Summer 2014

# Sugarcane harvest logistics

Kamal Lamsal
*University of Iowa*

# SUGARCANE HARVEST LOGISTICS

by

Kamal Lamsal

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Business Administration (Management Sciences)
in the Graduate College of
The University of Iowa

August 2014

Thesis Supervisors: Professor Philip C. Jones
Associate Professor Barrett W. Thomas

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

---

PH.D. THESIS

---

This is to certify that the Ph.D. thesis of

Kamal Lamsal

has been approved by the Examining Committee for the thesis requirement for the Doctor of Philosophy degree in Business Administration (Management Sciences) at the August 2014 graduation.

Thesis committee: ______________________________
Philip C. Jones, Thesis Supervisor

______________________________
Barrett W. Thomas, Thesis Supervisor

______________________________
Ann M. Campbell

______________________________
Pavlo Krokhmal

______________________________
Jeffrey W. Ohlmann

## ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my advisors Dr. Philip C. Jones and Dr. Barrett W. Thomas for the continuous support of my Ph.D study and research, for their patience and motivation. Their guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisors and mentors for my Ph.D study.

Besides my advisors, I would like to thank the rest of my thesis committee: Dr. Ann M. Campbell, Dr. Jeffrey W. Ohlmann and Dr. Pavlo Krokhmal, for their encouragement, insightful comments, and hard questions. I would also like to thank my fellow classmates Amit Kumar Verma, Michael Rechenthin, Guanglin Xu, Fahrettin Cakir, Preethi Issac, Huan Jin, Stacy Voccia, Shu Zhang, Wenjun Wang and Bhupesh Shetty for the stimulating discussions and for all the fun we have had in the last five years. I would like to thank John Deere for financial support. I am also grateful to Craig Wenzel and Brian Gilmore for their support of this research and their help in developing our knowledge of sugarcane harvests and logistics. I also thank Jean-Francois Cordeau for C++ code and Andrew Higgins for Fortran code that provided helpful insight for my own implementation. Last but not the least, I would like to thank my parents, Sophiya, Barbarra Carr and Prof. Nick Street for being there for me through this journey.

## ABSTRACT

Sugarcane is an important crop around the world. With each producing country having a different infrastructure, the logistics for each country are unique. In this work, we study the sugarcane harvest and transport operations in two major sugar producing nations – Brazil and the United States.

## TABLE OF CONTENTS

# LIST OF TABLES

Table

# LIST OF FIGURES

Figure

## LIST OF ALGORITHMS

Algorithm

# CHAPTER 1
# INTRODUCTION

Despite a growing global appetite for sugar as both a foodstuff and a fuel source there exists limited literature that explores sugarcane operations. In this paper, we look at harvest operations in the US and Australia which account for significant portions of the total sugarcane production costs in both countries. We develop a framework for coordinating harvest and transport of sugarcane to reduce the waiting time at the mill by maximizing the minimum gap between two successive arrivals at the mill. Our results show that arrivals can easily be coordinated to reduce truck waiting time at the mill.

Sugarcane is an important crop around the world. With each producing country having a different infrastructure, the logistics for each country are unique. In this paper, we focus on harvest operations in Louisiana in the United States. The key issue in Louisiana is the excessive queueing of trucks at the mill. The delayed turn around time disrupts harvest operations on farms and also increases the number of trucks required to haul the mill's daily quota of sugarcane. In this paper, we present a model that spreads the cane arrival throughout the day to match the mill's processing capacity. Our work generalizes previous work in the literature and better reflects the actual operation while maintaining tractability. We introduce new datasets based on the existing sugarcane infrastructure in Louisiana and present computational results that demonstrate our approach improves upon results in the literature.

Sugar mills in Brazil represent significant capital investments. To maintain appropriate returns on their investment, sugar companies aim to run the mills at full capacity over the entire nine months of the sugarcane harvest season. Because the sugar content of cane degrades considerably once it is cut, maintaining inventories of cut cane is undesirable. Instead, mills want to coordinate the arrival of cut cane with production. In Chapter 2, we present a model of the sugarcane harvest logistics problem in Brazil. We introduce a series of valid inequalities for the model, introduce heuristics for finding an initial feasible solution, and for lifting the lower bound. Computational results establish the effectiveness of the inequalities and heuristics. In addition, we explore the value of allowing trucks to serve multiple rather than single locations and demonstrate the value of permitting the harvest speed to vary.

In Chapter 3, we focus on harvest operations in Louisiana in the United States. The fundamental issue in Louisiana is the excessive queueing of trucks at the mill. Delayed turn around time disrupts harvest operations at farms and also increases the number of trucks required to haul the mill's daily quota of sugarcane. We present a model that spreads the cane arrival throughout the day to match the mill's processing capacity. Our work generalizes previous work in the literature and better reflects the actual operation while maintaining tractability. We introduce new datasets based on the existing sugarcane infrastructure in Louisiana and present computational results that demonstrate our approach improves upon results in the literature.

In Chapter 4, we build on the work in Chapter 3 and propose a different objective function for the US problem. In particular, we maximize the minimum gap between the consecutive mill arrivals. We believe this objective proxies the goal of reducing congestion at the mill and the number of trucks needed better than the other objectives proposed in the literature while also maintaining tractability. We have identified sets of valid inequalities that help to solve the model efficiently. We use the results from Chapter 3 to find initial feasible solutions.

In Chapter 5, we return to the problem of harvesting and delivery sugarcane to a mill in Brazil. This problem deals with harvesting operations (cutting and shipping) at the fronts, transportation by the trucks via road network, and unloading of the cane at the mill yard while accounting for the stochasticity in day-to-day operations.

In practice, the sugarcane harvest and transport are not deterministic, as we have described in Chapter 2. There are uncertainties in both operations. Harvesters break down at the fronts. The weather causes the shut down of harvesting at the front. The vehicles break down en-route to the mill or to the fronts. Many of these events need a change in the original plan of actions.

We develop a rolling horizon model that uses the math program in Chapter 2 to solve certainty equivalent problems as new information becomes available. We update constraints to account for changes in trucks or fronts.

# CHAPTER 2
# SUGARCANE HARVEST LOGISTICS IN BRAZIL

Sugar mills in Brazil represent significant capital investments. To maintain appropriate returns on their investment, sugar companies seek to run the mills at capacity over the entire nine months of the sugarcane harvest season. Because the sugar content of cane degrades considerably once it is cut, maintaining inventories of cut cane is undesirable. Instead, mills want to coordinate the arrival of cut cane with production. In this chapter, we present a model of the sugarcane harvest logistics problem in Brazil. We introduce a series of valid inequalities for the model, introduce heuristics for finding an initial feasible solution, and for lifting the lower bound. Computational results demonstrate the effectiveness of the inequalities and heuristics. In addition, we explore the value of allowing trucks to serve multiple rather than single locations and demonstrate the value of allowing the harvest speed to vary.

## 2.1 Introduction

Since 1989, the monthly global price of raw sugar has averaged US$ 0.1191 per pound. Starting in 2008, however, the price of raw sugar has steadily risen, peaking at US$ 0.3209 per pound in January of 2011. While prices have fallen from those highs, raw sugar was trading at US$ 0.2039 per pound in October 2012, the last month for which aggregate data is available [Economic Research Service, 2012]. There are two key sources of this rise in world sugar prices. First, consumption is increasing in countries such as India, China, Indonesia, and Turkey [McConnell et al., 2010].

Second, ethanol production diverts raw sugar from consumers. Between 2000 and 2010, world ethanol use increased by 300%, and as a result of increasing oil prices, economic growth, and new government mandates, the growth is expected to continue [Valdes, 2011]. Importantly, the recent lifting of an US import tariff on foreign-made ethanol led to a ninefold increase in US imports of Brazilian sugar-based ethanol in 2012. Additional growth is expected in 2013 [Wexler, December 17, 2012].

In the face of these high sugar prices, both consumers and producers have a keen interest in increasing world sugar supplies. As the world's largest exporter of both raw and refined sugar and the country whose production costs drive world sugar prices [McConnell et al., 2010], Brazil deserves particular focus. Yet surprisingly, the authors are not aware of any research that addresses logistics in the Brazilian sugar-cane industry, and the work from other countries does not apply due to significant industry differences.

The focal point of the Brazilian industry is its sugar mills that crush raw cane to extract the juice from which raw sugar is eventually made. Sugar mills represent significant capital investments, and to maintain appropriate returns on their investment, sugar companies seek to run the mills near or at capacity over the entire nine months of the sugarcane harvest season.

Running the mills at capacity requires an adequate supply a sugar cane. We define this adecute supply of truck loads of cane needed to keep the mill continuously running as mill needs. To overcome the challenges of coordinating harvest and transport operations with the mill needs, the obvious solution is to decouple the mill operation from the supply operation by carrying a stock of raw cane. However, raw cane presents a complication. Because of evaporation and bacterial growth, the sugar content of cut cane degrades considerably over time [Salassi et al., 2004, Saska et al., 2009, Saxena et al., 2010]. In areas of the world where cane is cut whole stalk, the cut stalk can last several days without significant lost of sugar content. However, in Brazil and some other countries, cane is primarily mechanically harvested, resulting in 12 - 18 inch billets of cane stalk. The multiple exposed ends increase the degradation in comparison to whole stalk cane [Salassi et al., 2004].

Because of the degradation, sugar producers want to reduce the cut-to-crush time, the time between when the sugar cane is cut in the field and when it is crushed at the mill. According to our conversations with our industry partners, a conservative estimate of average cut-to-crush time in a typical mill area in Brazil is three hours, even though the average travel time between the fronts and the mill is less than an hour (Personal Communication with Jose Coelho, Sugar Cane Segment Manager, John Deere, April 13, 2010). Improved logistics coordination, particularly coordinating the rate of harvest with the availability of trucks, offers an opportunity to reduce the cut-to-crush time and improve sugar yields while maintaining required service levels at the mill.

Estimating the cost of sugar loss is challenging. Sugar loss is affected by the cane variety, temperature, whether the cane is cut whole stock or billeted, whether it is burnt or clean, and the amount of debris entering the cane stalk during cutting and storage. These attributes differ by country, and only limited research explores the issue. As a rough estimate, for 2012/2013, Brazil is expected to crush 570 metric tons of sugarcane [Barros, 2012]. According to Saska et al. [2009], sugar loss per hour is linear over 24 hours and depends on the temperature range of the cane. Saska et al. [2009] finds that, for billeted cane (the case in Brazil) in the temperature range of 22-27 ℃ a reasonable range for temperatures during the harvest season in Brazil's largest sugar producing region, the center-south, is 0.03 tons per 100 tons of sugarcane per hour. At $0.20 per pound of sugar, the cost of sucrose loss alone is over $225 million and over $330 million if prices were again to rise near $0.30 per pound. However, if consider the possibility that the storage temperatures are much higher than the ambient temperature, notably over 27 ℃ the sugar loss increases to 0.32 tons per 100 tons of sugarcane per hour. In that scenario, the losses increase to $2.4 billion and $3.6 billion at $0.20 and $0.30 per pound of sugar, respectively. As noted in Salassi et al. [2004], dextran formation over time can additionally impact sugar value. Regardless, even small reductions in cut-to-crush time are likely to have an important monetary impact.

The sugarcane harvest is composed of three operations that must be coordinated: infield operations, over-the-road transport, and the mill operations. The infield operations usually occur in several pre-specified fronts a day. A front is a cluster of geographically close, but not necessarily contiguous, fields. The infield operations have several components. First, the cane is cut in the field, usually using a machine known as a harvester that processes the cane into uniformly sized billets (12-18 inches). While in operation, the harvester continuously feeds billets into an infield storage unit known as a cart. The cart is pulled by an infield transporter. This infield transporter and cart combination runs along side the harvester during the harvest operations, and when the cart is filled, the transporter and cart combination must be rotated with another infield vehicle and its associated cart to allow for continuous harvest operations. Filled carts are transported to an area known as the trans-loading zone that serves all of the fields in a front.

The second operation of the harvest begins at the trans-loading zone. At the trans loading zone in the fronts, the contents of the filled carts are transferred to over-the-road transport vehicles. These vehicles take the harvested cane from the fronts to the mill. The final operation of the harvest takes place at the mill where the over-the-road vehicles are unloaded. Once an over-the-road vehicle is unloaded, it can return to a front for its next load. Figure 2.1 illustrates the sugarcane harvest logistics problem. Note that, in practice, there are multiple fronts at varying distances from the mill even though only one representative front is shown in Figure 2.1.

In this chapter, we present a mixed linear integer programming model for the deterministic sugarcane harvest logistics problem in Brazil. The decision variables are the speed of harvesters at the fronts and the assignment of trucks to each of the loads produced at the fronts. Our objective is to minimize the cut-to-crush time of the sugarcane subject to the constraint that the mill is never starved of raw material. We present valid inequalities that allow us to strengthen the original formulation of the problem and also a heuristic for finding an initial feasible solution. In addition, we introduce a heuristic for lifting the lower bound of the linear relaxation. The

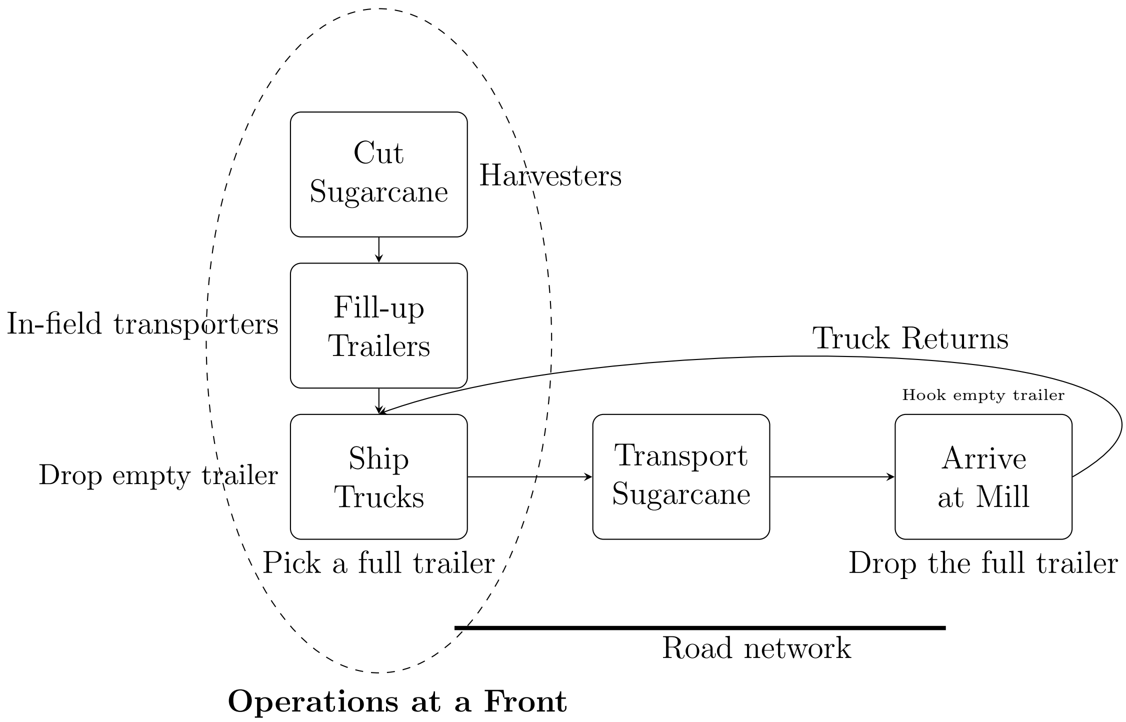


Figure 2.1: Mill Area Operation

result is particularly important in proving optimality as the linear relaxation of the original model often returns an objective value of zero. With the valid inequalities and heuristics, we demonstrate the ability to solve real-world sized problems in reasonable time. We demonstrate the value of our approach by contrasting it with the potentially managerially desirable approach of allowing trucks to serve a single front throughout the day. We also demonstrate the value of allowing the harvest rates to vary.

The rest of the paper is organized as follows. In Section 2.2, we survey related literature. Section 2.3 presents our math programming formulation. Section 2.4.2 presents the valid inequalities that strengthen the formulation, develops a heuristic for generating a feasible solution to the math program, and also introduces a heuristic for lifting the lower bound of the linear relaxation. Section 2.5 presents our experimental design and includes a description of the datasets on which we perform our computational experiments. The section also presents bounds on the number trucks needed to a serve a particular mill area. These bounds are useful in designing our experiments. In Section 2.6, we present the results of our computational experiments. Section 2.7 provides conclusions and presents future work.

## 2.2 Literature Review

Agricultural applications have a long history in the Operations Research literature, beginning with Heady [1954] that introduced the use of linear programming to the agricultural sector. Recent reviews of this broad field can be found in Lowe and Preckel [2004] and Ahumada and Villalobos [2009]. The supply of feedstock to sugar mills has garnered significant attention in the academic literature. Giles [2009] gives an overview of logistics issues stressing the need for the coordination between harvesting, transport, and storage for the smooth operation and the profitability of a sugar mill. Yet, to the best of the authors' knowledge, none of the literature addresses the degradation of sucrose during the cut-to-crush delay, a key concern in Brazil.

Most work on the sugarcane industry has been done in the last 15 years. The work can be more or less divided into value chain optimization and harvest scheduling. The only work related to the Brazilian sugarcane industry of which the authors' are aware focuses on value chain optimization. Kawamura et al. [2006] and Paiva and Morabito [2009] focus on production planning across multiple sugar mills, and da Silva et al. [2013] considers production planning at the individual mill level. Jena and Poggi [2013] present an optimization model for scheduling fields for harvest. Additional value-chain related literature includes Higgins et al. [1998], Grunow et al. [2008], and Kostin et al. [2011], which cover the Australian, Venezuelan, and Argentinian industries, respectively.

In this chapter, we focus on harvest and scheduling. This subset can be divided by country, notably Australia, South Africa, Brazil, Cuba and Thailand. There is very little similarity among the work from different countries because of differences in industry structure and operation. The harvesting practices and resultantly the transportation of the cane from the field is influenced by the terrain, the relative cost of labor and capital, the state of road and rail networks, and the distribution of sugarcane fields relative to the mill. In Brazil, in 2009, 70.0% of the cane supply was harvested from mill controlled fields, with 55.5% being mill owned [Neves et al., 2010], giving mill operators significant control over the supply process for the mill. In this situation, the mill does not face the problem of being oversupplied during the peak season as in Thailand nor is the issue of integrating the decision making process between growers and harvesters of the importance that it is in South Africa. Further, harvesting in Brazil is done 24 hours a day, and as a result, storage is not an issue as it is Australia, Cuba, and the United States.

Most closely related to the work presented here is Salassi et al. [2009a]. Salassi et al. [2009a] focus on sugar harvest logistics in Louisiana in the United States. As in the Brazilian case, each farm supplies a pre-determined number of truckloads each day. A key difference is that, once assigned to a farm, a truck serves loads at that farm until the harvest is completed at the farm. Further, harvest rates are fixed, and because harvesting only takes place over 14 hours in a day, the objective is designed to reduce truck congestion at the mill rather than to minimize cut-to-crush delay. One similarity is that Salassi et al. [2009a] groups the many farms harvested in a day into farm groups. These farm groups reduce the problem size as fronts do in the research presented here. Previously, Salassi et al. [2004] had explored the value of extending the hours of harvesting in the day. The goal was to reduce the degradation of cane resulting from storage.

In South Africa, the mill neither owns nor controls a big share of the farms. Hansen et al. [2002] develop a simulation model and conduct sensitivity analysis to investigate and reduce the delays in the South African sugarcane harvest and delivery systems. Their study shows that an integrated system comprising the harvest, transport, and mill process can lead to significant reductions in delay times and cost. Le Gal et al. [2009] develop a simulation to investigate the impact of increased mechanization of the sugarcane harvest.Lejars et al. [2008] also develop a simulation model to see the effects of centralized decision-making among the various stakeholders (sugar cane growers, harvesters, haulers, and millers) in the South African industry versus decentralized decision-making. McDonald et al. [2008] also develop a similar simulation model to simulate the sugarcane harvesting, transport and mill-yard activities for a mill supply area.

In Thailand, the sugar industry has a large number of small-sized, independent farms. This industry configuration causes uneven supplies throughout the harvesting season. Thus, research addresses supply issues to Thai sugar mills. Supsomboon and Yosnual [2004] present a stochastic model that helps mills optimize their order quantities, given the uncertainties of farmers' delivery lead times and quantities. Prichanont et al. [2005] use discrete-event simulation to demonstrate that the number of trucks should be reduced in order to avoid excess supply. They show that the excessive transportation cost is due to inefficient cane delivery truck utilization and extensive truck waiting time at the mill. Using the simulation model for one of the mills they studied, they show that as much as 600 of the existing 1000 trucks can be eliminated while the mill's needs remain statistically unchanged.

Diaz and Perez [2000] present a simulation model for the transportation of sugarcane in Cuba. For the same country, Lopez Milan et al. [2006] develop a linear programming model to pick up cane from different farms and storage locations to minimize transportation cost. The model accounts for more than one mode of transport. Besides road transport, they consider inter-modal transports which include first, transportation via trucks to warehouses, and then subsequent shipments to the mill by train. In Cuba, the train system is a cheaper alternative to road transport and also serves as a buffer because the mill operates for 24 hours a day while harvesting is done only 14 hours a day. Lopez-Milan and Pla-Aragones [2013] introduce a decision support system that builds on the work in Lopez Milan et al. [2006] . Unlike in this paper, neither Lopez Milan et al. [2006] nor Lopez-Milan and Pla-Aragones [2013] directly address the operational level scheduling of individual transport vehicles.

Historically, the Australian sugar industry has been the most exposed to the "world market price" because Australia neither has a large domestic demand like India nor access to protected high-priced European Union or United States domestic markets [Hildebrand, 2002]. To stay competitive in the market, the Australian sugar industry focused on lowering the transportation cost, and this is the main theme of the literature related to the Australian sugar industry. While some areas in the Australian sugar industry use rail transport [Higgins and Postma, 2004, Higgins and Davies, 2005, Higgins and Laredo, 2006], most relevant to the discussion in this paper is the use of truck transportation. Unlike Brazil, in Australia, harvesting is limited to daylight hours. Higgins [2006] develops an approach for scheduling the individual vehicles in a road-bound transportation system. Two alternative solution methods based on meta-heuristics are proposed for the model to solve problems of practical sizes. Both meta-heuristics were able to find solutions with an average reduction in vehicle queue time of about 90% compared to the manual methods used by the traffic officers in the mills. This helped reduce the required number of vehicles. Unlike the work presented in this paper, Higgins [2006] is concerned with reducing the waiting time for the trucks only at the unloading zone at the mill. We are concerned with decreasing the overall cut to crush time for the cane, while fulfilling the mill requirements at all times. We also have the option of controlling the harvester rate. Unlike Higgins, we provide an exact solution method.

## 2.3 Model

In this section, we present a linear, mixed integer programming formulation for coordinating sugarcane logistics harvests. For ease of exposition, we assume that all trucks start at the mill and harvesting has not yet started at the fronts. We also assume that the number of loads available at the fronts exactly matches the needs of the mill for the time horizon in question. We first introduce the notation for the problem and then present the math program.

### 2.3.1 Notation

$\mathcal{N}$**:** Set of needs at the mill, where the cardinality of $\mathcal{N}$ is $N$.

$\mathcal{T}$**:** Set of trucks available to service loads, where the cardinality of $\mathcal{T}$ is $T$.

$\mathcal{F}$**:** Set of fronts, where the cardinality of $\mathcal{F}$ is $F$.

$n_i$ : Time at which a truck load is needed at the mill to ensure the mill can continue operating at the desired rate, $i = 1, \ldots, N$.

$m$ : Time between two consecutive mill needs, assumed constant.

$d_i$ : Dispatch time to pick up the $i^{th}$ need, $i = 1, \ldots, N$.

$h_i$ : Time at which the harvest of the $i^{th}$ mill need is completed (ready time), $i = 1, \ldots, N$.

$p_i$ : Pick-up Time of the $i^{th}$ need, $i = 1, \ldots, N$.

$a_i$ : Arrival time of the $i^{th}$ load to the mill, $i = 1, \ldots, N$.

$\beta_f$ : Earliest time that harvesting can begin at front $f$, $f = 1, \ldots, F$.

$L_f$ : Number of loads to be produced from front f, $f = 1, \ldots, F$.

$t_f$ : Travel time to the front $f$, $f = 1, \ldots, F$.

$s_f$ : Travel time to mill from front $f$, $f = 1, \ldots, F$.

$r_f$ : Round trip time for front $f$, $f = 1, \ldots, F$.

