$\min \quad \sum_{i\in I}\sum_{s\in S}\sum_{k\in K} kT_s x_{isk} \qquad (6.1)$$

subject to

$$\sum_{s\in S} x_{isk} \le 1 \qquad \forall i\in I, \forall k \in K \qquad (6.2)$$

$$\sum_{i\in I}\sum_{k\in K} x_{isk} = p_s + q_s \qquad \forall s \in S \qquad (6.3)$$

$$x_{isk} \le \sum_{s'\neq s\in S} x_{is'k-1} \qquad \forall i\in I, \forall s \neq s' \in S, \forall k\in K, k>1 \qquad (6.4)$$

$$\sum_{s\in S}\sum_{k\in K} T_s x_{isk} + u_i \le L \qquad \forall i \in I \qquad (6.5)$$

$$x_{isk} \in \{0,1\} \qquad \forall i\in I, \forall s\in S, \forall k\in K \qquad (6.6)$$

where $$T_s = (f_s + \gamma + g_s + \beta + \delta + d_s + \alpha)p_s + (\delta + 2d_s + \alpha)(1-p_s)q_s, \forall s \in S \qquad (6.7)$$

In the above formulation, (6.1) is the objective function. The objective function minimizes the total weighted service time with weight as the trip number $k$ of the truck $i$ serving store $j$. Constraints (6.2) ensure that each truck serves at most one store in each trip. Equations (6.3) make sure every store that requests service is served. Constraints (6.4) take care of the consecutive order of the trips for every truck and every order. For example, trip $k$ for truck $i$ will not happen unless it has already completed $k$–1 trips. Constraints (6.5) enforce the time limit on the total service time of each truck so that there is no overtime. Constraints (6.6) are the binary constraints for the variable. Equations (6.7) give the value of the parameters $T_s$, total trip time or service time required for store $s$. The first term gives the service time when $p_s$ is 1 and the second term gives the service time when $p_s$ is 0 and $q_s$ is 1. That is, when $p_s$ is 1, the store needs delivery and the time involved in going through the designated ADC is included in the first term. When $p_s$ is 0 and $q_s$ is 1 the store needs only pickup and the truck travels directly from the GRC to the store and the ADC is not involved. Note that equations (6.7) use only parameters and no variable.

### 6.5.3 Test Example Minimizing the Sum of Weighted Service Time (*GWST*)

The GRC do not use computers to store the data on their service to stores. Rather, the service is tracked using signup sheets. There is no real data available on that end. Based on the information obtained from the visit, other parameters are approximately assigned using averages. The model was tested using the real network data combined with other assumptions. GAMS/CPLEX 11.2.0 is used to run the model.

In this example, it is assumed that all eight trucks are working and that 90% of the stores request delivery and pickup and 90% of the rest of the stores request only pickup. The stores were picked using a uniform random variable. With this assumption all the stores were included in the service. Stores 11 and 13 requested only for pickup and all the other stores needed delivery and pickup. All the trucks leave at 7:30 am for their first trip. The different time parameters are in minutes. The limit on the time a truck can work is ten hours which is taken as 600 minutes. Time to unload an inbound truck is 20, time to unload the truck at the store is 40, time to load the truck at ADC is 30 and time to load the truck at the store is 40.

The parameter $T_s$, total trip time or service time required for store *s* as calculated by the model for the 23 stores is 207, 182, 164, 184, 181, 174, 169, 184, 195, 192, 128, 205, 130, 213, 250, 228, 328, 202, 178, 214, 218, 188 and 372 respectively. The results are as follows: The sum of weighted trip times is 8520 minutes. The arrival time after all their trips for the eight trucks is 590, 593, 573, 578, 598, 594, 574 and 586 respectively. Table 6.3 shows the order in which the stores were served by the trucks and we can see that the stores with large trip time are mostly served during trip 1.

**Table 6.2:** Stores *s* served by truck *i* during trip *k* deduced from the variable $x_{isk}$

| Trip\Truck | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 16 | 1 | 14 | 20 | 15 | 21 | 23 | 17 |
| 2 | 22 | 12 | 2 | 9 | 4 | 10 | 18 | 13 |
| 3 | 6 | 5 | 19 | 7 | 3 | 8 | | 11 |

## 6.6 Uncertainty in Number of Trucks Available

Suppose that one of the trucks broke down or one of drivers was not available to work. Assume that there are only seven trucks available. If we use the above model 6.5.2, together with the data used in test problem example 6.5.3 we have an infeasible solution. To deal with this situation, we first tried to relax the time limit constraints (6.5) in example 6.5.2 using all eight trucks. In the absence of a time limit, two trucks had overtime whereas no truck had overtime in example 6.5.2. Then we considered the possibility of changing the percentage of stores served in test example 6.6.1. After that in Section 6.6.2, we consider another formulation using overtime to deal with the uncertainty in number of trucks available.

### 6.6.1 Test Example with Available Trucks

In this example, it is assumed that only seven trucks are available and that 70% of the stores request delivery and pickup and 50% of the rest of the stores request only pickup. The stores were picked using a uniform random variable. With this assumption twenty stores were included in the service. Stores 2, 11 and 14 do not request service, stores 8 and 13 request only for pickup and all the other eighteen stores needed delivery and pickup. Other parameters remain the same as in 6.5.3.

The parameter $T_s$, total trip time or service time required for store *s* as calculated by the model for the 20 stores is 207, 164, 184, 181, 174, 169, 102, 195, 192, 205, 130, 250, 228, 328, 202, 178, 214, 218, 188 and 372 respectively. The results are as follows: The sum of weighted trip times is 7970 minutes. The arrival time after all their trips for the seven trucks is 599, 564, 600, 553, 598, 573 and 594 respectively. There is no overtime if the service is reduced or they can choose to serve a percentage of the stores that request service. Table 6.3 shows the order in which the stores were served by the trucks and we can see that the stores with large trip time are served during trip 1.

**Table 6.3:** Stores *s* served by truck *i* during trip *k* deduced from the variable $x_{isk}$

| Trip\Truck | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| 1 | 7 | 5 | 16 | 9 | 21 | 18 | 20 |
| 2 | 8 | 12 | 23 | 6 | 10 | 3 | 15 |
| 3 | 17 | 19 | | 4 | 22 | 1 | 13 |

In addition to the above methods, there is another way to deal with this situation. Only a percentage of the stores that request service receive service. For example only

80% of the stores that request service receive service. This can be accomplished by changing the constraints (6.3) to

$$\sum_{i \in I} \sum_{k \in K} x_{isk} = 0.8(p_s + q_s), \forall s \in S \tag{6.3a}$$

With the use of these constraints overtime can be avoided by adjusting the percentage according to the number of available trucks.

### 6.6.2 Modeling with Overtime

Another option to deal with uncertainty in the number of trucks available is to introduce overtime while serving all the stores that request service. This will be useful during peak donation periods and peak shopping periods. In this model, we want to minimize the total overtime pay which is the same as minimizing the total overtime. This model minimizes the sum of the number of minutes each truck works above the time limit *L* (600 minutes) so that most of the trucks return back to GRC as close as possible to 5:30 pm. For that reason we define the objective to minimize the sum of overtime over all their trucks.

We define $V_i$ to be the overtime worked by truck *i.* The overtime worked by truck *i* will be the difference between the arrival time of truck *i* at GRC after all the trips and *L,* provided truck *i* arrives after the time limit. We call this problem as Goodwill Overtime (*GWOT*) problem. The model is a simple modification of the (*GWST*) problem. In (*GWOT*) model we change the objective function to be the total overtime by all the trucks and one set of constraints to account for the overtime. This model is a mixed integer programming problem.

**Model Formulation:**

**Sets**:

| | |
|---|---|
| $I$ | set of trucks (inbound and outbound) indexed by $i$ |
| $A$ | set of ADCs |
| $S$ | set of stores to be served indexed by $s$ |
| $K$ | set of serving order indexed by $k$ |

**Parameters:**

| | |
|---|---|
| $\alpha$ | Fixed time to unload an inbound truck |
| $\beta$ | Fixed time to unload a truck at the store |
| $\gamma$ | Fixed time to load an outbound truck at the ADC |
| $\delta$ | Fixed time to load a truck at the store |
| $L$ | Limit on the arrival time of inbound truck at the end of the day |
| $d_s$ | Distance between the GRC and store $s$ (or time) |
| $f_s$ | Distance between the GRC and the ADC designated to serve store $s$ |
| $g_s$ | Distance between the ADC designated to serve store $s$ and store $s$ |
| $u_i$ | Departure time of truck $i$ for its first trip from the GRC |
| $p_s$ | 1 if store $s$ needs delivery from the ADC (and pickup unsold inventory) and 0 otherwise |
| $q_s$ | 1 if store $s$ needs to dispose unsold inventory only and 0 otherwise |

