INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®

University of Alberta

Optimal Haul Truck Allocation in the Syncrude Mine

by

Chung Huu Ta ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**

Department of Electrical & Computer Engineering

Edmonton, Alberta

Spring 2002

National Library of Canada
Acquisitions and Bibliographic Services
395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada
Acquisitions et services bibliographiques
395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-69767-3

Canada

University of Alberta

Library Release Form

Name of Author: Chung H. Ta

Title of Thesis: Optimal Haul Truck Allocation in the Syncrude Mine

Degree: Master of Science

Year this Degree Granted: 2002

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

....

Chung H. Ta
11 Woodvale Village
Edmonton, Alberta
Canada, T6L-1W4

Date: Nov...19., 2001

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled ***Optimal Haul Truck Allocation in the Syncrude Mine*** submitted by ***Chung Huu Ta*** in partial fulfillment of the requirements for the degree of **Master of Science**.

....
Horacio J. Marquez

....
J. Fraser Forbes

....
Tongwen Chen

Date: .Nov..16.,.2001.

To

my parents.

my wife. Phi-Mỵ.

our children. Duy-Khiêm. Minh-Thư. and Duy-Đăng

Abstract

Optimal resource allocation is an important industrial problem. In the mining industry where the truck-and-shovel technology is used, allocating an optimal number of haul trucks is essential in reducing the overall mining cost. The objective of this thesis is to investigate the use of stochastic programming techniques in the truck allocation problem. Two stochastic methods were considered: recourse-based and chance-constrained based. The recourse method is not suitable because the truck allocation is not a two-stage problem and requires heavy computation. The thesis also studies the benefits of the implementation of the chance-constrained method with parameter update. These benefits include meeting production with a specified degree of confidence and the ability to recover from negative changes in mining environment.

Acknowledgements

I would like to thank a number of people, who have helped me throughout the thesis.

First, I owe a great deal to Dr Jim Kresta, my colleague and also my thesis supervisor in Syncrude Research. Jim has continually provided valuable guidance and encouragement from the start of the project and is especially the key person responsible for the formation of this thesis.

I also would like to thank Dr. Forbes (Chem. Eng) and Dr. Marquez (Elect. Eng) both from the University of Alberta. Dr. Forbes has provided important technical guidance and valuable help throughout the project and Dr. Marquez has provided much needed support in my home department during my four-year part-time study at U of A.

Thanks also go to my colleagues, Dr. Parsons and Mr. Mueller, for valuable ideas, and suggestions during the writing stage of the thesis. Last but not least, I would like to thank the management at Syncrude Research for supporting me during my graduate study, especially Dr. Czarnecki, and Mr. Oxenford, my current team leader. Thanks also go to our past and current Research directors, Dr. Clark and Mr. McKee, respectively.

In general, I also would like to acknowledge Syncrude Canada Ltd. and PRECARN Associates for funding this project work.

Table of Contents

List of Tables

List of Figures

1. Introduction

The aim of the thesis is to investigate methods to find an "optimal" solution to the *truck allocation*[1] problem, while recognizing the uncertainty that is present in the data supplied to the optimizer. Current truck allocation methods presume certainty in the data supplied to the decision system (e.g. truck productivity report, haul distance, etc.) and the inefficiency of the truck allocation is due to the conservatism of the mine planners in dealing with various uncertainties in the allocation problem.

1.1. Syncrude Operation

In Syncrude, or in any open-pit mine operation, the truck and shovel technology is predominantly used to mine ore and transport it to other locations for further processing. Shovels are deployed to load materials onto large trucks, which haul the material to various destinations. These large mobile pieces of equipment operate continuously to feed ore to the plant. Figure 1.1 illustrates the process of moving the ore in the Syncrude mine operation. The figure does not show the hauling of waste material, which is identical to the ore except that there is no restriction on the receiving capacity.

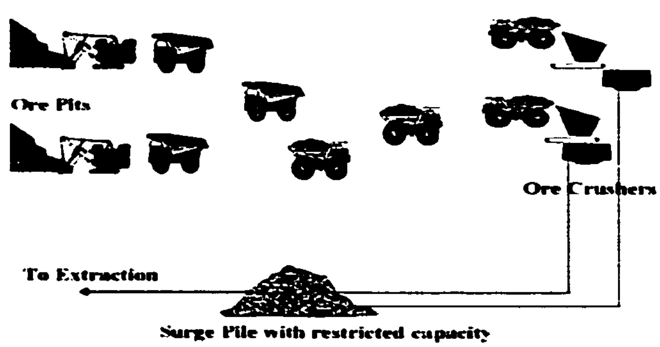


Figure 1.1 – Ore handling operation with trucks and shovels

Syncrude mine materials are categorized into two types: ore and waste. Waste material can be further sub-divided into inter-burden[2] and over-burden[3]. Overburden resides on top of the ore body and it must be removed to make way for the ore to be mined. Inter-burden layers lie close to the ore body below the overburden and they too have to be moved away before the ore can be retrieved. However, due to the location of the inter-burden, the ore shovel can also act as a waste shovel. Except when used as the construction material to build haul road, all waste material is transported to dumping areas while ore is transported to the ore crusher.

Ore usually has higher hauling priority than waste because it is required as a continuous feed into the Extraction plant for the running operation. This priority difference will be reflected in the truck allocation process in that the production constraint is placed on the rate of ore delivered by trucks.

Efficient use of haul trucks translates to fewer trucks being required for the hauling operation, resulting in lower maintenance cost as well as deployment cost, thus reducing overall mining costs. Current truck allocation is not always optimized, as it is calculated based on the production targets and an estimate of truck productivity for a given haul distance. Extra truck capacity is incorporated to account for uncertainty (e.g. truckload, truck cycle time) and upsets. The solutions obtained this way, work well when they are derived by an experienced planner, but as the operation becomes more integrated and the surge[4] capacity is decreased the truck allocation task will become more difficult. Experience

---

[1] Truck allocation is the process of determining a number of trucks required to haul mine material.

[2] This oil sand layer contains ore material with low bitumen content. They are either used to build hauling roads or hauled to the waste dump.

[3] The layers of sand, gravel and shale, which overlie the oil sands

[4] The location where the delivered ore is accumulated before it is sent to the Extraction plant, and the bitumen is extracted from the oil sand. Bitumen is a molasses-like substance, which comprises 6-14% of oil sand. Even after extraction, bitumen is still too thick for any practical purpose and must be upgraded to synthetic crude oil.

combined with appropriate decision support tools could provide more efficient truck allocation and cost reduction.

The mine operation can be divided into two main stages: planning stage and operating stage. The planning stage can be further broken down into long-term plan and short-term plan. Mine planning is done on an annual, quarterly, monthly or daily basis. The mine planners are responsible for determining the total number of trucks needed for a given production requirement. The number of trucks is determined using a special formula, which is derived with the curve-fitting method using historical data and operation experience. During the operation, the truck dispatcher only uses this truck solution as a guide and is free to alter the truck solution appropriately.

The focus of this study will be on the daily plan, which includes the identification of the working shovels and the allocation of haul trucks to these shovels. Top priority is to allocate enough trucks to satisfy the ore demand, i.e., rate of ore (tph) to Extraction. Hauling waste is a secondary but also important task, since the waste must be moved eventually. In general, the daily plan identifies the amount of material to be moved in the mine for that day and allocates resources to fulfill this hauling requirement.

An optimization study on the truck fleet was performed in 1991 [Coward, 1991], which reported the benefit of optimizing the truck fleet deployment using linear programming. The truck-hauling problem was implemented as a linear deterministic network flow model, in which shovels and dumps were source and destination nodes respectively. The objective was to maximize the amount of material moved. This material included rejects, ore and overburden in decreasing order of importance. The optimal solution reported by Coward [1991] was associated with the total number of trucks allocated, but rather with the real time deployment of a fixed number of trucks in a specific road network. In contrast, this thesis work focuses on the task of allocation truck resource to haul ore. The desired optimal solution corresponds to a minimum number of trucks required to satisfy the ore-hauling requirement.

1.2. Thesis Objective

The objective of this thesis is to formulate an appropriate optimization model and investigate the solution techniques that can be used for the truck allocation problem with uncertainty in the Syncrude mining operation.

1.3. Thesis Scope

This thesis is concerned with applying a stochastic solution method to solving an optimization problem for the truck allocation. Current allocation methods are not optimized and rely mostly on certain information such as average haul distance, average truck productivity information, etc. The number of trucks, which is found as fractional numbers are approximated and deployed as discrete numbers. Inefficiency in the truck deployment is tolerated in exchange for the assurance of the absolute satisfaction of the production constraint. Planners are more willing to accept inefficiency in truck deployment than to face the risk of not meeting the ore production constraint due to the uncertainty in the process.

It is important to work with a deterministic optimization model of the truck allocation problem to establish the basic understanding of the problem as well as to build a baseline for the subsequent comparison to the stochastic model. The deterministic model, as presented in Chapter 2, will lay an important framework of the truck allocation problem as a whole. The optimal truck solution will be determined. In addition, the sensitivity of the solution to some model parameters will be examined, with a focus on the most likely uncertain parameters in the model.

Solving optimization problems with uncertain parameters requires the application of stochastic programming. The field of stochastic programming is based on the theory of probability and has been applied successfully in many industrial applications. Chapter 3 will present a brief introduction to stochastic programming and two primary stochastic methods that are used to solve optimization problems with uncertainty. Since each of the two methods will have both strengths and weaknesses, the decision to use either technique will be driven by the nature of the application. Comparison of these two techniques will be made in the context of the truck allocation problem and the selected technique will be used consistently for the remainder of the thesis work.

The work in both Chapters 2 and 3 is based on a simplified model. In these chapters, the main production constraint is to meet the minimum required ore rate to Extraction. The truck models here do not include the surge pile from which the ore stream to Extraction actually originates.

Chapter 4 studies the truck allocation problem in the context of a parameter update approach. In this study, the multi-period optimization problem will be solved for a truck allocation solution, which in turn, will be implemented in a simulated environment. Statistical information gathered during a simulation period will be used as input to the optimization problem in the next period. The aim of the study is to investigate the mechanism with which uncertain information, as it is revealed with time, can be used in the decision-making process in the future. Modification is made to the model to include the effects of the surge pile, whose function is to regulate the ore rate flowing to Extraction. This addition of the surge pile will make this model more realistic and representative of the actual operation.

Chapter 5 contains the summary of the study and concludes with the suggestion of what method that is most appropriate for the truck allocation problem in Syncrude.

Although the scope of thesis is limited to the optimization problem for the truck allocation task, the stochastic programming technique being covered can be used in many industrial optimization applications, which are characterized with uncertainty. While Stochastic Programming have been used successfully in many industrial problems, optimization practitioners must still investigate individual problem and decide if the stochastic technique is more beneficial than the commonly used deterministic programming technique.

1.4. Thesis Contribution

The main contribution of this study includes the formulation of an appropriate model for the truck allocation problem. Other important contribution is to determine a suitable stochastic formulation for this class of allocation problems. Additional benefits can also be gained from learning the common problems, often encountered in the process of finding for the optimal solution, and methods that can be used to overcome such problems.

1.5. Thesis Conventions

240T, 320T, 360T	Short form reflecting the capacity-based category of trucks, e.g., 240T trucks, 320T trucks, or 360T trucks
Tph	Tonnes per hour
j	Truck category index based on truck capacity: $j = 1$ for 240-Tonne trucks, $j = 2$ for 320-Tonne trucks, $j = 3$ for 360-Tonne trucks
H	Time period (hours)
τ_{w_j}	cycle time of the waste trucks of type j (minutes)
τ_{o_j}	cycle time of the ore trucks of type j (minutes)
L_{w_j}	waste truckload of truck of type j (Tonnes)
L_{o_j}	ore truckload of truck of type j (Tonnes)
W_m	Minimum amount of waste required to be moved over the time period
R_j	Truck resource limitation of type j
x_{o_j}	Part of the solution variables: number of ore trucks per hour for truck of type j.
x_{w_j}	Part of the solution variables: number of waste trucks per hour for truck of type j.
$\mathbf{A}$	Left-hand side coefficient matrix in the linear model
$\mathbf{B}$	Right-hand side matrix term in the linear model
$\mathbf{c}^T$	Cost coefficient matrix term in the objective function
$\boldsymbol{\alpha}$	Vector of confidence limit used in the chance constrained

	model
σ	Matrix of standard deviations of the associated uncertain parameters in the chance constrained model
$F(z)$	Cumulative distribution function (monotonically increasing)
$P[.]$	Probability of *[.]*
$N(\mu,\sigma^2)$	Normal distribution with mean μ and standard deviation σ

2. Deterministic Truck Allocation

Linear programming has a long history in the field of mathematical programming, and is widely applied to many industrial applications due to its simplicity. An optimization application usually starts with the formulation of the linear model, which is then solved for desired decision variables. The application results are then analyzed to determine if the model truly reflects the behavior of the actual process. This chapter deals mainly with the formulation of the linear deterministic model for the truck allocation problem, while leaving the discussion beyond the linear deterministic model for later chapters. A brief theoretical background on linear programming concept is given in Appendix A.