$\underline{H}_f$ : Minimum time needed to harvest a load at front $f$, $f = 1, \ldots, F$.

$b_t$ : Earliest time time that truck $t$ can be dispatched, $t = 1, \ldots, T$.

$$x_{it} = \begin{cases} 1 & \text{if truck } t \text{ brings the } i^{th} \text{ need to the mill.} \\ 0 & \text{otherwise} \end{cases}$$

$$y_{if} = \begin{cases} 1 & \text{if the } i^{th} \text{ need is fulfilled by front } f \\ 0 & \text{otherwise} \end{cases}$$

### 2.3.2 Formulation

$$\textbf{(P)} \text{ minimize } \sum_{i \in \mathcal{M}} [(p_i - h_i) + (n_i - a_i)] \tag{2.1}$$

$$\text{subject to } d_i \le p_i \le a_i \le n_i \qquad , \forall\, i \in \mathcal{N} \tag{2.2}$$

$$h_i \le p_i \qquad , \forall\, i \in \mathcal{N} \tag{2.3}$$

$$\sum_{t \in \mathcal{T}} x_{it} = 1 \qquad , \forall\, i \in \mathcal{N} \tag{2.4}$$

$$\sum_{f \in \mathcal{F}} y_{if} = 1 \qquad , \forall\, i \in \mathcal{N} \tag{2.5}$$

$$d_i + \sum_{f \in \mathcal{F}} [t_f * y_{if}] \le p_i \qquad , \forall\, i \in \mathcal{N} \tag{2.6}$$

$$p_i + \sum_{f \in \mathcal{F}} [s_f * y_{if}] = a_i \qquad , \forall\, i \in \mathcal{N} \tag{2.7}$$

$$a_i - d_{i'} \le (2 - x_{it} - x_{i't}) \times M \qquad , \forall\, i, i' \in \mathcal{N},\ i' > i,\ \forall\, t \in \mathcal{T} \tag{2.8}$$

$$h_i + \underline{H}_f - h_{i'} \le (2 - y_{if} - y_{i'f}) \times M, \forall\, i, i' \in \mathcal{N},\ i' > i,\ \forall\, f \in \mathcal{F} \tag{2.9}$$

$$\sum_{i \in \mathcal{M}} y_{if} \le L_f \qquad , \forall\, f \in \mathcal{F} \tag{2.10}$$

$$d_t \ge b_t \qquad , \forall\, t \in \mathcal{T} \tag{2.11}$$

$$h_i - \sum_{f \in \mathcal{F}} \underline{H}_f y_{if} \ge \sum_{f \in \mathcal{F}} \beta_f y_{if} \qquad , \forall\, i \in \mathcal{N} \tag{2.12}$$

$$x_{it} \text{ binary }, y_{if} \text{ binary} \qquad , \ \forall\, i \in \mathcal{N},\ \forall\, f \in \mathcal{F},\ \ \forall\, t \in \mathcal{T}. \tag{2.13}$$

Equation (2.1) is the problem objective. We note that we can ignore travel time as, to maintain feasibility, all of the loads from each front must be transported to the mill and by assumption there are only enough loads available at each front such that the total is required to meet the mill's needs. Constraints (2.2) enforce the order of events associated with each load. For the load that fulfills the $i^{th}$ need, the dispatch time of the truck that services the $i^{th}$ need must occur before the actual pick up, and the load that fulfils the $i^{th}$ need at the mill should arrive to the mill before the $i^{th}$ need. Similarly, pickup at the front must occur before the arrival at the mill. Constraints (2.3) stipulate that harvesting for a load must be completed before it is picked up. Constraints (2.4) require that each load is picked up only once and constraints (2.5) say each load is harvested only once. Constraints (2.6) connect dispatch time and pick-up time, and the constraints (2.7) link pick-up time with arrival time at the mill. Constraints (2.8) say that, if two loads are picked up by the same truck, the dispatch time of the latter load is at least as large as the previous load's arrival time at the mill. Constraints (2.9) are the analogy of 2.8 when two loads are from the same front. i.e. if two loads are harvested at the same front, we can start harvesting the latter only after the first one is harvested. For both constraints (2.8) and (2.9), $M$ is a large number. Constraints (2.10) enforce the harvest cap on each front. Constraints (2.11) require that no truck is dispatched before it is available. Constraints (2.12) ensure that harvesting does not begin at a front before that front is available. Constraints (2.13) ensure binary decision variables.

## 2.4 Valid Inequalities, Initial Feasible Solutions, and Lifted Lower Bounds

While the model has obvious similarities to a constrained assignment problem, the addition of the variable harvest rates and thus variable ready times, creates challenges. Preliminary work demonstrated that it was not possible to solve instances of the above model using commercial solvers. In fact, it was often not possible to find even a feasible solution in a reasonable time. Thus, in this section, we present valid inequalities, a heuristic for finding a feasible solution, and a heuristic for lifting the lower bound of the linear relaxation. As we demonstrate in our computational results, these enhancements allow us to solve realistically-sized problems in reasonable time.

### 2.4.1 Valid Inequalities

In this section, we present results that help strengthen the formulation in the presented previously. We first note that we can strengthen the formulation by replacing Constraints (2.2) with

$$(d_i + \underline{t}) \leq (p_i + \underline{s}) \leq a_i \leq n_i, \ \forall \ i \ \in \ \mathcal{N}, \tag{2.14}$$

where $\underline{t} = \min_{f \in \mathcal{F}}[t_f]$ and $\underline{s} = \min_{f \in \mathcal{F}}[s_f]$.

Next, we state and prove a proposition that demonstrates the loads arriving to the mill can be processed in a first-in-first-out manner.

**Proposition 1 (FIFO Arrival Times).** *Let $a_i$ and $a_j$ denote the arrival times of loads $i$ and $j$, respectively, at the mill. Let load $i$ be assigned to the mill need occurring at time $n_i$ and let load $j$ be assigned to the mill need occurring at time $n_j$. If $a_i < a_j$*

*and the preceding assignment is feasible, then we may assume without loss of generality that* $n_i \leq n_j$.

*Proof.* Because a load cannot fill a mill need that occurs prior to its arrival, it follows that $a_i \leq n_i$ and $a_j \leq n_j$ as we have assumed feasibility. Note that the total waiting time for the two loads at the mill is $(n_i - a_i) + (n_j - a_j)$. With $a_i < a_j$, it follows that $a_i < n_j$. If it were the case that $n_j < n_i$, then it would also follow that $a_j < n_i$, and it would be possible to switch the loads so that load $i$ is assigned to mill need $n_j$ and load $j$ is assigned to mill need $n_i$. The total waiting time for the two loads would be $(n_j - a_i) + (n_i - a_j)$. Because the total waiting time is the same given either assignment, the proposition follows.

The result offers a way to break symmetry among the arrival of loads to the mill. Symmetry occurs when a group of variables forms a "symmetry group," a group of variables can be permuted without changing the value of the solution. As noted in [Margot, 2010], breaking symmetry can turn a computationally intractable problem into one that is easily solved. Consequently, we make use of the Proposition 1 and add the following constraints to the formulation:

$$a_i \leq a_{i+1}, \ \forall \ i \ \in \ \mathcal{N} \setminus N. \tag{2.15}$$

The next result bounds from below the arrival time values for a solution to **P**. To facilitate presentation, we first present additional notation. Let $\underline{H}_f$ be the minimum time required to harvest a load at front $f$ and let $\alpha_{fl}$ be the arrival time to the mill of the $l^{th}$ load harvested at front $f$. Then, for a front $f$, consider the sequence of arrival times such that $\alpha_{f1} = \beta_f + \underline{H}_f + s_f$, $\alpha_{f2} = (\beta_f + \underline{H}_f) + \underline{H}_f + s_f = \alpha_{f1} + \underline{H}_f$, and thus $\alpha_{fl} = \alpha_{f,l-1} + \underline{H}_f$ for $l = 2, \ldots, L_f$. Let $\mathcal{LB}_f$ be the ordered set of such arrival times for front $f$ and let $\mathcal{LB} = \cup_{f \in \mathcal{F}} \mathcal{LB}_f$, ordered in ascending order. We let $a'_i$ be the $i^{th}$ element in $\mathcal{LB}$. The following result follows directly from the construction of $\mathcal{LB}$ and Proposition 1.

**Proposition 2.** *The value $a'_i$ is a lower bound on $a_i$ for all $i \in \mathcal{N}$ resulting from a solution of $\boldsymbol{P}$.*

As a result of Proposition 2, we can add the following constraints to **P**:

$$a'_i \leq a_i, \; \forall \; i \; \in \; \mathcal{N}. \tag{2.16}$$

The following corollary follows from the construction of $\mathcal{LB}$ and Proposition 2.

**Corollary 1.** *If there exists a need $i \in \mathcal{N}$ such that $a'_i > n_i$, then $\boldsymbol{P}$ is infeasible.*

We next present an upper bound on the arrival time values for a solution to **P**. Again, we introduce new notation. Let $\eta_{fl}$ be the time at which harvesting of the $l^{th}$ load at front $f$ is completed and let $\hat{\alpha}_{fl}$ be the arrival time to the mill of the $l^{th}$ load harvested at front $f$. For each front $f \in \mathcal{F}$, let $\eta_{fL_f} = n_N - s_f$ and $\hat{\alpha}_{fL_f} = \eta_{fL_f} + s_f$. Then, $\eta_{fl} = \eta_{f,l+1} - \underline{H}_f$ and $\hat{\alpha}_{fl} = \eta_{f,l+1} - \underline{H}_f + s_f = \hat{\alpha}_{f,l+1} - \underline{H}_f$, for $l = 1, \ldots, L-1$. Let $\mathcal{UB}_f$ be the ordered set of such arrival times for front $f$ and let $\mathcal{UB} = \cup_{f \in \mathcal{F}} \mathcal{UB}_f$, ordered in ascending order. We let $a''_i$ be the $i^{th}$ element in $\mathcal{UB}$. The following result also follows directly from the construction of $\mathcal{UB}$ and Proposition 1.

**Proposition 3.** *The value $a''_i$ is an upper bound on $a_i$ for all $i \in \mathcal{N}$ resulting from a solution of $\boldsymbol{P}$.*

As a result of Proposition 3, we can add the following constraints to **P**:

$$a''_i > a_i, \ \forall \ i \ \in \ \mathcal{N}. \tag{2.17}$$

In addition to the FIFO result and bounds on the earliest and latest arrivals of loads to the mill, we can also fix some of the initial assignments of trucks to loads. Because we assume a homogeneous fleet of vehicles, we begin by noting that, without loss of generality, we can enforce:

$$x_{11} = 1 \text{ and} \tag{2.18}$$

$$x_{1t} = 0, \ \forall t \in \mathcal{T}, t \neq 1. \tag{2.19}$$

As a consequence of Proposition 2, we know that the earliest time at which truck 1 can possibly return back to the depot after picking up load 1 is $a'_1$. Based on this information, we introduce Proposition 4 that characterizes an initial set of truck assignments.

First, it is useful to define the following notation. Recall that the round trip time for front $f$ is $r_f$. Let $\rho = \min_{f \in F} r_f$. Define the set $\mathcal{S}$ such that $\mathcal{S} = \{i : i \in \mathcal{N}, n_i < a'_1 + \rho\}$.

**Proposition 4.** *Every $i$ in $\mathcal{S}$ requires a different truck.*

*Proof.* As $\rho$ is the smallest possible round trip time, having arrived back to the mill at the earliest at time $a'_1$, truck 1 cannot arrive back to the mill with its second load any earlier than $a'_1 + \rho$. Further, no truck serving a load $i$ such that $n_i < a'_1 + \rho$, the loads in $\mathcal{S}$ can return from serving its second load before $a'_1 + \rho$. Consequently, if a truck $t$ in $\mathcal{T}$ serves a load $i$ in $\mathcal{S}$, $t$ cannot also serve a load $i'$ in $\mathcal{S}$.

As a consequence of Proposition 4, we add the follow constraints to **P**, incorporating constraints (2.18) and (2.19):

$$x_{ii} = 1, \forall i \in \mathcal{S} \text{ and} \tag{2.20}$$

$$x_{ij} = 0,\ \forall i \in \mathcal{S},\ \forall t \in \mathcal{T}, t \neq i. \tag{2.21}$$

Even with the strengthened formulation, the lower bound provided by the linear relaxation of the math program above is not tight. We note that the value of the relaxation can be raised by solving a relaxed integer program that excludes the truck assignment constraints (constraints 2.4 and 2.8). This integer program is easier to solve than that presented above. The lower bound thus calculated gives the total wait if we could pick up all the loads at their ready times. Any constraint on number of trucks only increases the wait time.

### 2.4.2 A Heuristic for Generating a Feasible Solution

Starting a branch-and-bound procedure with an initial feasible solution often improves computation time. In this section, we develop a heuristic that is capable of generating feasible solution to problem **P**. The heuristic operates by decomposing problem **P** into its harvest and truck assignment components. While the heuristic does not guarantee a feasible solution, our computational experiments demonstrate that the method does so for all of our test cases and that the feasible solutions that are found are effective in improving computation time.

We begin by presenting an algorithm that generates a set of assignments of loads to fronts and corresponding arrival times to the mill. Our approach is motivated by the construction of the set $\mathcal{UB}$ in the previous section. It is straightforward to see that if the values in $\mathcal{UB}$ are feasible, then the arrival times in $\mathcal{UB}$ are optimal.

The algorithm can be found in Algorithm 2.1. The algorithm returns a set of assignments of fronts to mill needs, denoted $\bar{y}$, and a set of arrival times for each mill need $i$, denoted $\bar{a}$. Throughout, using the variable $l_f$, the algorithm tracks the number of loads at front $f$ that are remaining to be assigned. For each front $f$, the algorithm uses the value $\beta_f$ to track the next time at which a load from front $f$ can reach the mill.

The algorithm begins with the last mill need and assigns a front to meet that need and seeks to assign loads in descending order of the time at which the load is needed at the mill. To help maintain feasibility, before choosing a front to meet the $N^{th}$ and final need of the mill, we test all fronts $f$ to ensure that, if front $f$ is not assigned the $N^{th}$ load, then $f$ still has enough time to harvest all $L_f$ loads required by front $f$. To meet the $m^{th}$ need, the algorithm chooses the front that can deliver the load as close to $n_m$ as possible. In the case of the need to break a tie, the algorithm chooses the front $f$ that minimizes the ratio $\frac{n_m}{\underline{H}_f \times l_f}$, where $\underline{H}_f$ is the minimum time required to harvest a load at front $f$. The ratio is a measure of a front's flexibility to meet future loads and choosing the front with the maximum ratio chooses the least flexible front to break the tie. Once the assignment for the $m^{th}$ need is chosen, $\bar{a}$ and $\bar{y}$ are updated. The algorithm then updates the latest time at which the front chosen to fill the $m^{th}$ need can supply a feasible load to the mill. This updated time reflects the fact that load $m$ arriving at time $\bar{a}_m$ and supplied by front $f$ must have been harvested by time $\bar{a}_m - s_f$. So, the latest time at which front $f$ could finish harvesting its next load is $\bar{a}_m - s_f - \underline{H}_f$. Then, the latest time at which a load from front $f$ could arrive at the mill after supplying the $m^{th}$ load is $\bar{a}_m - s_f - \underline{H}_f + s_f = \bar{a}_m - \underline{H}_f$. In the next step, the algorithm updates the next available delivery time of all fronts to reflect the the time at which the $(m-1)^{st}$ need is required by the mill.

**Algorithm 2.1** Assignment of Fronts to Mill Needs

**Output:**
A vector of assignments of fronts to needs, $\bar{y}$, a vector a arrival times for the mill needs, $\bar{a}$
**Initialization:**
Set $m = M$, $\beta_f = n_N \forall f \in \mathcal{F}$, $l_f = L_f \forall f \in \mathcal{F}$, and $\bar{y}_{if} = 0 \forall i \in \mathcal{N}, f \in \mathcal{F}$
**while** $m \neq 0$ **do**
  $temp \leftarrow \underset{f \in F, l_f \neq 0}{\arg\max}\{\beta_f\}$
  $\bar{a}_m \leftarrow \beta_{temp}$
  $\bar{y}_{m,temp} \leftarrow 1$
  $\beta_{temp} \leftarrow \bar{a}_m - \underline{H}_{temp}$
  $\beta_f \leftarrow \min\{\beta_f,\ n_{m-1}\} \quad \forall f \in \mathcal{F}$
  $l_{temp} \leftarrow l_{temp-1}$
  $m \leftarrow m - 1$
**end while**

An alternative to Algorithm 2.1 is to solve a relaxation of the **P** that removes constraints 2.4 and 2.8. The same relaxation is discussed in the previous section for finding an improved root node bound. However, preliminary experiments found that the proposed algorithm more often leads to feasible and better solutions when coupled with the following truck assignment phase.

We next present an algorithm for assigning the loads in $\bar{a}$ to vehicles. From $\bar{a}$ and $\bar{y}$ returned by Algorithm 2.1, it is straightforward to compute a set of corresponding harvest completion times $\bar{h}$. The algorithm is presented formally in Algorithm 2.2. Throughout, the algorithm uses the value $\delta_t$ to represent the time at which vehicle $t$ is available to service its next load. For each load $i \in \mathcal{N}$, the algorithm chooses the vehicle that has been at the mill the longest. Ties are broken arbitrarily. Once a vehicle has been selected for assignment, the algorithm updates $\bar{x}$ accordingly, and then for the assigned vehicle $t$, the algorithm updates $\delta_t$ by computing the time at which the vehicle will return to the depot.

---

**Algorithm 2.2** Assignment of Vehicles to Harvested Loads

---

**Output:** Assignment of vehicles to loads, $\bar{x}$.
**Initialization:**
$\delta_t = 0 \forall t \in \mathcal{T}$ and $\bar{x}_{it} = 0 \forall i \in \mathcal{N}, t \in \mathcal{T}$
**for** $i = 1$ to $N$ **do**
  $temp \leftarrow \arg\min_{t \in \mathcal{T}} \{\delta_t\}$
  $x_{i,temp} \leftarrow 1$
  $\delta_{temp} \leftarrow \max\{\bar{h}_i, \delta_{temp} + \sum_{f \in \mathcal{F}} (\bar{y}_{if} + t_f)\} + \sum_{f \in \mathcal{F}} (\bar{y}_{if} + s_f)$
  $i \leftarrow i + 1$
**end for**

---

### 2.4.3 A Heuristic to Lift the Lower Bound

As the linear relaxation of the proposed model, which we call $\mathbf{P}'$, is often zero, proving optimality is challenging. We develop a heuristic that exploits the structure of the solution to raise the lower bound. For ease of exposition, we describe the heuristic with regard to the front-assignment variables, the $y$ variables. Algorithm 2.3 formally presents the heuristic. The algorithm for the truck-assignment variables, the $x$ variables, is analogous. A lower bound can be found by taking the minimum of the wait times computed from the two.

The method takes as input a solution to $\mathbf{P}'$, with assignment variables $y_{if}$, which we will refer to as $y'$ to represent that the binary condition has been relaxed. For each $f = 1, \ldots, F$, we let $\mathcal{R}_f = \{i \mid i \in \mathcal{N}, y'_{if} \notin \{0,1\}\}$, the set of mill needs for which the front $f$ fulfills a partial load in the solution to $\mathbf{P}'$. We assume that the set $\mathcal{R}_f$ is ordered from smallest to largest, and for algorithmic convenience, we assume that 0 is an element of $\mathcal{R}_f$ for every front $f$. Thus, given any two consecutive elements in $\mathcal{R}_f$, say $\mathfrak{r}_j$ and $\mathfrak{r}_k$, there exists a sequence of mill needs $i = \mathfrak{r}_j + 1, \ldots, \mathfrak{r}_k - 1$ such that $y'_{if} \in \{0, 1\}$.

Now, consider some sequence $i = \mathfrak{r}_j + 1, \ldots, \mathfrak{r}_k - 1$. Let $\bar{i}$ be the largest value in $i = \mathfrak{r}_j + 1, \ldots, \mathfrak{r}_k - 1$ such that $y'_{if} = 1$. Using an idea similar to that used in the heuristic for finding a feasible solution, we recognize that the latest possible time at which the harvest of the $\bar{i}^{th}$ mill need could be completed while still maintaining the feasibility of the solution is $n_{\bar{i}} - s_f$. Ignoring any constraints on vehicles, this completion time would also imply that the $\bar{i}^{th}$ mill need is satisfied with no wait.

Now, if it exists, we let $\bar{i}-1$ be the next largest $i = \mathfrak{r}_j+1,\ldots,\mathfrak{r}_k-1$ such that $y'_{if}=1$. Given the previous logic, the load $\bar{i}-1$ could not have been harvested any later than $n_{\bar{i}}-s_f-\bar{H}_f$. Thus, if front $f$ satisfied mill need $\bar{i}$ by harvesting the load at $n_{\bar{i}}-s_f$, the latest that front $f$ can harvest load $\bar{i}-1$ is $\min\{n_{\bar{i}}-s_f-\bar{H}_f, n_{\bar{i}-1}-s_f\}$. If the minimum is obtained by $n_{\bar{i}}-s_f-\bar{H}_f$, then the $(\bar{i}-1)^{th}$ load incurs a minimum wait of $n_{\bar{i}-1}-s_f-[n_{\bar{i}}-s_f-\bar{H}_f]$ time units. The correctness of the lifted lower bound follows from these feasibility arguments. The algorithm sums these waits over all fronts and all mill needs to compute the lifted lower bound.

## 2.5 Experimental Design

In this section, we describe the computational experiments designed to test our approach as well as to gain insight into the sugarcane harvest logistics problem. To aid the description of the experiments, we first describe how we determine the number of trucks that should be used in a dataset and then the datasets that we use.

### 2.5.1 Bounds on the Number of Trucks

As the number of available trucks is fixed in the short term, the number of trucks is a parameter in our sugarcane logistics model. We test our approach on a range of the number of trucks. It is obvious that, if there are too few trucks available, there is no feasible solution to our problem. The number of trucks necessary for feasibility depends on how far the fronts are from the mill and the times when the mill needs the loads. As the number of available trucks increases, we can find better solutions in that the average wait time per load decreases. A sufficient increase in the

**Algorithm 2.3** Heuristics to raise lower bound

**Output:** Lower bound to the integer solution
**Initialization:** $Wait = 0$
**Input:** A solution to $\mathbf{P}'$
**for** $f = 1$ to $F$ **do**
  **for** $k = 1, \ldots, \mid \mathcal{R}_f \mid$ **do**
    **if** $k \neq \mid \mathcal{R}_f \mid$ **then**
      $i \leftarrow \mathfrak{r}_{k+1} - 1$
    **else**
      $i \leftarrow N$
    **end if**
    $Flag \leftarrow$ TRUE
    **while** $i \neq \mathfrak{r}_k$ && $Flag =$ TRUE **do**
      $temp \leftarrow \infty$
      **if** $y'_{if} = 1$ **then**
        $Flag \leftarrow$ FALSE
        $Next \leftarrow n_i - s_f - \bar{H}_f$
        **for** $j = i - 1, \ldots, \mathfrak{r}_k + 1$ **do**
          **if** $y'_{jf} = 1$ **then**
            **if** $Next < n_j - s_f$ **then**
              $Wait \leftarrow Wait + n_j - s_f - Next$
              $Next \leftarrow Next - \bar{H}_f$
            **else**
              $Next \leftarrow n_j - s_f - \bar{H}_f$
            **end if**
          **end if**
        **end for**
      **end if**
      $i \leftarrow i - 1$
    **end while**
  **end for**
**end for**

number of trucks guarantees we can pick up all the loads at their ready times so each load arrives at the mill exactly when needed. Any further increase in the number of trucks beyond that point cannot further reduce the wait time.

Thus, we present bounds on the number of trucks required to achieve feasibility and the number of trucks necessary to achieve a wait time of zero. A corollary of the upper bound is that it guarantees a solution for which each truck need serve only one front. This number of trucks provides a managerially attractive solution, but requires more trucks to achieve the same level of average wait time per load than the alternative in which we allow trucks to serve multiple fronts.

We begin by bounding the number of trucks necessary for feasibility. The amount of truck time associated with each load $i$ depends on the front at which the load is harvested. For each truck t, $t$'s shift ends when the last load it transports arrives at the mill. Let $end_t$ be that time for truck $t$ such that $end_t = a_{(i|\text{i is the last load picked up by truck t})}$. The start time for the truck $t$ is denoted by $b_t$.