$T_s$ Total trip time to serve store $s$

**Positive variable:**

$V_i$ Overtime worked by truck $i$

**Binary variable:**

$x_{isk}$ 1 if truck $i$ serves store $s$ on trip $k$ and 0 otherwise

**Mathematical Model:**

$$\min \quad \sum_{i \in I} V_i \qquad (6.8)$$

subject to

$$\sum_{s \in S} x_{isk} \le 1 \qquad \forall i \in I, \forall k \in K \qquad (6.9)$$

$$\sum_{i \in I} \sum_{k \in K} x_{isk} = p_s + q_s \qquad \forall s \in S \qquad (6.10)$$

$$x_{isk} \le \sum_{s' \ne s \in S} x_{is'k-1} \qquad \forall i \in I, \forall s \ne s' \in S, \forall k \in K, k > 1 \qquad (6.11)$$

$$\sum_{s \in S} \sum_{k \in K} T_s x_{isk} + u_i - L \le V_i \qquad \forall i \in I \qquad (6.12)$$

$$x_{isk} \in \{0,1\} \qquad \forall i \in I, \forall s \in S, \forall k \in K \qquad (6.13)$$

$$V_i \ge 0 \qquad \forall i \in I \qquad (6.14)$$

where $T_s = (f_s + \gamma + g_s + \beta + \delta + d_s + \alpha) p_s + (\delta + 2d_s + \alpha)(1 - p_s) q_s, \forall s \in S$ (6.15)

In the above formulation, (6.8) is the objective function. The objective function minimizes the sum of overtime by all trucks. Constraints (6.9) ensure that each truck serves at most one store in each trip. Equations (6.10) make sure every store that requests service is served. Constraints (6.11) take care of the consecutive order of the trips for every truck and every order. Constraints (6.12) enforce that the overtime for a truck is considered only if the total time for all its trips or the service time for all the stores served by the truck is more than the limit. Otherwise, the overtime will be zero for that truck. Constraints (6.13) are the binary constraints for the variable and (6.14) are the non-negativity constraints for overtime. Equations (6.15) give the trip time or the service time required for each store. The first term gives the service time when $p_s$ is 1 and the second term gives the service time when $p_s$ is 0 and $q_s$ is 1. Note that equations (6.15) use only parameters and no variable. The difference between the models in 6.5.2 and 6.6.2 are the objective functions and the constraints (6.4) and (6.12). Also, if we want to minimize the cost, we can include the cost in the objective function and we can modify the objective function as:

$$\min \quad \sum_{i \in I} C_i V_i \qquad (6.8a)$$

where $C_i$ is the cost of overtime per minute for truck *i*.

### 6.6.3 Test Examples Minimizing Overtime (*GWOT*)

#### *6.6.3.1 Example One*

We used the same data as in test example 6.6.1 to run model 6.6.2. The results were slightly different but no overtime was incurred. The parameters remained the same. The value of the objective function was 0 since there was no overtime. The arrival time after all their trips for the seven trucks is 542, 579, 585, 597, 597, 588 and 593 respectively. Total service time by all trucks is 4081 in test example 6.6.1 and in example 6.6.3.1. Table 6.4 shows the order in which the stores were served by the trucks and we can see that the stores with large trip time are served during trip 1.

**Table 6.4:** Stores *s* served by truck *i* during trip *k* deduced from the variable $x_{isk}$

| Trip\Truck | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| 1 | 20 | 23 | 22 | 3 | 4 | 21 | 6 |
| 2 | 17 | 1 | 18 | 12 | 13 | 19 | 15 |
| 3 | | | 9 | 16 | 8 | 10 | 7 |
| 4 | | | | | 5 | | |

#### *6.6.3.2 Example Two*

In this example, we try to serve more stores than in example 6.6.3.1 with the same number of trucks available. We assumed that only seven trucks are available to serve and that 90% of the stores request delivery and pickup and 50% of the rest of the stores request only pickup. Store 11 did not request to be served. Store13 requests only for pickup and all the other stores needed delivery and pickup. All the parameters remained the same. The total overtime was 396 minutes with trucks 1, 3, 4, 5 and 7 doing an overtime of 25, 66, 42, 153, 110 minutes respectively. The arrival time after all their trips for the seven trucks is 625, 572, 666, 642, 753, 590, and 710 respectively. Considering the fact that the same number of stores are served as in example 6.5.3, the total overtime is only 396 minutes. Table 6.5 shows the order in which the stores were served by the trucks, and we can see that the stores with large trip times are served during trip 1. In the next section we consider the scenario-based approach to deal with uncertainty in travel time, loading time, and unloading time.

**Table 6.5:** Stores $s$ served by truck $i$ during trip $k$ deduced from the variable $x_{isk}$

| Trip\Truck | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| 1 | 12 | 5 | 15 | 13 | 7 | 2 | 3 |
| 2 | 10 | 8 | 20 | 17 | 19 | 14 | 6 |
| 3 | 16 | 1 | 18 | 4 | 22 | 9 | 23 |
| 4 | | | | | 21 | | |

### 6.7 Uncertainty in Trip Time and Scenario-Based Robust Approach

As mentioned earlier in Section 6.4.1 there is uncertainty in travel time and loading and unloading time in the dock and at the stores. These uncertainties results in changing the trip time for a truck or service time for a store. To address these issues we consider the scenario-based approach as in Chapter 3 which minimizes the worst case scenario. We only consider the absolute robust measure. In this section, we present the scenario-based robust models and examples.

### 6.7.1 The Robust Problem of *GWST*

Let $\omega \in \Omega$ to be a set of scenarios and $T_s^{\omega}$ be the trip time to serve store $s$ under scenario $\omega$. Also, we let $y$ be the worst case minimum or the minimum of the maximum sum of weighted service time. Then the objective function will be $\min \max_{\omega\in\Omega} \sum_{i\in I}\sum_{s\in S}\sum_{k\in K} kT_s^{\omega} x_{isk}$ and the scenario-based robust goodwill weighted service time problem (*RGWST*) can be formulated as follows:

$$\min \quad y \tag{6.16}$$

subject to

$$y \geq \sum_{i\in I}\sum_{s\in S}\sum_{k\in K} kT_s^{\omega} x_{isk} \qquad \forall \omega \in \Omega \tag{6.17}$$

$$\sum_{s\in S} x_{isk} \leq 1 \qquad \forall i \in I, \forall k \in K \tag{6.18}$$

$$\sum_{i\in I}\sum_{k\in K} x_{isk} = p_s + q_s \qquad \forall s \in S \tag{6.19}$$

$$x_{isk} \leq \sum_{s'\neq s\in S} x_{is'k-1} \qquad \forall i \in I, \forall s \neq s' \in S, \forall k \in K, k > 1 \tag{6.20}$$

$$\sum_{s\in S}\sum_{k\in K} T_s^{\omega} x_{isk} + u_i \leq L \qquad \forall i \in I, \forall \omega \in \Omega \tag{6.21}$$

$$x_{isk} \in \{0,1\}, \forall i \in I \qquad \forall i \in I, \forall s \in S, \forall k \in K \tag{6.22}$$

#### *6.7.1.1 Example One*

We considered eight trucks and five scenarios with 70% of the stores requesting delivery and pickup and 50% of the rest of the stores requesting only pickup. The stores were picked using a uniform random variable. With this assumption, twenty stores were included in the service. Stores 2, 11 and 14 do not request service, stores 8 and 13 request only for pickup, and all the other eighteen stores need delivery and pickup.

Other parameters remain the same as in 6.6.1. The individual trip time to serve a store $T_s^{\omega}$ increases from the usual trip time by the uniform random variable (0, 60) under each scenario with a probability of 30%. Table 6.6 shows the trip times for each store under different scenarios. The robust sum of min max weighted trip times is 7751 minutes and the scenario is two. Table 6.7 shows the order in which the stores were served by the trucks.

**Table 6.6:** $T_s^{\omega}$, Trip time for store *s* under scenario $\omega$.

| $s \backslash \omega$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 207 | 207 | 207 | 246 | 207 |
| 3 | 164 | 211 | 174 | 164 | 164 |
| 4 | 186 | 184 | 184 | 184 | 184 |
| 5 | 195 | 237 | 181 | 181 | 181 |
| 6 | 218 | 186 | 190 | 174 | 184 |
| 7 | 169 | 169 | 169 | 169 | 169 |
| 8 | 102 | 102 | 102 | 102 | 102 |
| 9 | 239 | 229 | 195 | 195 | 195 |
| 10 | 192 | 215 | 201 | 192 | 192 |
| 12 | 222 | 229 | 228 | 205 | 211 |
| 13 | 130 | 130 | 176 | 130 | 130 |
| 15 | 289 | 250 | 309 | 290 | 250 |
| 16 | 228 | 245 | 242 | 228 | 228 |
| 17 | 337 | 328 | 369 | 328 | 351 |
| 18 | 202 | 202 | 202 | 202 | 213 |
| 19 | 210 | 222 | 207 | 178 | 178 |
| 20 | 214 | 246 | 214 | 214 | 223 |
| 21 | 228 | 218 | 218 | 218 | 229 |
| 22 | 188 | 188 | 188 | 226 | 215 |
| 23 | 372 | 372 | 372 | 372 | 372 |

**Table 6.7:** Stores *s* served by truck *i* during trip *k* deduced from the variable $x_{isk}$

| Trip\Truck | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 8 | 4 | 6 | 23 | 17 | 13 | 5 | 15 |
| 2 | 16 | 9 | 22 | 19 | 1 | 10 | 12 | 20 |
| 3 | 18 | 7 | 3 | | | 21 | | |

#### *6.7.1.2 Example Two*

In this example we consider seven trucks and five scenarios with 60% of the stores requesting delivery and pickup and 30% of the rest of the stores requesting only pickup. The stores were picked using a uniform random variable. With this assumption

19 stores were included in the service. Stores 2, 11, 14 and 16 do not request service, stores 8, 13 and 19 request only for pickup and all the other sixteen stores needed delivery and pickup. Other parameters remain the same as in 6.6.1. The individual trip time to serve a store $T_s^{\omega}$ increases from the usual trip time by uniform (0, 60) random variable under each scenario with a probability of 30%. Table 6.8 shows the trip times for each store under different scenarios. The robust sum of min max weighted trip times is 6906 minutes and the scenario is two. Table 6.9 shows the order in which the stores were served by the trucks.