2.1. Truck Allocation Model Development

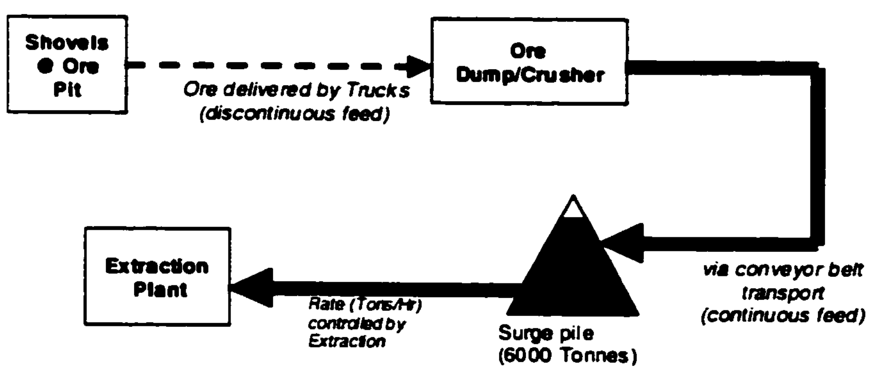


Figure 2.1 – Typical Ore Flow

Ore shovels at various locations throughout the mine load oil sand onto trucks to be hauled to the crusher, whose job is to crush the ore into smaller pieces (Figure 2.1). The ore is then moved via the conveyor belt to the surge pile and on to the Extraction plant. The surge pile is an intermediate storage of oil sand and acts as a buffer between the Extraction plant and the Mine. In the Base Mine, the surge pile was large and able to supply enough oil sand to Extraction so that the Mine and Extraction could act relatively independently. The current surge pile in the North Mine is much smaller and can supply ore to Extraction for about 20 minutes at the maximum rates of 12000 tph from the normal operating point of *80%* with no feed from the Mine. As a result, in the North Mine, the Extraction operation is more closely coupled with the mining process.

Parallel with the ore delivery to Extraction is the removal of waste material (over-burden and inter-burden). Waste material is loaded by shovels onto trucks and hauled to dump locations. Waste hauling requires the same valuable truck resources and is therefore, a major cost component in the hauling process (approximately one volume unit of waste must be moved to recover one volume unit of ore). While the timing for the removal of the waste material is not as important as the timing of the delivery of oil sand, it still needs to be removed so that the ore can be mined.

Due to the flexibility in waste removal, this operation acts as a buffer for the ore delivery by releasing trucks when they are needed to ensure production and making profitable use of equipment when production is constrained upstream of Mining. Since the ore is processed by a continuous system with only a limited surge capacity, the ore delivery must be at relatively constant rate. For this reason the production constraints used in this formulation are based on an hourly ore target which is set and controlled by Extraction.

The waste is deposited in dumps and the goal is to move as much waste with the remaining trucks as possible. No constraints on rates or variability are imposed on a shift basis. However, there will be delays at dumps as they are leveled and prepared.

For the allocation of trucks for upcoming shifts, the best estimate for the demand of Extraction is a constant hourly rate for the entire shift. Unforeseen changes from this estimate cannot be predicted and will need to be accommodated by the dispatcher and the panel operators. If there are planned changes in the extraction demand then the demand constraint can be easily modified.

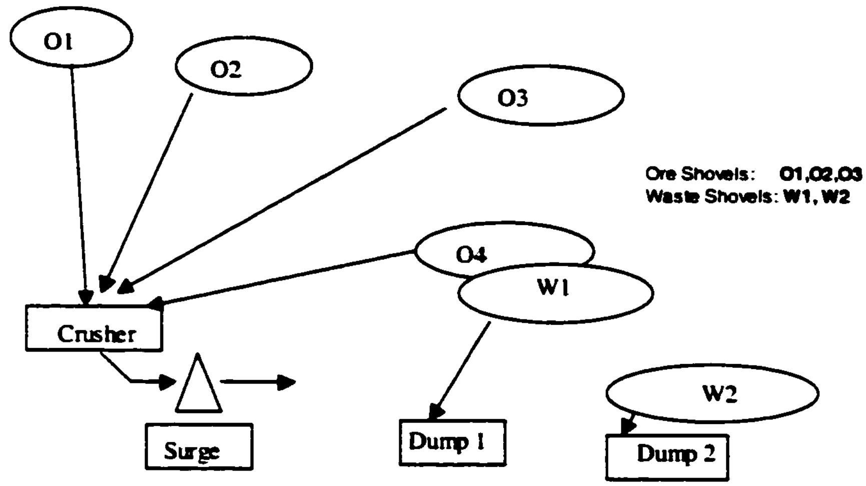


Figure 2.2 – A Simple Mine Layout

A simplified representation of the mine plan information is presented in Figure 2.2. This figure shows that ore is brought to the crusher from different locations over varying distances. Four shovels are designated to mine material either ore or waste (O1, O2, O3, W2) while one shovel (O4/W1) handles both materials. Ore trucks will be allocated to haul ore between this ore shovel group and the crusher. The remaining trucks are used for waste transport.

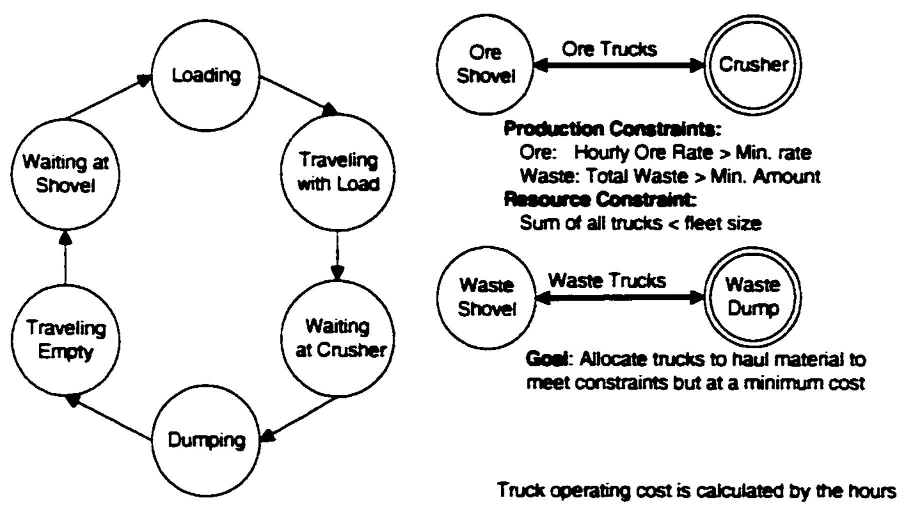


Figure 2.3 – Elements of a Truck Cycle

Figure 2.3 shows the states of a haul truck for a complete cycle and it is applied to both ore trucks and waste trucks. The model of the cycle is appropriate for trucks moving between a single shovel and crusher or between multiple shovels and a single crusher depending on the configuration specified by the dispatcher. For simplicity of presentation the model is shown with only one shovel in each cycle with two groups of shovels: ore shovel group and waste shovel group.

2.1.1. Model Simplification & Limitations

This chapter is focused on the formulation of a simple truck allocation model so that a quick assessment of the optimization application can be established. This model follows the simplified structure outlined previously and while it does not reflect all the complexity of the actual mine haulage operation, it is representative and its simplicity allows clear demonstration of the optimization problem. The simple deterministic optimization problem presented here will serve as a good base for the subsequent stochastic formulation, which will be considered in the future work. The goal is not to construct the model that matches the exact operation, but rather to show the benefits of solving the optimization problem, to indicate why it is important to account for uncertainty, and finally to identify the additional benefits that can be gained by applying stochastic solution methods.

The objective of the truck allocation problem is to determine the optimal number of trucks to haul oil sand and waste materials for a given mine plan such that ore production is satisfied and waste removal maximized. This is not the only possible formulation for the optimization objective that could be used; however, such an objective matches the current truck allocation practice reasonably well. Trucks contribute to the cost of producing Syncrude Sweet Blend whether or not they are actually being used. The first priority is to allocate truck to haul ore to meet the production demand specified by the downstream Extraction plant. The remaining trucks are allocated to haul waste. One possible truck allocation solution corresponds to a maximum truck resource left to haul waste material. This resource quantity is measured as the amount of waste material (in Tonnes) that can be hauled by the remaining trucks after the required number of trucks are sent to haul ore. The constraints of this problem can be generally classified into production constraints and resource constraints.

The Extraction plant controls the rate at which ore is pulled from the surge pile. This hourly rate (Tph) as specified by Extraction is the main ore constraint that the mining operation must satisfy. The goal of the truck allocation group is to assign just enough trucks to deliver the required ore tonnage. Extra trucks will provide greater assurance of meeting the ore constraint at the expense of the higher overall cost.

In this model, truck resource is subdivided into 3 fleets of trucks based on their loading capacity: 360T, 320T and 240T trucks. The 240T truck fleet is the oldest but also the largest while the 360T fleet is newest but smallest. These types of trucks are different in the cost of maintenance, mechanical performance, and loading capacity. But for the purpose of simplicity, only the difference in their loading capacity is considered in this thesis work.

One of the most important elements in the allocation problem is the time component, which is modeled as the truck cycle time. This cycle time is the summation of all elapsed time periods of the stages in a complete truck cycle. Figure 2.3 enumerates all the stages that a truck has to go through in a cycle. The truck cycle time typically includes the loading time at the shovel, traveling time (empty and full), dumping time, queuing time (both at shovel locations and at dump locations). The loading time depends on the capacity and efficiency of the operating shovels as well as the truck capacity. The traveling time depends on a number of parameters such as the driving habits of the driver, the truck speed, the road distance, the road condition, the weather, the weight of the payload, and the health condition of the truck. The queuing time depends on the number of deployed trucks and truck payload. All of these elements combine to determine the ultimate truck cycle time.

In the first simplification, the truck cycle time is modeled as a constant parameter, which is extracted from the online truck dispatch system. Average truck cycle times were collected from the WENCO[5] database during a selected time period and used as a representative cycle time in this model. Although truck cycle time varies among the trips for every single truck in the fleet, only an average cycle time is modeled and used in the allocation problem. One cycle value is used for the ore trucks and one for the waste trucks.

2.1.2. Production Constraints

The most important production constraint in the problem is to satisfy the hourly ore demand (tph) specified by the Extraction. Trucks are allocated to ensure consistent satisfaction of this demand. To some extent, the surge pile can help maintain a steady feed of ore; but due to its small size, the effect is negligible. Any disruption in the amount of ore delivered by haul trucks will quickly translate to operational problems in the Extraction plant.

The second production constraint, with lower order of importance, is the waste handling and movement. While the ore constraint is hourly based, this waste constraint is usually given over a longer time period (e.g., 12-hour shift or days). As a result, this waste production constraint is more flexible with respect to truck requirements. This is particularly helpful when truck allocation is done over a period time where there is a shortage of truck resources. As such, trucks can be temporarily used for ore hauling to meet the ore demand while the waste movement can be fulfilled at a later time when the problem of truck resources is resolved.

2.1.3. Resource Constraints

The haul trucks are allocated from a fixed pool of trucks. In practice, truck shortages can be resolved through renting of additional trucks from contracting companies. However, the work of this thesis does not include this renting

[5] WENCO dispatching software system is designed by Wenco International Inc. Further information about the product and company can be found at http://www.wencomine.com

option, but rather imposes an upper bound limit on the truck resources that can be allocated.

2.2. Formulation of Objective Function

The objective function can be formulated in many ways: minimize costs, maximize profit, maximize production, maximize truck utilization, etc. Each formulation has its advantages and drawbacks. The objective of the allocation problem is to maximize the waste removal while satisfying the ore requirement set by Extraction. Adoption of such an objective is based on the fact that the ore cannot be overproduced and stored and moreover, this objective closely reflects current operating practice. Objectives based on cost were investigated but it was difficult to formulate the waste constraint using this criterion.