**Proposition 5.** *The number of trucks needed to fulfill all the mill needs at the mill need times, $\{n_1, n_2 \ldots n_N\}$, is bounded on the lower side by a positive number $k$ such that $k$ is the smallest integer satisfying $\sum\limits_{t \in 1 \ldots k} (n_{N-t+1}) \geq \sum\limits_{f \in \mathcal{F}} L_f r_f$.*

*Proof.* Because of the assumption that the total number of loads harvested across all fronts is exactly the number of loads needed to fulfill the mill needs, $\sum\limits_{f \in \mathcal{F}} L_f r_f$ is the same regardless of what need is served by what front. Further, we note that the minimum possible time needed to serve all loads, the time when the trucks do not wait at either the mill or the fronts, is the same sum $\sum_{f \in \mathcal{F}} L_f r_f$.

Then, for $k$ available trucks, the maximum available truck time is

$$\begin{aligned}
&\sum_{t \in 1 \ldots k} n_{(N-t+1)} \geq \sum_{t \in 1 \ldots k} n_{(N-t+1)} - \sum_{t \in 1 \ldots k} b_t \\
&\geq \sum_{t \in 1 \ldots k} a_{(N-t+1)} - \sum_{t \in 1 \ldots k} b_t = \sum_{t \in 1 \ldots k} end_t - \sum_{t \in 1 \ldots k} b_t \\
&= \sum_{t \in 1 \ldots k} (end_t - b_t).
\end{aligned}$$

The second inequality follows from the the fact that a load must arrive before the mill need time that it satisfies. The first equality holds by definition of $end_t$. The second equality rearranges terms and represents the time that $k$ trucks needed to serve all of the mill needs.

Thus, the smallest $k$ such that $\sum_{t \in 1 \ldots k} n_{(N-t+1)} \geq \sum_{i \in 1 \ldots N} \rho_i$ is the smallest number of trucks that could cover the minimum possible time needed to serve all loads.

If the mill needs are evenly spaced, we can rewrite $\sum_{t \in 1 \ldots k} n_{(N-t+1)}$ as $k \times \left( \frac{n_N + n_{(N-k+1)}}{2} \right)$.

We also note that, by treating each front as a separate mill area, Proposition 5 allows us to compute a lower bound on the number of trucks required to serve each front with dedicated vehicles. We call this separability. We formally present the result in Corollary 2. The proof follows directly from Proposition 5 and is omitted.

**Corollary 2.** *For $f \in \mathcal{F}$, let $k_f$ be the smallest integer that satisfies*

$$\sum_{t \in 1 \ldots k_f} n_{(N-t+1)} \geq L_f r_f.$$

*Then,* $k = \sum_{(f \in \mathcal{F})} k_f$ *is the minimum number of trucks that could possibly serve each front separably.*

We next present an upper bound on the number of vehicles required to serve the loads. The proof of the upper bound requires the realization that, if each load is picked up at its ready time, the solution can be improved only by changing the harvest times.

**Proposition 6.** *The upper bound on the number of trucks needed meet all mill needs among fronts is given by* $\hat{k}$ *such that* $\hat{k} = \sum_{(f \in \mathcal{F})} \hat{k}_f$ *where* $\hat{k}_f = \min\{L_f, \lceil \frac{r_f}{\bar{H}_f} \rceil\}$.

*Proof.* The shortest possible time between the ready times of two consecutive loads at front $f$ is $\bar{H}_f$. To serve at their ready times loads whose ready times differ by $\bar{H}_f$, it is clear that we need at least $\hat{k} = \lceil \frac{r_f}{\bar{H}_f} \rceil$ trucks. For any front, however, we never need more than $\hat{k}_f = L_f$ trucks, because $L_f$ trucks is enough to serve each load at front $f$ with its own vehicle. If each load is served by its own vehicle, then a vehicle is always capable to serving the load at its ready time.

### 2.5.2 Datasets

Our computational analysis uses several datasets generated based on our conversations with our industry partners (Personal Communication with Jose Coelho, Sugar Cane Segment Manager, John Deere, April 13, 2010; Personal Communication with Craig Wenzel, Staff Engineer, Worksite Systems and Productivity Group, John Deere, May 28, 2010). All of the data is available from `myweb.uiowa.edu/bthoa/iowa/Research.html`.

Each instance represents a typical mill area operations in Brazil . A mill area has four to eight fronts. We generate two mill areas each for four through eight fronts.

The front to mill travel time is on average one and half times higher than the mill to front travel time. The increase in return trip time reflects the impact of a load trailer. In each mill area, the closest front has the trip time between 30 minutes and 50 minutes whereas the farthest front has the round trip time between 70 minutes and 120 minutes.

Each front also has a harvest quota, the number of loads to be produced from the front in the given problem instance. The sum of the harvest quotas across all fronts in an instance is equal to the number of the total mill needs. The closest front serves 5% to 10% of the total mill needs and the farthest front serves between 35% and 60%.

The minimum time required to harvest a load at each front is based on the number of loads each front serves. We first compute the time that would elapse if loads from a front were evenly spaced. We then assume that the minimum time is 70% of that time. The 70% reflects the fact that, in practice, a capacity cushion is used as reactive capacity. Given the previously described harvest quotas, the closest front has the highest minimum time required to harvest a load. The minimum harvest time of the closest front ranges from 15 minutes to 50 minutes. The farthest front has the lowest minimum harvest time which ranges between 10 minutes and 30 minutes.

In all our instances, all the fronts and the trucks are allowed to start their operations 100 minutes before the the mill's first need. If the mill's first need is at the

$100^{th}$ minute, all the fronts can potentially start harvesting at $0^{th}$ minute and trucks are available to leave mill for a pick up at $0^{th}$ minute. We call this gap between when fronts can start harvesting and the time of mill's first need the "warm-up period." This warm-up period is analogous to having a setup time in manufacturing setting and eases the construction of feasible datasets.

Each geography is solved for four different inter-mill need times, three, four, five, and 10 minutes, for a total of 480 loads. These inter-mill need times reflect the range of values that might be encountered. For each instance and inter-mill need time combination, we also solve for the number of trucks in the range (lower bound to upper bound) as calculated in Section 2.5.1. Altogether, we generate 204 instances.

### 2.5.3 Experiments

In this chapter, we seek to address three questions. First, what is the computational value of the proposed valid inequalities and initial solution heuristic in finding optimal solutions in reasonable computation times? Two, what is the value of coordinating vehicles across fronts rather than assigning vehicles to specific fronts for the entire horizon? Third, what is the value of allowing variable harvest rates?

We address the first question by running our datasets using the initial model. We then add the valid inequalities and finally combine the valid inequalities with the initial feasible solution. We note that we tried various subsets of the valid inequalities, but were able to achieve provably optimal solutions in reasonable runtime only by using the entire set. The results of these experiments are discussed in Section 2.6.1.

Our second question is motivated by our review of the literature. We found that, in sugarcane harvesting and transportation operations, one common practice was to assign the trucks to a single front for the entire horizon. The practice arises particularly in countries in which the industry is not vertically integrated. In such countries, the growers are responsible for the transport of cane to the mill, and consequently growers employ a dedicated set of vehicles. In Brazil and as modeled in this chapter, however, the high level of vertical integration allows the coordination across the fronts. While coordination can reduce the number of trucks, separability decreases the managerial complexity of the operation. The results of these experiments are presented in Section 2.6.2.

Our third question is motivated by the knowledge, that in practice, harvesters are run at approximately 70% of their capacity. As noted previously, this extra capacity is used as reactive capacity, but on average this reactive capacity is unused. Thus, on average, the capacity cushion could also be used facilitate coordination, as we have modeled in this chapter. However, there is a managerial challenge to such coordination, particularly when decisions are being made without automated decision support.

To assess the value of varying harvest rates to reduce cut-to-crush time, we consider a case in which harvest rates are fixed. We create fixed harvest rates by setting the harvest time at each front to the average harvest time required to meet each front's quota of loads. This choice of harvest rates causes infeasibility in most instances. We overcome this issue by increasing the warm-up period. To determine this warm-up period, we iterated through warm-up times seeking the lowest warm-up time that achieved feasibility. We did not consider the cases where only a subset of fronts' warm-up time is increased. Consequently, the average wait time that we report for the fixed harvest rates is using the minimum warm-up period needed to meet all the mill needs. The results of these experiments are presented in Section 2.6.3.

## 2.6 Computational Results

This section presents the results of our computational experiments. We first demonstrate the value of our valid inequalities and the use of an initial feasible solution. We then explore the cost of the managerially attractive separable solution. Finally, we explore the value of variable harvest rates.

The math programs are solved using GUROBI OPTIMIZER 5.1. The experiments were performed on a 3.40 GHz Intel Core i7-3770 CPU running the Ubuntu 12.04 operating system. For all of the reported results, we implement the lifting heuristic described in Section 2.4.3 and Gurobi's relaxation-induced-neighborhood-search routine, called RINS. The lifting heuristic was implemented in C++ and communication with Gurobi was achieved through Gurobi's C++ Interface. In our initial experiments, these heuristics alone did not improve the performance of branch-and-bound, but proved valuable in proving optimality once the valid inequalities of Section 2.4.1 and the initial solution of Section 2.4.2 were implemented. The heuristics are run on all relaxed solutions for which 5% or fewer of the binary variables are fractional. The 5% value represents a compromise between the runtime of the heuristic and their value in reducing overall runtime. Branching was set to give priority to the front assignments or $y$ variables. All runs were terminated when the optimality gap was 1% or less.

### 2.6.1 Algorithmic Performance

Tables 2.1 through 2.4 present the results of the experiments testing the value of the valid inequalities and the initial feasible solution. As noted previously, without the valid inequalities, the instances rarely found a feasible solution and never return an optimal solution, even with significant runtime. Thus, the tables report the computation times for runs with just the valid inequalities and then runs with both the valid inequalities and an initial feasible solution. We label these computation times as "VI" and "Both," respectively. For each instance, the table also reports the average wait time per load. We also report the runtimes for each instance. In almost all cases, we were able to prove optimality with reasonable runtimes. Instances marked with "*" are instances for which we could not prove optimality even with 10,000 seconds of runtime. In those cases, the reported result is the best found feasible solution with the integer gap reported in the brackets after 1,000 seconds of runtime.

Finally, for some instances, such as instance 5a with 24 trucks, we do not report any values. In these cases, the reported number of trucks is the number of trucks that achieves the lower bound on the number of trucks for the instance. However, even after 10,000 seconds of runtime, we were unable to find a feasible solution for such instances. Given that the lower bound presented in Section 2.5.1 does not guarantee feasibility, we believe that it is likely that the lower bound is infeasible in these instances, but we were also unable to prove infeasbility. We mark these instances with "–".

The results in the tables demonstrate that valid inequalities alone are almost always able to achieve optimal solutions. In only ten cases out of 204 did the solver return a solution without being able to prove optimally in 10,000 seconds. Only four instances require more than 1,000 seconds to prove optimality.

In most cases, the addition of the initial feasible solution has a positive impact on runtimes. Using results for the instances with an inter-mill need time of 3, the initial feasible solution improves runtime by an average of almost 17% for the instances proved to optimality. We are also able to prove optimality for three cases for which it was not previously proven.

In terms of problem characteristics, run times increase in the number of fronts but decrease in the number of trucks. This result is not surprising. The problem size grows as the number of fronts grows, increasing runtime. However, the fronts are not the challenge in determining feasibility. Feasibility is driven by the trucks and their assignments. Thus, while the problem size also grows in the number of trucks, the computation time associated with the growth in problem size is overcome by the decrease that comes from the reduced challenge in finding feasible solutions and the resulting increase in fathoming in the branch-and-bound algorithm.

Finally, from a managerial perspective, it is also valuable to consider the trade-off between the number of trucks and the average wait of each load. As an example,

Figure 2.2 presents a graph of the average waiting time per load and the number of trucks. It is clear that most of the reduction in wait time comes from the addition of one truck over the number required for feasibility. This relationship is evident throughout the instances. Given the lower bound is analytically computable and generally feasible, the result offers a potential “rule of thumb” for planning purposes.

Table 2.1: Intermill-Needtime = 3, 480 loads

| Geography | # of trucks | Average wait | VI | Both |
|---|---|---|---|---|
| 4a | 29 | 25.00 | * (7.4 %) | * (6%) |
| | 30 | 22.63 | 728 | 600 |
| | 31 | 18.38 | 716 | 537 |
| | 32 | 11.225 | 599 | 454 |
| | 33 | 8.53 | 421 | 414 |
| | 34 | 4.79 | 297 | 274 |
| | 35 | 4.2 | 206 | 239 |
| | 36 | 1.36 | 193 | 181 |
| | 37 | 0.2 | 184 | 180 |
| 4b | 30 | 34.2 | * (4.5%) | * (4%) |
| | 31 | 30.1 | 928 | 680 |
| | 32 | 14.7 | 730 | 543 |
| | 33 | 5.03 | 695 | 435 |
| | 34 | 3.31 | 547 | 356 |
| | 35 | 0.5375 | 539 | 281 |
| | 36 | 0.2625 | 356 | 248 |
| | 37 | 0.09 | 310 | 217 |
| 5a | 24 | – | – | – |
| | 25 | 23.05 | 846 | 827 |
| | 26 | 19.15 | 784 | 741 |
| | 27 | 13.31 | 516 | 497 |
| | 28 | 9.375 | 425 | 408 |
| | 29 | 5.13 | 394 | 342 |
| | 30 | 0.10 | 341 | 325 |
| 5b | 16 | 17.02 | 909 | 855 |
| | 17 | 15.2 | 823 | 693 |
| | 18 | 7.6 | 735 | 672 |
| | 19 | 2.1 | 556 | 422 |
| | 20 | 0.5 | 556 | 523 |
| | 21 | 0.3 | 380 | 347 |
| 6a | 17 | 14.41 | 768 | 711 |
| | 18 | 9.78 | 832 | 645 |
| | 19 | 4.25 | 564 | 518 |
| | 20 | 1.20 | 510 | 454 |
| | 21 | 0.425 | 398 | 386 |
| | 22 | 0.25 | 319 | 366 |
| 6b | 17 | 17.39 | 660 | 646 |
| | 18 | 9.70 | 485 | 431 |
| | 19 | 4.17 | 470 | 422 |

Table 2.1 – Continued

| **Geography** | **# of trucks** | **Average wait** | **VI** | **Both** |
|---|---|---|---|---|
| | 20 | 0.41 | 400 | 356 |
| | 21 | 0 | 205 | 190 |
| 7a | 16 | 8.1 | * (5.6%) | * (2.4%) |
| | 17 | 6.98 | * (1.1%) | 956 |
| | 18 | 1.46 | 951 | 895 |
| | 19 | 1.10 | 813 | 608 |
| | 20 | 0.12 | 728 | 340 |
| | 21 | 0.033 | 567 | 361 |
| | 22 | 0 | 507 | 226 |
| 7b | 26 | – | – | – |
| | 27 | 6.27 | 946 | 811 |
| | 28 | 4.13 | 909 | 840 |
| | 29 | 1.56 | 871 | 729 |
| | 30 | 0.67 | 723 | 626 |
| | 31 | 0.31 | 701 | 597 |
| | 32 | 0.17 | 664 | 501 |
| 8a | 23 | 21.81 | * (7.5%) | * (2.6%) |
| | 24 | 12.3 | 992 | 889 |
| | 25 | 9.22 | 831 | 714 |
| | 26 | 6.19 | 982 | 648 |
| | 27 | 4.27 | 765 | 682 |
| | 28 | 1.29 | 722 | 561 |
| | 29 | 0.54 | 710 | 513 |
| | 30 | 0.44 | 557 | 340 |
| | 31 | 0.18 | 522 | 296 |
| 8b | 31 | 9.56 | 1536 | 1077 |
| | 32 | 6.725 | 937 | 918 |
| | 33 | 2.8 | 639 | 581 |
| | 34 | 1.70 | 599 | 394 |
| | 35 | 0.95 | 564 | 336 |
| | 36 | 0.67 | 465 | 432 |

Table 2.2: Intermill-Needtime = 4, 480 loads

| **Geography** | **# of trucks** | **Average wait** | **VI** | **Both** |
|---|---|---|---|---|
| 4a | 22 | 27.39 | * (5.3%) | * (2.2%) |
| | 23 | 7.51 | 924 | 876 |
| | 24 | 3.26 | 780 | 764 |
| | 25 | 2.84 | 543 | 510 |
| | 26 | 1.45 | 388 | 334 |
| | 27 | 0.416 | 306 | 276 |
| | 28 | 0.0625 | 289 | 254 |
| 4b | 23 | 26.32 | * (1.1%) | 1030 |
| | 24 | 10.9 | 952 | 873 |
| | 25 | 8 | 928 | 715 |
| | 26 | 5.07 | 837 | 680 |
| | 27 | 0.996 | 782 | 490 |
| | 28 | 0.375 | 650 | 406 |
| | 29 | 0.083 | 468 | 362 |
| | 30 | 0 | 355 | 227 |
| 5a | 18 | – | – | – |
| | 19 | 22.34 | * (1.9%) | * (1.7%) |
| | 20 | 16.28 | 471 | 401 |
| | 21 | 8.125 | 418 | 380 |
| | 22 | 4.08 | 391 | 358 |
| | 23 | 0.13 | 184 | 179 |
| 5b | 12 | 18.81 | 879 | 776 |
| | 13 | 7.56 | 742 | 708 |
| | 14 | 0 | 194 | 182 |
| 6a | 13 | 17.70 | 878 | 637 |
| | 14 | 11.28 | 707 | 581 |
| | 15 | 1.25 | 462 | 497 |
| | 16 | 0 | 256 | 201 |
| 6b | 13 | 10.90 | 985 | 917 |
| | 14 | 1.50 | 446 | 372 |
| | 15 | 0.19 | 258 | 229 |
| | 16 | 0 | 218 | 231 |
| 7a | 13 | 5.99 | * (3.2%) | * (1.6%) |
| | 14 | 4.01 | 1103 | 905 |
| | 15 | 2.64 | 845 | 706 |
| | 16 | 1.27 | 615 | 543 |
| | 17 | 0.96 | 638 | 566 |
| | 18 | 0.85 | 517 | 421 |
| 7b | 20 | 22.9 | * (1.5%) | 812 |

Table 2.2 – Continued

| Geography | # of trucks | Average wait | VI | Both |
|---|---|---|---|---|
| | 21 | 9.09 | 885 | 711 |
| | 22 | 1.85 | 685 | 547 |
| | 23 | 0.57 | 600 | 433 |
| | 24 | 0.29 | 610 | 306 |
| 8a | 17 | 25.05 | * (2.1 %) | * (1.8 %) |
| | 18 | 16.96 | 903 | 877 |
| | 19 | 4.97 | 850 | 762 |
| | 20 | 1.575 | 776 | 655 |
| | 21 | 0.85 | 680 | 519 |
| | 22 | 0.33 | 585 | 458 |
| | 23 | 0.21 | 617 | 364 |
| 8b | 21 | 30.86 | * (4.7 %) | * (3.8 %) |
| | 22 | 22.02 | 879 | 764 |
| | 23 | 11.18 | 709 | 676 |
| | 24 | 8.15 | 760 | 619 |
| | 25 | 2.47 | 469 | 246 |
| | 26 | 1.17 | 342 | 221 |
| | 27 | 0.09 | 305 | 197 |

Table 2.3: Intermill-Needtime = 5, 480 loads

| **Geography** | **# of trucks** | **Average wait** | **VI** | **Both** |
|---|---|---|---|---|
| 4a | 17 | 28.44 | 970 | 879 |
| | 18 | 14.92 | 575 | 617 |
| | 19 | 8.89 | 538 | 512 |
| | 20 | 4.95 | 421 | 418 |
| | 21 | 0 | 272 | 259 |
| 4b | 19 | 26.02 | 1811 | 1750 |
| | 20 | 15.01 | 945 | 932 |
| | 21 | 2.93 | 863 | 606 |
| | 22 | 0.5875 | 717 | 596 |
| | 23 | 0.35 | 532 | 356 |
| | 24 | 0 | 547 | 301 |
| 5a | 15 | 21.09 | 946 | 803 |
| | 16 | 15.37 | 734 | 613 |
| | 17 | 0 | 342 | 303 |
| 5b | 9 | – | – | – |
| | 10 | 10.89 | 818 | 713 |
| | 11 | 0.47 | 602 | 510 |
| 6a | 11 | 10.59 | 917 | 835 |
| | 12 | 1.08 | 771 | 625 |
| | 13 | 0.16 | 337 | 299 |
| | 14 | 0.04 | 261 | 240 |
| 6b | 10 | 26.01 | 963 | 872 |
| | 11 | 17.15 | 876 | 774 |
| | 12 | 8.68 | 677 | 531 |
| | 13 | 0.22 | 668 | 594 |
| 7a | 11 | 4.15 | * (1.5%) | * (1.2%) |
| | 12 | 2.21 | 630 | 512 |
| | 13 | 0.575 | 587 | 369 |
| | 14 | 0.125 | 557 | 290 |
| 7b | 16 | 19.62 | 1708 | 1286 |
| | 17 | 12.41 | 962 | 855 |
| | 18 | 1.83 | 878 | 729 |
| | 19 | 1.22 | 749 | 561 |
| | 20 | 1.05 | 701 | 473 |
| | 21 | 0.34 | 623 | 414 |
| 8a | 13 | 19.87 | 1025 | 901 |
| | 14 | 14.80 | 816 | 785 |
| | 15 | 4.66 | 728 | 632 |
| | 16 | 2.24 | 714 | 563 |

Table 2.3 – Continued

| **Geography** | **# of trucks** | **Average wait** | **VI** | **Both** |
|---|---|---|---|---|
| | 17 | 0.64 | 698 | 501 |
| | 18 | 0.18 | 688 | 317 |
| 8b | 17 | 12.66 | 1757 | 1491 |
| | 18 | 9.56 | 932 | 859 |
| | 19 | 5.39 | 888 | 702 |
| | 20 | 1.67 | 867 | 623 |
| | 21 | 1.01 | 607 | 434 |
| | 22 | 0.2 | 556 | 406 |

Table 2.4: Intermill-Needtime = 10, 480 loads

| Geography | # of trucks | Average wait | VI | Both |
|---|---|---|---|---|
| 4a | 9 | 51.875 | 788 | 714 |
| | 10 | 0.5 | 335 | 321 |
| | 11 | 0 | 206 | 194 |
| 4b | 9 | 54.8375 | 992 | 717 |
| | 10 | 16.06 | 863 | 600 |
| | 11 | 0.746 | 620 | 596 |
| | 12 | 0.416 | 477 | 218 |
| 5a | 8 | 17.37 | 833 | 621 |
| | 9 | 0 | 247 | 202 |
| 5b | 5 | 4.10 | 768 | 605 |
| | 6 | 0 | 200 | 196 |
| 6a | 6 | 7.51 | 451 | 319 |
| | 7 | 0.67 | 298 | 284 |
| | 8 | 0.32 | 290 | 264 |
| | 9 | 0.16 | 244 | 231 |
| 6b | 5 | 17.75 | 953 | 786 |
| | 6 | 1.47 | 813 | 802 |
| | 7 | 0.1 | 342 | 343 |
| 7a | 5 | 7.03 | * (6.4 %) | * (2.1 %) |
| | 6 | 1.86 | 812 | 726 |
| | 7 | 0.325 | 875 | 574 |
| | 8 | 0 | 465 | 257 |
| 7b | 10 | 10.35 | 706 | 634 |
| | 11 | 2.6 | 517 | 369 |
| | 12 | 0.02 | 408 | 321 |
| 8a | 7 | 3.22 | 917 | 573 |
| | 8 | 0 | 267 | 202 |
| 8b | 9 | 9.56 | 856 | 632 |
| | 10 | 3.7 | 800 | 665 |
| | 11 | 0.48 | 711 | 533 |

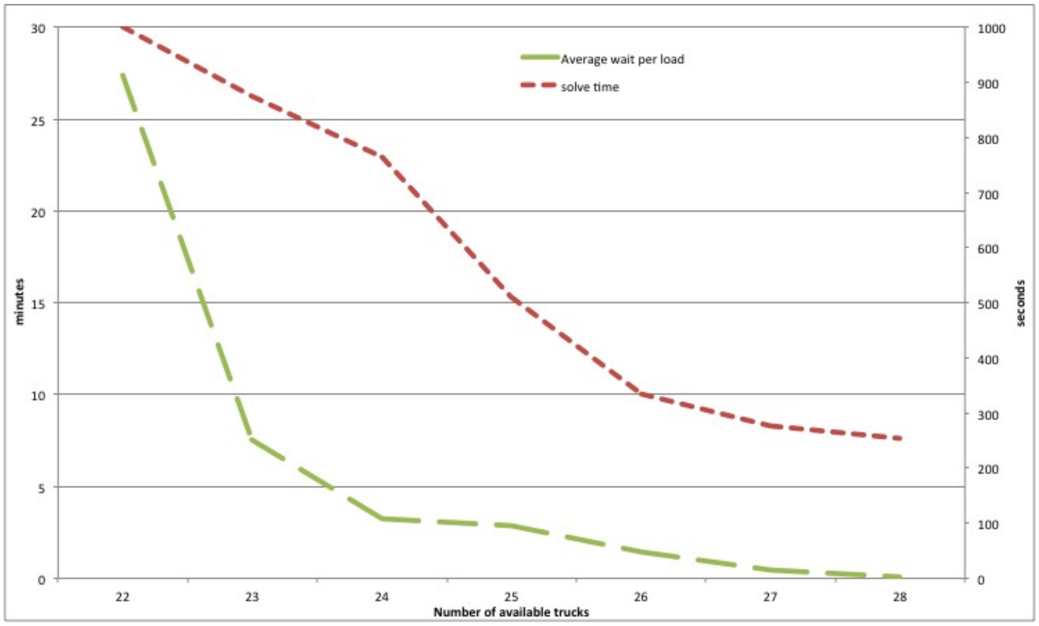


Figure 2.2: Relationship between Number of available trucks and average wait times (4a- 4 mins)

### 2.6.2 Cost of a Separable Solution

Table 2.5 presents the results of the experiment to demonstrate the value of coordination. For each geography and intermill need time, the table presents the number of trucks needed to find a feasible separable solution, or a solution for which the trucks serve only one front, and the number of trucks needed to find a feasible solution when coordination across fronts is allowed. These columns are labeled "Separable" and "Coordination," respectively.