**Table 6.8:** $T_s^{\omega}$, Trip time for store *s* under scenario $\omega$

| $s \backslash \omega$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 207 | 207 | 207 | 246 | 207 |
| 3 | 164 | 211 | 174 | 164 | 164 |
| 4 | 186 | 184 | 184 | 184 | 184 |
| 5 | 195 | 237 | 181 | 181 | 181 |
| 6 | 218 | 186 | 190 | 174 | 184 |
| 7 | 169 | 169 | 169 | 169 | 169 |
| 8 | 102 | 102 | 102 | 102 | 102 |
| 9 | 239 | 229 | 195 | 195 | 195 |
| 10 | 192 | 215 | 201 | 192 | 192 |
| 12 | 222 | 229 | 228 | 205 | 211 |
| 13 | 130 | 130 | 176 | 130 | 130 |
| 15 | 289 | 250 | 309 | 290 | 250 |
| 17 | 328 | 328 | 337 | 328 | 369 |
| 18 | 202 | 225 | 202 | 202 | 202 |
| 19 | 90 | 101 | 122 | 134 | 119 |
| 20 | 214 | 214 | 214 | 246 | 214 |
| 21 | 218 | 227 | 228 | 218 | 218 |
| 22 | 188 | 199 | 188 | 188 | 188 |
| 23 | 410 | 399 | 372 | 372 | 372 |

**Table 6.9:** Stores $s$ served by truck $i$ during trip $k$ deduced from the variable $x_{isk}$

| Trip\Truck | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| 1 | 9 | 10 | 20 | 12 | 23 | 15 | 17 |
| 2 | 18 | 6 | 3 | 1 | 22 | 5 | 21 |
| 3 | 13 | 4 | 7 | 19 |  | 8 |  |

We were able to run CPLEX for these two problems in two minutes. In the above two examples we tried to increase the number of scenarios to ten, due to the size of the problem and the limitations of the license we could not run CPLEX. In addition to that, when we tried example one using 90% of the stores requesting service the problem became infeasible because of the time limit constraints. Then we decided to use the model with overtime. In the next section we formulate the scenario-based problem with overtime and present a few examples.

### 6.7.2 The Robust Problem of *GWOT*

Let $V_i^\omega$ be the overtime by truck $i$ under scenario $\omega$. Also, we let $y$ be the worst case minimum or the minimum of the maximum sum of overtime by all trucks. Then the objective function will be $\min \max_{\omega\in\Omega} \sum_{i\in I} V_i^\omega$ and the scenario-based robust goodwill overtime problem (*RGWOT*) can be formulated as follows:

$$\min \quad y \tag{6.23}$$

subject to

$$y \geq \sum_{i\in I} V_i^\omega \qquad \forall \omega \in \Omega \tag{6.24}$$

$$\sum_{s\in S} x_{isk} \leq 1 \qquad \forall i \in I, \forall k \in K \tag{6.25}$$

$$\sum_{i \in I} \sum_{k \in K} x_{isk} = p_s + q_s \qquad \forall s \in S \qquad (6.26)$$

$$x_{isk} \leq \sum_{s' \neq s \in S} x_{is'k-1} \qquad \forall i \in I, \forall s \neq s' \in S, \forall k \in K, k > 1 \qquad (6.27)$$

$$\sum_{s \in S} \sum_{k \in K} T_s^{\omega} x_{isk} + u_i - L \leq V_i^{\omega} \qquad \forall i \in I, \forall \omega \in \Omega \qquad (6.28)$$

$$x_{isk} \in \{0,1\} \qquad \forall i \in I, \forall s \in S, \forall k \in K \qquad (6.29)$$

$$V_i^{\omega} \geq 0 \qquad \forall i \in I, \forall \omega \in \Omega \qquad (6.30)$$

#### *6.7.2.1 Example One*

We consider eight trucks and five scenarios with 90% of the stores requesting delivery and pickup and 50% of the rest of the stores requesting only pickup. The stores were picked using uniform random variable. With this assumption twenty-two stores were included in the service. Store 11 does not request service; store 13 requests only for pickup and all the other twenty-one stores need delivery and pickup. Other parameters remain the same as in 6.6.1.

The individual trip time to serve a store $T_s^{\omega}$ increases from the usual trip time by uniform (0, 60) random variable under each scenario with a probability of 30%. Table 6.10 shows the trip times for each store under different scenarios. It took about fifteen minutes to run the model in CPLEX. The robust min max sum of overtimes is 69 minutes and the scenario is four. Table 6.11 shows the order in which the stores were served by the trucks.

**Table 6.10:** $T_s^{\omega}$, Trip time for store *s* under scenario $\omega$.

| $s \backslash \omega$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 207 | 207 | 207 | 246 | 207 |
| 2 | 182 | 187 | 220 | 182 | 229 |
| 3 | 174 | 164 | 164 | 166 | 164 |
| 4 | 184 | 184 | 184 | 198 | 240 |
| 5 | 181 | 181 | 181 | 225 | 193 |
| 6 | 190 | 174 | 184 | 174 | 174 |
| 7 | 169 | 169 | 169 | 169 | 169 |
| 8 | 184 | 184 | 184 | 228 | 218 |
| 9 | 195 | 195 | 195 | 195 | 218 |
| 10 | 201 | 192 | 192 | 215 | 192 |
| 12 | 205 | 228 | 205 | 211 | 205 |
| 13 | 130 | 176 | 130 | 130 | 130 |
| 14 | 213 | 213 | 213 | 213 | 252 |
| 15 | 250 | 309 | 290 | 250 | 250 |
| 16 | 245 | 242 | 228 | 228 | 237 |
| 17 | 328 | 369 | 328 | 351 | 328 |
| 18 | 202 | 202 | 202 | 213 | 234 |
| 19 | 222 | 207 | 178 | 178 | 178 |
| 20 | 246 | 214 | 214 | 223 | 224 |
| 21 | 218 | 218 | 218 | 229 | 218 |
| 22 | 188 | 188 | 226 | 215 | 188 |
| 23 | 372 | 372 | 372 | 372 | 372 |

**Table 6.11:** Stores *s* served by truck *i* during trip *k* deduced from the variable $x_{isk}$

| Trip\Truck | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 8 | 21 | 18 | 20 | 5 | 7 | 13 | 17 |
| 2 | 3 | 23 | 19 | 6 | 12 | 1 | 15 | 14 |
| 3 | 16 | | 22 | 10 | 2 | 9 | 4 | |

#### *6.7.1.2 Example Two*

In this example we assume that one of the trucks is not available to serve. We consider seven trucks and five scenarios with 70% of the stores requesting delivery and pickup and 50% of the rest of the stores requesting only pickup. The stores were picked using a uniform random variable. With this assumption twenty stores were included in the service. Stores 2, 11 and 14 do not request service, stores 8 and 13 request only for pickup and all the other sixteen stores needed delivery and pickup. Other parameters remain the same as in 6.6.1.

**Table 6.12:** $T_s^{\omega}$, Trip time for store $s$ under scenario $\omega$

| $s\backslash\omega$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 207 | 207 | 207 | 246 | 207 |
| 3 | 164 | 211 | 174 | 164 | 164 |
| 4 | 186 | 184 | 184 | 184 | 184 |
| 5 | 195 | 237 | 181 | 181 | 181 |
| 6 | 218 | 186 | 190 | 174 | 184 |
| 7 | 169 | 169 | 169 | 169 | 169 |
| 8 | 102 | 102 | 102 | 102 | 102 |
| 9 | 239 | 229 | 195 | 195 | 195 |
| 10 | 192 | 215 | 201 | 192 | 192 |
| 12 | 222 | 229 | 228 | 205 | 211 |
| 13 | 130 | 130 | 176 | 130 | 130 |
| 15 | 289 | 250 | 309 | 290 | 250 |
| 16 | 228 | 245 | 242 | 228 | 228 |
| 17 | 337 | 328 | 369 | 328 | 351 |
| 18 | 202 | 202 | 202 | 202 | 213 |
| 19 | 210 | 222 | 207 | 178 | 178 |
| 20 | 214 | 246 | 214 | 214 | 223 |
| 21 | 228 | 218 | 218 | 218 | 229 |
| 22 | 188 | 188 | 188 | 226 | 215 |
| 23 | 372 | 372 | 372 | 372 | 372 |

The individual trip time to serve a store $T_s^{\omega}$ increases from the usual trip time by uniform (0, 60) random variable under each scenario with a probability of 30%. Table 6.12 shows the trip times for each store under different scenarios. The robust min max sum of overtimes is 186 minutes and the scenario is two. Table 6.13 shows the order in which the stores were served by the trucks. In the two robust models (6.7.1) and (6.7.2) note that we have a set of constraints (6.21) and (6.28) which vary according to each scenario in addition to the min max constraints.