2.3. Linear Programming

A simple formulation of the truck allocation problem is:

$$maximize\ H\sum_j \frac{60}{\tau_{Wj}} L_{Wj} x_{Wj} \qquad \text{(Tonnes)}$$

Subject to

Ore Constraint	$\frac{60}{\tau_O}\sum_j L_{Oj} x_{Oj} \geq T$	*Tph*
Minimum Waste Constraint	$\sum_j \frac{60}{\tau_{Wj}} L_{Wj} x_{Wj} \geq W_m$	*Tonnes/Period*
Resource Constraint	$x_{Oj} + x_{Wj} \leq R_j$	For $j=1,2,3$

where H denotes the time period (hours); $\tau_{Wj}, \tau_{Oj}, L_{Wj}, L_{Oj}$ represent the truck cycle time (minutes) and the truckloads (Tonnes) of the waste trucks and ore trucks respectively. W_m corresponds to the minimum amount of waste material required to be moved over the period while R_j is the truck resource constraint for trucks of type j. Finally, the non-negative decision variables x_{Oj}, x_{Wj} denote the number of trucks per hour that are required. Three different types of trucks are considered in the model with *j* being the index reflecting the truck type (e.g., *1,2,3 corresponds 240T, 320T, 360T* trucks respectively).

The presented linear optimization model, which has 6 degrees of freedom, is based on all deterministic parameters. The Simplex method [Dantzig, 1955] can be used to solve this problem and has been widely implemented in computer software.

2.4. Deterministic Results

The numerical truck solution is obtained by solving the linear optimization problem in the GAMS[6] software environment. While the number of trucks, which is the decision variables in this problem, does not reflect the actual mining operation, it helps establish a useful basis for subsequent optimization studies.

2.4.1. Model Parameters

Table 2.1 summarizes the data used to solve the simplified deterministic optimization problem. The truck cycle times are assumed to be constant over the time period of interest. Truckload is assumed to be equal to the rated capacity and constant throughout the period. The case data being used in this study also assumes an over-trucking condition, that is more than enough trucks are available to satisfy the ore requirement.

[6] The General Algebraic Modeling System (GAMS) is specifically designed for modeling linear, nonlinear and mixed integer optimization problems. The GAMS software provides users with a programming environment where they can construct optimization models and solve for them using a number of well-known mathematical solvers. The GAMS system is equipped, by default, with a certain number of standard solvers, which are capable of solving linear and nonlinear models. One of the main strength is its adaptability to work with new solvers. Further detailed information about GAMS can be found at http://www.gams.com

Ore Truck Cycle Time	$\tau_O = 25$ *minutes*
Waste Truck Cycle Time	$\tau_W = \{35,30,25\}$*minutes*
Deterministic truckload (in order of 240T, 320T, 360T trucks)	$L_{Oj} = \{220,290,327\}$ *Tonnes* $L_{Wj} = \{220,290,327\}$ *Tonnes*
Hourly Ore Constraint (Tph)	$T = 7{,}000$ *Tonnes*
Minimum Waste Constraint during the period	$W_m = 60000$ Tonnes
Time Period	H = 12 hours
Truck Fleet Size (240T, 320T, 360T)	$R_j = \{18,9,5\}$

Table 2.1 – Model Parameters (Deterministic Linear Model)

2.4.2. Results

The optimal solution of the linear deterministic truck problem is shown in Table 2.2. It is noted that the number of trucks is found as a continuous number. Although this optimal solution is not deployable due to the fractional number of trucks, this continuous, but "relaxed" solution does reveal approximately where the discrete truck solution lies. The corresponding GAMS program that generated the results in Table 2.2, 2.3, and 2.4 is listed in Appendix G1.

	240T Trucks	320T Trucks	360T Trucks
Ore Trucks	13.26	0.0	0.0
Production Ore Rate	7,000 Tph		
Maximizing waste	131,191 Tonnes (240T: 4.74, 320T: 9.00, 360T: 5.00)		

Table 2.2 – Continuous Solution

The optimal solution to the truck allocation problem, as shown in Table 2.2, is not realistic because it allocates fractional trucks. Another simplifying assumption is the fact that the truck cycle time is constant in this optimization run. Constant truck cycle time does not account for realistic events and conditions that occur during a complete truck cycle. These components include the effect of the road conditions, road distance, mechanical condition of the trucks, which all affect travel time. The truck cycle time is also affected by other time components such as the time trucks are queuing at the shovels, at the dumps, etc. This truck allocation model does not fully reflect the actual process as the current truck allocation also takes into account other issues such as shovel capacity, ore grade requirement, etc. Yet, it still can be used as a good basic model for future studies with added realistic components.

Table 2.3 shows a more realistic solution with the addition of rounding to the fractional solution found in Table 2.2. The corresponding remaining truck resource is 127,831 Tonnes. However the values in Table 2.3 are more practical than those in Table 2.2. The difference between the results in Table 2.2 and 2.3 does not appear to be significant. This suggests that it is possible to solve the problem with continuous variables and to round up the fractional solutions as required.

	240T Trucks	320T Trucks	360T Trucks
Ore Trucks	14	0	0
Actual Ore Rate	7,392 Tph		
Maximizing waste	127,831Tonnes (240T: 4, 320T: 9, 306T: 5)		

Table 2.3 – Discrete Solution (Continuous optimizer + Rounding)

The same model can be resolved using a mixed-integer[7] solver to obtain a more efficient result (Table 2.4) compared to the rounding method, which is easy but inefficient. The drawback is an increased computational requirement.

	240T Trucks	320T Trucks	360T Trucks
Ore Trucks	12	1	0
Resulted Ore Rate	7,032 tph		
Maximizing waste	129,922 Tonnes (240T: 6, 320T: 8, 360T: 5)		

Table 2.4 – Discrete Solution (Discrete Optimizer)

The truck allocation based on the results from solving the continuous problem yields the highest efficiency of truck usage. This allocation is impractical since a fractional number of trucks cannot be allocated. In practice, these fractional numbers are rounded up to the next closest integer, reflecting the actual, practical number of trucks to be allocated for the ore hauling requirement. Table 2.4 clearly shows that it is more efficient to implement the solution obtained by solving the discrete optimization problem directly.

[7] Mixed-Integer optimizer involves both continuous and discrete quantities in the calculation.

However, the focus of the current chapter is on the initial deterministic truck model with continuous truck solution and the benefit of working with discrete model is briefly mentioned without further investigation. The discussion on the formulation of the discrete truck model and its corresponding solution methods will be covered in Chapter 4.

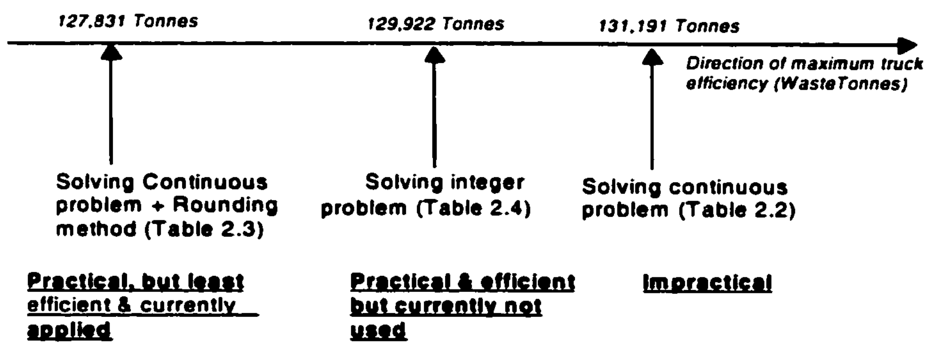


Figure 2.4 – Solution Summary of The Allocation Problem

2.5. Sensitivity Analysis Results

The remaining part of this section will focus on the sensitivity analysis based on the deterministic linear allocation truck model that corresponds to the results in Table 2.2. The sensitivity data, which is obtained as part of the GAMS results, is shown in Table 2.5. The marginal values show how sensitive the optimal solution is to certain model parameters.

	240T Trucks ($x \leq 18$)	320T Trucks ($x \leq 9$)	360T Trucks ($x \leq 5$)
Ore Trucks	13.26	0.00	0.00
Max Amount of waste that can be moved	131,191 Tonnes (4.74 240T, 9.0 320T, 5.0 360T)		
Active Constraints	**Marginal Values**[8]		
240T truck Resource	4,526 Tonnes/truck (upper limit)		
320T truck resource	6,960 Tonnes/truck (upper limit)		
360T truck resource	9,418 Tonnes/truck (upper limit)		
Hourly Ore Production Demand	-8.571 Tonnes/Ore Tons (lower limit)		

Table 2.5 – Optimal Truck Solution (Linear Det. Model)

Since changes are assumed to be small enough that no new constraints become active, it is possible to predict the effect of these changes to the overall objective function value without solving the optimization problem. At the optimal solution, the truck resource constraint, and the hourly ore demand constraint are active. Therefore, the addition of one more 240T truck to the operating fleet results in an extra 4,526 Tonnes of waste material that can be handled in the hauling operation. Similarly, 1 320T truck and 1 360T truck correspond to an extra 6,960 Tonnes, and *9,418 Tonnes* respectively. Also, if the hourly ore demand is increased, fewer trucks are available to haul waste.

It is found that each one-tonne increase of the hourly ore demand will translate in an amount of 8.57 Tonnes of waste material that is not hauled by waste trucks. An increase in ore demand corresponds to more truck resource required to haul ore, leaving less truck resource for waste hauling. (*Caution should be exercised in interpreting these values as they correspond only to the model in this study*).

[8] These *marginal* values are often generated as the output of the optimizer. They provide some indications of how sensitive the objective function value is to the certain model parameters. In linear models, the marginal values are only meaningful for *active* constraints (on the constraint boundary).

Changes to the ore truckload in tons (240T)	200 Tonnes	210 Tonnes	220 Tonnes	230 Tonnes	240 Tonnes	*Linear* Rate = **370** Tonnes / Truckload
Truck resource left for waste handling (Tonnes) and Truck Solution : [240T,320T,360T]	123,785 [14.58,0,0]	127,488 [13.89,0,0]	131,191 [13.26,0,0]	134,894 [12.68,0,0]	138,597 [12.15,0,0]	
Changes to the ore truckload in tons (320T)	270 Tonnes	280 Tonnes	290 Tonnes	300 Tonnes	310 Tonnes	*Linear* Rate = **216** Tonnes / Truckload
Truck resource left for waste handling (Tonnes) and Truck Solution: [240T,320T,360T]	126,871 [13.26,0,0]	129,031 [13.26,0,0]	131,191 [13.26,0,0]	133,351 [13.26,0,0]	135,511 [13.26,0,0]	
Changes to the ore truckload in tons (360T)	297 Tonnes	317 Tonnes	327 Tonnes	337 Tonnes	347 Tonnes	*Linear* Rate = **144** Tonnes / Truckload
Truck resource left for waste handling (Waste Tons) and Truck Solution: [240T,320T,360T]	126,871 [13.26,0,0]	129,751 [13.26,0,0]	131,191 [13.26,0,0]	132,631 [13.26,0,0]	134,071 [13.26,0,0]	
Changes to the ore truck cycle time (minutes)	23	24	25	26	27	*Linear* Rate = - **2,400** Tonnes / Minute
Truck resource left for waste handling (Tonnes) and Truck Solution: [240T,320T,360T]	135,991 [12.20,0,0]	133,591 [12.73,0,0]	131,191 [13.26,0,0]	128,791 [13.79,0,0]	126,391 [14.32,0,0]	

Table 2.6 – Effect of Changes in Truckload and Cycle Time

Other changes to the model parameters such as truckload or truck cycle time are not readily obtained without solving the optimization problem. Table 2.6 tabulates the results found by repeatedly solving the problem with different values of the model parameters in question (only one single change can be examined at a time). It must be pointed out that the efficiency of the truck allocation is measured as the amount of waste material that can be handled with the remaining truck resource. Therefore, high efficiency of truck allocation corresponds to a high amount of waste material hauled.