As the table shows, on average, when coordination is allowed, almost 27% fewer vehicles are needed to achieve feasibility compared to the number needed to achieve feasibility when requiring separability. This result is consistent regardless of the number of fronts. However, the value of coordination increases as the time between mill needs increases. When the intermill need time is three, the average difference in the number of trucks is 18.1%. At an intermill need time of 10, the difference increases to 40.3%.

Table 2.5: Number of trucks needed for separable solution

| Geography | InterMill Need Time | Separable | Coordination |
|---|---|---|---|
| 4a | 3 | 32 | 29 |
| | 4 | 27 | 22 |
| | 5 | 21 | 17 |
| | 10 | 14 | 9 |
| 4b | 3 | 34 | 30 |
| | 4 | 26 | 23 |
| | 5 | 23 | 19 |
| | 10 | 13 | 9 |
| 5a | 3 | 32 | 25 |
| | 4 | 24 | 19 |
| | 5 | 20 | 15 |
| | 10 | 11 | 8 |
| 5b | 3 | 20 | 16 |
| | 4 | 17 | 12 |
| | 5 | 13 | 10 |
| | 10 | 10 | 5 |
| 6a | 3 | 22 | 17 |
| | 4 | 17 | 13 |
| | 5 | 14 | 11 |
| | 10 | 10 | 6 |
| 6b | 3 | 21 | 17 |
| | 4 | 17 | 13 |
| | 5 | 15 | 11 |
| | 10 | 9 | 5 |
| 7a | 3 | 19 | 16 |
| | 4 | 18 | 13 |
| | 5 | 16 | 11 |
| | 10 | 9 | 5 |
| 7b | 3 | 33 | 27 |
| | 4 | 25 | 20 |
| | 5 | 22 | 16 |
| | 10 | 16 | 10 |
| 8a | 3 | 39 | 23 |
| | 4 | 24 | 17 |
| | 5 | 19 | 13 |
| | 10 | 13 | 7 |
| 8b | 3 | 38 | 31 |
| | 4 | 29 | 21 |
| | 5 | 23 | 17 |

Table 2.5 Continued:

| | | | |
|---|---|---|---|
| | 10 | 16 | 9 |

### 2.6.3 Value of Variable Harvest Rates

Table 2.6 presents the results of our experiments designed to demonstrate the value of variable harvest rates. For each instance, geography and intermill need time, we present the number of trucks required to achieve a feasible solution when harvest rates are constant, called “Const Trucks.” We also presented the average wait time per load associated with this number of trucks, called “Const Avg Wait.” Finally, in Table 2.6 we present the number of trucks required to achieve a zero wait time using the variable harvest rates, called “Var Trucks.”

As the table shows, allowing variable harvest rates leads to solutions that have often considerably less wait time per load, almost 24 minutes on average, than the constant harvest rate solutions. Further, the constant harvest rate solutions require 12% more trucks than the variable harvest rate case to pick up all of the loads at their ready times. In only one case does the constant harvest rate case achieve feasibility with fewer trucks than the number required to achieve no wait with variable harvest rates.

Table 2.6: Number of trucks needed to pick up loads at ready times and the corresponding average wait times

| Geography | InterMill Need Time | Const Trucks | Const Avg Wait | Var Trucks |
|---|---|---|---|---|
| 4a | 3 | 39 | 43 | 37 |
| | 4 | 31 | 19 | 28 |
| | 5 | 24 | 26 | 21 |
| | 10 | 13 | 21 | 11 |
| 4b | 3 | 42 | 20 | 39 |
| | 4 | 31 | 44 | 30 |
| | 5 | 24 | 16 | 21 |
| | 10 | 14 | 15 | 12 |
| 5a | 3 | 34 | 64 | 30 |
| | 4 | 25 | 38 | 23 |
| | 5 | 20 | 26 | 17 |
| | 10 | 11 | 45 | 9 |
| 5b | 3 | 22 | 23 | 21 |
| | 4 | 17 | 15 | 14 |
| | 5 | 14 | 11 | 11 |
| | 10 | 9 | 19 | 6 |
| 6a | 3 | 24 | 9 | 22 |
| | 4 | 19 | 9 | 16 |
| | 5 | 14 | 5 | 14 |
| | 10 | 8 | 18 | 9 |
| 6b | 3 | 23 | 13 | 21 |
| | 4 | 18 | 17 | 16 |
| | 5 | 15 | 23 | 13 |
| | 10 | 9 | 29 | 7 |
| 7a | 3 | 24 | 28 | 22 |
| | 4 | 19 | 13 | 18 |
| | 5 | 16 | 20 | 14 |
| | 10 | 8 | 33 | 8 |
| 7b | 3 | 34 | 10 | 32 |
| | 4 | 26 | 17 | 24 |
| | 5 | 22 | 25 | 21 |
| | 10 | 13 | 35 | 12 |
| 8a | 3 | 34 | 20 | 31 |
| | 4 | 25 | 18 | 23 |
| | 5 | 20 | 15 | 18 |
| | 10 | 12 | 25 | 8 |
| 8b | 3 | 43 | 36 | 36 |
| | 4 | 33 | 29 | 27 |

Table 2.6 Continued:

| | | | | |
|---|---|---|---|---|
| | 5 | 25 | 23 | 22 |
| | 10 | 13 | 15 | 11 |

## 2.7 Conclusion

In this chapter, we introduce the sugarcane harvest logistics problem for Brazil. We introduce valid inequalities and a heuristic to generate a feasible initial solution. Our results show that the presented valid inequalities and heuristic for finding feasible solutions provide significant computational advantages. We also introduce a lifting heuristic that was advantageous in proving optimality. We further demonstrate that coordinating the trucks across fronts offers the opportunity to dramatically reduce the number of vehicles needed to serve the mills. Finally, our results show that variable harvest rates reduce cut-to-crush times while also reducing the number of vehicles needed to serve the loads.

There are two important directions for future work. The first direction builds on the work of Salassi et al. [2009b] that explores a sugarcane harvest logistics problem in the United States. While there are key differences between Brazilian and US-based operations, most notably the fact that harvests in Brazil run 24 hours a day, the US operations have many more farms involved in a daily harvest operation. Even with the enhancements introduced in this chapter, the size of the US problem requires additional research.

Second, the problem introduced here is deterministic. Not surprisingly, a harvest operation has a number of uncertainties. Importantly, there are isolated weather events and breakdowns that require attention. Yet, this research is an important foundation for the stochastic case. For one, the research presented here can be used to solve for perfect information solutions for the stochastic case. These solutions can be helpful for evaluating heuristic approaches. Second, the methods here can be valuable in a rolling horizon procedure. When one considers a horizon of four to six hours rather than 24 as we have here, the proposed solution method runs in a few seconds and can thus be amenable for use in real time.

# CHAPTER 3
# SUGARCANE HARVEST LOGISTICS IN THE US

## 3.1 Introduction

The sugarcane industry in the United States produces an annual sugarcane crop with an estimated economic value of over $5 billion annually. This chapter focuses more specifically on the sugarcane industry within the State of Louisiana for two reasons: first, Louisiana accounts for approximately 50% of the cane sugar production in the US with an annual estimated economic value of $2.7 billion, and second, data for sugarcane production in Louisiana is publicly available since production is spread throughout 475 (mostly) family farms on approximately 440,000 acres [United States Department of Agriculture, Economic Research Service, 2010]. In contrast, sugarcane production in other states is less significant economically and is sometimes controlled by a single private owner, making the availability of data problematic.

In recent years, US sugar prices have fallen, due largely to the North American Free Trade Agreement (NAFTA) which allows Mexico to export unlimited amounts of sugar to the United States without tariff: in the first 9 months of 2013, Mexican sugar exports to the US total is approximately 1.9 million metric tons [Polopolus et al., 2010, United States Department of Agriculture, Economic Research Service, 2010]. As a result of these imports, US sugar prices have fallen below government support levels, but production costs have actually increased at the same time due to increasing costs of fuel used for harvesting and transporting the sugarcane to mills

[Salassi and Deliberto, 2012].

Salassi and Barker [2008] estimate that the total harvest cost per acre is \$241.32 per acre (variable cost is \$144.54 per acre), and for every additional minute harvesting resources are forced to wait for the truck returning from the mill, the variable cost increases by approximately \$1.30 using 2007 harvest fuel and labor prices [Barker, 2007]. Because mills pay for the cost of hauling the cane from the farms, growers thus have an incentive to use enough trucks to pick up all the loads as soon as they become ready. The situation is further complicated by the fact that harvest operations among farms are often uncoordinated. The result is congestion at the mill, and a vicious cycle that leads growers to use even more trucks. To reduce the costs associated with congestion at the mill, the mill operators have two choices. First, they can increase the unloading capacity at the mill. Second, they can seek to coordinate the harvests. Salassi and Barker [2008] and Salassi et al. [2009a] suggest that coordination can be achieved simply by coordinating the start times of the harvests so truck arrivals to the mill are evenly spread throughout the day. By allowing that farms can continue to harvest at a constant rate and that loads are picked up at their ready times, the proposed solution structure minimally disrupts current practices in the operations while offering the promise of reducing the number of trucks. In this chapter, we generalize Salassi and Barker [2008] and Salassi et al. [2009a] and develop a model that can solve real-sized problems without making the aggregating and discretization assumptions required for tractability by Salassi and Barker [2008] and Salassi et al. [2009a]. As a result, we present a model that more closely matches the operational environment of sugarcane harvest logistics in Louisiana and reduces the number of trucks required to pick up the loads. Like Salassi and Barker [2008] and Salassi et al. [2009a], we employ an approach that first determines the start times for the harvest at each farm supplying the mill and then determine the number of trucks required.

This chapter makes several contributions to the literature. First, taking advantage of ideas from the modeling of piecewise linear functions, we develop a model that overcomes the limitations of existing models for the problem such that realistic problem sizes can be solved by modern integer program solvers. Relatedly, this model eliminates the need to discretize problem parameters as is required in previous work and allows us to penalize not only having too many arrivals to a time block but also too few. Second, via computational experiments, we demonstrate that forcing the loads to be spread as evenly as possible through out the day has the desired effect of reducing the number of trucks needed to serve the loads.

Section 3.2 of the paper discusses previous work on sugarcane logistics and importantly details the contributions of Salassi and Barker [2008] and Salassi et al. [2009a]. Section 3.3 presents our model. In section 3.4, we present the results of two studies using our model. The first study uses a set of benchmark problems developed by Salassi and Barker [2008] and Salassi et al. [2009a] and compares the effectiveness of solutions developed by our approach to those developed by Salassi and Barker [2008] and Salassi et al. [2009a] for the same problems. Since these benchmark problems do not correspond to any particular real instance, we also present a second study in which we use publicly available data on the geographical locations of each of Louisiana's 456 sugarcane farms and 11 sugarcane mills as well as their production and processing rates to construct a set of 11 sugarcane logistics problems (1 for each of the 11 mills in Louisiana) based upon real data. We shall show that our modeling approach improves upon existing approaches in the literature and can easily solve realistically sized problems. In section 3.5.2, we develop managerial insights and section 3.6 presents our conclusions.

## 3.2 Literature Review

Sugarcane is an important agricultural commodity grown around the world. A key challenge in studying the logistics of sugarcane harvests is that almost every sugarcane producing country has a different infrastructure and differing levels of vertical integration. Hansen et al. [2002], Lejars et al. [2008], Le Gal et al. [2009], and McDonald et al. [2008] explore supply chain issues in the South African sugarcane industry. Supsomboon and Yosnual [2004] and Prichanont et al. [2005] study uncertainty arising in the Thai sugarcane industry. Diaz and Perez [2000], Lopez Milan et al. [2006], and Lopez-Milan and Pla-Aragones [2013] explore the multi-model sugarcane logistics in Cuba. Hildebrand [2002], Higgins and Postma [2004], Higgins and Davies [2005], Higgins and Laredo [2006], and Higgins [2006] explore sugarcane supply logistics in Australia. Lamsal et al. [2013a] study the coordination of harvest logistics in Brazil and provide a detailed review of harvest logistics around the world.

As noted previously, the most closely related to the work in this chapter is the work by Salassi and Barker [2008] and Salassi et al. [2009a]. With the intent of reducing congestion at the mill and the associated increase in trucks, Salassi and Barker [2008] and Salassi et al. [2009a] suggest coordinating harvest start times in the fields so that the number of deliveries to the mill are evenly spread over the mill's delivery time window. In concept, smoothing the arrivals to the mill reduces the congestion at the mill, and fewer trucks are needed to pick up the loads at their respective ready times. In both Salassi and Barker [2008] and Salassi et al. [2009a], the daylight hours, during which harvesting takes place, are divided into thirteen hourly blocks. The mill is assumed to have the same unloading capacity during each block.

To maintain tractability of the real-sized problem, Salassi and Barker [2008] and Salassi et al. [2009a] make modeling choices that are more restrictive than the actual operational constraints. First, they assume that the travel time between the farms and the mill are in multiples of 15 minutes. Specifically, in their model, they have some farms 15 minutes away from the mill, some farms 30 minutes away for the mill and others 45 minutes away from the mill. They also assume that the harvest time for every load in every farm is exactly 45 minutes. Finally, they assume that the harvesting at each farm can start only on the hour, 15 minutes past the hour, 30 minutes past the hour, and 45 minutes past the hour.

It is also worth noting Higgins [2006] who studies the problem of truck congestion at an Australian sugar mill. His objective is to minimize the sum of mill's idle time and the trucks' queue time. The main difference with the problem studied in this chapter is that, in the Australian case, the cane filled trailers can wait for the trucks returning from the mill at the fields. We also note that Higgins [2006] explicitly models the queue at the mill. Unfortunately, this modeling approach impacts the tractability of the model. Given that we restrict loads to being picked up at their ready-times in the field, as discussed subsequently, we do not explicitly model the queue.

## 3.3 Model Formulation

In this section, we present a formal model of the problem. Our ultimate goal is to reduce the number of trucks needed to pick up loads at the times that they are ready. However, directly modeling to minimize the number of trucks leads to an intractable problem. As a result, we seek an alternative objective that minimizes congestion at the mill and thus should have the effect of minimizing the number of trucks. That is, we seek to spread the load arrivals among the sections in the partition as per the mill's predefined capacity.

The first step of the model is that we divide the daylight hours into blocks of time. A block can be of arbitrary length. For each of these blocks, the mill has a predefined unloading capacity, expressed as the number of loads that the mill can process in the block of time. The unloading capacity need not be the same for each block, reflecting potentially changing capacity throughout the day. We call this partition of the day $P$.

Suppose we have the partition $P$ of the delivery window. As shown in Figure 3.1, a partition $P$, with $n$ blocks can be specified by the set of $n+1$ points. Specifically, $\{a_k\}$, for $k = 0, 1, \cdots n$. We enforce $a_0 < a_1 < a_2 \cdots < a_n$. The start time of the delivery window at the mill is $a_0$ and $a_n$ is the end time of the delivery window.

For each block $k$, the time between $a_{k-1}$ and $a_k$, we have a predefined unloading capacity of $N_k$ loads. We solve for a harvest schedule that matches load arrivals with the mill's capacity in each block and penalizes any deviation from the mill's desired unloading capacity in each block.

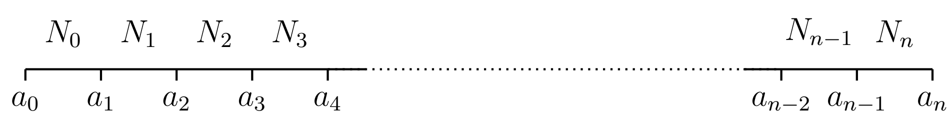


Figure 3.1: Partition of the delivery window and the unloading capacity

Let $x_{ij}$ be the arrival time at the mill of $i^{th}$ farm's $j^{th}$ load. Let $y_i$ be the time when harvesting starts at farm $i$, and $h_i$ be the time it takes to harvest a load at farm $i$. Let the travel time between farm $i$ and the mill be $t_i$, and $Q_i$, the daily load quota of the farm $i$ in $F$. The arrival time of a load to the mill depends on the harvest start time for the originating farm, the harvest rate at that farm, and the travel time between the farm and the mill. Mathematically,

$$x_{ij} = y_i + j \cdot h_i + t_i \qquad \forall (i,j) \mid i \in F, j \in \{1 \ldots Q_i\}$$

To distribute the loads among blocks, we need to count the number of load arrivals in each of these time blocks. A straightforward approach is to introduce binary variables $b_{ij}^k$ for $k = 1 \ldots n$, that indicate whether or not load $j$ from field $i$ arrives between $a_{k-1}$ and $a_k$. With $M$ as a large number and using the constraints

$$a_{k-1} \leq x_{ij} + (1 - b_{ij}^k) \times M \qquad \forall (i,j) \forall i \in F, j \in \{1 \ldots Q_i\}, k \in 1...n \tag{3.1}$$

$$x_{ij} + (1 - b_{ij}^k) \times (-M) \leq a_k \qquad \forall (i,j) \forall i \in F, j \in \{1 \ldots Q_i\}, k \in 1...n \tag{3.2}$$

$$\sum_{k=1}^{n} b_{ij}^k = 1 \qquad \forall (i,j) \forall i \in F, j \in \{1 \ldots Q_i\}, \tag{3.3}$$

we can determine in which block each arrival occurs.

While constraints (3.1) to (3.3) may be straightforward, they lead to a formulation with a poor relaxation. Consequently, we propose an alternative that expresses the arrival time $x_{ij}$ as a convex combination of the beginning and the end time of the section $k$ in which the arrival time lies. Specifically, $x_{ij}$ in section $k$ can be expressed as:

$$\begin{aligned} x_{ij} = \lambda_{ij}^{k-1} \times a_{k-1} + \lambda_{ij}^k \times a_k \\ \lambda_{ij}^{k-1} + \lambda_{ij}^k = 1 \\ 0 \leq \lambda_{ij}^{k-1} \leq 1 \\ 0 \leq \lambda_{ij}^k \leq 1. \end{aligned}$$

More generally, we can write

$$x_{ij} = \sum_{k=1}^{n} \lambda_{ij}^{k} \times a_k, \qquad \sum_{k=1}^{n} \lambda_{ij}^{k} = 1, \qquad \lambda_{ij}^{k} \in R^{+} \qquad \forall i \in F, j \in \{1 \ldots Q_i\},$$

if for given $i$ and $j$, we force at most two $\lambda_{ij}^{k}$ variables among $\lambda_{ij}^{k}$ variables, $k = 0, 1, \cdots n$, to be positive. If $\lambda_{ij}^{k}$ and $\lambda_{ij}^{l}$ are positive, then $k = l - 1$ or $k = l + 1$. This situation can be modeled using binary variables $b_{ij}^{k}$ for $k = 1 \ldots n$, (where $b_{ij}^{k} = 1$ if $a_{k-1} \leq x_{ij} \leq a_k$ and $b_{ij}^{k} = 0$, otherwise), and the following constraints:

$$\begin{aligned}
\lambda_{ij}^{0} &\leq b_{ij}^{1} && \forall (i,j) \forall i \in F, j \in \{1 \ldots Q_i\} \\
\lambda_{ij}^{k} &\leq b_{ij}^{k-1} + b_{ij}^{k} && \forall k \in 1, \cdots n-1, i \in F, j \in \{1 \ldots Q_i\} \\
\lambda_{ij}^{n} &\leq b_{ij}^{n} && \forall (i,j) \forall i \in F, j \in \{1 \ldots Q_i\} \\
\sum_{k=1}^{n} b_{ij}^{k} &= 1 && \forall (i,j) \forall i \in F, j \in \{1 \ldots Q_i\}.
\end{aligned}$$

To account for the situation when not all the required loads can be delivered or extra loads must be delivered during any section in the partition, we define slack variables, $s_k^{+}$ and $s_k^{-}$ for $k = 1 \ldots n$. Then,

$$\sum_{i \in F} \sum_{j \in 1 \cdots n_i} b_{ij}^{k} + s_k^{+} - s_k^{-} = N_k \qquad \forall k \in 1 ... n.$$

Now, we can express a complete math program as:

$$\min \quad \sum_{k \in 1 \cdots n} (s_k^+ + s_k^-)$$

$$x_{ij} = y_i + j \cdot h_i + t_i \qquad \forall (i,j) \mid i \in F, j \in \{1 \ldots Q_i\} \tag{3.4}$$

$$x_{ij} = \sum_{k=1}^{n} \lambda_{ij}^k \times a_k \qquad \forall i \in F, j \in \{1 \ldots Q_i\} \tag{3.5}$$

$$\sum_{k=1}^{n} \lambda_{ij}^k = 1, \qquad \forall i \in F, j \in \{1 \ldots Q_i\} \tag{3.6}$$

$$\lambda_{ij}^0 \leq b_{ij}^1 \qquad \forall (i,j) \forall i \in F, j \in \{1 \ldots Q_i\} \tag{3.7}$$

$$\lambda_{ij}^k \leq b_{ij}^{k-1} + b_{ij}^k \qquad \forall k \in 1, \cdots n-1, i \in F, j \in \{1 \ldots Q_i\} \tag{3.8}$$

$$\lambda_{ij}^n \leq b_{ij}^n \qquad \forall (i,j) \forall i \in F, j \in \{1 \ldots Q_i\} \tag{3.9}$$

$$\sum_{k=1}^{n} b_{ij}^k = 1 \qquad \forall (i,j) \forall i \in F, j \in \{1 \ldots Q_i\} \tag{3.10}$$

$$\sum_{i \in F} \sum_{j \in 1 \cdots n_i} b_{ij}^k + s_k^+ - s_k^- = N_k \qquad \forall k \in 1 ... n \tag{3.11}$$

$$b_{ij}^k \in \{0, 1\} \qquad \forall (i,j) \forall i \in F, j \in \{1 \ldots Q_i\}, k \in 1 ... n \tag{3.12}$$

$$\lambda_{ij}^k \in [0, 1] \qquad \forall (i,j) \forall i \in F, j \in \{1 \ldots Q_i\}, k \in 1 ... n. \tag{3.13}$$

The objective function penalizes both positive and negative deviation from the mill's unloading capacity, but this does not always have to be the case. We can choose to just penalize the number of arrivals that are above the mill's capacity in the given period. Such a scenario might be appropriate when adding one extra arrival significantly affects the average turn around time for trucks. We can also choose to penalize just the deviation on the lower side. This might be appropriate when we want to achieve high utilization for the resources at the mill yard. We can also choose to penalize deviations in one direction more heavily than the deviation on the other side. We could also give more weight to the deviations in one period than the deviations in other periods.

## 3.4 Comparison with Results in the Literature

In this section, we compare our modeling approach with that proposed by by Salassi and Barker [2008] and Salassi et al. [2009a]. Table 3.1 summarizes the two instances from Salassi and Barker [2008] and Salassi et al. [2009a]. In both instances, the harvest time for a load for each grower is set at 45 minutes. In the first instance, there are 45 growers and 360 loads, and in the second scenario, there are 48 growers and 432 loads. In both instances, harvesting at the farms can start as early as 6:00 a.m. and loads arrive to the mill between 7:00 a.m. and 8:00 p.m. The hourly limit for the number of arrivals is set at 30 and 36, respectively, for the two instances. Salassi and Barker [2008] and Salassi et al. [2009a] only penalize the arrivals above the limit. We will refer these two instances collectively as "Salassi instances." Because the Salassi instances include groups of identical fields, we add symmetry breaking constraints to the model when running these instances.