**Table 6.13:** Stores $s$ served by truck $i$ during trip $k$ deduced from the variable $x_{isk}$

| Trip\Truck | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| 1 | 21 | 15 | 16 | 22 | 9 | 17 | 6 |
| 2 | 19 | 1 | 23 | 18 | 20 | 4 | 12 |
| 3 | 3 | 5 | 10 | 13 | 8 | 7 | |

In this chapter we looked at the problem of serving the retail stores from the GRC and truck allocation. The major expense of paying overtime to the drivers is addressed. Two models with small variation and their robust problems are formulated and the formulations were tested using partial real data and random data. In the future we plan to model their recycling operations inside the cross-dock. Also, we will try using protection-based approach. The next chapter covers a combined cross-dock problem in which we consider satisfying demand and scheduling inbound and outbound operations, including different costs and penalties.

# Chapter 7: Combined Cross-dock Operations at H-E-B

## 7.1. Introduction

In this chapter, we present a model for the operations of a distribution center in a major Texas grocery chain, H-E-B. H-E-B started as a small, family-owned store by Florence Butt in 1905 in Kerrville in the Texas Hill Country. During 1920's her youngest son, Howard E. Butt started expanding the business by opening two more stores. Since then H-E-B has opened stores all over Texas. Today, H-E-B is a major regional grocery chain serving 155 communities with more than 329 stores in Texas and Mexico. As a central Texas resident for more than 21 years, I have been a loyal customer to H-E-B ever since I moved to Austin. The visits to their headquarters in San Antonio and my observations of their operations are the motivation for this paper.

For a grocery chain of this size, all the products are distributed using trucks. Their distribution centers are located in San Antonio, San Marcos, and Houston. Recently, in 2010 they opened a warehouse and transportation facility in Temple, in the Central Texas area. As a cost-cutting measure, they are proceeding in the direction of cross-docking by reducing their warehousing. Most of the products in the San Marcos facility that come from other vendors are distributed through cross-docking. Also, they use cross-docking for the products that come from different facilities within H-E-B.

The model we discuss in this chapter is concerned with the distribution of perishable goods involving cross-dock operations while satisfying the store's demand. The mathematical model is a result from the real life observations of the operations of different distribution facilities. Consequently, the model closely follows the real constraints that arise on a daily basis in those facilities. The model is illustrated using real-life data representative of an operational day at the distribution facility. In particular, we consider the problem of satisfying the demand from each grocery store by cross-docking and allocating the products to outbound trucks and sequencing the inbound and outbound trucks to minimize several costs associated with that together with time constraint.

The rest of this chapter is organized as follows: Section 2 describes the problem in detail, section 3 presents the mathematical formulation and section 4 provides the sample data and results. Finally, section 5 gives the conclusion and future research directions.

### 7.2 PROBLEM DESCRIPTION

The distribution of food goods needs to be carefully planned because it involves perishable items with strict guidelines and specific shelf lives. From the San Antonio facility, H-E-B ships pallets consisting of perishable items from the Perishable Distribution Center (PDC) together with pallets of a combination of food items from the Combo Distribution Center (Combo), pallets of milk cartons, pallets of egg cartons, and pallets with flowers. The milk and eggs come on forklifts because they are located in buildings connected to PDC. Also, with milk and eggs there is nothing to sort. The floral department completes their order and they arrive early and are normally ready to be loaded before the loading of the outbound trucks starts. The stores place orders via computers to each department.

The Combo is located away from the PDC and smaller trucks are used to bring pallets from the Combo to the PDC. The Combo starts working two hours earlier than the PDC. The Combo fills out orders and packs them in pallets and labels them with the pallet number and store identification in a Pre-C manner. Then they are transported to the PDC. The logistics department works out the order. At the PDC all the above items are cross-docked. Other perishable items are stored at the PDC, and they are sorted and packed into pallets in a Post-C manner for each store. Since major state and federal highways are conveniently located near the facility, trucks are able to serve more than one store at a time in the same route. In this paper, we consider the stores in one major corridor so that we can just use the distance from the dock to the store without using the distance between stores. Also, they need to avoid sending many trucks to one store unless their order exceeds more than one truck load. There will be loaders, and each loader will be assigned almost the same number of trucks. The number of available outbound doors depends on the number of available loaders. Each loader has the same amount of workers and forklifts available.

We want to consider the following problem: Given customer orders for different products, how do we allocate the products to outbound trucks in order to reduce the cost and send them within a reasonable amount of time? At the same time, we want to reduce the number of trucks to a store and minimize the weighted departure time for each store served. A full truck sometimes averages only 2.5 miles per gallon and the mileage depends on the weight of the truck. The cost per pallet per mile is about $0.07 per pallet if the fuel price is about $4 a gallon. Thus each of these costs is separately considered in the objective function. Also, we add a term to the objective function by introducing a cost for weighted distance which is the product of the departure time of the last truck serving the store and the distance of the store from the PDC. We consider the following assumptions and formulation based on the observation of the facilities in San Antonio, Texas.

### 7.3 PROBLEM FORMULATION

Assumptions:

1. Stores are along the corridor of one route so that a truck can serve more than one store to minimize the cost.
2. Customer orders are known a priori.
3. Each inbound truck transports ten pallets and the last inbound truck may have less than ten pallets. They are unloaded in the same order they arrive.
4. The unloading time and the changeover time are the same for each inbound truck.
5. The number of pallets for each store in an inbound truck is known *a priori*.
6. The empty outbound trailers are cleaned and ready for loading at time 0.
7. The time to start loading and the time to load an outbound truck vary.
8. The changeover time is the same for each outbound truck.
9. Not all the outbound trailers are used.
10. The inbound and outbound trucks do not leave the dock until they are completely unloaded or loaded.

11. The maximum number of pallets for an outbound truck is 28.

12. All outbound trucks must arrive at the stores before a fixed time.

Next, we will look at the problem formulation. We will call this problem as Combined Cross-dock Cost Minimization Problem (*CCC*).

**Formulation:**

**Sets**:

| | |
|---|---|
| $I$ | set of inbound trucks indexed by $i$ |
| $J$ | set of outbound trucks indexed by $j$ |
| $A$ | set of inbound doors indexed by $a$ |
| $B$ | set of outbound doors indexed by $b$ |
| $S$ | set of stores to be served indexed by $s$ |
| $K$ | set of serving order indexed by $k$ |
| $N$ | set of product types indexed by $n$ |

**Parameters:**

| | |
|---|---|
| $\alpha$ | Fixed time to unload an inbound truck |
| $\beta$ | Fixed changeover time for an inbound truck |
| $\gamma$ | Fixed changeover time for an outbound truck |
| $\delta$ | Fixed transportation cost per pallet per mile |
| $\lambda$ | Fixed penalty cost for each truck used for one store |
| $\eta$ | Fixed cost for weighted distance completion time per mile-minute |

$\theta$ Fixed time in minutes to load one pallet from combo into an outbound truck

$\omega$ Extra time to serve a store on the way when one truck serves multiple stores

$D$ Latest time the truck can arrive at the store

$L$ Limit on the number of pallets in one truck

$M$ Big number

$d_s$ Distance between PDC and store $s$

$u_i$ Arrival time of inbound truck $i$

$c_j$ Cost of operating outbound truck $j$

$com_s$ Demand of store $s$ in number of pallets from CDC

$type_{ns}$ Demand of store $s$ in number of pallets of perishable product of type $n$

$t_n$ Fixed time in minutes to load one pallet of type $n$ into an outbound truck

$f_{si}$ Number of pallets for store $s$ in inbound truck $i$

$\mu_{si}$ 1 if inbound truck $i$ has a pallet for store $s$ and 0 otherwise (1 if $f_{si} > 0$)

**Decision variables:**

**Non-negative variables:**

$p_{sj}$ Fraction of the demand of store $s$ from CDC satisfied by outbound truck $j$

$e_{nsj}$ Fraction of the demand of store *s* for product type *n* satisfied by outbound truck *j*

$g_i$ Time at which unloading of inbound truck *i* is completed

$r_j$ Time at which all products needed for outbound truck *j* are ready for loading

$PT_j$ Processing time for truck *j*

$h_{jbk}$ Starting time of loading inbound truck *j* at outbound door *b* as *k*th truck

$dep_s$ Latest departure time for store *s*

$y_j$ Departure time of outbound truck *j*

**Binary variables:**

$o_j$ 1 if outbound truck *j* is used and 0 otherwise

$x_{js}$ 1 if outbound truck *j* is used for store *s* and 0 otherwise

$q_{jbk}$ 1 if outbound truck *j* is served at door *b* as *k*th truck and 0 otherwise