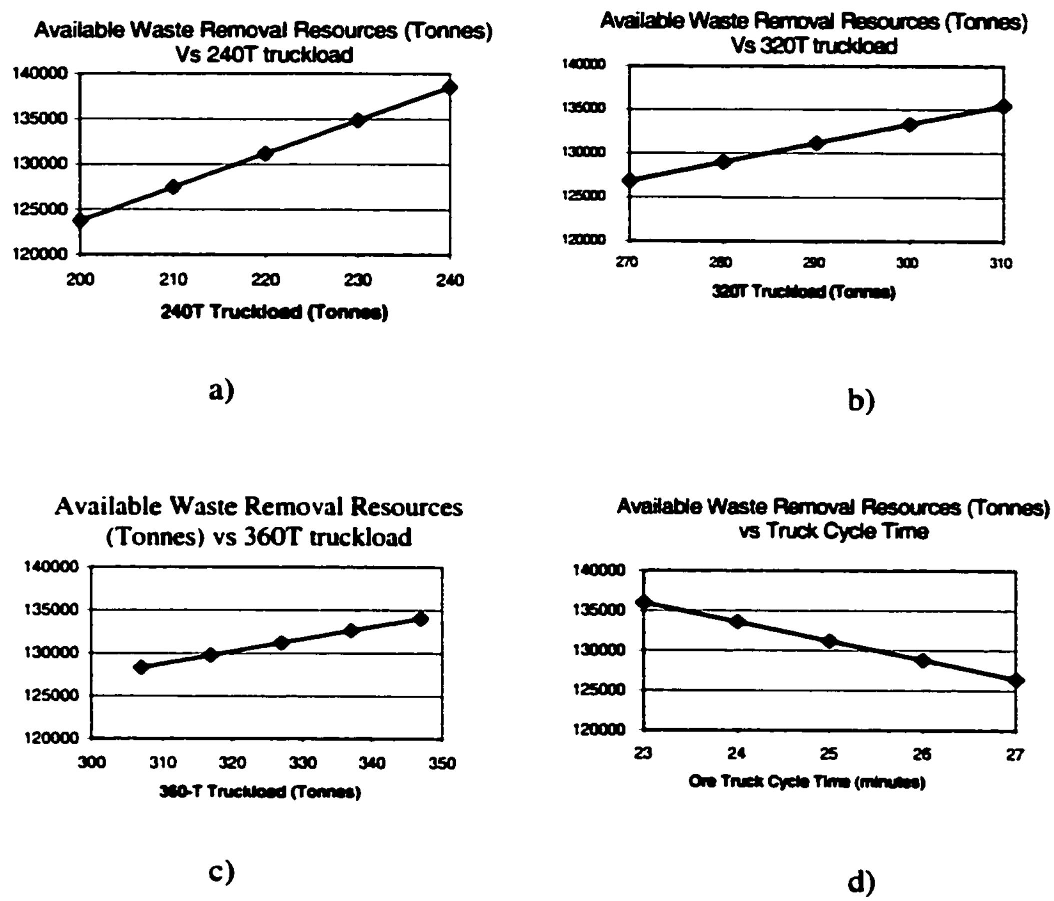


Figure 2.5 – Remaining Truck Resource vs. Truckload and Truck Cycle Time

It should be noted that the truck resources remaining is directly proportional to the changes in truckload of the ore truck (*Figure 2.5-a,b,c*). Changes in the 240T truckloads have the most impact on the overall remaining truck resource (e.g., an one-Tonne increase of the truckload of 240T trucks results in an extra amount of 370 Tonnes of waste that can be hauled, 216 Tonnes for 320T trucks and 144 Tonnes for 360T trucks). This result agrees with the fact the 240T trucks belong to the largest fleet in the overall truck resource pool.

Of all the changes, ore truck cycle time shows the highest degree of influence on the final outcome. As shown in Table 2.6, a change of 1 minute in the cycle time of the ore truck results in the change of 2,400 Tonnes of waste. Based on the current model, a *1-minute reduction* of the cycle time of the ore truck will add an equivalent of *~0.9* 240T trucks back to the resource pool ($\frac{2400\,Tons}{220\,Tonnes\,/\,240T\;Truck} = 0.9\;240T\;Truck$).

In summary, this discussion has thus far laid out the framework for the optimization study of the truck allocation problem in the mine operation. The linear deterministic model has shown to be simple in the formulation as well as in the solution effort, but it does represent a numerical approach of determining the optimal number of trucks to be deployed at the mine site. However, the allocation method currently used during the planning stage is predominantly a heuristic approach, e.g., truck is allocated based on a formula that was developed using historical data, leaving ample opportunity for the optimization effort.

Despite good initial results, the linear deterministic model is still based on continuous variables. As a result, the continuous truck solution must be rounded to the nearest feasible, discrete solution before it can be implemented. This step of rounding the non-discrete solution leads to a sub-optimal truck solution. A direct method of obtaining the discrete truck solution requires formulating a model with discrete numbers of trucks and solving for the solution using an integer solver. The formulation of such a model is omitted because the basic model is unchanged except an added restriction on the domain of the variables. Implementing the solution obtained from the integer solver was shown to provide a better solution than the rounded continuous solution. The treatment of discrete truck solution will be discussed in more detail in Chapter 4 with the parameter update approach.

The sensitivity analysis being studied so far is only valid in the model with continuous variables. Therefore, for the purpose of the performing the analysis, the model has been relaxed to involve only continuous variables. Full sensitivity analysis study involving discrete variables is outside of the scope of the thesis, and thus is omitted.

3. Stochastic Programming

In Chapter 2, the truck allocation problem was formulated as a simple linear deterministic optimization model, which was easily solved for the optimal solutions. Unfortunately, the parameters used in the model are not constant and well-known, but rather can vary randomly according to some distribution. Random changes in the model parameters can make the deterministic optimal solution non-optimal, and in the worst case, can cause the original optimal solution to become infeasible. This weakness of the deterministic solution gives rise to the need of applying Stochastic Programming where random variations of parameters are accounted for in the formulation of the optimization model. Since the 1950's, the field of Stochastic Programming has grown and has witnessed many new contributions in terms of theory and solution techniques from many operations research scientists. The material in this chapter will contain only a brief introduction to the stochastic optimization problems, focusing on two types of optimization problems: two-stage problems with simple recourse and chance-constrained problems.

3.1. Introduction to Stochastic Programming

Stochastic Programming is a branch of mathematical programming that deals with theory and methods that incorporates stochastic variations into a mathematical problem. Here, the study of Stochastic Programming is based on a class of underlying linear problem as

$$\begin{aligned}&\textit{minimize } h(\mathbf{x},\zeta)\\&\textit{subject to } g(\mathbf{x},\zeta)\geq 0\\&\textit{where } \mathbf{x}\geq \mathbf{0}\end{aligned} \tag{3.1.1}$$

or in special form:

$$\begin{aligned}&\textit{minimize } h(\mathbf{x},\zeta)\\&\textit{subject to } \mathbf{Tx}\geq \zeta\\&\qquad\qquad \mathbf{Ax}=\mathbf{b}\\&\textit{where } \mathbf{x}\geq \mathbf{0}, \mathbf{x}\in R^{n}, \mathbf{T}\in R^{n\times w}, \mathbf{A}\in R^{n\times m}, \mathbf{b}\in R^{m}, \zeta\in R^{w}\end{aligned} \tag{3.1.2}$$

The symbol ζ designates a random vector and all elements of $\mathbf{T}$ are also random. The stochastic behavior is then introduced into components embedded in the objective function $h(\mathbf{x},\zeta)$ or in the constraints $g(\mathbf{x},\zeta)\geq 0$.

Generally, uncertainty occurs in either the constraints or the objective function (or both). Uncertainty in the objective function appears in the price coefficients in many problems. Uncertainty on the right-hand-side of the constraints is often encountered in economic models where the demand or availability of either a particular resource or product is uncertain. Uncertainty in the parameters on the left-hand side of the constraints relates to uncertainty in model parameters.

3.2. Uncertainty in Constraints

This class of stochastic programming problems involves uncertainty in the constraints of the model. The first type of model involves the formulation of the probabilistic constrained stochastic programming problems:

$$\begin{aligned}&\textit{minimize } h(\mathbf{x})\\&\textit{subject to } P\begin{Bmatrix} g_1(\mathbf{x},\zeta)\geq 0\\ \vdots \\ g_r(\mathbf{x},\zeta)\geq 0\end{Bmatrix}\geq \alpha\\&\textit{where } \mathbf{x}\geq \mathbf{0}, \mathbf{x}\in R^{n}\end{aligned} \tag{3.2.1}$$

Or

$$\begin{aligned}
&\textit{minimize } h(\mathbf{x})\\
&\textit{subject to } P\{\mathbf{Tx} \geq \zeta\} \geq \alpha_i, i = 1,2,\dots,r\\
&\qquad\qquad \mathbf{Ax} = b_j, j = 1,2,\dots,m\\
&\textit{where } \mathbf{x} \geq \mathbf{0}, \mathbf{x} \in R^n, \mathbf{A} \in R^n, \mathbf{T} \in R^n
\end{aligned} \tag{3.2.2}$$

The probability α reflects the reliability of the system, especially for engineering type problems. Reliability, or safety, is a well-known and important concept in other applied problems such as finance, inventory control, resource allocation, etc.

Problem (3.2.1) corresponds to the joint-probabilistic chance-constrained model and is more difficult than problem (3.2.2) where the probabilistic constraints are imposed individually. Problem (3.2.1) can be replaced with a simpler problem, which contains individual probabilistic constraint:

$$\begin{aligned}
&\textit{minimize } h(\mathbf{x})\\
&\textit{subject to } P\{g_i(\mathbf{x},\zeta) \geq 0\} \geq \alpha_i, \ i = 1,2,\dots,r\\
&\textit{where } \mathbf{x} \geq \mathbf{0}, \mathbf{x} \in R^n
\end{aligned} \tag{3.2.3}$$

As long as the individual operations, each of which corresponds to the constraint, $g_i(\mathbf{x},\zeta) \geq 0$, are independent, such replacement is justified.

It is important to determine the individual α_i (with respect to α) such that Problem (3.2.3) is equivalent with Problem (3.2.1). If for every **x**, the random variables $g_1(\mathbf{x},\zeta), g_2(\mathbf{x},\zeta),\dots, g_r(\mathbf{x},\zeta)$ are independent of each other, then the probabilistic constraint in (3.2.1) can be simplified

$$\begin{aligned}
&P\{g_1(\mathbf{x},\zeta) \geq 0, g_2(\mathbf{x},\zeta) \geq 0,\dots, g_r(\mathbf{x},\zeta) \geq 0\} =\\
&\quad P\{g_1(\mathbf{x},\zeta) \geq 0\} P\{g_2(\mathbf{x},\zeta) \geq 0\} \dots P\{g_r(\mathbf{x},\zeta) \geq 0\} \geq \alpha
\end{aligned}$$

If $\alpha_1, \alpha_2,\dots,\alpha_r$ can be chosen such that $\sum_{i=1}^{r}(1-\alpha_i) \leq 1-\alpha$, then any **x** that satisfies (3.2.3) also satisfies the probabilistic constraint in problem (3.2.1) (based on Boole's inequality[9])

[9] Boole's inequality: $P(A_1 \cup A_2 \cup \dots \cup A_n) \leq P(A_1) + P(A_2) + \dots + P(A_n)$

The common method of solving these problems (individual probabilistic constraints) involves the conversion of the probabilistic constraint into the deterministic equivalent. The solution technique is derived based on Theorem 3.2.1 [Prékopas, 1995], which was first proved by Kataoka [1963], and van de Panne and Popp [1963].

Theorem 3.2.1

If $\zeta_1, \zeta_2, \ldots, \zeta_n$ have a joint Normal distribution, then the set of $\mathbf{x} \in R^n$ vectors satisfying

$$P\{x_1\zeta_1 + x_2\zeta_2 + \ldots + x_n\zeta_n \leq 0\} \geq \alpha \tag{3.2.4}$$

is the same as those satisfying

$$\boldsymbol{\mu}^T\mathbf{x} + F^{-1}(\alpha)\sqrt{\mathbf{x}^T\mathbf{C}\mathbf{x}} \leq 0 \tag{3.2.5}$$

where $\mu_i = E(\zeta_i), i = 1,2,\ldots,n$, $\boldsymbol{\mu} = (\mu_1 \ \ldots \ \mu_n)^T$, $\mathbf{C}$ is the covariance matrix of the random vector $\boldsymbol{\zeta} = (\zeta_1 \ \ \zeta_2 \ \ \ldots \ \ \zeta_n)^T$, F is the cumulative distribution function of the uncertainty, and α is a fixed probability, $0<\alpha<1$.