To calculate the number of trucks needed, Salassi and Barker [2008] and Salassi et al. [2009a] do not model the queue at the mill. In their models, a truck becomes available for another dispatch one hour after the arrival to the mill. This is true regardless of how many trucks arrive at the mill in the given hour. We propose an alternative that explicitly models the queueing at the mill in determining the number of trucks. First, we assume a FIFO queue at the mill and that growers start harvesting at the optimal start times given by the model. Then, we assign the trucks to pick up the loads at their respective ready times. The trucks pick up the cane, travel to the mill, wait in the queue, unload, and become available for next dispatch when the cane is unloaded (unloading time of two minutes per load for the first instance and and 1.66 minutes per load for the second instance). Because Salassi and Barker [2008] and Salassi et al. [2009a] set their maximum loads per hour at 30 and 36 for the two instances, we set unloading times of two minutes and 1.66 minutes, respectively.

In this section and the next, given a solution to the math program, we can compute the optimal number of trucks needed to pick up the loads at their ready times at the growers by using a slightly modified version of an algorithm presented in Lamsal et al. [2013a]. The truck assignment algorithm is coded in Python and runs instantaneously on the subsequently described hardware.

In this section and the next, all computational results for instances of the proposed math program are obtained using GUROBI OPTIMIZER 5.6. The experiments were performed on a 2.3 GHz Intel Core i7 CPU running OS X 10.9. Communication with Gurobi was achieved through Gurobi's C++ interface.

In Table 3.2, we report the objective value for the math program and the number of trucks needed to pick up all the loads at their respective ready times with a FIFO queue at the mill. Using the approach, we find the number of trucks needed are 32 and 43 for our solutions. Evaluating the solutions in Salassi and Barker [2008] in a similar manner, that is if we model the queue at the mill and use the optimal truck

Table 3.1: Summary of Salassi instances

| | Group A | Group B | Group C | Group D | Group E | Group F |
|---|---|---|---|---|---|---|
| **Instance 1** | | | | | | |
| Grower numbers | 24 | 12 | 3 | 2 | 3 | 1 |
| Daily loads | 6 | 12 | 6 | 12 | 6 | 12 |
| Total daily loads | 144 | 144 | 18 | 24 | 18 | 12 |
| Travel time to the mill (mins) | 15 | 15 | 30 | 30 | 45 | 45 |
| **Instance 2** | | | | | | |
| Grower numbers | 12 | 12 | 8 | 8 | 4 | 4 |
| Daily loads | 6 | 12 | 6 | 12 | 6 | 12 |
| Total daily loads | 72 | 144 | 48 | 96 | 24 | 48 |
| Travel time to the mill (mins) | 15 | 15 | 30 | 30 | 45 | 45 |

assignments according to the optimal algorithm, the solutions found by Salassi and Barker [2008] need 53 and 72 trucks for the first and second instances, respectively. In an attempt to spread the loads equally across the hours of the day, for the first

Table 3.2: Solution comparison for Salassi instances

| | Objective (Salassi) | Trucks(Salassi) | Objective | Trucks |
|---|---|---|---|---|
| Scenerio 1 | 9 | 53 | 0 | 32 |
| Scenerio 2 | 9 | 72 | 0 | 43 |

instance, we also consider an the hourly limit of 27 between 7:00 a.m. and 11:00 a.m. and the hourly limit of 28 between 11:00 a.m. and 8:00 p.m. Again, we return an objective value of zero for the math program. Further, as a result, we are able to spread the loads across the day. Figure 3.2 shows the number of loads per hourly block for our solution. As is shown Figure 3.3, the solution given in Salassi and Barker [2008] for the first instance does not spread the loads as evenly across the day. For example, there are 39 arrivals between 3:00 p.m. and 4:00 p.m. and nine for the last hour of the day. While it is only one instance, it suggests that there is an advantage to spreading the loads as evenly through the day as possible. We will revisit this question in the next section.

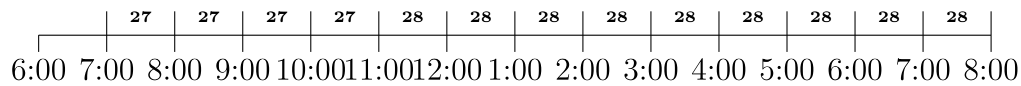


Figure 3.2: Our Solution

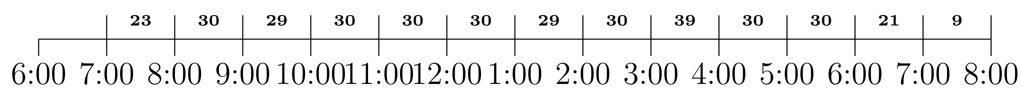


Figure 3.3: Salassi and Barker [2008] Solution

Again considering the first instance and the case where we constrain the load limit in each block to spread the loads across the day, Figure 3.4 compares the time spent by each load arrival at the mill yard in our solution and the solution in Salassi and Barker [2008]. In Figure 3.5, we plot the number of trucks at the mill yard for our solution and for the solution in Salassi and Barker [2008]. In these comparisons we make the truck assignments for both our and Salassi and Barker [2008]'s solution by assuming a FIFO queue with unloading time of two minutes per load at the mill. In Figure 3.5, each line on the graph is a moving moving average of the previous 10 minutes. It is noticeable that the waiting time for each load is significantly shorter in our solution despite of having the same unloading time. This fact is also evident by comparing the number of trucks at the mill yard throughout the day. With our solution, the number of trucks in the queue at the mill is much more consistent through out the day.

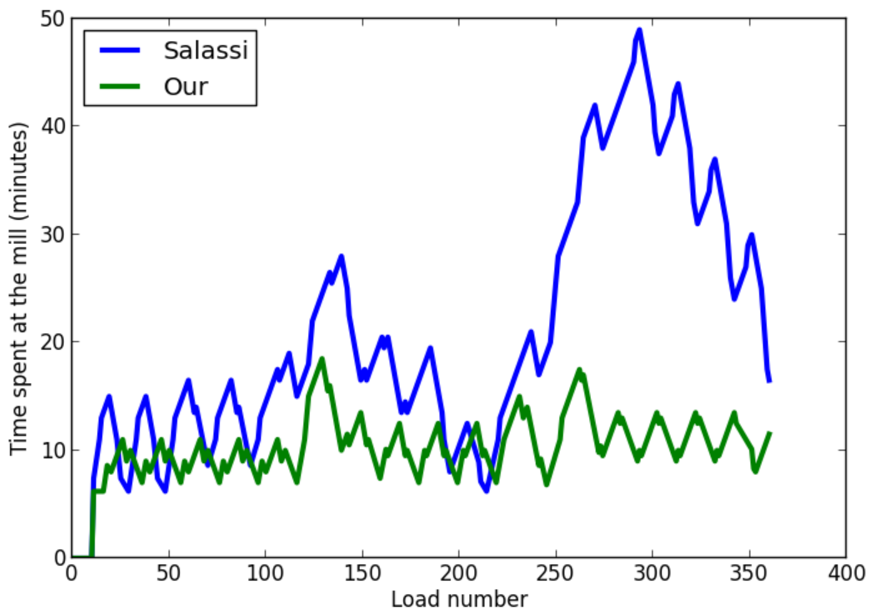


Figure 3.4: Comparison of time spent at the mill by each load

## 3.5 Computational Experiments

In this section, we present a series of experiments to test both the computational limits our our modeling approach and the value of spreading the loads evenly throughout the day. For these tests, we develop a class of instances using publicly

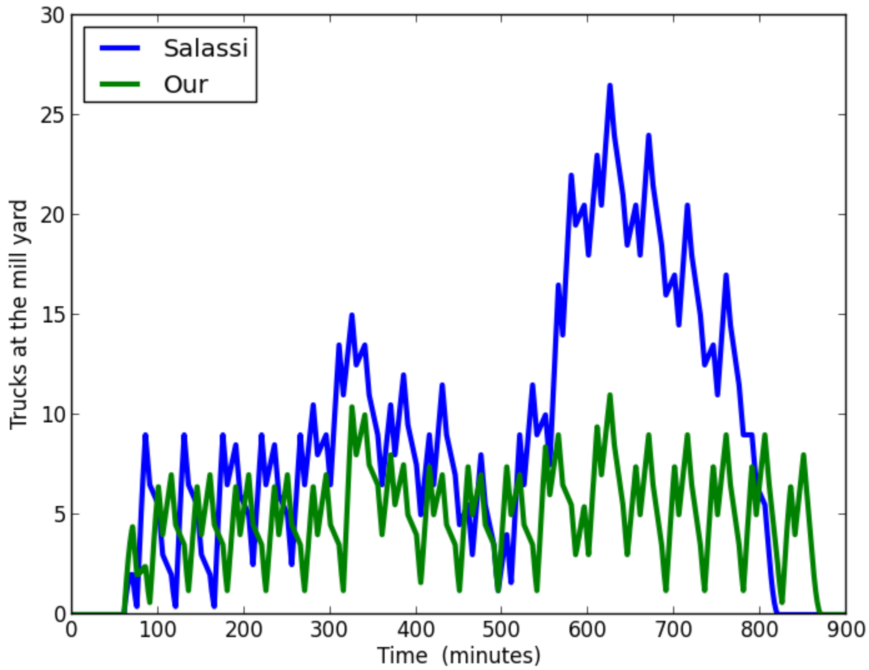


Figure 3.5: Comparison of number of trucks at the mill yard

available mill and farm data.

### 3.5.1 Datasets

There are 11 mills and approximately 475 farms in Louisiana. United States Department of Agriculture, Economic Research Service [2010] and League [2013] provide zip-code level addresses for 456 farms and exact addresses of the 11 mills. We also have the county level data on sizes of the farms that puts them into buckets of various sizes [United States Department of Agriculture, Economic Research Service, 2010]. First, we calculate the distances between the farms and the mills. Then, we randomly assign the sizes for the individual farms according to the distribution of farm sizes in the respective counties. We then assume that farms that harvest more than 750 acres of cane a year have two combine harvesters and the ones that harvest less than 750 acres have one combine harvester. This harvester distribution is motivated by the fact Salassi and Barker [2008] found the average number of combines to be 1.5. Each combine harvester takes approximately 45 minutes to fill a load. So, the time to harvest a load in the farm with one harvester is 45 minutes plus a small random component (chosen from uniform random between negative 5 and positive 5) and the time to harvest a load in the farm with two harvesters is 22.5 minutes plus a small random component (chosen from uniform random between negative 2.5 and positive 2.5) [Barker, 2007, Salassi and Barker, 2008]. In total, we have 456 farms in 85 zip codes with a daily capacity of 4044 loads. The number of farms in the respective counties and the location of the mills are shown in Figure 3.6. Each dot represents the location of a mill and the number inside a county is the number of farms in the county.

To create mill assignments for each farm, we solve a capacitated assignment problem. The objective is to minimize the sum of the distances between the farms and the mill that serves the respective farms. We assume the mills are of approximately the same size, each receiving between 365 and 370 loads. Table 3.3 provides the summary of the 11 farm scenarios. The first mill is served by 55 farms and has a daily quota of 370 loads and so forth.

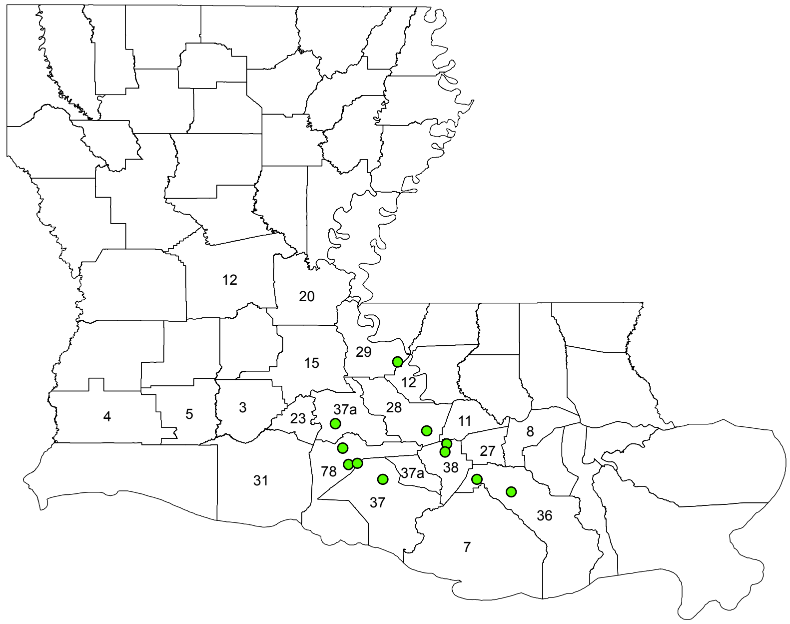


Figure 3.6: Distribution of farms and the mills in Louisiana

### 3.5.2 Computational Results

In this section, we present the results of our computational experiments with the newly created datasets. In our first test, presented in section 3.5.2.1, we demonstrate that simply controlling the start of the harvest time can have a remarkable effect in reducing the number of trucks needed to serve the loads at their ready times. In section 3.5.2.2, we explore the impact on solution quality when we reduce the size of the time blocks from an hour to half an hour to 15 minutes. This experiment also demonstrates the limits of the proposed math programming formulation.

Table 3.3: Distribution of farms and total loads

| Mill Area | # of farms | # of loads |
|---|---|---|
| 1 | 55 | 370 |
| 2 | 66 | 367 |
| 3 | 29 | 369 |
| 4 | 39 | 365 |
| 5 | 23 | 365 |
| 6 | 69 | 370 |
| 7 | 28 | 370 |
| 8 | 53 | 370 |
| 9 | 27 | 365 |
| 10 | 26 | 365 |
| 11 | 41 | 368 |

#### 3.5.2.1 Changing the Load Arrival Limits

Salassi and Barker [2008] and Salassi et al. [2009a] suggested that mill operators could coordinate harvests simply by controlling the starting times of harvests at the farms that supply the mill. In this section, we demonstrate this value of this coordination. We consider hourly load limits of 29, 30, 31, 32, 35, and 40. A limit of 29 represents the smallest integer limit that is equal across all hours and offers the chance of a solution that does not violate the load limit. Increasing the limit mimics an increasingly uncoordinated solution. Table 3.4 presents the number of trucks needed to pick up all the loads at their ready times corresponding to optimal solutions of the math program for each of the 11 instances.

The results clearly show a steady increase in the number of vehicles required as the solution becomes less coordinated. The solution also demonstrates how simply controlling the start of harvests at the farms can have a remarkable impact on the number of trucks.

Table 3.4: Relationship between load arrival limits and number of trucks needed

| | Arrival Limits | | | | | |
|---|---|---|---|---|---|---|
| Mill Area | 29 loads | 30 loads | 31 loads | 32 loads | 35 loads | 40 loads |
| 1 | 32 | 39 | 40 | 45 | 48 | 59 |
| 2 | 36 | 40 | 41 | 43 | 51 | 64 |
| 3 | 31 | 31 | 37 | 39 | 42 | 57 |
| 4 | 34 | 37 | 39 | 40 | 54 | 66 |
| 5 | 35 | 40 | 36 | 49 | 47 | 86 |
| 6 | 48 | 53 | 55 | 55 | 61 | 72 |
| 7 | 30 | 32 | 34 | 34 | 50 | 68 |
| 8 | 35 | 39 | 41 | 43 | 46 | 47 |
| 9 | 38 | 41 | 40 | 45 | 49 | 79 |
| 10 | 28 | 32 | 33 | 35 | 41 | 70 |
| 11 | 33 | 33 | 36 | 40 | 48 | 58 |

#### 3.5.2.2 Size of Time Blocks

In this section, we seek to demonstrate that reducing the size of the time blocks considered can lead to improved solutions in terms of the number of trucks required. To make a fair comparison among the solutions using various time blocks, we want to have the total number of loads be divisible by the number of one hour, half an hour, and 15 minute time blocks. For our instances, we have 13, 26 and 52 blocks with one hour, half an hour, and 15 minute blocks. The first number greater 370 such that we have a whole number as target in each block is 416. With the total loads of 416, we have an arrival limit of 32 loads, 16 loads and 8 loads, respectively, for one hour, 30 minutes and 15 minute blocks. To make a scenario with 416 loads, we take scenario 1, which has 55 farms and 370 total loads, and add one load each to the first 46 growers. Scenario 2 has 29 farms and 369 loads. So, we add two loads each to the first 18 farms and one load each to the remaining 11 farms. We proceed in a similar fashion for the remaining nine instances, thus creating 11 scenarios.

We solve these 11 scenarios for one hour, half an hour, and 15 minute blocks. To find the number of trucks, we assume a FIFO queue at the mill and that growers start harvesting at the optimal start times given by the model. Then, we assign the trucks to pick up the loads at their respective ready times. The trucks pick up the cane, travel to the mill, wait in the queue, unload and become available for next dispatch when the can is unloaded. We assume the unloading time for each load is 1.875 minutes, because the arrival window is 780 minutes and each instance has 416 loads.

In Table 3.5, we report the solution time for one hour, half hour, and 15 minute blocks for the 11 instances with 416 loads and the number of trucks needed to meet the ready times for the corresponding best solution. The instances marked with "*" are not solved to optimality, and we present the number of trucks or the best solution obtained at 3600 seconds. All but one of the one hour block instances were solved to optimality. The solutions obtained using smaller time blocks better spread the load across the arrival window which in turn reduces the number of trucks needed to haul all the loads. This seems to be consistent even when the solutions obtained using smaller time blocks are not optimal.

To better understand the advantage of the narrower time blocks, in Table 3.6, we describe Scenario 1's one hour, half hour, and 15 minute block solutions. We divide the arrivals in each solution into 52 different 15-minute bins. All but the last bin (7:45 p.m. - 8:00 p.m.) is half-open. In other words, in counting the total arrivals in the first bin (7:00 a.m. - 7:15 a.m.), we assume the bin is the interval [7:00 a.m. -

7:15 a.m.) and thus include arrivals 7:00 a.m., but exclude the arrivals at 7:15 a.m. The last bin, however, is [7:45 p.m. - 8:00 p.m.], which includes 8:00 p.m. This differs from how time blocks are defined in the math program above. For the math program with 15 minute blocks, the arrival at 7:15 a.m. can be counted in the first block (7:00 a.m. -7:15 a.m.) or the second block (7:15- 7:30 a.m.). More, specifically, if we have two arrivals at 7:15, one arrival could be counted towards the first block and the second arrival could be counted towards the second block. Not including the half-open intervals in the model maintains a continuous solution space, reducing computation time. We make the change in the presentation in Table 3.6 to simplify the presentation of the table. Because of this modeling choice, in the first 4 bins, (7:00 a.m. - 7:15 a.m., 7:15 a.m. - 7:30 a.m., 7:30 a.m. - 7:45 a.m., 7:45 a.m. - 8:00 a.m.), we have only 23 arrivals (5+3+8+7), even though the instance with hour long blocks is solved to an objective of zero. In the optimal solution, there are nine arrivals at 8:00 a.m. that the math program counts for the block (7:00 a.m - 8:00 a.m).

Analysis of the arrivals in each of the 15-minutes bins offers a clue as to why the number of trucks decreases for instances run with the smaller 15-minute blocks. Having divided the arrivals for each solution into bins, we calculate the standard deviation of arrivals across the 52 bins for the one hour, half hour and 15 minute blocks, respectively, obtaining 3.02, 2.03 and 1.20, respectively. As queueing analysis predicts, the queueing at the mill increases as the variability in arrivals increases. Because of the increased queueing, more trucks are then required to serve the loads at their ready times.

Table 3.5: Relationship between size of time blocks and number of trucks needed

| Scenario | Time blocks | Solution time (secs) | # Trucks needed Unloading time $= 1.875\ {}^{min}/_{load}$ |
|---|---|---|---|
| 1 | Hour blocks | 88 | 48 |
| | 30 minutes blocks | 1922 | 37 |
| | 15 minutess blocks | * | 34 |
| 2 | Hour blocks | 272 | 41 |
| | 30 minutes blocks | 2514 | 33 |
| | 15 minutess blocks | * | 32 |
| 3 | Hour blocks | 989 | 33 |
| | 30 minutes blocks | * | 31 |
| | 15 minutess blocks | * | 33 |
| 4 | Hour blocks | 52 | 40 |
| | 30 minutes blocks | * | 39 |
| | 15 minutess blocks | * | 38 |
| 5 | Hour blocks | * | 58 |
| | 30 minutes blocks | * | 52 |
| | 15 minutess blocks | * | 51 |
| 6 | Hour blocks | 118 | 52 |
| | 30 minutes blocks | 3279 | 50 |
| | 15 minutess blocks | * | 48 |
| 7 | Hour blocks | 237 | 35 |
| | 30 minutes blocks | * | 35 |
| | 15 minutess blocks | * | 34 |
| 8 | Hour blocks | 94 | 43 |
| | 30 minutes blocks | * | 37 |
| | 15 minutess blocks | * | 31 |
| 9 | Hour blocks | 1473 | 42 |
| | 30 minutes blocks | * | 41 |
| | 15 minutess blocks | * | 38 |
| 10 | Hour blocks | 509 | 34 |
| | 30 minutes blocks | * | 34 |
| | 15 minutess blocks | * | 32 |
| 11 | Hour blocks | 361 | 41 |
| | 30 minutes blocks | 3410 | 30 |
| | 15 minutess blocks | * | 32 |

Table 3.6: Break-down of arrivals

| Time | Hour | 30 Minutes | 15 Minutes |
|---|---|---|---|
| 7:00 a.m. - 7:15 a.m. | 5 | 7 | 6 |
| 7:15 a.m. - 7:30 a.m. | 3 | 5 | 7 |
| 7:30 a.m. - 7:45 a.m. | 8 | 10 | 8 |
| 7:45 a.m. - 8:00 a.m. | 7 | 7 | 7 |
| 8:00 a.m. - 8:15 a.m. | 10 | 8 | 9 |
| 8:15 a.m. - 8:30 a.m. | 8 | 9 | 7 |
| 8:30 a.m. - 8:45 a.m. | 13 | 9 | 10 |
| 8:45 a.m. - 9:00 a.m. | 5 | 9 | 7 |
| 9:00 a.m. - 9:15 a.m. | 8 | 8 | 8 |
| 9:15 a.m. - 9:30 a.m. | 14 | 5 | 8 |
| 9:30 a.m. - 9:45 a.m. | 8 | 12 | 9 |
| 9:45 a.m. - 10:00 a.m. | 3 | 7 | 8 |
| 10:00 a.m. - 10:15 a.m. | 17 | 8 | 8 |
| 10:15 a.m. - 10:30 a.m. | 5 | 6 | 8 |
| 10:30 a.m. - 10:45 a.m. | 9 | 13 | 8 |
| 10:45 a.m. - 11:00 a.m. | 5 | 5 | 8 |
| 11:00 a.m. - 11:15 a.m. | 11 | 6 | 7 |
| 11:15 a.m. - 11:30 a.m. | 10 | 8 | 6 |
| 11:30 a.m. - 11:45 a.m. | 6 | 9 | 8 |
| 11:45 a.m. - 12:00 p.m. | 5 | 8 | 8 |
| 12:00 p.m. - 12:15 p.m. | 13 | 7 | 8 |
| 12:15 p.m. - 12:30 p.m. | 5 | 6 | 6 |
| 12:30 p.m. - 12:45 p.m. | 4 | 12 | 10 |
| 12:45 p.m. - 1:00 p.m. | 6 | 7 | 6 |
| 1:00 p.m. - 1:15 p.m. | 11 | 10 | 6 |
| 1:15 p.m. - 1:30 p.m. | 6 | 7 | 9 |
| 1:30 p.m. - 1:45 p.m. | 9 | 11 | 7 |
| 1:45 p.m. - 2:00 p.m. | 6 | 5 | 7 |
| 2:00 p.m. - 2:15 p.m. | 11 | 12 | 10 |
| 2:15 p.m. - 2:30 p.m. | 3 | 3 | 9 |
| 2:30 p.m. - 2:45 p.m. | 12 | 9 | 7 |
| 2:45 p.m. - 3:00 p.m. | 6 | 7 | 9 |
| 3:00 p.m. - 3:15 p.m. | 8 | 8 | 8 |
| 3:15 p.m. - 3:30 p.m. | 8 | 7 | 6 |
| 3:30 p.m. - 3:45 p.m. | 6 | 9 | 6 |
| 3:45 p.m. - 4:00 p.m. | 9 | 8 | 9 |
| 4:00 p.m. - 4:15 p.m. | 9 | 7 | 8 |
| 4:15 a.m. - 4:30 a.m. | 8 | 9 | 9 |
| 4:30 p.m. - 4:45 p.m. | 11 | 8 | 9 |
| 4:45 p.m. - 5:00 p.m. | 7 | 7 | 7 |
| 5:00 p.m. - 5:15 p.m. | 9 | 11 | 9 |
| 5:15 p.m. - 5:30 p.m. | 10 | 5 | 8 |
| 5:30 p.m. - 5:45 p.m. | 9 | 8 | 10 |
| 5:45 p.m. - 6:00 p.m. | 4 | 9 | 7 |
| 6:00 p.m. - 6:15 p.m. | 12 | 9 | 9 |
| 6:15 p.m. - 6:30 p.m. | 8 | 7 | 9 |
| 6:30 p.m. - 6:45 p.m. | 9 | 9 | 9 |
| 6:45 p.m. - 7:00 p.m. | 5 | 7 | 7 |
| 7:00 p.m. - 7:15 p.m. | 11 | 9 | 9 |
| 7:15 p.m. - 7:30 p.m. | 9 | 6 | 9 |
| 7:30 p.m. - 7:45 p.m. | 6 | 8 | 9 |
| 7:45 p.m. - 8:00 p.m. | 6 | 10 | 10 |

### 3.6 Conclusion

For sugarcane farms and mills in Louisiana to remain profitable when sugar prices are falling and production cost is high and rising, increasing efficiency is important. By using a new modeling technique, this chapter demonstrates that a tractable model can be developed that reduces operational restrictions necessary for tractability in existing models. With this new model, truck costs can be reduced with coordination that requires minimal changes to the operating standards of the current harvest operations.