**Model:**

$$\min \quad \sum_{j\in J} c_j o_j + \sum_{j\in j}\sum_{s\in S}\left(\lambda x_{js} + \delta com_s d_s p_{sj}\right) + \sum_{j\in J}\sum_{k\in K}\sum_{n\in N}\delta e_{nsj} type_{ns} d_s + \sum_{j\in J}\eta dep_s d_s \tag{7.1}$$

subject to

$$\sum_{j\in J} p_{sj} = 1 \qquad \forall s\in S \tag{7.2}$$

$$p_{sj} \le x_{js} \qquad \forall s\in S, \forall j\in J \tag{7.3}$$

$$\sum_{j \in J} e_{nsj} = 1 \qquad \forall s \in S, \forall n \in N \qquad (7.4)$$

$$e_{nsj} \le x_{js} \qquad \forall s \in S, \forall n \in N, \forall j \in J \qquad (7.5)$$

$$o_j \ge x_{js} \qquad \forall s \in S, \forall j \in J \qquad (7.6)$$

$$PT_j = \sum_{s \in S} \theta p_{sj} com_s + \sum_{n \in N} \sum_{s \in S} e_{nsj} type_{ns} t_n \quad \forall j \in J \qquad (7.7)$$

$$\sum_{n \in N} \sum_{s \in S} e_{nsj} type_{ns} + \sum_{s \in S} p_{sj} com_s \le L \qquad \forall j \in J \qquad (7.8)$$

$$g_i \ge u_i + \alpha \qquad \forall i \in I \qquad (7.9)$$

$$g_i \ge g_{i-|A|} + \beta + \alpha \qquad \forall i \in I, i > |A| \qquad (7.10)$$

$$r_j \ge \mu_{si} g_i - M(1 - x_{js}) \qquad \forall s \in S, \forall i \in I, \forall j \in J \qquad (7.11)$$

$$\sum_{b \in B} \sum_{k \in K} q_{jbk} = o_j \qquad \forall j \in J \qquad (7.12)$$

$$1 \le \sum_{j \in J} \sum_{k \in K} q_{jbk} \le \sum_{j \in J} o_j / |B| \qquad \forall b \in B \qquad (7.13)$$

$$\sum_{j \in J} q_{jbk} \le 1 \qquad \forall b \in B, \forall k \in K \qquad (7.14)$$

$$q_{jbk} \le \sum_{j' \ne j \in J} q_{j'bk-1} \qquad \forall j \ne j' \in J, \forall b \in B, \forall k \in K, k > 1 \qquad (7.15)$$

$$h_{jbk} \ge r_j - M(1 - q_{jbk}) \qquad \forall j \in J, \forall b \in B, \forall k \in K \qquad (7.16)$$

$$h_{jbk} \ge h_{j'bk-1} + PT_{j'} - M(1 - q_{jbk}) \qquad \forall j \ne j' \in J, \forall b \in B, \forall k \in K, k > 1 \qquad (7.17)$$

$$y_j \ge h_{jbk} + \sum_{s \in S} \theta p_{sj} com_s + \sum_{n \in N} \sum_{s \in S} e_{nsj} type_{ns} t_n \qquad \forall j \in J \qquad (7.18)$$

$$dep_s \geq y_j - M(1 - x_{js}) \qquad \forall s \in S, \forall j \in J \qquad (7.19)$$

$$y_j + \sum_{s \in S} 2d_s x_{js} + \omega\left(\sum_{s \in S} x_{js} - 1\right) \leq D \qquad \forall j \in J \qquad (7.20)$$

$$0 \leq p_{sj} \leq 1 \qquad \forall s \in S, \forall j \in J \qquad (7.21)$$

$$0 \leq e_{nsj} \leq 1 \qquad \forall s \in S, \forall n \in N, \forall j \in J \qquad (7.22)$$

$$g_i \geq 0 \qquad \forall i \in I \qquad (7.23)$$

$$r_j \geq 0 \qquad \forall j \in J \qquad (7.24)$$

$$PT_j \geq 0 \qquad \forall j \in J \qquad (7.25)$$

$$h_{jbk} \geq 0 \qquad \forall j \in J, \forall b \in B, \forall k \in K \qquad (7.26)$$

$$dep_s \geq 0 \qquad \forall s \in S \qquad (7.27)$$

$$y_j \geq 0 \qquad \forall j \in J \qquad (7.28)$$

$$x_{js} \in \{0,1\} \qquad \forall s \in S, \forall j \in J \qquad (7.29)$$

$$o_j \in \{0,1\} \qquad \forall j \in J \qquad (7.30)$$

$$q_{jbk} \in \{0,1\} \qquad \forall j \in J, \forall b \in B, \forall k \in K \qquad (7.31)$$

In the above formulation, (7.1) is the objective function. The first term $\sum_{j \in J} c_j o_j$ denotes the total cost to operate the trailers. The second term $\sum_{j \in j} \sum_{s \in S} \left(\lambda x_{js} + \delta com_s d_s p_{sj}\right)$ has two components: The first component is the penalty for using each truck and this leads to reduction of the number of trucks used per store. That is, any store is served by two trucks only if their order exceeds one truck load. The second component and the third term $\sum_{j\in J}\sum_{k\in K}\sum_{n\in N}\delta e_{nsj} type_{ns} d_s$ are the total transportation cost of the pallets. The fourth term $\sum_{j\in J}\eta dep_s d_s$ is the cost of weighted distance, and by decreasing the value of this term the stores with large distances get served first. Also, note that only a part of the objective function is the actual cost for transporting the pallets and the other costs are added to avoid some situations such as: sending two trucks to the same store when it is not necessary.

Constraints (7.2) and (7.4) ensures that the total number of pallets from combo and other types of products served by all outbound trucks to a store is equal to the demand of the store, (7.3) and (7.5) make sure a pallet is served to a store by an outbound truck only if that outbound truck serves that store. Constraints (7.6) indicate that an outbound truck is used only if it serves a store. Constraints (7.7) give the total processing time for each outbound truck and constraints (7.8) say that the total number of pallets loaded to an outbound truck does not exceed the limit of the truck.

From constraints (7.9) the completion time of unloading an inbound truck $i$ is greater than the sum of the arrival time and processing time. From (7.10) the completion time of unloading an inbound truck $i$ is greater than the completion time of the previous truck served at the same inbound door. Constraints (7.11) make sure that an outbound truck is ready to load only when all the pallets needed are available to load. (7.12), (7.13), (7.14) and (7.15) are constraints to schedule the door for an outbound truck, (7.16) and

(7.17) give the time to start loading an outbound truck at a particular order. Constraints (7.18) and (7.19) give the departure time for an outbound truck and the latest departure time for a store. Constraint (7.20) takes care of the latest time a truck can arrive at a store. Constraints (7.21) to (7.28) are non-negativity constraints and the rest of the constraints are binary constraints.

## 7.4 EXAMPLES AND RESULTS

### 7.4.1 Example One

In order to illustrate our model first we employed a small test problem with six stores, six available outbound trailers, two inbound doors, and two outbound doors. GAMS/CPLEX 11.2.0 was employed to arrive at the following results. We used the following data:

$\alpha = 10$, $\beta = 5$, $\gamma = 10$, $\delta = 0.07$, $\lambda = 100$, $\eta = 0.01$, $\theta = 3$, $\omega = 60$, $D$=400, $L$=28, $M$ =10000.

$d_s$ : 15, 8, 25, 75, 35, 40 $u_i$ : 0, 5, 30 $c_j$ : 180 $t_n$ : 2, 1, 3

**Table 7.1:** $type_{ns}$ Demand of store *s* in number of pallets of perishable product of type *n*

| Types\Store | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Combo | 8 | 5 | 7 | 2 | 6 | 2 |
| Type 1 | 10 | 3 | 7 | 2 | 4 | 6 |
| Type 2 | 7 | 4 | 2 | 5 | 1 | 0 |
| Type 3 | 6 | 5 | 3 | 4 | 2 | 1 |

**Table 7.2:** $f_{si}$ Number of pallets for store *s* in inbound truck *i*

| Truck\Store | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 7 | 2 | 0 | 0 |
| 2 | 8 | 0 | 0 | 0 | 0 | 2 |
| 3 | 0 | 4 | 0 | 0 | 6 | 0 |

The results are as follows:

It took one minute to run this data. The objective function value - the minimum cost, is $1829. Four outbound trailers were used and there were two trucks serving store one since the demand was more than 28 pallets. The following table shows the stores served by each truck. Trucks 5 and 3 were served consecutively at door 1 and trucks 2 and 1 were served consecutively at door 2.

**Table 7.3:** Outbound trucks used and stores served by each truck deduced from $x_{js}$

| Truck | 1 | 2 | 3 | 5 |
|---|---|---|---|---|
| Stores served | 1 | 3, 6 | 1, 2 | 4, 5 |

### 7.4.2 Example Two

Next we employed the real-life data for the demand from ten stores. There were ten outbound trailers, two inbound doors and four outbound doors available. We used the following data:

$\alpha = 10$, $\beta = 5$, $\gamma = 10$, $\delta = 0.07$, $\lambda = 100$, $\eta = 0.01$, $\theta = 3$, $\omega = 60$, $D$ = 900, $L$ = 28, $M$ =10000.

$d_s$ : 15, 8, 250, 75, 35, 180, 63, 120, 220, 55 $u_i$ : 0, 5, 30, 45 $c_j$ : 180

$t_n$ : 2, 1

**Table 7.4:** $type_{ns}$ Demand of store *s* in number of pallets of perishable product of type *n*

| Types\Store | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Combo | 3 | 5 | 7 | 0 | 4 | 0 | 4 | 5 | 4 | 5 |
| Type 1 | 12 | 7 | 16 | 9 | 15 | 11 | 11 | 8 | 10 | 9 |
| Type 2 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |

**Table 7.5:** $f_{si}$ Number of pallets for store *s* in inbound truck *i*

| Truck\Store | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 |
| 2 | 3 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 4 | 0 | 4 | 0 | 2 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 5 |

The results are as follows:

It took about fifteen minutes to run the data for ten trucks in GAMS. The objective function value - the minimum cost is $4300.87. Eight outbound trailers were used and there were two trucks serving two stores each. No store was served by two trucks, since the demand was less than 28 pallets. The maximum *k* was assigned as three. But since there were only eight trucks used, each outbound door served two trucks. The following table shows the stores served by each truck. Trucks 9 and 10 were served consecutively at door 1, trucks 5 and 4 were served consecutively at door 2, trucks 3 and 7 were served consecutively at door 3 and trucks 8 and 2 were served consecutively at door 4.