Programming under probabilistic constraints as a decision model under uncertainty has been introduced by Charnes, Cooper, and Symonds [1958]. These chance-constrained models are based on individual chance constraints. Many types of chance-constrained optimization models exist in the class of problems observed by Thompson *et al.* [1963]. They are categorized according to the objective function, namely E-model, V-model, and P-model. The E-model involves maximizing the expected value of the objective function, the V-model aims to minimize the variance of the objective function and the P-model's objective is to maximize the probability of the objective function. For example, the problem can be formulated to maximize the expected value of the amount of truck resource remaining (E-model); to minimize the expected value of the variance of the ore throughput (V-model); or to maximize the probability of meeting the specified ore throughput (P-model).

$$
\begin{aligned}
&E\text{-}model: maximize\ z = E\left[\mathbf{c}^{\mathbf{T}}\mathbf{x}\right] \\
&V\text{-}model: minimize\ z = E\left[\mathbf{c}^{\mathbf{T}}\mathbf{x} - \mathbf{c}_{\mathbf{0}}^{\mathbf{T}}\mathbf{x}_{\mathbf{0}}\right]^{2} \\
&P-model: maximize\ z = P\left[\mathbf{c}^{\mathbf{T}}\mathbf{x} \geq \mathbf{c}_{\mathbf{0}}^{\mathbf{T}}\mathbf{x}_{\mathbf{0}}\right] \\
&subject\ to \quad P\left[\mathbf{Ax} \geq \mathbf{b}\right] \geq \boldsymbol{\alpha}
\end{aligned}
$$

The common characteristic of these types of models is the fact that they are all chance-constrained based. The chance constraint method implicitly allows violation of the constraints up to a prescribed frequency (α), which is typically specified as a managerial input. However, the choice of α is frequently arbitrary [Prékopa, 1995] and was criticized as a point of weakness due to the difficulty in determining its appropriate value [Hogan *et al.*, 1981].

The second class of optimization models with uncertainty in the constraints involves conditional expectation to ensure safety [Prékopa, 1970, 1973]. With the assumption of the existence of the conditional expectations, the problem with this type of constraint is written as

$$
\begin{aligned}
&minimize\ h(\mathbf{x}) \\
&subject\ to\ E\ \{\ -g_i(\mathbf{x},\zeta) \mid g_i(\mathbf{x},\zeta) < 0\ \} \leq d_i,\ t = 1,2,\dots,r \\
&where\ \boldsymbol{x} \geq \boldsymbol{0}
\end{aligned}
\tag{3.2.6}
$$

The i^{th} inequality constraint means that the average measure of violation of the inequality $g_i(\mathbf{x},\zeta) \geq 0$, which is defined as $-g_i(\mathbf{x},\zeta)$, is limited by d_i where this average is taken only for those cases in which violations exist.

If Problem (3.1.2) is used as the underlying problem and the rows of matrix **T** is denoted by $\mathbf{T}_1, \mathbf{T}_2, \dots, \mathbf{T}_n$, the constraints in (3.2.6) are rewritten as

$$
E\ \{\ \zeta_i - \mathbf{T}_i\mathbf{x} \mid \{\ \zeta_i - \mathbf{T}_i\mathbf{x} > 0\ \} \leq d_i\ ,\ i=1,2,\dots,r \tag{3.2.7}
$$

Let $L_i(t)$ denote $E\ \{\ \zeta_i - \mathbf{T}_i\mathbf{x} \mid \{\ \zeta_i - \mathbf{T}_i\mathbf{x} > 0\ \}$ where $t = \mathbf{T}_i\mathbf{x}$, the inequality (3.2.7) becomes

$$L_i(\mathbf{T}_i\mathbf{x}) \le d_i, \quad i = 1,2,\ldots,r \tag{3.2.8}$$

or $$\mathbf{T}_i\mathbf{x} \ge L_i^{-1}(d_i), \quad i = 1,2,\ldots,r \tag{3.2.9}$$

The inequality (3.2.9) is equivalent to (3.2.8) if $L_i(t)$ is a decreasing function with respect to t. This condition can be established with the help of Theorem 3.2.2.

Theorem 3.2.2

If ζ has continuous probability distribution in R^1, its probability density function is logconcave, and $E\{\zeta\}$ exists, then $E\ \{\zeta_i - \mathbf{T}_i\mathbf{x} \mid \{\zeta_i - \mathbf{T}_i\mathbf{x} > 0\}$ where $i=1,2,\ldots,r$ also exists and is a decreasing function of $\mathbf{T}_i\mathbf{x} \in R^1$ [Prékopas, 1995].

Two types of constraints with uncertainty have been introduced: the probabilistic (chance) constraints and the expectation-based constraints. While both types of models are formulated to ensure safety (reliability) of the constraint satisfaction, each model approaches the problem differently. In chance-constrained models, the degree of the reliability is more important than the degree of the violation itself. On the other hand, expectation-based constrained models do not concern the degree of the reliability, but rather focus on limiting the average amount of violation, if any.

3.3. Uncertainty in Objective Function

Uncertainty in the objective functions is found in economic problems where prices of products or cost of building materials fluctuate. The company profit is affected by the change of the selling price of the product and the cost to building materials. Uncertainty in these prices can cause random changes in the overall profit of a company. Problems with uncertainty in the objective function can be handled in different methods. Let $h(\mathbf{x},\zeta)$ be the objective function to be minimized, the following three methods can be used to handle uncertainty [Prekopas, 1995].

3.3.1. Method I

This method involves the conversion of the original objective function, $h(\mathbf{x},\zeta)$, into a new objective function using the expectation of the objective function, $E\{h(\mathbf{x},\zeta)\}$. For example, if $h(\mathbf{x},\zeta)$ is linear in $\mathbf{x}$ (e.g., $h(\mathbf{x},\zeta)=\xi^T\mathbf{x}$), its expectation is also linear in $\mathbf{x}$ i.e., $E\{h(\mathbf{x},\zeta)\}=[E(\xi)]^T\mathbf{x}$. This method can only be used under two conditions (self-explanatory):

- The system performance has to be repeated in a large number of cases to ensure that the average of the outcomes is close to the expectation.
- The magnitude of the variation of the outcomes is not large.

3.3.2. Method II

Conceptually similar to Method I, this method also accounts for the variance of the outcomes in addition to the use of the expectation. The objective function equivalent is a linear combination of the expectation and the standard deviation:

$$pE\{h(\mathbf{x},\zeta)\}+q\sqrt{Var[h(\mathbf{x},\zeta)]}$$

where $p\succ 0, q\succ 0$ are constants. The choice of p and q is arbitrary and depends on the actual problems.

3.3.3. Method III

This method introduces a new constraint and a new objective function. Here, the uncertainty is now moved into the constraint:

Minimize d

Subject to $P\{h(\mathbf{x},\zeta)\le d\}\ge p$, $\mathbf{x}\in D$ [Kataoka, 1963]

where p is a prescribed probability, $0 < p < 1$, and D is a set determined by the remaining constraints in the problem. The new problem is equivalent to the original problem when p is large.

3.4. Chance-Constrained Programming

The chance-constrained programming models are used in various fields ranging from economic to engineering. Charnes, Cooper, and Symonds [1958] first introduced the chance-constrained concept while working on the problem of scheduling heating oil production where the uncertainties were in the weather and the demand for oil. It was established early that the solution method generally involved constructing a deterministic equivalent constraint [Symond, 1966]. In these early days, uncertainty was limited the right-hand-side matrix (**b**) even though the same analysis can be applied to cases where uncertainty occurs in the left-hand-side matrix (**A**). Almost exclusively, the objective function, which involves uncertain elements, was also optimized as an expected value of the return (minimizing cost or maximizing profit). In the problem of minimum-cost cattle feed under probabilistic protein constraint, Pann and Popp [1963] relied on the CCP method to solve for an optimal cattle feed product mix under the uncertainty of the nutritive content of various inputs. Rao [1980] and Fózwiak [1985] successfully applied the CCP method in their work related to structural optimization. Most of these original CCP problems contain the random components in the right-hand-side vector **b**.

Regardless of where uncertainty is located in the model, the main goal is to convert the probabilistic constraint into an equivalent deterministic constraint. This conversion usually results in a non-linear deterministic model, which can be solved easily with many nonlinear solvers. For a given linear probabilistic constraint, such as $P[\mathbf{Ax} \geq \mathbf{b}] \geq \boldsymbol{\alpha}$, $(\mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{1}, \mathbf{x} \geq 0)$, the conversion results in different equivalent nonlinear constraints, subject to the location of uncertainty.

In the general form of the chance-constrained problem shown in (3.41), the probabilistic constraints ensures that the inequality $\mathbf{Ax} \geq \mathbf{b}$ will be satisfied no less than $\boldsymbol{\alpha}$% of the times. The probability $\boldsymbol{\alpha}$ is also considered the reliability of the system. Other literature refers $(1-\boldsymbol{\alpha})$ as the limit of the constraint violation (i.e. violation of the respected constraint cannot exceed $(1-\boldsymbol{\alpha})$ of the times).

For linear stochastic problem, problem (3.2.3) can be rewritten as in Problem (3.4.1). The probabilistic constraints are considered individually independent (as opposed to jointly probabilistic based). The objective function $h(\mathbf{x})$ is assumed to be free from the uncertain components at this time, whereas **A, b** both are dependent on random parameters.

$$\begin{aligned}&\textit{minimize } h(\mathbf{x})\\&\textit{subject to } P\{\mathbf{Ax} \geq \mathbf{b}\} \geq \boldsymbol{\alpha}\\&\textit{where } \mathbf{A} \in R^{m \times n}, \mathbf{b} \in R^{m}, \mathbf{x} \in R^{n}, \boldsymbol{\alpha} \in R^{m}, 0 \leq \boldsymbol{\alpha} \leq 1, \mathbf{x} \geq \mathbf{0}\end{aligned} \quad (3.4.1)$$

At row i, the probabilistic constraint can be written as

$$P\{\mathbf{A}_i \mathbf{x} \geq b_i\} \geq \alpha_i$$

Common solution technique has been to convert this probabilistic inequality into a deterministic equivalent, based on the principle in the derivation of (3.2.5). Two cases are considered:

Case 1: Uncertainty in the left-hand side of the constraint

The deterministic equivalent:

$$\overline{\mathbf{A}}_i \mathbf{x} + F^{-1}(1-\alpha_i)\sqrt{\mathbf{x}^T \mathbf{C} \mathbf{x}} \geq b_i$$

where $\overline{\mathbf{A}}_i$ denotes the i^{th} row vector that contains the mean values of the uncertain parameters, $F(t)$ the accumulative distribution function, α_i the degree of confidence of meeting the constraint, b_i the right-hand-side coefficient, and **C** the covariance matrix related to the uncertain parameters.

Case 2: Uncertainty in the right-hand side of the constraint

The deterministic equivalent:

$$\overline{\mathbf{A}}_i \mathbf{x} + F^{-1}(\alpha_i)\sqrt{\mathbf{x}^T \mathbf{C} \mathbf{x}} \geq \overline{b}_i$$

where $\overline{b}_i$ denotes the mean value of the uncertain parameter on row i.

In special cases, where the uncertain parameters are considered independently distributed, the covariance matrix can be omitted in the conversion process.

Recourse-based programming is another type of Stochastic Programming method, and receives much attention in the field of Operations Research. While most of the discussion so far involved probabilistic (chance) constraints, some problems involve decisions, which are made at different times. In these problems, first-stage decisions are made before the uncertain events become realized and subsequent decisions must be made appropriately to offset any negative impact caused by the first decision given the realized events. Simple recourse problem with two-stage decisions is the main focus in this thesis and will be further investigated in the next section.

3.5. Two-Stage Recourse Programming

Two-stage recourse stochastic models are used widely in the industry. They are applied in economic problems [Dantzig *et al.*, 1994], production planning problems [Jagannathan, 1991], capacity planning [Eppen *et al.*, 1989; Dantzig *et al.*, 1992] and resource allocation problems. In these problems, first-stage decisions are made without the knowledge of the outcome of future events. These future events are uncertain, but the distribution of the uncertainty may be known or guessed. For example, when a decision is made to produce a given quantity of product **G** while the demand for **G** in the market is uncertain, managers are taking a certain amount of risk in the decision process. A low demand for **G** will result in either higher inventory cost or the loss of selling below cost. High demand for **G** will result in high sale and potentially a supply shortage of **G**. When the production level is lower than the demand level, the company is losing the chance to maximize the profit, and incurs an opportunity cost. If the distribution of the demand uncertainty is known and the cost of the alternatives like inventory or opportunity costs can be estimated, then the company can work with a recourse-based stochastic model to determine the optimal production level of product **G** to maximize the expected profit.