There are two areas of future work. First, as is demonstrated in the results, even small-sized blocks have some variability in arrivals and importantly in the time between arrivals. Ideally, arrivals throughout the day would be equally spaced. Preliminary work showed such a model to be intractable. In future work, we propose to explore valid inequalities and cuts for such a model in the hope of obtaining tractability. It is notable that the model proposed in this chapter can be used to generate initial feasible solutions for the cases of equally spacing arrivals. A second area of future work rests in methods for managing real-time harvest logistics. The proposed model offers a planning tool for determining the start times of harvests, but is less helpful in handling the unknown events that are certain to arise throughout a day's operations.

# CHAPTER 4
# A CONTINUOUS TIME MODEL FOR SUGARCANE HARVEST LOGISTICS IN THE UNITED STATES AND AUSTRALIA

## 4.1 Introduction

Despite a growing global appetite for sugar as both a foodstuff and a fuel source [Shapouri and Salassi, 2006, Jacobs, 2006, Service/USDA, 2014], there exists limited literature that explores sugarcane operations. In this chapter, we look at harvest operations in the US and Australia. Sugarcane harvests in the US and Australia have three operations that must be coordinated: infield operations, over-the-road transport, and mill operations. Infield operations usually occur in several pre-specified farms and have numerous components. First, the cane is cut in the field, usually using a mechanical harvester that cuts the cane into uniformly sized billets (12-18 inches). While in operation, the harvester continuously feeds billets into a cart pulled by an infield transporter. This infield transporter and cart combination runs alongside the harvester, and, when the cart is filled, the transporter and cart combination must be rotated with another infield vehicle and its associated cart for continuous harvest operations. Filled carts are transported to a loading pad that serves the farm. At the loading pad, the sugarcane is transferred to trucks that take the harvested cane from the farms to the mill. The final operation of the harvest takes place at the mill where the trucks are unloaded. Once a truck is unloaded, it can return to a farm for its next load.

Harvest operations on farms are generally conducted only during daylight hours, and most farms begin harvesting operations as early in the morning as possible. One of the key challenges in both countries is the lack of coordination among growers as well as between growers and the mill. As a result, there can be a long queue of trucks waiting to be unloaded at the mill yard. This extra waiting time at the mill reduces the number of loads that can be hauled by each individual truck. Thus, the existing harvest and transport arrangement increases the number of trucks required to haul the mill's daily quota of sugarcane. Collaboration between farmers on the one hand and the mill on the other could improve the overall efficiency of harvested cane transport operations by reducing number of trucks required to haul the cut cane.

In this chapter, we seek to reduce congestion at the mill and as well as the number of trucks required to serve the harvest. We seek to reduce mill congestion rather than to model the trucks directly because the latter leads to intractable models. We consider a set of fields which provide a pre-specified set of loads to the mill. The farms harvest at a fixed rate. All the trucks start their shifts at the mill. The travel time between the farms and the mill is deterministic. The trucks arriving at the mill form a single first in first out queue. When a truck is unloaded, it is available for the next dispatch. The cycle continues until all the loads are picked up from the farms are unloaded at the mill. Our objective is to maximize the minimum time between consecutive truck arrivals to the mill. The objective is maximized by setting the start times of the harvests at the farms. Given the solution to the math program, we generate truck assignments.

This chapter makes two contributions to the literature. We demonstrate that the mills and the growers can achieve significant savings by spreading the harvesting throughout the daylight hours. We show that we can achieve this savings by coordinating start times at the fields. Through start time coordination, we spread arrivals of trucks, reducing congestion, and thus reducing the number of trucks required to serve the harvest. Our computational results show that setting the start times of harvests at the various farms is sufficient to achieve the necessary coordination. These validate Salassi and Barker's conjecture (2008) that truck congestion at the mill could be reduced by coordinating the start times of the harvests at the farms. Second, we introduce a model that eliminates the discretization required in Salassi and Barker [2008] and in Lamsal et al. [2013b]. We demonstrate that eliminating discretization reduces the number of trucks. We also introduce a series of valid inequalities that lead to a practical model. As a minor contribution, we demonstrate the value of using the model presented in Lamsal et al. [2013b] to generate initial feasible solutions.

Section 4.2 of the paper discusses previous work on sugarcane logistics. Section 4.3 presents our model as well as valid inequalities and optimality cuts. In Section 4.4, we describe the solution approach. In Section 4.5, we present the results of two computational studies using our model. The first study uses the benchmark problems developed by Lamsal et al. [2013a]. These benchmark datasets use publicly available data on the geographical locations of each of Louisiana's 456 sugarcane farms and 11 sugarcane mills as well as their production and processing rates to construct a set of 11 sugarcane logistics problems (one for each of the 11 mills in Louisiana). The second study uses a set of benchmark problems used by Higgins and Laredo [2006] to study harvest logistics in Australia. We compare the effectiveness of solutions developed by our approach to those developed by Higgins and Laredo [2006] for the same problem. Section 4.6 presents our conclusions.

## 4.2 Literature Review

Trying to maintain the desired delivery schedule at the mill while reducing the number of harvesters and the size of the fleet of the trucks is a recurring theme in the literature related to sugarcane harvesting and transportation system. In different countries, the sugarcane harvesting and transportation system has varying divisions of decision making between farm and mill levels. A lack of coordination among the decision makers affects the efficiency of the whole system. For a more detailed discussion of the literature, specially on the other objective functions used in the sugarcane harvest and transportation models and the discussion on the literature related to the infrastructure different from those in the US and Australia, we refer the reader to Lamsal et al. [2013a].

Most closely related to the work in this chapter are Salassi et al. [2009a] and Lamsal et al. [2013b]. Both papers use mixed integer mathematical programming models to evaluate the impact of alternative harvest schedules at the farms that result in shorter queues at the mill of the trucks waiting to be unloaded thus reducing the total truck hours and the number of trucks needed to haul the cane. Salassi et al. [2009a] and Lamsal et al. [2013b] divide the day into blocks of time and use discretization techniques to spread arrivals among these blocks of time. Lamsal et al. [2013b]

show that as the time blocks become smaller, the model produces more desirable results, in the sense that the loads arrivals are spread more uniformly throughout the day and also require fewer trucks. On the flip side, the complexity of the problem increases when the size of the time blocks decreases, eventually leading to a computationally intractable problem. Our objective is motivated by the results in Lamsal et al. [2013b]. In this chapter, we make the problem continuous by removing the notion of time blocks and maximally spread the load arrivals by maximizing the smallest gap between two successive arrivals at the mill.

Also closely related to the work in this chapter is Higgins et al. [2004] and Higgins and Laredo [2006]. The two papers develop a framework for integrating a complex harvesting and transportation system for sugar production in Australia. They seek to reduce the congestion at the mill. They use heuristic methods to produce transportation schedules such that mill idle time, queue length and the number of trucks needed to haul the cut cane are reduced. We add to their work by coordinating harvest schedules with the transport schedules to further reduce the queue length and number of trucks needed.

Also related to the work in this chapter is Lamsal et al. [2013a]. However, Lamsal et al. [2013a] focuses on sugarcane operations in Brazil where the infrastructure, notably the level of vertical integration, differs from that in the US and Australia. Consequently, Lamsal et al. [2013a] develop a model that coordinates the load arrivals at the mill with the objective of minimizing the cut-to-crush delay.

Other papers in the literature have explored sugarcane harvests from a tactical rather than operational level. One of the first such models for sugarcane harvesting and transportation is described in Whitney and Cochran [1976]. They use queuing theory to predict delivery rates of the harvested cane. Their model is meant for a transport system of tractors and wagons and one continuous road between the farm and the sugar factory. Loading times are assumed to follow an exponential distribution, and arrival times of the trucks at the farms and the mill are Poisson distributed. They use their model outputs to forecast the rate at which sugarcane can be delivered from the farms. Arjona et al. [2001] developed a discrete event simulation model of the harvesting and transportation system of a sugarcane farm in Mexico that covers all processes from the burning of the cane to its unloading at the processing line. Their solutions show that machinery is underutilized. They suggest methods to improve the efficiency of machinery in use, thereby allowing a reduction in the number of machinery without increasing sugarcane processing times. Both of these papers look at the harvest and transportation problem from a broader approach. Instead of developing a harvest schedule and truck assignments, they study the impact of a given set of resources on the on delivery rate, and how varying these available resources would impact on the delivery rate. In our paper, we are more concerned with developing an implementable harvest and truck assignment schedule.

## 4.3 Model

In this section, we present a formal model of the problem. Our goal is to minimize the number of trucks required to pick up loads at the times when they become ready. However, directly modeling the minimization of the number of trucks results in intractable problem. As a result, we propose an alternative objective that minimizes congestion at the mill and has the effect of greatly reducing the number of trucks. Our objective maximizes the minimum time between arrivals to the mill. In this model, we eliminate the notion of time blocks and maximally spread the load arrivals by maximizing the smallest gap between the two successive arrivals to the mill. We introduce the notation and a basic model and discuss the constraints.

### 4.3.1 Base Model

**Sets:**

$F$ set of farms.

**Parameters:**

| | | |
|---|---|---|
| $h_i$ | $i \in F$ | time to harvest one load at the farm $i$ |
| $t_i$ | $i \in F$ | travel time from mill to the farm $i$ |
| $n_i$ | $i \in F$ | daily load quota of the farm $i$ |
| $[a_{ij}, b_{ij}]$ | $i \in F, j \in \{1 \dots n_i\}$ | feasible harvest bounds for load $ij$ |
| $U_{iji'j'}$ | $i, i' \in F, j \in \{1 \dots n_i - 1\}, j' \in \{1 \dots n_i' - 1\}, i < i'$ | upper bound to the difference of $x_{ij}$ and $x_{i'j'}$ |
| $L_{iji'j'}$ | $i, i' \in F, j \in \{1 \dots n_i - 1\}, j' \in \{1 \dots n_i' - 1\}, i < i'$ | lower bound to the difference of $x_{ij}$ and $x_{i'j'}$. |

**Variables:**

| | | |
|---|---|---|
| $y_i$ | $i \in F$ | time when harvesting starts at farm $i$ |
| $z_{ij}$ | $i \in F, j \in \{1 \dots n_j\}$ | ready time for load $j$ from farm $i$ |
| $x_{ij}$ | $i \in F, j \in \{1 \dots n_j\}$ | arrival time at the mill for load $j$ from farm $i$ |
| $S^+_{iji'j'}$ | $i, i' \in F, j \in \{1 \dots n_i\}$, $j' \in \{1 \dots n_{i'}\}, i < i'$ | dummy variable that takes the value as the difference between $x_{ij}$ and $x_{i'j'}$ if $x_{ij} > x_{i'j'}$ and zero otherwise |
| $S^-_{iji'j'}$ | $i, i' \in F, j \in \{1 \dots n_i\}$, $j' \in \{1 \dots n_{i'}\}, i < i'$ | dummy variable that takes the value as the difference between $x_{ij}$ and $x_{i'j'}$ if $x_{ij} < x_{i'j'}$ and zero otherwise |
| $B_{iji'j'}$ | $\in \{0, 1\}$ $i, i' \in F$, $j \in \{1 \dots n_i\}$, $j' \in \{1 \dots n_{i'}\}, i < i'$ | binary variable that takes the value of 1 if $x_{ij}$ is larger than $x_{i'j'}$ and 0 if $x_{ij}$ is smaller than $x_{i'j'}$ |
| $obj$ | | objective value, minimum gap between two consecutive arrivals. |

**Objective:**

$$\max \quad Obj$$

**Constraints:**

$$z_{ij} = y_i + j \times h_i \qquad \forall (i,j) \mid i \in F,\ j \in \{1 \ldots n_i\} \tag{4.1}$$

$$x_{ij} = z_{ij} + t_i \qquad \forall (i,j) \mid i \in F,\ j \in \{1 \ldots n_i\} \tag{4.2}$$

$$a_{ij} \leq x_{ij} \leq b_{ij} \qquad \forall (i,j) \mid i \in F,\ j \in \{1 \ldots n_i\} \tag{4.3}$$

$$x_{ij} - x_{i'j'} = S^+_{iji'j'} - S^-_{iji'j'} \qquad \forall (i,j),(i',j') \mid i,i' \in F,\ j \in \{1 \ldots n_i\}, j' \in \{1 \ldots n_{i'}\}, i < i' \tag{4.4}$$

$$0 \leq S^+_{iji'j'} \leq U_{iji'j'} \cdot B_{iji'j'} \qquad \forall (i,j),(i',j') \mid i,i' \in F,\ j \in \{1 \ldots n_i\}, j' \in \{1 \ldots n_{i'}\}, i < i' \tag{4.5}$$

$$0 \leq S^-_{iji'j'} \leq |L_{iji'j'}| \times (1 - B_{iji'j'}) \quad \forall (i,j),(i',j') \mid i,i' \in F,\ j \in \{1 \ldots n_i\}, j' \in \{1 \ldots n_{i'}\}, i < i' \tag{4.6}$$

$$Obj \leq S^+_{iji'j'} + S^-_{iji'j'} \qquad \forall (i,j),(i',j') \mid i,i' \in F,\ j \in \{1 \ldots n_i\}, j' \in \{1 \ldots n_{i'}\}, i < i'. \tag{4.7}$$

Constraints 4.1 relate the harvest start times of the farms to the ready times of all the loads from the respective farms. Constraints 4.2 relate the ready times of the loads with the loads' arrival times at the mill. Constraints 4.3 set lower and upper bounds for the arrival times for each load. In Contraints 4.4, we represent the difference between two arrival times as the difference of two non-negative variables. We note that we do not define Contraints 4.4 for $i' \leq i$. Such constraints are unnecessary. We do not consider the situation when $i = i'$ because the difference between two closest arrivals from the same farm is fixed. Further, for two arbitrary arrival times $x_{ij}$

and $x_{i'j'}$ and $i' < i$, $x_{ij} - x_{i'j'} = -\left(x_{i'j'} - x_{ij}\right)$ and $S^{+}_{iji'j'} - S^{-}_{iji'j'} = -\left(S^{+}_{i'j'ij} - S^{-}_{i'j'ij}\right)$.

Constraints 4.5 and Constraints 4.6 force one of the two non-negative variables from Constraints 4.4 to be zero. Unlike in min-max formulations, in the max-min objective, increasing $S^{+}_{iji'j'}$ or $S^{-}_{iji'j'}$ improves the objective value. Thus, we need to introduce constraints to force one of the variables in each pair to be zero. The variable $S^{+}_{iji'j'}$ is positive and $S^{-}_{iji'j'}$ is zero if $x_{ij}$ is larger than $x_{i'j'}$ , and if $x_{ij}$ is smaller than $x_{i'j'}$, $S^{-}_{iji'j'}$ is positive and $S^{+}_{iji'j'}$ is zero. The binary variable $B_{iji'j'}$ takes the value 1 when $x_{ij}$ is larger than $x_{i'j'}$ and zero when $x_{ij}$ is smaller than $x_{i'j'}$. Constraints 4.7 forces the objective to be larger than the absolute difference of any two arrivals.

### 4.3.2 Valid inequalities and Optimality Cuts

In this section, we present results that help strengthen the present formulation. In the first result, we state and prove a proposition demonstrating monotonicity among the binary variables $B$. The result takes advantage of the fact that all loads from any given farm must picked up at their ready time and the physical constraint of the harvest time for each load. These cuts are added at the root nodes.

**Proposition 7 (Monotonicity).** *For all $i, i', j$, and $j'$ such that $i, i' \in F$, $i < i'$, $j \in 1, \ldots, n_i - 1$, and $j' \in 1, \ldots, n_{i'}$*

$$B_{iji'j'} \leq B_{i(j+1)i'j'}.$$

*Similarly, for all $i, i', j$, and $j'$ such that $i, i' \in F$, $i < i'$, $j \in 1, \ldots, n_i$, and $j' \in$*

$1, \ldots, n_{i'} - 1$

$$B_{iji'j'} \geq B_{iji'(j'+1)}.$$

*Proof.* Consider a series of arrivals from farm $i$, $x_{i,1}, x_{i,2}, \ldots, x_{i,n_i}$. By Constraints 4.1 and 4.2, we know that $x_{i,1} < x_{i,2} < \cdots < x_{i,n_i}$. Next, consider any load from farm $i'$. Let this be load $j'$. The arrival time for the $j'^{th}$ load from farm $i'$ is $x_{i',j'}$. Subtracting the arrival time $x_{i',j'}$ from the arrival times of each of the loads from farm $i$ gives us $(x_{i,1} - x_{i',j'}) < (x_{i,2} - x_{i',j'}) < \cdots < (x_{i,n_i} - x_{i',j'})$.

As a result of constraints 4.5 and 4.6, for any load $j$ from farm $i$, $B_{i,j,i',j'} = 1$ if $x_{i,j} - x_{i',j'}$ is positive and 0 otherwise. Then, because $(x_{i,j} - x_{i',j'}) < (x_{i,j+1} - x_{i',j'})$ for every $j \in \{1, \ldots, n_i - 1\}$, $B_{iji'j'} \leq B_{i(j+1)i'j'}$.

The second part of the proof follows analogously. Again as a result of Constraints 4.1 and 4.2, we have the following series of inequalities $(x_{i,j} - x_{i',1}) > (x_{i,j} - x_{i',2}) > \cdots > (x_{i,j} - x_{i',n_{i'}})$, which implies $B_{i,j,i',1} \geq B_{i,j,i',2} \geq \cdots \geq B_{i,j,i',n'_i}$.

As a result of Proposition 7, we add following sets of valid inequalities to the base model:

$$B_{iji'j'} \leq B_{i(j+1)i'j'} \; \forall (i,j),(i',j') \mid i,i' \in F, j \in \{1 \ldots n_i - 1\}, j' \in \{1 \ldots n_{i'}\}, i < i' \tag{4.8}$$

$$B_{iji'j'} \geq B_{iji'(j'+1)} \; \forall (i,j),(i',j') \mid i,i' \in F, j \in \{1 \ldots n_i\}, j' \in \{1 \ldots n_{i'} - 1\}, i < i'. \tag{4.9}$$

We next present two optimality cuts that use the value of a feasible solution to bound the number of arrivals to the mill that can occur between to successive arrivals from a given farm. The first result bounds the number of arrivals that occur from a single farm in the interval between two successive arrivals from another. The second results bounds the number of arrivals from all farms that can occur in the interval between two successive arrivals from any farm. In both cases, we take advantage of the objective value of a feasible solution and also the fact that the harvest rates at each farm are constant and that we require loads to be picked up when they are ready.

**Proposition 8.** *Given a feasible solution value $\underline{obj}$ and for two successive loads arriving to the mill from farm $i$, the number of maximum arrivals originating from any farm $i' \neq i$ is bounded by $\left(\left\lfloor \frac{h_i - 2\times\underline{obj}}{h_{i'}} \right\rfloor + 1\right)$.*

*Proof.* Let $x_{i,j}$ and $x_{i,j+1}$ be any two successive arrivals to the mill from the farm $i$ . By construction, we know that $x_{i,j+1} - x_{i,j} = h_i$. Further, the time between any two arrivals must also be greater than the given objective value of a feasible solution $\underline{obj}$. Then, there exists at most $h_i - 2 \times \underline{obj}$ units of time in which loads can arrive. We also know from the data that farm $i'$ produces a load every $h_{i'}$ time units and thus all arrivals from farm $i'$ are separated by at least $h_{i'}$.

Thus, if $\frac{h_i - 2\times\underline{obj}}{h_{i'}}$ is non-integer, no more than $\left(\left\lceil \frac{h_i - 2\times\underline{obj}}{h_{i'}} \right\rceil\right)$ loads can arrive from farm $i'$ between two successive loads from farm $i$. However, if $\frac{h_i - 2\times\underline{obj}}{h_{i'}}$ is integer, we must account for the fact that a load can arrive exactly at time $x_{i,j} + \underline{obj}$ and resultantly the bound becomes $\left(\left\lceil \frac{h_i - 2\times\underline{obj}}{h_{i'}} \right\rceil + 1\right)$. However, this bound is not tight in the non-integer case. We can tighten the bound by instead using $\left(\left\lfloor \frac{h_i - 2\times\underline{obj}}{h_{i'}} \right\rfloor + 1\right)$.

To introduce inequalities that take advantage of the result in Proposition 8, we first note that $\sum\limits_{j' \in 1..n_{i'}} B_{i(j+1)i'j'}$ counts the total number of arrivals prior to $x_{i,j+1}$ from farm $i'$. Similarly, the term $\sum\limits_{j' \in 1..n_{i'}} B_{iji'j'}$ counts the number of arrivals prior to $x_{i,j}$ from farm $i'$. Thus, the sum

$$\left( \sum_{j' \in 1..n_{i'}} B_{i(j+1)i'j'} - \sum_{j' \in 1..n_{i'}} B_{iji'j'} \right)$$

reflects the total number of loads from farm $i'$ that arrive to the mill between $x_{i,j}$ and $x_{i,j+1}$. Thus, as a result of Proposition 8 and when a feasible solution exists, we add the following set of optimality cuts to the base model:

$$0 \leq \sum_{j' \in 1..n_{i'}} B_{i(j+1)i'j'} - \sum_{j' \in 1..n_{i'}} B_{iji'j'} \leq \left( \left\lfloor \frac{h_i - 2 \times \underline{\underline{obj}}}{h_{i'}} \right\rfloor + 1 \right)$$

$$\forall (i,j),\ \&i' \mid i, i' \in F, j \in \{1 \ldots n_i - 1\}, i < i'. \tag{4.10}$$

Similar to Proposition 8, we can bound the number of arrivals from all farms that can occur between two successive loads from a given farm. The proof is analogous to that of Proposition 8 and is omitted.

**Proposition 9.** *Given a feasible solution value $\underline{obj}$ and for two successive loads arriving to the mill from farm $i$, the number of maximum arrivals originating from all other farms is bounded by Given a feasible solution value $\underline{obj}$ and for two successive loads arriving to the mill from farm $i$, the number of maximum arrivals originating* *from any farm* $i' \neq i$ *is bounded by* $\left(\left\lfloor \frac{h_i - 2\times \underline{\underline{obj}}}{\underline{obj}} \right\rfloor + 1\right)$.

As was the case with Constraints 4.10, to implement Proposition 9, we need to count the arrivals that occur between two successive loads from the same farm. We make use of the following sums:

$$\sum_{i'>i} \sum_{j'\in 1..n_{i'}} \left(B_{i(j+1)i'j'}\right), \tag{4.11}$$

$$\sum_{i'>i} \sum_{j'\in 1..n_{i'}} \left(B_{iji'j'}\right), \tag{4.12}$$

$$\sum_{i'<i} \sum_{j'\in 1..n_{i'}} \left(1 - B_{i'j'i(j+1)}\right), \text{ and} \tag{4.13}$$

$$\sum_{i'<i} \sum_{j'\in 1..n_{i'}} \left(1 - B_{i'j'ij}\right). \tag{4.14}$$

The sum 4.12 counts the total number of arrivals prior to $x_{i,j+1}$, and the sum 4.13 counts the total number of arrivals prior to $x_{i,j}$ from the farms with index $i'$ greater than $i$. The sum 4.14 counts the total number of arrivals prior to $x_{i,j+1}$, and the sum 4.14 counts the total number of arrivals prior to $x_{i,j}$ from the farms with index $i'$ less than $i$.