**Table 7.6:** Outbound trucks used and stores served by each truck deduced from $x_{js}$

| Truck | 2 | 3 | 4 | 5 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|
| Stores served | 5 | 7 | 9 | 4, 6 | 1, 2 | 3 | 8 | 10 |

### 7.4.3 Example Three

We used one more data set from H-E-B to test the problems. These are for 11 different stores and the distances and the demand vary from example 7.4.2.

$d_s$ : 278, 281, 255, 278, 257, 241, 244, 250, 250, 231, 239 $D$ = 1200

**Table 7.7:** $type_{ns}$ Demand of store *s* in number of pallets of perishable product of type *n*

| Types\Store | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Combo | 7 | 5 | 0 | 3 | 0 | 5 | 4 | 5 | 4 | 0 | 5 |
| Type 1 | 16 | 7 | 11 | 12 | 6 | 8 | 11 | 9 | 5 | 4 | 12 |
| Type 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

**Table 7.8:** $f_{si}$ Number of pallets for store *s* in inbound truck *i*

| Truck\Store | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 |
| 2 | 7 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 5 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 4 | 0 | 0 |

All other parameters are the same. The results are as follows:

It took about twenty minutes to run the data for eleven trucks in GAMS. The objective function value - the minimum cost is $7021.44. Eight outbound trailers were used and there were three trucks serving two stores each. No store was served by two trucks, since the demand was less than 28 pallets. The maximum *k* was assigned as three. But since there were only eight trucks used each outbound door served two trucks. The following table shows the stores served by each truck. Trucks 5 and 3 were served consecutively at door 1, trucks 2 and 4 were served consecutively at door 2, trucks 7 and

8 were served consecutively at door 3, and trucks 1 and 10 were served consecutively at door 4.

**Table 7.9:** Outbound trucks used and stores served by each truck deduced from $x_{js}$

| Truck | 1 | 2 | 3 | 4 | 5 | 7 | 8 | 10 |
|---|---|---|---|---|---|---|---|---|
| Stores served | 8 | 4 | 6, 9 | 7 | 2, 10 | 3, 5 | 11 | 1 |

### 7.5 Uncertainties in Combined Cross-dock Operation

There are a lot of uncertainties that happen in H-E-B cross-dock operations. Thursdays are always very busy for them since they send products to their stores before the weekend. The products need to be there by Friday at noon. Similarly, Sundays are also heavy work days. One of the days I observed the H-E-B cross-dock operations happened to be the Thursday before Easter, so they already had heavy demand on all products, especially floral goods. The cross-dock was to be closed for Easter on Saturday night and Sunday. I went to the Combo around 2:30 pm. Due to heavy demand, the logistics department delayed their completion time. In turn, at the Combo warehouse the operations were delayed and all the workers were waiting for about an hour. When I reached the PDC from the Combo to observe their cross-dock at 6:30 pm, their workers were getting ready to load. But they had to wait for another hour to start loading since the trucks were not coming from the Combo. There were 22 loaders and all 44 outbound doors were open on that day at the PDC. Each loader loaded two doors at a time using two forklifts and four workers.

The delay in the inbound trucks from the Combo causes a delay in loading and a delay in the departure time of the outbound trucks. According to the cross-dock manager at the PDC, this situation is frequently encountered and a large volume of products to be shipped is not the only factor. Another factor for the lateness of these Combo trucks is that the first few trucks leave the Combo during the evening peak traffic time. Also, the Combo ships other products directly from the warehouse, and they need to load those trucks simultaneously. So it is very common for the trucks from the Combo to arrive late. Thus the major uncertainty in PDC operations is the arrival time of the inbound trucks. We can model these in a robust approach.

### 7.6 Discussion on Modeling Under Uncertainty

Next, we discuss if we can formulate a robust model addressing the uncertainty in the arrival time of the inbound trucks using the approach of Bertsimas and Sim (2004). In the formulation of the (*CCC*) problem, we have $u_i$ as the arrival time of inbound truck *i.* The constraint using the arrival time of inbound trucks is

$$g_i \geq u_i + \alpha \qquad \forall i \in I \tag{7.9}$$

where $g_i$ is the time at which unloading of inbound truck *i* is completed and $\alpha$ is the fixed time to unload an inbound truck. The data uncertainty affects a lot of other variables in the formulation. The variables $g_i$ affect $r_j$ that in turn affect $h_{jbk}$, $y_j$ and $dep_s$. As we can see $u_i$ is not a coefficient of any variable, and we will not be able to use the protection-based approach of Bertsimas and Sim. In the scenario-based method, we could try using $g_i \geq u_i^{\tau} + \alpha, \forall i \in I, \forall \tau \in \mathrm{T},$ where $\mathrm{T}$ is the set of scenarios in (7.9). The size of the problem may increase drastically, so that task is left as a future exercise.

A mathematical model and formulation for a real life logistics problem based on empirical observations is presented in this Chapter. This formulation was tested and demonstrated using a small example and real-life sample data. In the future, we plan to test the model with comprehensive data from H-E-B including uncertainty. Also, applying decomposition methods will be considered to solve large-scale instances of the problem.

# Chapter 8: Conclusions and Future Research

Cross-docking is used in many industries as a cost cutting measure because it reduces storage and inventory cost together with other costs. Cross-dock operations are prone to uncertainty. Several uncertainties that happen in cross-dock area include delay in inbound trucks, number of working machines and available number of workers. Transportation delays are inevitable due to congestion or incidents. These uncertainties have been accounted for in the problems discussed in this dissertation. The main contribution of this research is applying uncertainty in cross-dock problems and in practice. A simple unloading problem in cross-dock was considered and two different robust approaches, as well as a meta-heuristic approach were applied. Optimization of cross-dock operations in two Texas industries with their major uncertainty was discussed.

First in chapter 3, a scenario-based robust optimization model for the unloading problem using a min max objective is presented with examples. The model optimizes the worst case solution. The polynomially solvable *TWC* problem becomes NP-Complete as the problem grows quickly with the increase in number of scenarios. A surrogate heuristic procedure is used to find a robust solution. Along the way, the heuristic procedure gives a lower bound and an upper bound for this problem. This solution method is very conservative. Next, a meta-heuristic procedure is applied to the unloading problem. For this dissertation we use a two-space genetic algorithm combined with the bounds obtained by the heuristic procedure in Chapter 3. The results are much closer to the optimal solution than the results obtained by the two-space genetic algorithm without bounds. When compared with the regular GA combined with bounds, two-space algorithm performs well considering it only searches 50% of the scenarios. Another protection-based approach is considered in which the limit on the number of coefficients allowed to change with data uncertainty gives a protection against the degree of conservatism. Though there are some sudden jumps in the nominal solution and variance, overall this robust approach works well for the problem.

The management of trucks and reduction of overtime pay in the cross-dock operations of Central Texas Goodwill Industries is addressed through two models and uncertainty is applied to those models. Real stores and distances are used along with random data. A combined cross-dock operation model together with demand is formulated and the uncertainties are discussed for H-E-B operations. Real data is used to test the model. The model takes care of product allocation, assignment of trucks to doors, the order in which the trucks are served, trucks to serve stores and uses a time constraint.

This dissertation does not address the recycling operation within the cross-dock of GICT, but a model can be developed to consider that. The H-E-B model is complex and it can be decomposed in the future. We have not done sensitivity analysis on these applications. The methods used in this dissertation do not address the uncertainty in H-E-B data and different approaches can be developed to solve problems involving uncertainties. This dissertation encourages more research to consider uncertainty in different types of cross-dock problems.

# References

Agustina, D., Lee, C.K.M. and Piplani R. (2010). A Review: Mathematical models for cross docking planning. *International Journal of Engineering Business Management*, Vol. 2, No. 2, pp. 47-54.

Alvarez-Pérez, G.A., González-Velarde, J.L. and Fowler, J.W. (2009). Crossdocking—Just in time scheduling: An alternative solution approach. *Journal of the Operational Research Society,* Vol. 60, pp. 554-564.

Apte, U.M., and Viswanathan, S. (2000). Effective cross docking for improving distribution efficiencies. *International Journal of Logistics Research and Applications,* Vol. 3, No. 3, pp. 291-302.

Arabani, A.B., Ramtin, F. and Rafienejad, S. N. (2009). Applying simulated annealing algorithm for cross docking system. *Proceedings of the World Congress on Engineering and Computer Science,* Vol. II, WCECS, San Francisco, USA.

Bachlaus, M., Pandey, M.K., Mahajan, C., Shankar, R. and Tiwari, M.K. (2008). Designing an integrated multi-echelon agile supply chain network: a hybrid taguchi-particle swarm optimization approach. *Journal of Intelligent Manufacturing,* Vol. 19, No. 6, pp.747–761.

Ballou, R.H. (1999). *Business Logistics Management*, 4th ed (New Jersey, Prentice Hall).

Bartholdi J.J. III, and Gue, K.R. (2000). Reducing labor costs in an LTL crossdocking terminal. *Operation Research,* Vol. 48, No. 6, pp. 823-832.

Bartholdi, J.J., III and Gue, K.R. (2004). The best shape for crossdock. *Transportation Science*, Vol. 38, No. 2, pp. 235-244.

Bartholdi, J. J. III, Gue, K.R. and Kang, K. (2001). Staging freight in a cross dock. *Proceedings of the International Conference on Industrial Engineering and Production Management*, (May), Quebec City, Canada, 10 pages.