The first-stage decision is a single solution while the second-stage decision does not correspond to a unique realization but rather an expected value over the range of all possibilities. Each second-stage decision corresponds to a recourse action that is taken after the uncertainty is realized. The core of the recourse algorithm is to examine as many realizations as possible to determine the most appropriate first-stage decision that corresponds to the optimal objective. Uncertain model parameters are modeled as random variables with known distributions.

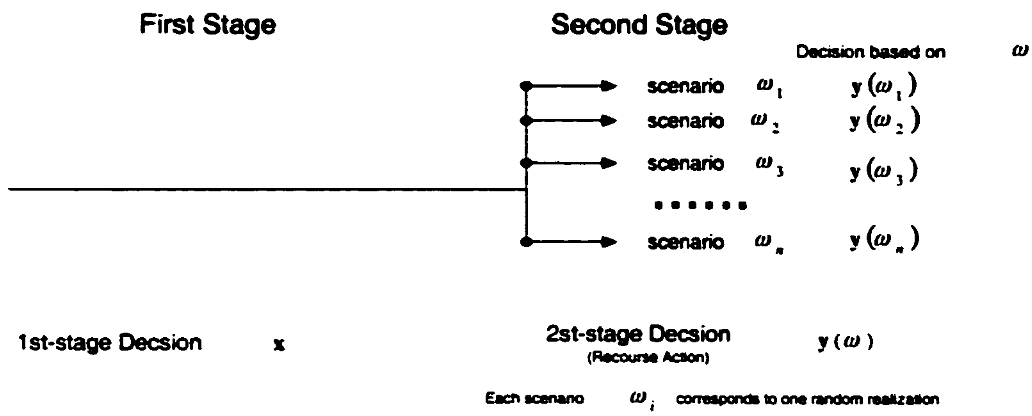


Figure 3.1 – Two-Stage Recourse Formulation

The two-stage recourse problems involve two distinctive stages and decisions. The first-stage decisions are made in the first period and the second-stage decisions, or recourse actions, are made in the second period after the uncertainty is realized. Figure 3.1 represents a classical two-stage recourse model. The objective of solving the stochastic two-stage problem is to determine the optimal first-stage decision. The second-stage decisions vary and are only important during the model formulation.

In a classical two-stage recourse problem such as the farmer problem [Birge and Louveaux, 1997], the farmer needs to make the planning decision on the sizes of the crops (*wheat, sugar beet, corn*) he plants at the beginning of the year. Since the crop yields are not known until harvest time, the farmer will face either crop shortages, which require him to buy additional crops at a higher purchase price, or an oversupply of crops, which he has to sell at a lower price. Further, a selling quota is imposed on sugar beets such that selling an amount, which is beyond the specified quota, will be done at a much lower price. Unfortunately, since he does not know the yield ahead of time, he may have to resolve to recourse actions such as buying what he needs and selling the extra crop.

The objective of the recourse-based program is to determine the optimal first-stage decision while minimizing the total cost, which includes the basic cost and the recourse cost. The basic cost in the farmer problem involves the first-stage costs such as fertilizer cost, labor cost, seed cost, etc. The recourse costs include the purchasing cost of buying needed crops and loss of selling the over-quota amount at a lower price. Figure 3.2 shows the farmer problem formulated as a two-stage recourse model. If three yields are assumed to happen with equal probability: *good* yield, *average* yield and *poor* yield. The farmer can apply the two-recourse stochastic technique to solve for the optimal crop size such that he can minimize the overall expected cost. In Figure 3.2, $\omega_1, \omega_2, \omega_3$ represent three equally likely yield cases: *good, average, poor* (each with a probability of 0.33). The objective is to determine the optimal first-stage decision on the crop sizes that would maximize the *expected* profit for the farmer.

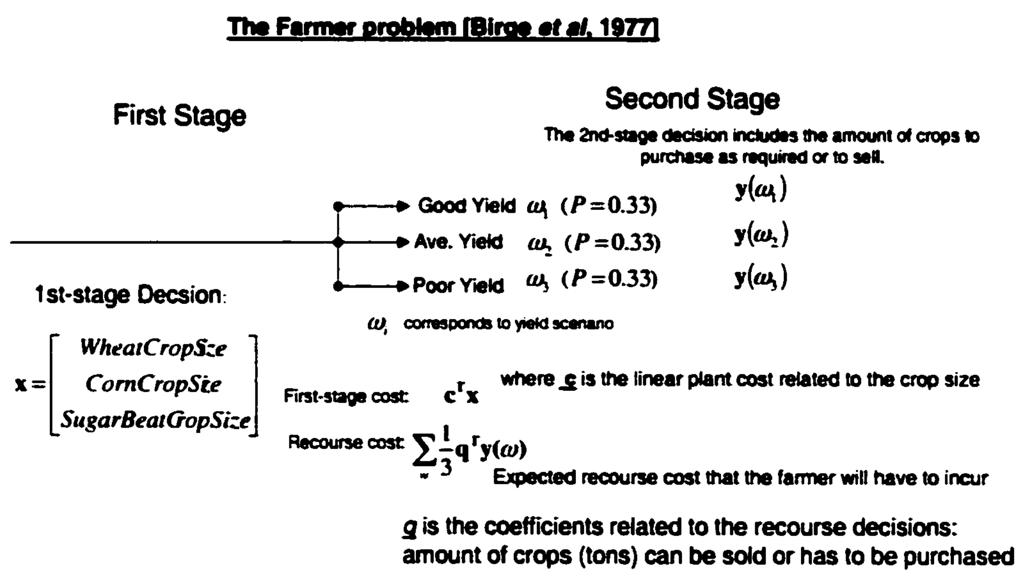


Figure 3.2 – Two-Stage Recourse Farmer Problem

Recourse-based stochastic programming traditionally demands heavy computer resources, particularly if there are many realizations to be evaluated. This computation problem limits the use of the recourse-based technique in many industrial optimizing applications [Hogan *et al.*, 1981]. While it is simple to model the recourse actions, the constraints and the objective function, complete analysis of the problem requires that recourse strategies be modeled and computed for all possible realizations of the random variables. This can make the model computationally intractable. Yet, much progress has been made in solving large stochastic programming problems. Huang *et al.* [1977] devised a method to approximate solutions to large-scale problems. This method could be applied in simple linear recourse problems to solve realistically sized problems easily [Wets, 1979]. Many recourse techniques were derived to overcome the resource limitation and approximate solutions were obtained [Birge and Wets, 1987], [Kall *et al.*, 1988].

Advances in computer technology have led to more widespread use of the recourse technique in many industrial and commercial applications [Dantzig and Infanger, 1993], [Dantzig and Infanger, 1994]. Continuing increase in computer power helps alleviate much of the resource limitation previously encountered by recourse-based problems. Further discussion on the two-stage recourse solution technique can be found in Appendix B.

In summary, the two-stage simple recourse model involves making decisions in two stages. The first-stage decision can be regarded as proactive, as it is made before the uncertainty is revealed. The second-stage decision is reactive, serves as the recourse actions, and is made after the uncertain events are realized. The second-stage decision is governed by the recourse policy, which must be set up for a practical problem. It is also dependent on the first-stage decision as well as the random events being realized. The two-stage recourse problem, though simple in principle, can be difficult to model, particularly on the recourse policy side and the appropriate penalty function in the objective function. The combination of this modeling difficulty and the intensive demand of computation all make the approach less attractive to general practitioners.

3.6. EVPI, VSS & Stochastic Programming

An important question that needs to be answered in most industrial problems is whether working with stochastic programming will provide an appreciable benefit as a worthwhile trade-off for its complexity. It is intuitive that when there are large random changes in the variables to which the deterministic optimal solution is highly sensitive, stochastic programming may provide added benefit, since it can account for the uncertainty. Theoretically, many scientists have studied the concept of Expected Value of Perfect Information, or EVPI, ([Raiffa and Schlaifer, 1961] and the Value of Stochastic Solution, or VSS, [Birge and Louveaux, 1997]) and used these values as the measures of the value of stochastic solutions over their deterministic counterparts.

3.6.1. EVPI and VSS in Truck Allocation

All calculations for the EVPI and VSS values are based on a similar, but simpler, problem of allocating trucks to haul ore. Ore truckloads and ore truck cycle-times are two uncertain parameters that are assumed to vary independently[10], and according to a Normal distribution with known standard deviations (Table 3.1). These random variations are considered to be independent. To reduce the heavy demand of the computer resource required to solve the problem the allocation problem involves only one truck type (320T trucks).

[10] Truck cycle time and truckload are not correlated. Refer Appendix H for their correlation data.

Uncertainty Parameters	Independent Normal Distributions[11]
Mean truckload of 320T ore trucks	$\overline{L}_O = 290\ tonnes$ $\sigma = 25\ tonnes$
Mean cycle time of 320T ore trucks	$\overline{\tau}_O = 24\ minutes$ $\sigma = 5\ minutes$

Table 3.1 – Distribution Characteristics of Uncertain Parameters

EVPI and VSS are calculated from the *wait-and-see* (WS) solution [Madansky, 1960], the *Recourse value* (RP), and the *expected results* of using the *EV solution*, EEV. A brief introduction on how to determine these values is presented in the following section (a more thorough development of the EVPI and VSS values is provided by Birge and Louveaux [1997]).

First, let ξ be the random variable whose realizations correspond to the various scenarios. Define

$$\min z(\mathbf{x},\xi) = \mathbf{c}^{\mathbf{T}}\mathbf{x} + \min\{\mathbf{q}^{\mathbf{T}}\mathbf{y} \mid \mathbf{W}\mathbf{y} = \mathbf{h} - \mathbf{T}\mathbf{x}, \mathbf{y} \geq 0\}\} \qquad (3.4.1)$$

as the optimization problem associated with one particular scenario ξ. The wait-and-see solution is defined as the expected value of the optimal solution, i.e. $WS = E_{\xi}\left[\min_{\mathbf{x}} z(x,\xi)\right]$. The wait-and-see solution is rather theoretical because the decision is not made until the uncertainty becomes realized.

On the other hand, the recourse solution corresponds to the solution to the recourse problem (RP) and it is defined as

$$RP = \min_{\mathbf{x}} E_{\xi} z(x,\xi). \qquad (3.4.2)$$

In practice, it is common to simplify the problem by replacing all the random variables with their mean values and solve the problem using the deterministic approach. Such problem is called the expected value (EV) problem,

$$EV = \min_{\mathbf{x}} z(\mathbf{x},\overline{\xi}) \qquad (3.4.3)$$

---

[11] Uniform distributions and triangular distributions are possible, but not used since, in practice, normal distributions remain as the most popular distribution.

where $\bar{\xi} = E(\xi)$ denotes the expectation of ξ. Let $\bar{x}(\bar{\xi})$ be an optimal solution to (3.4.3), also called the *expected value solution*. The expected result of using the EV solution is defined as $EEV = E_{\xi}\left(z(\bar{x}(\bar{\xi}), \xi\right)$.

Now, the Expected Value of Perfect Information and the Value of Stochastic Solution[12] can be determined as

$$EVPI = RP - WS$$
$$VSS = EEV - RP$$

The parameters of the model for the truck allocation problem are summarized in Table 3.2.

[12] The concept of EVPI and VSS is more suitable for the recourse-based programming approach, where the amount of constraint violation is measurable. On the other hand, pure chance-constrained approach only controls the frequency of the constraint violation, not the extent of the violation.