Thus, as a result of Proposition 9, we add the following optimality cuts when a feasible solution is available:

$$\begin{aligned} 0 \leq & \sum_{i'>i} \sum_{j'\in 1..n_{i'}} B_{i(j+1)i'j'} - \sum_{i'>i} \sum_{j'\in 1..n_{i'}} B_{iji'j'} + \sum_{i'<i} \sum_{j'\in 1..n_{i'}} \left(1 - B_{i'j'i(j+1)}\right) \\ & - \sum_{i'<i} \sum_{j'\in 1..n_{i'}} \left(1 - B_{i'j'ij}\right) \leq \left(\left\lfloor \frac{h_i - 2\times \underline{\underline{obj}}}{\underline{obj}} \right\rfloor + 1\right) \forall (i,j), \mid i, i' \in F, j \in \{1 \ldots n_i - 1\}. \end{aligned} \tag{4.15}$$

## 4.4 Experimental Design and Solution Approach

We first, compare our solutions with the solutions in Lamsal et al. [2013b] for the number of trucks needed to pick up all the loads at their ready times. A side-by-side comparison of number of trucks needed is presented in Table 4.2. The approach presented in this chapter reduces the number of trucks in all but one instance (Instance 3), in which case the number of trucks are equal. On average, the number of trucks is reduced by 7%.

To understand why the approach presented in this chapter reduces the number of trucks, we compare the two solution methods with respect to truck utilization. Figure 4.1 compares the time spent by each truck arriving to the mill in our solution and the solution in Lamsal et al. [2013b] for the first instance. In this comparison, we make the truck assignments for both our and Lamsal et al.'s (2013b) solution by assuming a FIFO queue with unloading time of two minutes per load at the mill. It is noticeable that the waiting time for each load is shorter in our solution in-spite of having the same unloading time. The regular pattern for the hourly block solution is because of the staggering of the loads at the hour ends.

We next describe the Australian instance described in Higgins [2006]. Australian industry practices are different from those in the US. In the US, enough trucks are scheduled to pick up all loads as soon as they become ready. As a result, any waiting time for a given load occurs at the mill, not in the field. In Australia, even though the harvesting is done only during the day light hours, the hauling continues through out the night. So, the cut cane waits at the loading pads at the farms waiting to be picked up. In Higgins [2006], the harvesting hours and the harvest rates for each farm is treated as given and the model solves for the truck assignment. In this chapter, we take the length of harvesting hours and the harvest rates as given and solve for the time when the harvesting should begin to reduce the congestion at the mill, maintaining the restriction that harvesting must occur during daylight hours. In our truck assignment, we pick up the loads at their ready times and do not allow them to wait at the farms. As such, our truck assignments are more constrained than those in Higgins [2006].

We seek to answer two questions. First, what is the computational value of the proposed valid inequalities and optimality cuts? The computational value of the proposed valid inequalities and optimality cuts is readily apparent from the fact that the problem is insolvable without these additions to the base model. We show that each of the realistic instances can be solved to optimality in about two hours.

There is however a remaining question as to whether the continuous model has an advantage over the discrete model in Lamsal et al. [2013b]. This question is motivated by the observation in Lamsal et al. [2013b] that, as time blocks become smaller, the model produces solutions using fewer trucks and spreading load arrivals more uniformly throughout the day. Lamsal et al. [2013b] also shows that the complexity of the problem increases when the size of the time blocks decreases to the extent that making the size of the time blocks smaller than 10 minutes (thus resulting large number of blocks) produced unsolvable problems. For practical purposes, the continuous model proposed in this chapter is equivalent to having infinitesimal time blocks.

To compare the approach presented in this chapter to those in the literature, we use the 11 instances developed in Lamsal et al. [2013b] to represent the approximately 475 farms in 11 mill areas in Louisiana. United States Department of Agriculture, Economic Research Service [2010] and League [2013] provide zip-code level addresses for 456 farms and exact addresses of the 11 mills. In total, we have 456 farms in 85 zip codes with a daily capacity of 4044 loads. The zip codes with sugarcane farms and the location of the mills are shown in Figure 4.2. Each start represents the location of a mill and the represents the centroid of the zip code that has at least one sugarcane farm. Using this data as well as additional data from [Barker, 2007] and Salassi and Barker [2008], each farm was assigned a daily harvest volume, either one or two harvesters, and a per load harvest time determined. Table 4.1 summarizes the 11 mill areas.

Table 4.1: Distribution of farms and total loads

| Instance | # of farms | # of loads |
|---|---|---|
| 1 | 55 | 370 |
| 2 | 66 | 367 |
| 3 | 29 | 369 |
| 4 | 39 | 365 |
| 5 | 23 | 365 |
| 6 | 69 | 370 |
| 7 | 28 | 370 |
| 8 | 53 | 370 |
| 9 | 27 | 365 |
| 10 | 26 | 365 |
| 11 | 41 | 368 |

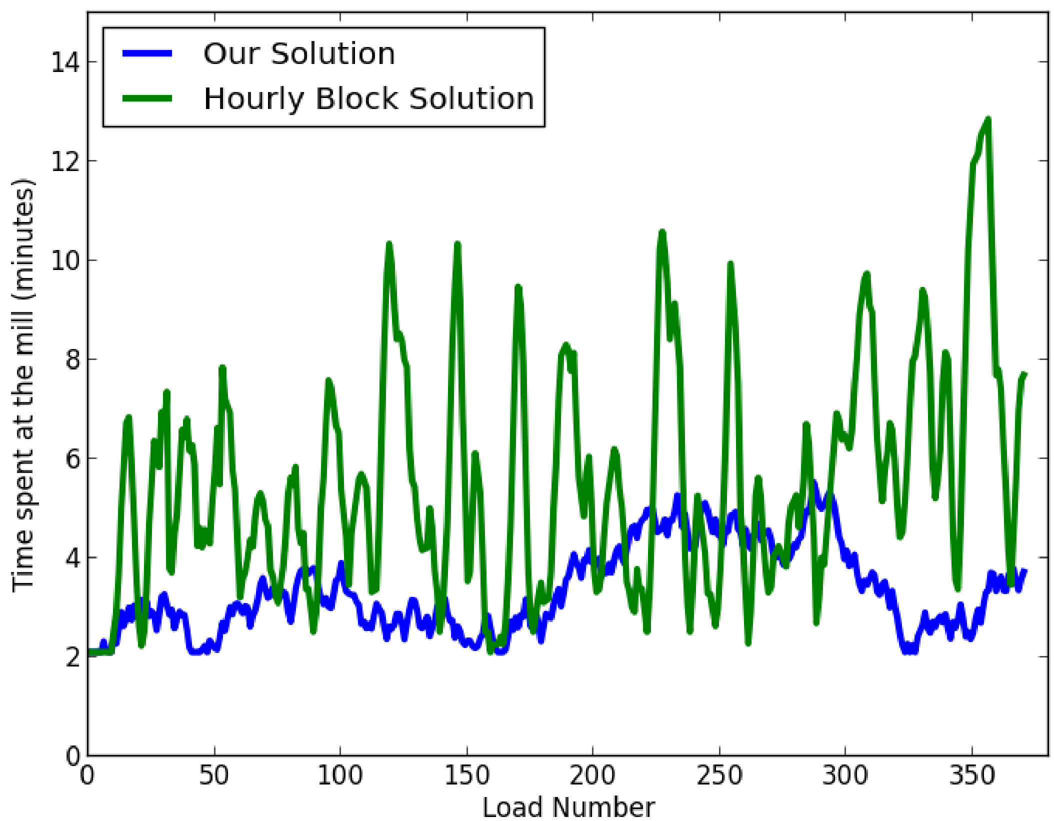


Figure 4.1: Time spent at the mill yard for individual load

Table 4.2: Comparison of number of trucks needed to haul the cane in ready times

| Instance | # of trucks needed (old) | # of trucks needed (new) |
|---|---|---|
| 1 | 32 | 31 |
| 2 | 36 | 33 |
| 3 | 31 | 31 |
| 4 | 34 | 29 |
| 5 | 35 | 32 |
| 6 | 48 | 42 |
| 7 | 30 | 30 |
| 8 | 35 | 34 |
| 9 | 38 | 34 |
| 10 | 28 | 26 |
| 11 | 33 | 30 |

To solve the instances, we use a branch-and-bound algorithm. We note that we significantly reduce solution time by using an initial feasible solution. Moreover, to instantiate the Constraints 4.10 and 4.15, we need a non-zero lower bound to the objective. We use the model and algorithm described in Lamsal et al. [2013b] with hourly time blocks to get a feasible solution to the model presented in this chapter. Because solutions to the model in Lamsal et al. [2013b] can have two arrivals that occur at the same time, we iteratively perturb the start times of the farms whose loads have the same arrival times until we have a solution in which no two loads have the same arrival times. As there are infinite real numbers, we are guaranteed to find a non-zero solution. In our implementation, we decrease the start time of the lower numbered farm greedily by $\epsilon = [0.05, 0.25]$ and increase the start time of the higher numbered farm by $\epsilon$, when two loads have same arrival times. Constraints 4.10 and 4.15 are instantiated using the objective of the initial feasible solution as the lower bound.. When a better lower bound is found, the constraints are updated and

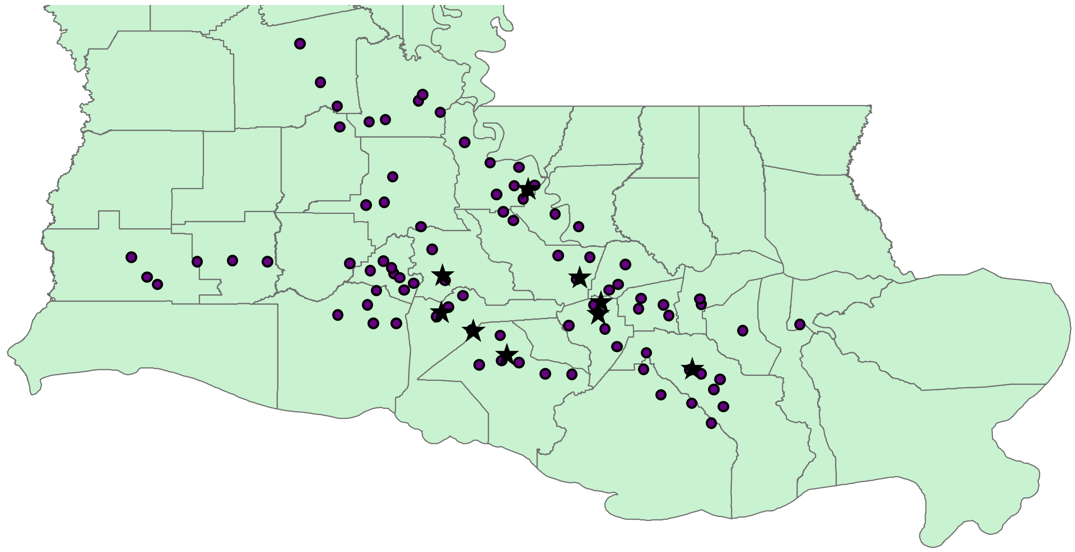

Figure 4.2: Distribution of farms and the mills in southern Louisiana

added as new optimality cuts. The algorithm stops when the optimality condition is satisfied.

Given a solution to the math program, we can compute the optimal number of trucks needed to pick up the loads at their ready times at the growers by using the algorithm presented in Lamsal et al. [2013a]. The truck assignment algorithm is coded in Python and runs instantaneously on the previously described hardware. The math programs are solved using GUROBI OPTIMIZER 5.6 using the Python interface. The experiments are performed on a 3.40 GHz Intel Core i7-3770 CPU running the Ubuntu 12.04 operating system.

## 4.5 Computational Results

This section presents computational results for both the US and Australian instances. Further evidence of the value of the method presented in this chapter can be seen by comparing variation in truck hours. We define truck hours for a truck as the time between when the last load hauled by the truck is unloaded at the mill and the time when the truck is dispatched from the mill to pick up the truck's first load. A better solution would reduce the variability in truck hours across all the trucks. That is, the trucks would all work about the same number of hours. One of the weaknesses of the solutions in Lamsal et al. [2013b] is that a significant number of trucks serve a single load. Thus, variability in truck hours is high in those solutions. Such reduction in variability in truck hours should be desirable because it would be useful to equitably divide work among drivers.

In Table 4.3, "old STDEV" refers to the standard deviation of the truck hours and "old Max - Min" refers to the difference between the maximum and minimum truck hours for each solution using the best solutions from [Lamsal et al., 2013b]. Similarly, "new STDEV" and "new Max - Min" refer to the standard deviation and the difference between the maximum and minimum of the truck hours for our solutions. The approach presented in this chapter reduces this variability by an average of 19% across the 11 mill areas. The difference between the maximum and minimum truck hours is also reduced by about 11.66 %.

Figure 4.3 plots the cumulative arrivals with three different solutions for the first instance. The line labelled as "Earliest Start for all farms," represents the so-

Table 4.3: Comparison of standard deviations and differences in working time

| Instance | old STDEV | old Max - Min | new STDEV | new Max - Min |
|---|---|---|---|---|
| 1 | 131.35 | 601 | 95.79 | 478 |
| 2 | 159.28 | 689 | 130.38 | 635 |
| 3 | 140.76 | 579 | 90.90 | 460 |
| 4 | 136.35 | 512 | 125.95 | 460 |
| 5 | 125.85 | 438 | 116.29 | 374 |
| 6 | 132.27 | 588 | 96.83 | 557 |
| 7 | 114.76 | 418 | 96.65 | 378 |
| 8 | 135.59 | 543 | 110.02 | 508 |
| 9 | 170.42 | 633 | 137.14 | 633 |
| 10 | 147.01 | 570 | 139.21 | 412 |
| 11 | 134.17 | 553 | 116.13 | 522 |

lution that simulates the current practice in which all farms start harvesting at the beginning of the day. The line labelled as "Hourly Block Solution" represents time block solution obtained using the solution method described in Lamsal et al. [2013b] (Hourly blocks and 29 loads per hour limit). The line labelled as "Our Solution" represents the solution from our proposed solution method. Our best estimate for the number of trucks needed to pick up all the loads for the first solution is 62 trucks. Similarly, we need 32 and 31 trucks, respectively for the second and third solution.

In "Earliest Start for all farms" solution, most loads arrive at the mill within the 500 minutes. This causes congestion at the mill increasing the turn around times for the trucks, thus increasing the number of trucks required to haul all the cane. In "Hourly Block Solution," the hourly truck arrival rate is constant but within the hour, truck arrivals are not spread out. So, there are times, when the unloading resource at the mill is idle and there are also times when there is congestion as loads arrive simultaneously. In "Our Solution," the trucks arrive at the constant rate

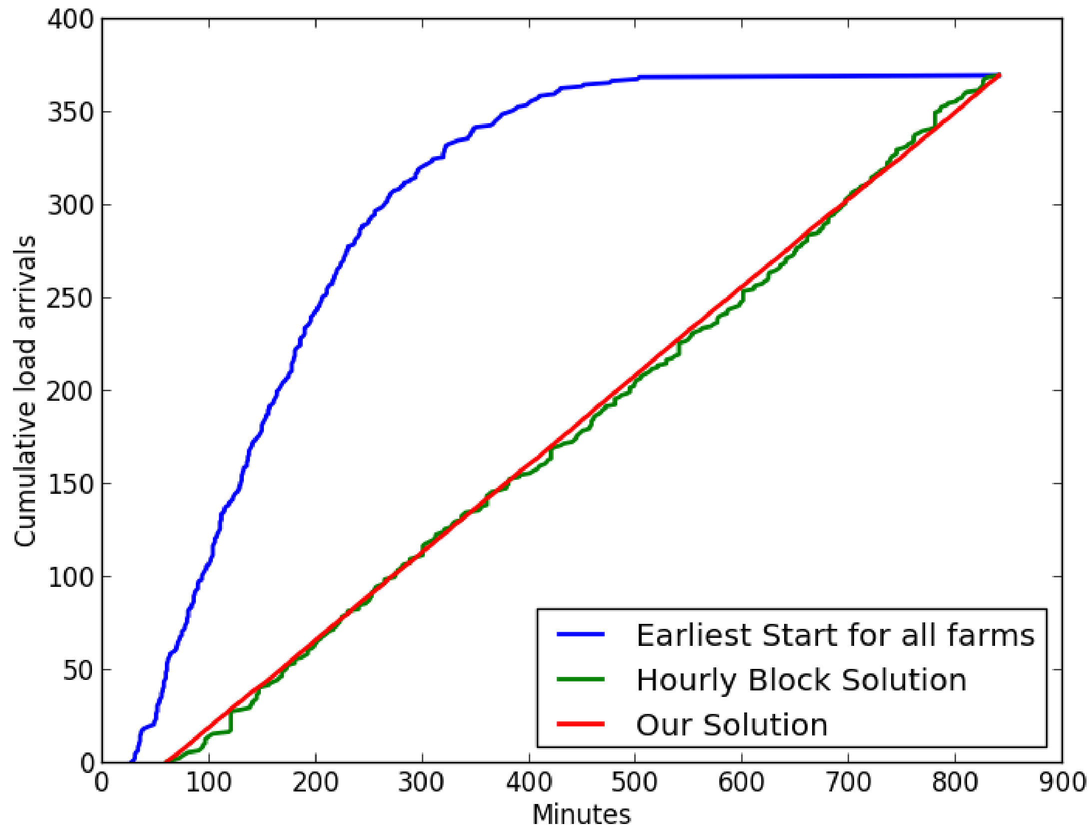


Figure 4.3: Cumulative arrivals at the Mill throughout the daylight hours

which reduces the chances of unloading resource at the mill being idle or the chances of congestion.

In the Australian instance of Higgins [2006], a total of 21 trucks are used for the total of approximately 505 hours of truck time, with the average shift length of each truck being slightly higher than 24 hours. By using the approach presented in this chapter, we can pick up all the loads with 19 trucks using approximately 438 hours of truck time and none of the trucks needing a shift longer than 24 hours.

## 4.6 Conclusions

Optimizing operations in a sugar mill area is a difficult task involving several stake holders with competing interests. Previous literature in the area uses a discrete time approach that results in problems becoming computationally intractable as the time discretization becomes finer. This chapter uses an objective function, maximizing the minimum difference between two consecutive arrivals at the mill, which allows the problem to be solved in continuous time; thereby obviating difficulties encountered using previous approaches. Our results show that this new approach provides solutions that not only reduce the number of trucks needed to conduct the harvest, but that also reduce variation of truck utilization. Reducing such variation is important for a variety of efficiency and operational reasons, but also because it spreads the workload more evenly amongst truck drivers, thereby increasing perceived fairness and equity. Additionally, our results show that these advantages can be obtained with only minimal coordination between the mills and farms: namely the farms must allow the mill to set the time of day at which the sugar cane harvest starts. Since the farms are independently owned, such minimal coordination requirements are important if the solution is to be workable in a practical setting.

There are two areas of future work. First, the model can be extended to include random events that commonly occur such as equipment breakdowns. Second, methods for managing harvest logistics in real time are needed to deal with randomly occurring events. Although the model in this chapter could be used to determine the start time of the daily harvest, additional work is needed before it could handle unknown events that might arise during a day's operations as they occur. Relaxing the strong assumption of no wait at the farm also offers area of future exploration.

# CHAPTER 5
# HARVEST LOGISTICS UNDER UNCERTAINTY IN BRAZIL

## 5.1 Problem Description

In this chapter, we return to the problem of harvesting and delivery sugarcane to a mill in Brazil. This problem deals with harvesting operations (cutting and shipping) at the fronts, transportation by the trucks via road network, and unloading of the cane at the mill yard while accounting for the stochasticity in day-to-day operations.

In practice, the sugarcane harvest and transport are not deterministic, as we have described in Chapter 2. There are uncertainties in both operations. Harvesters break down at the fronts. The weather causes the shut down of harvesting at the front. The vehicles break down en-route to the mill or to the fronts. Many of these events need a change in the original plan of actions.

We develop a rolling horizon framework that uses the math program in Chapter 2 to solve certainty equivalent problems as new information becomes available. We update constraints to account for changes in trucks or fronts.

## 5.2 Model

We model the problem as a Markov decision process. Let $G = (\mathcal{N}, \mathcal{E})$ be a *complete* graph where $\mathcal{N} = \{0, 1, ..., N\}$ is a set of $N+1$ nodes and $\mathcal{E} = \left\{ \left(n, n^{'}\right) : n, n^{'} \in \mathcal{N} \right\}$ a set of edges connecting the nodes in the network. Node *0* represents mill and nodes $1, \ldots, N$ represent the fronts. Travel times $t(n, n^{'})$ associated with each edge $(n, n^{'})$ in

$\mathcal{E}$ are known. Let $\mathcal{M} = \{1, \ldots, M\}$ be a set of $M$ identical vehicles initially located at the mill. Each front $n$ in $\mathcal{N} \setminus 0$ has $B_n$ harvesters. Let $H_n = (H_{n,1}, H_{n,2}, \ldots, H_{n,B_n})$ be a vector of binary variables, where 1 represents, the harvester is up and running and 0 means it is down and $B_n$ is the number of harvesters in the front $n$. let $R_n = (R_{n,1}, R_{n,2}, \ldots, R_{n,B_n})$ be a column vector, where $R_{n,1}$ is the capacity of harvester 1 in front $n$. The maximum harvest rate at front $n$ is given by a vector multiplication, $H_n^T R_n$. For the sake of brevity, let $H = (H_n)_{n \in \mathcal{N} \setminus 0}$ be a set containing harvester information for all the fronts. Let the vector, $H^T R = \left(H_1^T R_1, H_2^T R_2, \ldots, H_N^T R_N\right)$, represent the maximum possible harvest rates of all the fronts, such that the first element is the maximum possible harvest rate at front 1 and second element is the harvest rate at front 2 and so on.

A decision epoch is triggered by the arrival of one or more vehicles at the fronts or at the mill. The time of decision epoch $k$, $T_k$, characterizes the end of period $k-1$ and the beginning of the period $k$. Even though we assume deterministic travel times, $T_k$ is a random variable because the breakdowns and shut-downs at fronts impact our decisions and thus make it impossible to determine decision epochs a priori.

At decision epochs, for vehicles at the the mill, an action is selected prescribing the front where these vehicles will travel next. When a vehicle arrives at the front, an action is selected prescribing the vehicle's return to the mill. Both of these actions take place during the deterministic transition from pre-decision state $s_k$ to the post-decision state $s_k^a$. During the transition from post decision state $s_{k-1}^a$ to the next pre-decision state $s_k$, the status of the harvesters, $H_k$ becomes available which enables us to calculate the maximum harvest rate at each front for the next epoch. We assume that the the epochs are short enough that the status of the harvesters we received before the beginning of an epoch stays unchanged during that epoch. At decision epochs, we also decide the harvest rates for the fronts. (using the information contained in $H^T R$, among other state information. The next period begins when next decision epoch is triggered. The process repeats until the end of planning horizon is reached.

### 5.2.1 State

The state of the system captures all relevant information required to define available actions and to determine the transitions. We represent attributes of a vehicle $m \in \mathcal{M}$ by a triple $(d_m, t_m, u_m)$ where $d_m \in \mathcal{N}$ is the destination of the vehicle (or its current location if the vehicle has arrived) and $t_m \in [0, L]$ is the time at which the vehicle $m$ is scheduled to arrive at $d_m$. The third component $u_m$ is a binary variable which takes the value 1 if the vehicle has a loaded trailer and 0 when the trailer is empty. Let $(d, t, u) = (d_m, t_m, u_m)_{m \in \mathcal{M}}$ denote the vector of vehicle attributes.

We represent attributes of the nodes, $n \in \mathcal{N}$ by a vector $(l_n, h_n)$. For the fronts $(n \in \mathcal{N} \setminus 0)$, $l_n$ is the total number of loads already harvested in the planning period and $h_n$ is the harvest rate. Harvest rate at the front is expressed as trailer per unit time. For the mill $(n = \{0\} \subset \mathcal{N})$, $l_n$ is the number of loads waiting at the mill yard and $h_n$ is the crushing rate, which is constant in our model but could very well be a variable. Let $(l, h) = (l_n, h_n)_{n \in \mathcal{N}}$.