Ben-Tal, A. and A. Nemirovski. (2000). Robust solutions of linear programming problems contaminated with uncertain data. *Math. Programming,* Vol. 88, pp. 411–424.

Bertsimas, D. and Sim, M. (2003). Robust discrete optimization and network flows. *Math. Programming,* Ser. B, Vol. 98, pp. 49–71.

Bertsimas, D. and Sim, M. (2004). The price of robustness. *Operations Research,* Vol. 52, No. 1, pp. 35-53

Boysen, N. (2010). Truck scheduling at zero-inventory crossdocking terminals. *Computers and Operations Research,* Vol. 37, pp. 32 – 41.

Boysen, N., and Fliender, M. (2010). Cross-dock scheduling: Classification, literature review and research agenda. *Omega*, Vol. 38, No. 6, 2010, pp. 413-422.

Boysen, N., Fliedner, M. and Scholl, A. (2010). Scheduling inbound and outbound trucks at cross docking terminals. *OR Spectrum,* Vol. 32, pp. 135-161.

Bozer, Y.A. and Carlo, H.J. (2008). Optimizing inbound and outbound door assignments in less-than truckload crossdocks. *IIE Transactions,* Vol. 40, pp. 1007–1018.

Chen, F. and Lee, C.Y. (2009). Minimizing the makespan in a two-machine cross-docking flow shop problem. *European Journal of Operational Research*, Vol. 193, pp. 59–72.

Chen, F. and Song, K. (2009). Minimizing makespan in two-stage hybrid crossdocking scheduling problem. *Computers and Operations Research*, Vol. 36, pp. 2066-2073.

Chen, P., Guo, Y., Lim, A. and Rodrigues, B. (2006). Multiple crossdocks with inventory and time windows. *Computers and Operations Research*, Vol. 33, pp..43-63.

Chen, R., Fan, B. and Tang, G. (2009). Scheduling problems in cross docking. *Lecture Notes in Computer Science,* Vol. 5573 LNCS, pp. 421-429.

Daniels, R.L. and Kouvellis, P. (1995). Robust scheduling to hedge against processing time uncertainty in single machine production. *Management Science*, Vol. 41, pp. 363-376.

Derbes, D., Rabadi, D. and Musa, R. (2009). Transportation problem for a cross-dock network with time constraints. *Industrial Engineering Research Conference,* Miami, FL, pp. 1393-1398.

Donaldson, H., Johnson, E.L., Ratliff, H.D. and Zhang, M. (1998). Scheduled-driven cross-docking networks. *Georgia Institute of Technology Research Report*. http://www/isye.gatech.edu/research/files/misc9904.pdf

Duthie, J.C., Unnikrishnan, A. and Waller, S.T. (2011). Influence of demand uncertainty and Correlations on traffic predictions and decisions. *Computer-Aided Civil and Infrastructure Engineering,* Vol. 26, No. 1, pp. 16–29.

Fan, W. and Machemehl, R.B. (2008). Tabu search strategies for the public transportation network optimizations with variable transit demand. *Computer-Aided Civil and Infrastructure Engineering,* Vol. 23, No. 7, pp. 502–520.

Forger, G. (1995). UPS starts world's premiere cross-docking operation. *Modern Materials Handling*. November 1.

Goodwill Industries of Central Texas. http://www.austingoodwill.org/ems/txcounties.html (Last Accessed on Oct 24th 2011)

Gue, K.R. (1999). The effects of trailer scheduling on the layout of freight terminals. *Transportation Science,* Vol. 33, No. 4, pp. 419-428.

Gue, K.R. (2007). Warehouses without inventory. *International Commerce Review,* Vol. 7, No. 2, pp. 124-132.

Gümüs, M., and Bookbinder, J.H. (2004). Cross-docking and its implications in location-distribution system. *Journal of Business Logistics.*Vol.25, No.2, pp.199-228

Hauser, K. and Chuang, C.H. (2006). Genetic algorithms for layout optimization in crossdocking operations of a manufacturing plant. *International Journal of Production Research,* Vol 44, No. 21, 2006, pp. 4663-4680.

HEB Store information and history. Web http://www.heb.com/sectionpage/about-us/company/heb-history/3000002. (Last accessed on Oct 24th 2011).

Herrmann, J.W. (1999). A genetic algorithm for minimax optimization problems. *Proceedings of the 1999 Congress on Evolutionary Computation*, pp. 1099-1103, Washington, D.C., July 6-9.

Heragu, S.S., Du, L., Mantel, R.J. and Schuur, P.C. (2005). Mathematical model for warehouse design and product allocation. *International Journal of Production Research,* Vol. 43, No. 2, pp. 327-338.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor.

Huang, G.Q., Lau, J.S.K. and Mak, K.L. (2003). The impacts of sharing production information on supply chain dynamics: a review of the literature. *International Journal of Production Research,* Vol. 41, pp. 1483–1517.

Jayaraman, V. and Ross, A. (2003). A simulated annealing methodology to distribution network design and management. *European Journal of Operational Research,* Vol. 144, pp. 629-645.

Karoonsoontawong, A. and Lin, D. (2011). Time-varying lane-based capacity reversibility for traffic management. *Computer-Aided Civil and Infrastructure Engineering,* DOI: 10.1111/j.1467-8667.2011.00722.x

Khoury, J.E. and Hobeika, A. (2007). Incorporating uncertainty into the estimation of the passing sight distance requirements. *Computer-Aided Civil and Infrastructure Engineering*, Vol. 22, No. 5, pp. 347–357.

Kim, J. and Ok, C. (2008). Distributed feedback control algorithm for dynamic truck loading scheduling problem. *Applied Mathematics and Computation*, Vol. 199, No. 1, pp. 275-284.

Ko, C.S., Lee, H.K., Choi, E.J. and Kim, T. (2008). A genetic algorithm approach to dock door assignment in automated cross-docking terminal with restricted layout. *The 2008 International Conference on Genetic and Evolutionary Methods*, pp. 186-95.

Kouvellis P. and Yu, G. (1997). Robust discrete optimization and its applications. *Kulwer Academic Publications,* Boston.

Kreng, V.B. and Chen, F. (2008). The benefits of a crossdocking delivery strategy: a supply chain collaboration approach. *Production Planning & Control*, Vol. 19, No. 3, pp. 229-241.

Larbi, R., Alpan, G., Baptiste, P. and Penz, B. (2007). Scheduling of transshipment operations in a single strip and stack doors crossdock. *19th International Conference on Production Research ICPR*.

Larbi, R., Alpan, G. and Penz, B. (2009). Scheduling transshipment operations in a multiple inbound and outbound door crossdock. *International Conference on Computers and Industrial Engineering*, pp. 227-232.

Larbi, R., Alpan, G., Baptiste, P. and Penz, B. (2010). Scheduling cross docking operations under full, partial and no information on inbound arrivals. *Computers and Operations Research,* Vol. 38, pp. 889-900.

Lee, Y. H., Jung, J. W. and Lee, K. M. (2006). Vehicle routing scheduling for cross-docking in the supply chain. *Computers & Industrial Engineering*. Vol. 51, pp. 247–256.

Ley, S. and Elfayoumy, S. (2007). Cross Dock Scheduling Using Genetic Algorithms. *The 2007 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Jacksonville, FL, USA.

Li, Y., Lim, A. and Rodrigues, B. (2004). Crossdocking - JIT scheduling with time Windows. *Journal of the Operational Research Society*, Vol. 55, pp. 1342–1351.

Li, Z., Low, M.Y.H. and Lim, R.Y.G. (2009). Optimal decision-making on product allocation for crossdocking and warehousing operations. *International Journal of Services Operations and Informatics,* Vol. 4, No. 4, pp. 352-365.

Li, Z., Sim, C.H., Low, M.Y.H. and Lim, Y.G. (2008). Optimal product allocation for crossdocking and warehousing operations in FMCG supply chain. *IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI),* pp. 2963-2968

Lim, A., Ma, H. and Miao, Z. (2006). Truck dock assignment problem with time windows and capacity constraint in transshipment network through crossdocks. *Lecture Notes in Computer Science*, Vol. 3982 LNCS, pp. 688-697.

Lim, A., Miao, Z., Rodrigues, B. and Xu, Z. (2005). Transshipment through crossdocks with inventory and time windows. *Naval Research Logistics*, Vol. 52, pp. 724-733.

Lin, D., Valsaraj, V. and Waller, S.T. (2011). A Dantzig-Wolfe decomposition-based heuristic for off-line capacity calibration of dynamic traffic assignment. *Computer-Aided Civil and Infrastructure Engineering* Vol. 26, No. 1, pp. 1–15.

Louis Y., Tsui, L.Y. and Chang, C.H. (1990). A Microcomputer Based Decision Support Tool for Assigning Dock Doors in Freight Yards. *Proceedings of the 12th Annual Conference on Computers & Industrial Engineering*, Vol. 19, No. 1-4, pp. 309-312.

Maps of Goodwill store locations obtained from http://maps.google.com/

Maknoon, M.Y. and Baptiste, P. (2009). Crossdocking: Increasing platform efficiency by sequencing incoming and outgoing semi-trailers. *International Journal of Logistics Research and Applications*, Vol. 12, No. 4, pp. 249-261.

McWilliams, D.L. (2009). A dynamic load-balancing scheme for the parcel hub-scheduling problem. *Computers & Industrial Engineering,* Vol. 57, pp. 958-962.