	Optimization Models used to evaluate the WS, RP, and EEV values	
WS	Objective	Maximize $Z(s_1, s_2) = [R - x] L_W \dfrac{60}{\tau_W}$
	Production constraint (Tph)	$L_O(s_1) x \dfrac{60}{\tau_O(s_2)} \geq D$
	Resource Constraint	$x \leq R$
	Result	$WS = E_{(s_1, s_2)} \{Z(x(s_1, s_2))\}$
RP	Objective (Waste Tonnes)	Maximize $Z = [R - x] L_W \dfrac{60}{\tau_W} - \sum_{s_1, s_2} \dfrac{1}{N} Q(s_1, s_2)$
	Production Constraint (Tph)	$L_O(s_1) x \dfrac{60}{\tau_O(s_2)} + Q(s_1, s_2) \geq D$
	Resource Constraint	$x \leq R$
	Result	RP = Z
EEV	Objective (Waste Tonnes)	Maximize $Z = [R - x] L_W \dfrac{60}{\tau_W}$
	Production Constraint (Tph)	$\bar{L}_O x \dfrac{60}{\bar{\tau}_O} \geq D$
	Resource Constraint	$x \leq R$
	Result	$EEV = E_{(s_1, s_2)} \{Z - 2Q(s_1, s_2)\}$ where $Q(s_1, s_2) = \begin{cases} 0 & \text{if } \bar{L}_O x \dfrac{60}{\bar{\tau}_O} \geq D \\ D - \bar{L}_O x \dfrac{60}{\bar{\tau}_O} & \text{if } \bar{L}_O x \dfrac{60}{\bar{\tau}_O} \prec D \end{cases}$

Table 3.2 – Models Used to Determine WS, RP, and EEV

Parameters used in Table 3.2 include:

Z	Truck resource remaining to be maximized
$x \geq 0$	Number of trucks to be allocated
R	Size of truck fleet in use
L_W	Truckload of waste truck
$\bar{L}_O$	Mean truckload of ore trucks (tonnes)
s_1	Realization of the ore truckload
s_2	Realization of the ore truck cycle-time
$L_O(s_1)$	Truckload of ore trucks at realization s_1
τ_W	Mean cycle time of waste trucks (minutes)
$\bar{\tau}_O$	Mean cycle time of ore trucks (minutes)
$\tau_o(s_2)$	Mean cycle time of ore truck at realization s_2
$Q(s_1, s_2)$	Extra ore to added as a recourse action at realization (s_1, s_2)

D Ore rate (Tph)

N Total number of realizations (Number of unique pair of $L_O(s_1)$ and $\tau_o(s_2)$)

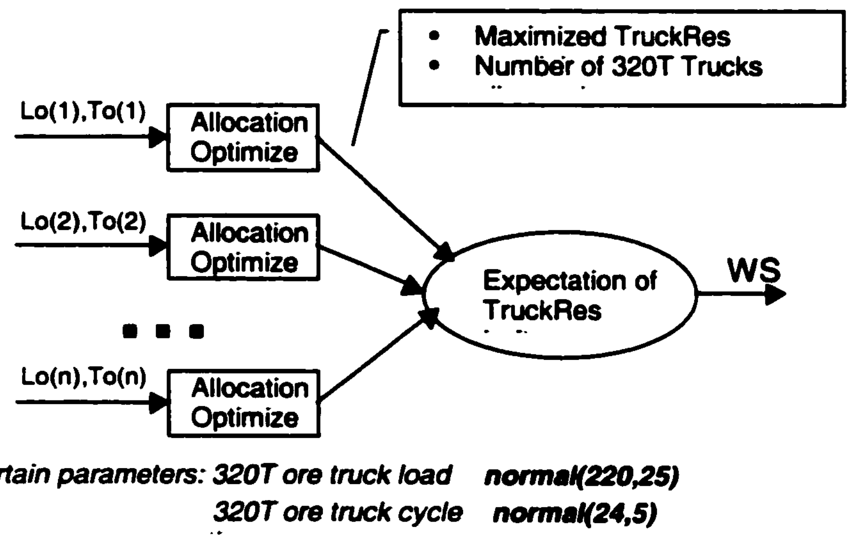


Figure 3.3 – Calculating the WS value

Calculating the *wait-and-see* value (WS) is determined as the expectation of the objective function values found by repeatedly solving the optimization problem. The number of times the model is solved is equal to the total number of realizations being considered. If s_1, s_2 are the numbers of realizations of the ore truckload and the ore truck cycle-time respectively, the total number of realizations is the product: $s_1 s_2$. Figure 3.3 illustrates the process of finding the WS value. Corresponding GAMS program to calculate the WS value can be found in Appendix G3. The GAMS calculation is done based on 100[13] Normal samples of ore truckload and 100 Normal samples of the ore truck cycle-time. The number of times the model is solved to determine the WS value is as high as 10,000.

The RP value is obtained as the optimal solution to the recourse-based problem. Table 3.2 shows the extensive form of the stochastic model in which all realizations of the uncertainty are accounted for. Because of the existence of uncertainty in the ore truckload and ore truck cycle-time, the amount of ore delivered by 320T trucks will fluctuate causing a shortage in the ore throughput. It is assumed that this shortage can be overcome by bringing ore from another source at a certain cost. This cost is twice[14] as high as the one that would have been incurred if the appropriate number of trucks were assigned for the task. This high penalty is embedded into the objective function so that the optimal solution is such that the need of the recourse actions is minimized.

The factor of 2 is chosen to add some weight to the recourse-based ore amount. This factor is chosen just high enough to ensure that the penalty of adding extra ore from an imaginary source is higher than the cost to haul ore from the mine. Too high a value for this factor can cause the EEV value to be insensitive to the objective value Z in the EEV problem (Table 3.2).

Figure 3.4 shows the process of determining the RP value. The model contains a high number of ore throughput constraints, each of which corresponds to a unique scenario (Number of constraints: $1+s_1 s_2$).

---

[13] The population size should be chosen large enough ($n \geq 30$) so that the sample set is representative of the population. while too large a sample size will make the problem computing intensive.

[14] This recourse-based component should carry such weight that original truck allocation is a controlling factor in the optimal direction of the objective function. In this truck allocation problem. this recourse-based component is somewhat fictitious. and not a realistic option. However. this factor should be greater than 1 to fulfill this purpose.

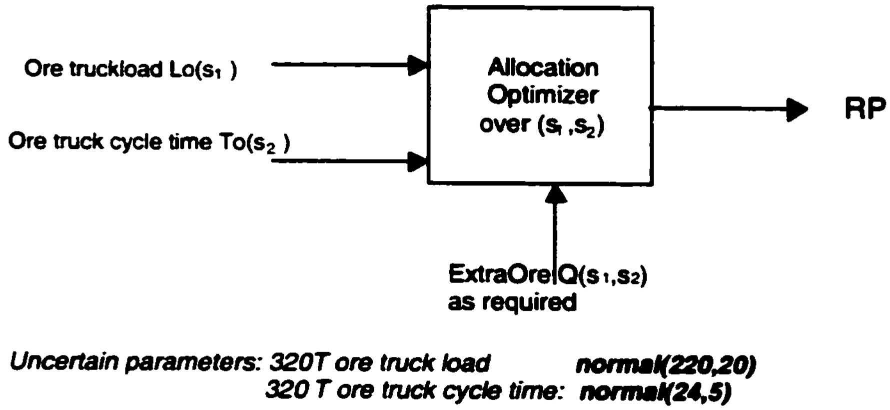


Figure 3.4 – Calculating the RP value

The EEV value corresponds to the worst case scenario where the truck allocation is done based on the mean value of the ore demand (Tph). This condition is very close to reality, where trucks are assigned to haul ore to satisfy an average ore rate. Any shortfall of the ore due to the uncertainty of the amount of ore delivered is made up by an extra ore amount. This amount, as in the RP-case, comes from the additional haul trucks that are added to the route. This results in a reduced amount of the overall truck resource remaining. The EEV value is calculated as the expectation of the truck resource remaining over the domain of all realizations (Figure 3.5).

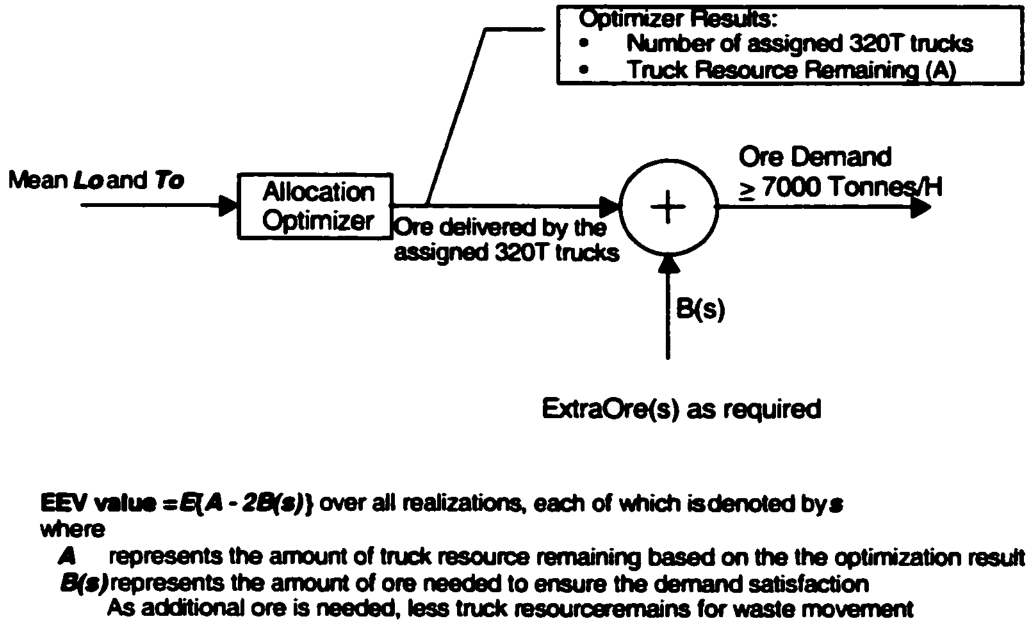


Figure 3.5 – Calculating the EEV value

Figure 3.6 shows that the expected value of the perfect information is equivalent to *8460 Tonnes/shift* of truck resource remaining for waste and that the value of the stochastic solution is *840 Tonnes/shift*. In other words, the price of knowing the perfection information is the savings of ~19 truckloads/shift, and the benefit of using stochastic solution is ~3 truckloads/shift. When either number is large enough to represent worthwhile economic benefit, the application of the Stochastic Programming approach is recommended.

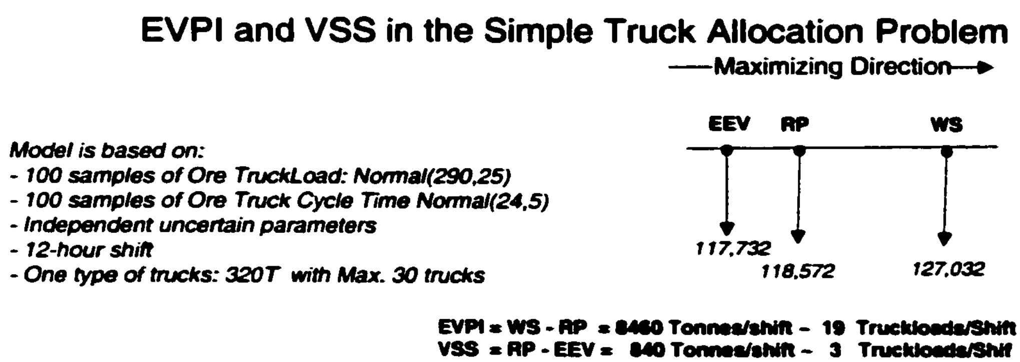
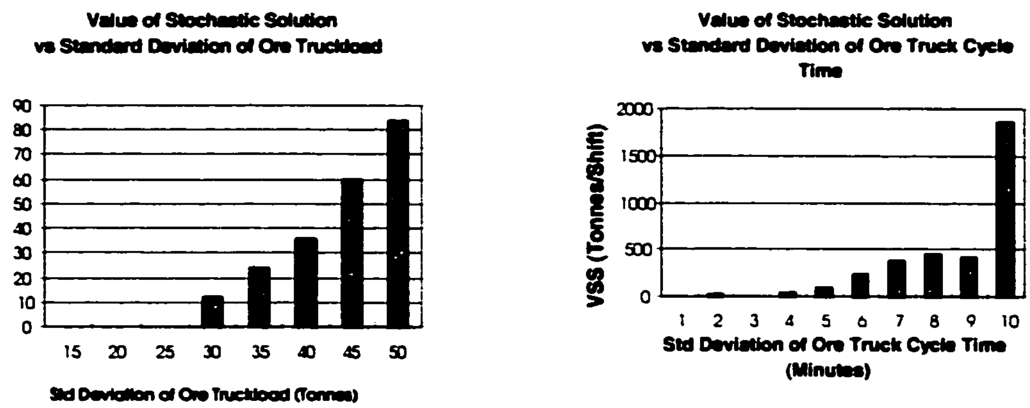


Figure 3.6 – The EVPI and VSS values



a) b)

Figure 3.7 – VSS vs. Std. Dev. of Truckload and Cycle Time

For the distributions used in this calculation (i.e., Normal with a specified standard deviation) the value of perfect information is 10 times as high as the solution from a better (more complicated) algorithm. If the standard deviation of the mean were larger, then the factor would be greater. This strongly points to the value of better information for problems with Normally distributed errors. Recourse algorithms will provide greater value for skewed distributions or skewed penalties, where taking this into account will provide a first stage answer which will minimize the cost of expensive recourse actions.