### 5.2.2 Actions

An action is taken at each decision epoch. A decision epoch is triggered by the arrival of a vehicle at the mill or one of the fronts. Our state information includes a vector of expected arrival times $(t_m)_{m\in\mathcal{M}}$ of all the vehicles to their respective destinations. Decision epoch occurs at time $T_k = \min_{m\in\mathcal{M}}\{t_m\}$. Let $\mathcal{M}' = \arg\min_{m\in\mathcal{M}}\{t_m\}$ be the set of vehicles currently at the fronts or at the mill.

There are two sets of actions available at each epoch. The first set of actions assigns the vehicles in the set $\mathcal{M}'$ to $n \in \mathcal{N}$. The second set of actions changes the harvest rates at the fronts.

Let $\mathcal{A}^1(s_k)$ be the action space for first set of actions. Each member of the set $\mathcal{A}^1(s_k)$ is a vector of length $M$. Let $a_m$ be the $m^{th}$ member of this vector. It can only take the value $n \in \mathcal{N}$ and is the new destination for the vehicle $m \in \mathcal{M}$. Even though each action is defined by a vector of length of $M$, in each decision epoch, only $|M'|$ members of the action vector change. Let $a \in \mathcal{A}^1(s_k)$, and $a$ is a vector of length of $M$. In each decision epoch we change only those members of this vector $a$ which belong to the set $\mathcal{M}'$. Remaining members of $a_k$ is same is that of $a_{(k-1)}$. The reason behind this is we assume that vehicles that are en-route cannot be re-routed. The action space for the first set of actions is given by

$$\mathcal{A}^1(s_k) = \left\{a : a \in \mathcal{N}^M\right\}.$$

Each member $a_m$ of an action vector $a \in \mathcal{A}^1(s_k)$ can be defined as follows:

$$a_m = d_m, \quad \forall m \in \{\mathcal{M} \setminus \mathcal{M}^{'}\}, \tag{5.1}$$

$$a_m = n \in \mathcal{N}, \quad \forall m \in \{\mathcal{M}^{'}, d_m = 0, T_k + t(d_m, n) + t(n, 0) \leq L\}, \tag{5.2}$$

$$a_m = 0, \quad \forall m \in \{\mathcal{M}^{'}, d_m = 0, T_k + t(d_m, n) + t(n, 0) > L\}, \tag{5.3}$$

$$a_m = 0 \quad \forall m \in \{\mathcal{M}^{'}, d_m \neq 0\} \tag{5.4}$$

Condition 5.1 requires that the vehicles en route continue to their destination. Condition 5.2 allows waiting at the mill and Condition 5.3does not allow assignment to the front that would violate the planning horizon limit and Condition 5.4 requires the vehicles that reach to the fronts to get back to the mill.

Let $\mathcal{A}^2(s_k)$ be the action space for second set of actions. Each member $b$ of the set $\mathcal{A}^2(s_k)$ is a vector of length $N$. Let $b_n$ be the $n^{th}$ member of this vector $b$. It can only take the value between $\left[0, H_n^T R_n\right]$ and is the new harvest rate of the front $n \in \mathcal{N} \setminus 0$.

The action space for this second set of actions is given by

$$\mathcal{A}^2(s_k) = \left\{ b : b \in \left[0, \max_{n \in \ \mathcal{N} \setminus 0} \left[H_n^T R_n\right]\right]^N \right\}.$$

If $\bar{l_n}$ is the pre-decided, number of loads to be harvested in front $n$ during the planning horizon, each member $b_n$ of an action vector $b \in \mathcal{A}^2(s_k)$ can be defined as follows:

$$b_n = h : \min\left[\min_h \left[(l_n, 1) + (t_m - T_k) * h \geq |m^{'} : d_{m^{'}} = n \;\&\; t_{m^{'}} \leq t_m|\right]_{\{m:\, d_m = \, n\}}, (H_n^T R_n)\right] \tag{5.5}$$

$b_n = 0$ when $l_n = \bar{l_n}$.(5.6)

Condition 5.5 requires the new harvest rate in all the fronts to be minimum of (a) minimum rate such that enough cane is harvested to fill the trailers for all the vehicles en-route to the front, just on time or before the vehicle arrives at the front (b) maximum harvest rate. Condition 5.6 requires to stop harvesting at the front when the limit for the number of loads from the front is reached.

Let $\mathcal{A}(s_k)$ be the complete action space such that each member is a vector of length $(M + N)$ where the first $M$ members come from the $a$ in $\mathcal{A}^1(s_k)$ and next $N$ members come from $b$ in $\mathcal{A}^2(s_k)$.

### 5.2.3 Exogenous Information

This is the information that first becomes known to us each epoch. In our model, in each epoch, we get the information on what are the maximum possible harvest rates in each front for the next epoch (in fact, we get the information on which harvesters are up or down and this allows us to calculate the maximum possible harvest rates).

We have defined status of each harvester by a binary variable variable, $H_{(n \in \mathcal{N} \setminus 0,\ i \in \{1 \ldots B_n\})}$. We can assume that $\left\{H_{(n \in \mathcal{N} \setminus 0,\ i \in \{1 \ldots B_n\})}(k),\ k = k_0, k_1, \ldots\right\}$ and $\left\{H_{n^{'} \in \mathcal{N} \setminus 0,\ i^{'} \in B_n}(k),\ k = k_0, k_1, \ldots\right\}$ are independent Markov chains for $n \neq n^{'}$. We assume the dynamics of all of these Markov chains are described independently by the one epoch transition matrix (the assumption that the epochs are almost equal allows us to assume this) Kim and Lee [2005].

$$\mathcal{T}^{(k,k+1)} = \begin{bmatrix} \gamma_k & 1-\gamma_k \\ 1-\beta_k & \beta_k \end{bmatrix}.$$

Let $P(H(k+1)|H(k))$ be the probability of a transition occurring from $H$ in epoch $k$ to $H(k+1)$ in epoch $k+1$. Due to our assumption of independence,

$$P(H(k+1)|H(k)) = \prod_{n \in \mathcal{N} \backslash 0} \prod_{i=1}^{B_n} P(H_{n \in \mathcal{N} \backslash 0,\ i \in B_n}(k+1)|H_{n \in \mathcal{N} \backslash 0,\ i \in B_n}(k))$$

If we assume that the time between two epochs are roughly constant, we can compute the transitions between all the states by

$$\mathcal{T}^{(k,k^{'})} = \begin{bmatrix} \gamma_k & 1-\gamma_k \\ 1-\beta_k & \beta_k \end{bmatrix} X \begin{bmatrix} \gamma_{k+1} & 1-\gamma_{k+1} \\ 1-\beta_{k+1} & \beta_{k+1} \end{bmatrix} X \begin{bmatrix} \gamma_{k+2} & 1-\gamma_{k+2} \\ 1-\beta_{k+2} & \beta_{k+2} \end{bmatrix}$$

$$\ldots X \begin{bmatrix} \gamma_{k'} & 1-\gamma_{k'} \\ 1-\beta_{k'} & \beta_{k'} \end{bmatrix}.$$

So,

$$P(H(k^{'})|H(k)) = \prod_{n\in\mathcal{N}\backslash 0}\prod_{i=1}^{B_n} P(H_{(n\in\mathcal{N}\backslash 0,\ i\in B_n)}(k^{'})|H_{(n\in\mathcal{N}\backslash 0,\ i\in B_n)}(k)).$$

### 5.2.4 Transition

The transition function determines how our system evolves from the state $s_k$ to the state $s_{(k+1)}$. The transition from $s_k$ to $s_{k+1}$ is a function of the current state, the chosen action $\alpha$, and the random information $w_{k+1}$ and is denoted by $S^M()$.

$$s_{(k+1)} = S^M(s_k, \alpha_k, w_{k+1}) \tag{5.7}$$

The transition function is given by:

$$(d_m)_{(k+1)} = a_m \tag{5.8}$$

$$(t_m)_{(k+1)} = \begin{cases} t_m & \text{if } m \in \{\mathcal{M} \setminus \mathcal{M}^{'}\} \\ \\ T_k + t(d_m, a_m) & \text{if } m \in \mathcal{M}^{'} \end{cases} \tag{5.9}$$

$$(u_m)_{(k+1)} = \begin{cases} u_m & \text{if } m \in \{\mathcal{M} \setminus \mathcal{M}^{'}\} \\ \\ 0 & \text{if } m \in \mathcal{M}^{'} \ \ \& \ \ d_m = 0 \\ \\ 0 & \text{if } m \in \mathcal{M}^{'} \ \ \& \ \ d_m \neq 0 \ \ \& \ \ H^T_{(d_m)} R_{(d_m)} = 0 \\ \\ 1 & \text{if } m \in \mathcal{M}^{'} \ \ \& \ \ d_m \neq 0 \ \ \& \ \ H^T_{(d_m)} R_{(d_m)} \neq 0 \end{cases} \tag{5.10}$$

$$(l_m)_{(k+1)} = \begin{cases} l_n + (T_{k+1} - T_K) \times (h_n)_k & \text{if } n \in \{\mathcal{N} \setminus 0\} \\ \\ l_n - ((T_{k+1} - T_K) \times h_0) + \sum_{m \in \mathcal{M}'} u_m & \text{if } n = 0 \end{cases} \tag{5.11}$$

$$(h_n)_{(k+1)} = b_n \tag{5.12}$$

Condition 5.8 says the destination of the vehicle $m$ is given by the $m^{th}$ component of action $a$. Condition 5.9 says the estimated arrival time for the vehicle $m$ at its destination. Condition 5.10 updates the status of the trailer of the vehicle $m$. Condition 5.11 updates the amount the cane harvested at the front $n$ and the amount of cane waiting at the mill yard.

### 5.2.5 Rewards and Objective

Our primary goal is to maintain a constant buffer at the mill while trying to minimize the total cut to crush time. Let $\tilde{\mathcal{R}}(l_0, k)$ be a strictly concave function with a positive coefficient for $l_0$. Let us also assume that this quadratic function achieves unique global maximum when $l_0 = \mathcal{B}$, where $\mathcal{B}$ is the optimal buffer at the mill. The strict convex nature of the function $\tilde{\mathcal{R}}(l_0, k)$ penalizes the deviation from optimal buffer.

## 5.3 Future Work

In our future work, we want to develop a dymanic model for sugarcane harvest and logistics. We want to prescribe a policy, $\pi$, that prescribes actions (where should a vehicle be dispatched and what should be the harvest rate at each front) until the end of planning horizon is reached. If the random variable K be the number of decision epochs before time T. The total reward under the policy $\pi$ is

$$v^{\pi}(s_k) = \tilde{\mathcal{R}}_k(l_0, k) + \mathbb{E}\left\{\sum_{i=0}^{K} \delta^{i+1} * \mathcal{R}_{k+i}(l_0, k)\right\} \tag{5.13}$$

where $\delta$ is the discount factor.

We seek an optimal policy such that $v^{\pi^\star} \geq v^{\pi}$ for all policies.

The existence of such a policy is guaranteed because from every state there is a path to the boundary condition of $t = T$. [Powell, 2007]

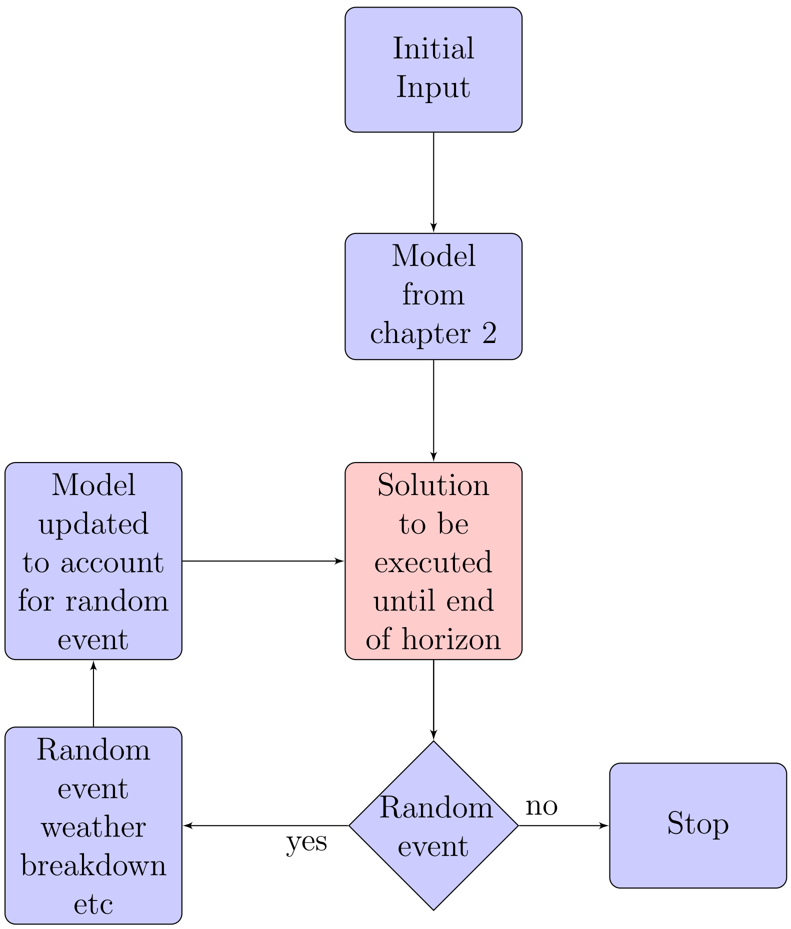


Figure 5.1: Solution Approach

## REFERENCES


Michael E Salassi and F Gil Barker. Reducing harvest costs through coordinated sugarcane harvest and transport operations in louisiana. *Journal Association Sugar Cane Technologists*, 28:32–41, 2008.

Economic Research Service. Sugar and sweetners yearbook tables. Technical report, United States Department of Agriculture, 2012. URL `http://www.ers.usda.gov/data-products/sugar-and-sweeteners-yearbook-tables.aspx`.

M. McConnell, E. Dohlman, and S. Haley. World sugar price volatility intensified by market and policy factors. *Amber Waves*, 8(3):28 – 35, 2010.

Constanza Valdes. Can Brazil meet the world's growing need for ethanol? *Amber Waves*, 9(4):38 – 45, 2011.

Alexandra Wexler. U.S. Throws Gas on Sugar Market. *The Wall Street Journal*, page C1, December 17, 2012.

Michael E Salassi, Mercedes Garcia, and Janis B Breaux. Impact of Sugarcane Delivery Schedule on Product Value at Raw Sugar Factories. *Journal of Agribusiness*, 22(1):61–75, 2004.

M. Saska, S. L. Goudeau, and M. Marquette. Determination of sucrose lose in clean unburnt billet cane. *Journal of the American Society of Sugar Cane Technologies*, 9:53 – 77, 2009.

P. Saxena, R. P. Srivastava, and M. L. Sharma. Impact of cut to crush delay and biochemical changes in sugarcane. *Australian Journal of Crop Science*, 4(9):692–699, 2010.

Sergio Barros. GAIN Report: Brazil Semi-Annual Sugar 2012. Technical Report BR12014, USDA Foreign Agricultural Service, October 2012. URL `http://gain.fas.usda.gov/RecentGAINPublications/SugarSemi-annual\_SaoPauloATO\_Brazil\_10-1-2012.pdf`.

E.O. Heady. Simplified presentation and logical aspects of linear programming technique. *Journal of Farm Economics*, 36(5):1035–1048, 1954.

T.J. Lowe and P.V. Preckel. Decision technologies for agribusiness problems: A brief review of selected literature and a call for research. *Manufacturing & Service Operations Management*, 6(3):201–208, 2004.

O. Ahumada and J.R. Villalobos. Application of planning models in the agri-food supply chain: A review. *European Journal of Operational Research*, 196(1):1–20, 2009. ISSN 0377-2217.

RC Giles. A simulation study of cane transport system improvements in the Sezela Mill area. Master's thesis, School of Bioresources Engineering and Environmental Hydrology, Univserity of KwaZulu-Natal, Pietermaritzburg, South Africa, December 2009.

M S Kawamura, D P Ronconi, and H Yoshizaki. Optimizing transportation and storage of final products in the sugar and ethanol industry : a case study. *International Transactions in Operational Research*, 13:425–439, 2006.

Rafael P. O. Paiva and Reinaldo Morabito. An optimization model for the aggregate production planning of a Brazilian sugar and ethanol milling company. *Annals of Operations Research*, 169(1):117–130, September 2009. ISSN 0254-5330. URL `http://www.springerlink.com/index/10.1007/s10479-008-0428-9`.

Aneirson Francisco da Silva, Fernando Augusto Silva Marins, and José Arnaldo Barra Montevechi. Multi-choice mixed integer goal programming optimization for real problems in a sugar and ethanol milling company. *Applied Mathematical Modelling*, 37(9):6146–6162, May 2013. ISSN 0307904X. doi: 10.1016/j.apm.2012.12.022. URL `http://linkinghub.elsevier.com/retrieve/pii/S0307904X13000061`.

Sanjay Dominik Jena and Marcus Poggi. Harvest planning in the Brazilian sugar cane industry via mixed integer programming. Technical report, Departamento de Informatica, Pontificia Universidade Catolica do Rio de Janeiro, Rio De Janeiro, 2013. URL `ftp://139.82.16.194/pub/docs/techreports/13\_02\_jena.pdf`.

A J Higgins, R C Muchow, A V Rudd, and A W Ford. Optimising harvest date in sugar production : A case study for the Mossman mill region in Australia I . Development of operations research model and solution. *Field Crops Research*, 57: 153 – 162, 1998.

M. Grunow, H.-O. Gunther, and R. Westinner. Supply optimization for the production of raw sugar. *International Journal of Production Economics*, 110:224–239, 2008.

A.M. Kostin, G. Guillén-Gosálbez, F.D. Mele, M.J. Bagajewicz, and L. Jiménez. A novel rolling horizon strategy for the strategic planning of supply chains. Application to the sugar cane industry of Argentina. *Computers & Chemical Engineering*, 35(11):2540–2563, November 2011. ISSN 00981354. doi: 10.1016/j.compchemeng.2011.04.006. URL `http://linkinghub.elsevier.com/retrieve/pii/S0098135411001402`.

Marco Fava Neves, Vinicius Gustavo Trombin, and Matheus Alberto Consoli. Measurement of sugar cane chain in Brazil. *International Food and Agribusiness Management Review*, 13(3):37–54, 2010.

M. E. Salassi, F. G. Barker, and M. A. Deliberto. Optimal scheduling of sugarcane harvest and mill delivery. *Sugar Cane International*, 27(3):87–90, 2009a.

AC Hansen, AJ Barnes, and PWL Lyne. Simulation modeling of sugarcane harvest-to-mill delivery systems. *Transactions of the ASABE*, 45(3):531–538, 2002.

P.Y. Le Gal, J. Le Masson, C. N. Bezuidenhout, and L. F. Lagrange. Coupled modelling of sugarcane supply planning and logistics as a management tool. *Computers and Electronics in Agriculture*, 68:168–177, 2009.

C. Lejars, P.Y. Le Gal, and S. Auzoux. A decision support approach for cane supply management within a sugar mill area. *Computers and Electronics in Agriculture*, 60(2):239–249, 2008. ISSN 0168-1699.

B. C. McDonald, E. Dube, and C. N. Bezuidenhout. TranSwarm: A sugarcane transport simulation model based on behavioural logistics. In *Proceedings of the South African Sugar Technologists' Association*, volume 81, pages 434–438, 2008.

S. Supsomboon and J. Yosnual. A stochastic model for optimal order quantity for sugarcane factory with uncertainties. In *Proceedings of the Fifth Asia Pacific Industrial Engineering and Management Systems Conference*, pages 21.2.1–21.2.11, 2004.

K. Prichanont, S. Prichanont, and N. Buransri. Improvement guidelines for sugar cane delivery systems. In *35th International Conference on Computers and Industrial Engineering*, pages 1585–1590, 2005.

J.A. Diaz and H.G. Perez. Simulation and optimization of sugar cane transportation in harvest season. *Winter Simulation Conference*, 2:1114–1117, 2000.

E. Lopez Milan, S. Miquel Fernandez, and L.M. Pla Aragones. Sugar cane transportation in Cuba, a case study. *European Journal of Operational Research*, 174 (1):374–386, 2006.

Esteban Lopez-Milan and Lluis Pla-Aragones. A decision support system to manage the supply chain of sugar cane. *Annals of Operations Research*, to appear, 2013.

C. Hildebrand. Independent assessment of the sugar industry. Report to the Hon. Warren Truss MP, Minister for Agriculture, Fisheries and Forestry, available from `http://www.apec.org.au/docs/06_TP_Sugar/04_report.pdf`, June 2002.

Andrew J. Higgins and Steve Postma. Australian sugar mills optimise siding rosters to increase profitability. *Annals of Operations Research*, 128:235–249, 2004. 10.1023/B:ANOR.0000019107.68291.36.

A. Higgins and I. Davies. A simulation model for capacity planning in sugarcane transport. *Computers and Electronics in Agriculture*, 47:85–102, 2005.

A J Higgins and L A Laredo. Improving harvesting and transport planning within a sugar value chain. *Journal of the Operational Research Society*, 57(4):367–376, July 2006. ISSN 0160-5682. doi: 10.1057/palgrave.jors.2602024. URL `http://www.palgrave-journals.com/doifinder/10.1057/palgrave.jors.2602024`.

A. Higgins. Scheduling of road vehicles in sugarcane transport: A case study at an Australian sugar mill. *European Journal of Operational Research*, 170(3):987–1000, 2006. ISSN 0377-2217.

Francois Margot. Symmetry in integer linear programming. In Michael Junger, Thomas M. Liebling, Dennis Naddef, George L. Nemhauser, William R. Pulleybank, Gerhard Reinelt, Giovanni Rinaldi, and Laurence A. Wolsey, editors, *50 Years of Integer Programming*, chapter 17, pages 647–686. Springer-Verlag, Berlin, 2010.

M. E. Salassi, F. G. Barker, and M. A. Deliberto. Optimal scheduling of sugarcane harvest and mill delivery. *Sugar Cane International*, 27(3):87 – 94, 2009b.

United States Department of Agriculture, Economic Research Service. Sugar and sweetners yearbook tables, December 2010. URL `http://www.ers.usda.gov/Briefing/Sugar/`.

Leo C Polopolus, Jose Alvarez, and William A Messina Jr. Sugar and the north american free trade agreement. 2010.

Michael E. Salassi and Michael A. Deliberto. Sugarcane production in louisiana. Online, 2012. URL `http://www.lsuagcenter.com/en/our_offices/departments/Ag_Economics_Agribusiness/extension_outreach/budgets/2012/2012-Projected-Louisiana-Sugarcane-Production-Costs.htm`.

Francis Gilbert Barker. *An economic evaluation of sugarcane combine harvester costs and optimal harvest schedules for Louisiana*. PhD thesis, Louisiana State University, 2007.

Kamal Lamsal, Philip C Jones, and Barrett W Thomas. Sugarcane harvest logistics in Brazil. 2013a. URL `http://ir.uiowa.edu/cgi/viewcontent.cgi?article=1061&context=tippie_pubs`. Submitted for publication.

American Sugar Cane League. Raw sugar factories. Online, 2013. URL `http://www.amscl.org/factories`.

Hosein Shapouri and Michael E Salassi. Economic feasibility of ethanol production from sugar in the united states. 2006.

James Jacobs. Ethanol from sugar: What are the prospects for us sugar co-ops? *Rural Cooperatives*, 73(5), 2006.

Foreign Agricultural Service/USDA. Sugar: World markets and trade. May 2014. URL `http://apps.fas.usda.gov/psdonline/circulars/sugar.pdf`.

Kamal Lamsal, Philip C Jones, and Barrett W Thomas. Sugarcane harvest logistics in us. 2013b. URL `http://ir.uiowa.edu/cgi/viewcontent.cgi?article=1061&context=tippie_pubs`. Submitted for publication.

Andrew Higgins, George Antony, Gary Sandell, Ian Davies, Di Prestwidge, and Bill Andrew. A framework for integrating a complex harvesting and transport system for sugar production. *Agricultural Systems*, 82(2):99–115, 2004.

RW Whitney and BJ Cochran. Predicting sugar-cane mill delivery rates [transport equipment, louisiana]. *Transactions of the ASAE*, 1976.

Enrique Arjona, Graciela Bueno, and Luis Salazar. An activity simulation model for the analysis of the harvesting and transportation systems of a sugarcane plantation. *Computers and electronics in agriculture*, 32(3):247–264, 2001.

Hyun Kim and Yeon Lee. Optimal watermarking of digital hologram of 3-d object. *Optics Express*, 13(8):2881–2886, 2005.

Warren B Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*, volume 703. John Wiley & Sons, 2007.