McWilliams, D.L., Stanfield, P.M. and Geiger, C.D. (2005). The parcel hub scheduling problem : A simulation – based solution approach. *Computers & Industrial Engineering,* Vol. 49, pp. 393-412.

Miao, Z., Fu, K., Fei, Q. and Wang, F. (2008). Metaheuristic algorithm for the transshipment problem with fixed transportation schedules. *Lecture Notes in Computer Science*, Vol. 5027 LNAI, pp. 601-610.

Miao, Z., Lim, A. and Ma, H. (2009). Truck dock assignment problem with operational time constraint within crossdocks. *European Journal of Operation Research,* Vol. 192, pp. 105-115

Mula, J., Peidro, D., Madronero, M.D. and Vicens, E. (2010). Mathematical programming models for supply chain production and transport planning. *European Journal of Operation Research,* Vol. 204, No. 3, pp. 377-390

Musa, R., Arnaout, J.P. and Jung, H. (2010). Ant colony optimization algorithm to solve for the transportation problem of cross-docking network. *Computers & Industrial Engineering*, Vol. 59, pp. 85–92.

Napolitano, M. (2000). Making the move to cross docking. *Oak Brook, IL*: Warehousing Education and Research Council.

Ng, M.W., Lin, D. and Waller, S.T. (2009). Optimal long-term infrastructure maintenance planning accounting for traffic dynamics. *Computer-Aided Civil and Infrastructure Engineering,* Vol. 24, No. 7, pp. 459–469.

Ng, M.W., Park, J. and Waller, S.T. (2010). A hybrid bilevel model for the optimal shelter Assignment in Emergency Evacuations. *Computer-Aided Civil and Infrastructure Engineering,* Vol. 25, No. 8, pp. 547–556.

Ng, M.W**.,** Sathasivan, K. and Waller, S.T., (2011). Modeling human error and in-vehicle navigation systems: Model formulation and properties. *Proceedings of the Transportation Research Board's 90th Annual Meeting,* Washington, DC.

Ng, M.W. and Waller, S.T. (2011). A dynamic route choice model considering uncertain capacities. *Computer-Aided Civil and Infrastructure Engineering,* DOI: 10.1111/j.1467-8667.2011.00724.x

Peng, F., Kang, S., Li, X., Ouyang, Y., Somani, K. and Acharya, D. (2011). A Heuristic approach to the railroad track maintenance scheduling problem. *Computer-Aided Civil and Infrastructure Engineering,* Vol. 26, No. 2, pp. 129–145.

Pinedo, M. (2002). Scheduling Theory, Algorithms and Systems. Second edition, *Prentice-Hall*, New Jersey.

Pinedo M. (2005). Planning and scheduling in manufacturing and services. *Springer*, USA.

Ratliff, H.D., Vate, J.V. and Zhang, M. (1998). Network design for load-driven cross-docking systems. *Georgia Institute of Technology Research Report,* http://www/isye.gatech.edu/research/files/misc9914.pdf

Ross, A. and Jayaraman, V. (2008). An evaluation of new heuristics for the location of cross-docks distribution centers in supply chain network design. *Computers & Industrial Engineering.* Vol. 55, pp. 64-79.

Sathasivan, K., Ng, M.W. and Waller, S.T. (2011). A robust heuristic for scheduling the loading and unloading of trucks. *Proceedings of the Transportation Research Board's 90th Annual Meeting,* Washington, DC.

Shakeri, M., Yoke, M., Hean Low, M.Y. and Li, Z. (2008). A generic model for crossdock truck scheduling and truck-to-door assignment problems. *IEEE International Conference on Industrial Informatics*. pp. 857-864

Soltani, R. and Sadjadi, S.F. (2010). Scheduling trucks in cross-docking systems: A robust meta-heuristics approach. *Transportation Research Part E: Logistics and Transportation Review*, Vol. 46, No. 5, pp. 650-666.

Song, K. and Chen, F. (2007). Scheduling cross docking logistics optimization problem with multiple inbound vehicles and one outbound vehicle. *Proceeding of the IEEE International Conference and Automation and Logistics,* Jinan, China, pp. 3089-3094.

Soyster, A. L. (1973). Convex programming with set-inclusive constraints and applications to inexact linear programming. *Opeations Research,* Vol. 21, pp. 1154–1157.

Sung, C.S. and Song, S.H. (2003). Integrated service network design for a cross-docking supply chain network. *The Journal of the Operational Research Society,* Vol. 54, No. 12, pp. 1283-1295.

Sung, C.S. and Yang, W. (2008). An exact algorithm for a cross-docking supply chain network design problem. *Journal of the Operational Research Society,* Vol. 59, pp. 119 –136.

Syarif, A., Yun, Y.S, and Gen, M. (2002). Study on multistage logistic chain network : a spanning tree-based genetic algorithm approach. *Computers & Industrial Engineering,* Vol. 43, pp. 299-314.

Szeto, W.Y., Jiang, Y. and Sumalee, A. (2011). A cell-based model for multi-class doubly stochastic dynamic traffic assignment. *Computer-Aided Civil and Infrastructure Engineering*, DOI: 10.1111/j.1467-8667.2011.00717.x

Tang S. and Yan H. (2010). Pre-distribution vs. post-distribution for cross-docking. *Omega,* Vol. 38, pp. 192-202.

Tsui, L.Y. and Chang, C.H. (1992). An optimal solution to a dock door assignment problem. *Computers & Industrial Engineering,* Vol. 23, pp. 283-286.

Tsui, L.Y. and Chang C-H. (1990). A Microcomputer based decision support tool for assigning dock doors in freight yards. *Computer & Industrial Engineering*, Vol. 19, No. 1-4, pp. 309-312.

Ukkusuri, S. V., Mathew, T.V. and Waller, S.T. (2007). Robust transportation network design under demand uncertainty. *Computer-Aided Civil and Infrastructure Engineering,* Vol. 22, No.1, pp. 6–18.

Unnikrishnan, A., Valsaraj, V. and Waller, S.T. (2009). Stochastic and dynamic shipper carrier network design problem. *Networks and Spatial Economics,* Vol. 9, No. 4, pp. 525-550.

U.S. Department of Transportation. (2010, November). Research and Innovative Technology Administration, Bureau of Transportation Statistics, *State Summaries: 2007 Commodity Flow Survey,* (Washington, DC)

Vahdani, B. and Zandieh, M. (2010). Scheduling trucks in cross-docking systems: Robust meta-heuristics. *Computers & Industrial Engineering,*Vol. 58, pp.12-24.

Vis, I.F.A. and Roodbergen, K.J. (2008). Positioning of goods in a cross-docking environment. *Computers & Industrial Engineering*, Vol. 54, pp. 677-689.

Vogt, J.J. (2010). The successful cross-dock based supply chain. *Journal of Business Logistics,* Vol. 31, No. 1, pp. 99-119.

Wei, J., Leung, S.C.H., Lee, S.C. and Kwok, B. (2009). Simulation of RFID-enabled loading strategy for outbound logistics: A case study in Hong Kong. *International Conference on Computers & Industrial Engineering*, CIE 2009, pp. 1769 - 1774.

Wen, M., Larsen, J., Clausen, J., Cordeau, J.F. and Laporte, G. (2009). Vehicle routing with cross-docking. *Journal of the Operational Research Society*, Vol. 60, pp. 1708-1718.

Yan H. and Tang, S. (2009). Pre-distribution and post-distribution cross-docking operations. *Transportation Research* Part E, Vol. 45, pp. 843-859.

Yanchang, L. and Min, D. (2009). Discussion of cross-docking design in modern logistics systems. *WRI World Congress on Computer Science and Information engineering*, Vol. 3, pp. 149-151.

Yang, J. and Yu, G. (2002). On the robust single machine scheduling problem. *Journal of Combinatorial Optimization,* Vol. 6, pp. 17-33.

Yu, G. and Yang, J. (1998). On the robust shortest path problem. *Computers and Operations Research,* Vol. 25, pp. 457-468.

Yu, W. and Egbelu, P. J. (2008). Scheduling of inbound and outbound trucks in cross docking systems with temporary storage. *European Journal of Operations Research,* Vol. 184, No. 1, pp. 377-396.

Yu, W. (2002). Operational strategies for cross-docking systems. Iowa State University, Ph.D. Dissertation.

Zhang, T., Saharidis, G., Theofanis, S. and Boile, M. (2010). Scheduling of inbound and outbound trucks at cross-docks: Modeling and analysis. *Proceedings of the Transportation Research Board's 89th Annual Meeting,* Washington, DC.

# Vita

Kanthimathi Sathasivan was born in Chennai, Tamilnadu, India to T. M. Nellaiyappan and N. Rathinaparvathi. She received a Bachelor of Science in Mathematics from Madurai Kamaraj University, India. After coming to the United States of America, she completed a Master of Science in Computational and Applied Mathematics from Southern University, Baton Rouge and a Master of Arts in Mathematics from Louisiana State University, Baton Rouge. In 2008 she finished a Master of Science in Engineering in Operations Research and Industrial Engineering at The University of Texas at Austin prior to enrolling in the doctoral program in Transportation Engineering under the supervision of Dr. S. Travis Waller. She is an Adjunct Professor of Mathematics at Austin Community College. She lives in Austin with her husband Kanagasabapathi Sathasivan, daughter Kanaka and son Sarathi.

Email: kanthisatha@gmail.com

This dissertation was typed by the author.