Both graphs in Figure 3.7 show that as the uncertainty becomes smaller, the value of stochastic solution decreases. These effects were investigated by repeating these calculations while the standard deviations of the ore truck cycle-time and the ore truckload are changed one at a time. These two charts consistently show the decreasing trend of the effect of the uncertainty as the standard deviations are reduced.

3.7. Two-Stage Recourse Results

The following section includes optimization results, obtained from solving the truck allocation model using two different methods (two-stage recourse and chance-constrained). While GAMS was selected as the main software environment solving the truck allocation problem, the numerical results obtained from using a different software tool are also presented. Comparison between the results using two different software programs will be done to draw further insight into the stochastic approach.

3.7.1. Two-Stage with Simple Recourse Model

The basis of recourse algorithm is to ensure a complete satisfaction of the constraints by relying on appropriate recourse actions. In the first stage, the trucks must be allocated to haul ore. This allocation decision, when made, will stay in effect for the whole 12-hour shift. Due to the variations in the ore truckload and the cycle times, the overall shift average may be different from that used in the original allocation and it is possible that the ore constraint cannot be met. Therefore, the second stage decision involves an addition of the ore amount of Q (Tonnes), which is needed at the end of shift (Figure 3.8). Such a recourse action is not suitable in practice since the ore rate must be maintained throughout the period and actions must be taken to avoid any ore shortfall. The recourse formulation allows users to calculate how much ore would need to be made up over the length of the shift and calculate an equivalent number of trucks, which need to be shifted from other duties to satisfy the ore constraint. In effect, extra trucks would be moved to the route to satisfy the ore demand as soon as it became apparent that they were needed.

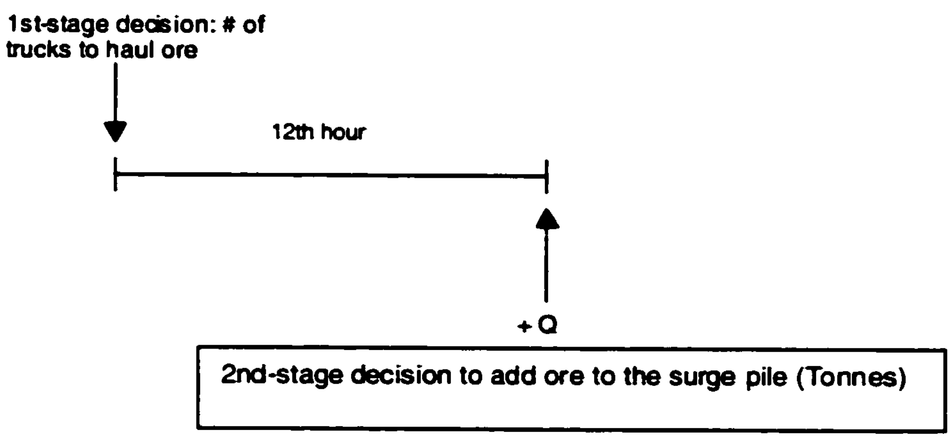


Figure 3.8 – Two-Stage Recourse Timing Diagram

The objective function used in the recourse-based formulation of the truck allocation problem is to maximize the total truck resource remaining after the production constraint has been satisfied. This truck resource is measured as the amount of waste that would be hauled if all remaining truck resources are devoted to the movement of waste. The stochastic recourse-based model is derived from the deterministic model in Chapter 2. The two differences are the introduction of uncertain variables and the second-stage variable Q, which represents the amount of ore that is required to make up the shortfall. Thus the recourse-based model is

$$maximize\ Z = \sum_{j=1}^{3} \left[R_j - x_j\right] L_{Wj} \frac{60}{\tau_w} - \eta Q$$

subject to the following constraints:

Production constraint: $H \sum_{j=1}^{3} x_j L_{Oj} \frac{60}{\tau_O} + Q \geq HD$ (Tonnes/Shift)

Truck Resource constraint on truck type *j*: $x_j \leq R_j$

Non-negativity constraints: $x_j \geq 0$ for all *j*, $Q \geq 0$

where *H* represents the number of hours in a shift, η reflects the weight of the penalty or increased cost of the recourse action and D denotes the hourly ore demand (Tonnes/Hour)

Adding extra trucks or moving them from another service during the shift comes at an additional cost. This additional cost is represented by η [15](η is equal to 2 for this example). The overall effect is to ensure that the optimal solution relies more on a proper initial truck allocation than more on recourse actions to satisfy the ore constraint.

The uncertainty being modeled is in the truckload and the truck cycle-time. It is reasonable to model this variability as random and independent from each other. The total number of random parameters in the model is dependent on

---

[15] This factor is the same factor as that found in previous section where the EEV value is determined.

the number of truck types being considered, e.g. 3 types of trucks correspond to 6 independent random parameters. These uncertain parameters are Normally distributed with given mean values and given standard deviations (Table 3.3) while the remaining deterministic parameters are shown in Table 3.4.

Uncertain parameters	Normal Distribution Characteristics	
	Mean	Standard Deviation
240T Ore Truckload (Tonnes)	220	20
320T Ore Truckload (Tonnes)	290	25
360T Ore Truckload (Tonnes)	327	35
240T Ore Truck Cycle-time (Minutes)	24	5
320T Ore Truck Cycle-time (Minutes)	24	5
360T Ore Truck Cycle-time (Minutes)	24	5

Table 3.3 – Deterministic, Uncertain Model Parameters

Deterministic parameters	Values
Constant waste truck cycle-time	30 minutes
The number of hours in the period	12 hours
Production Constraint	$OreRate \geq 7{,}000\ Tonnes/Hour$
Truck Total Resource Available	240T truck fleet: 18 trucks 320T truck fleet: 9 trucks 360T truck fleet: 5 trucks
Waste constraint per period	Waste per period $\geq 60{,}000\ Tonnes$

Table 3.4 – Deterministic Model Parameters

The two-stage recourse-based program for the truck allocation problem is developed in the GAMS modeling language and solved with the GAMS/DECIS[16] solver [Infanger, 1999], which was developed to solve large-scale stochastic programs that include uncertain parameters in the coefficients and in the demand. With the exception of some modification to the stochastic model, DECIS solver can work with the same GAMS program that was developed for the deterministic model, making an easy transition from the

[16] The DECIS solver is developed by Dr. Infanger to solve large-scale stochastic optimization problems that include parameters that are not known with certainty. It employs Benders decomposition and uses advanced Monte Carlo sampling techniques. The distribution characteristic of the uncertain parameters is not required as advanced Monte Carlo sampling technique is used to generate realizations that are mapped

deterministic model to the two-stage recourse model. The DECIS solver relies heavily on the probability information supplied in the GAMS input file. The probabilistic data is expected in a model file named ***model.stg***, which is to be created from within the GAMS program. Since DECIS solver can only work with discrete probability, continuous probability information must be converted to discrete data. Table 3.5 contains distribution characteristics with means and standard deviations for the uncertain parameters (3 types of truckloads and 1 truck cycle-time).

240T truckload $\mu = 220$ $\sigma = 20$ (Tonnes)		320T truckload $\mu = 290$ $\sigma = 25$ (Tonnes)		360T truckload $\mu = 327$ $\sigma = 35$ (Tonnes)		Ore Truck Cycle-time $\mu = 24$ $\sigma = 5$ (Tonnes)	
Truckload (Tonnes)	**Prob.**	**Truckload (Tonnes)**	**Prob.**	**Truckload (Tonnes)**	**Prob.**	**Cycle-time (mins)**	**Prob.**
135	0.0001	185	0.0002	177	0.0001	4	0.0003
152	0.0015	206	0.0017	207	0.0014	8	0.0025
169	0.015	227	0.0159	237	0.0144	12	0.0198
186	0.0846	248	0.0862	267	0.0834	16	0.0927
203	0.2339	269	0.2331	297	0.2345	20	0.2291
220	0.3296	290	0.3259	327	0.3322	24	0.3112
237	0.2339	311	0.2331	357	0.2345	28	0.2291
254	0.0846	332	0.0862	387	0.0834	32	0.0927
271	0.015	353	0.0159	417	0.0144	36	0.0198
288	0.0015	374	0.0017	447	0.0014	40	0.0025
305	0.0001	395	0.0002	477	0.0001	44	0.0003

Table 3.5 – Probability data[17] (used in GAMS Program)

Table 3.6 presents the truck solution found from solving the recourse-based truck model[18]. The remaining trucks can be used to move waste with the mean amount of 129,002 Tonnes moved in a 12-hour shift. The recourse amount Q is not considered as a part of the solution suggested by the optimizer because the recourse amount also varies as a random quantity. The main aim is not to

---

to the specified discrete probability. However, this solver is limited to the recourse-based stochastic problems and is designed currently to handle only *2-stage stochastic linear* programs.

[17] It is essential that the 10 probabilities add up to 1 (to ensure the correct execution of GAMS/DECIS program)

[18] Corresponding GAMS program is listed in Appendix F2

determine Q, but rather to determine the first-stage decision, the initial truck allocation in this case, such that the expected value of the remaining truck resource is maximized.

Truck Solution	Maximized Objective Function Value
240T Trucks: 2.28 320T Trucks: 4.15 360T Trucks: 4	Truck Resource Remaining: 129,002 Tonnes

Table 3.6 – Recourse-Based Optimal Result

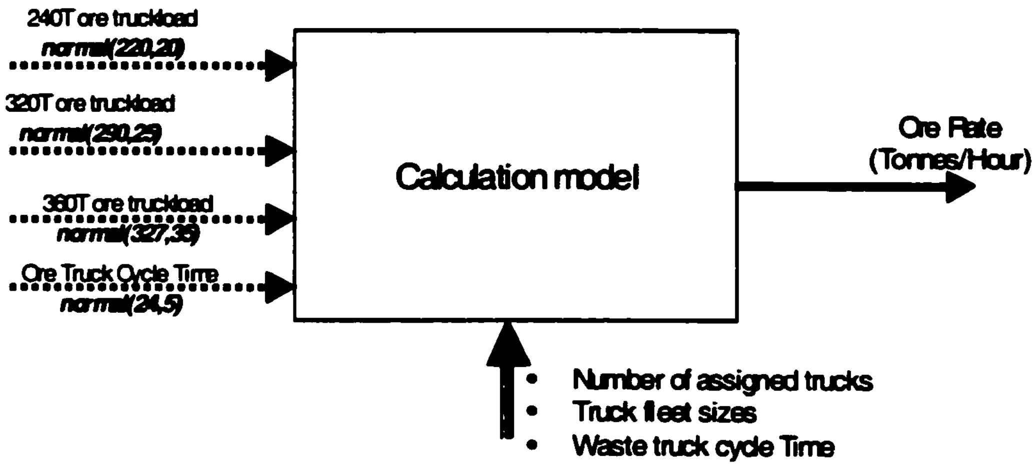


Figure 3.9 – Calculation Model Using Allocation Results

It is important to validate the performance of the truck solution that is found from using the stochastic methods. Figure 3.9 illustrates the process of determining the effectiveness of the truck solutions. The results of this calculation will provide additional insight into the performance of the allocation solution. The hourly ore rate, *OreRate* (Tph), and the truck resource remaining that can be devoted completely to the waste hauling are determined according to the following equations

$$OreRate = 60\sum_{j=1}^{3}\frac{L_{Oj}x_j}{\tau_O} \quad \text{(Tph)}$$

$$TruckLeft = 60\sum_{j=1}^{3}\frac{L_{Wj}\left(R_j - x_j\right)}{\tau_W} \quad \text{(Waste Tonnes)}$$

The calculation scheme, which is equivalent to the stochastic truck models, is implemented in Microsoft Excel with the Crystal Ball[19] add-on software. The uncertain data is simulated via the Crystal Ball software using known distribution characteristics. Under the calculation scheme, the hourly ore throughput is the forecast quantity. Frequency data of the ore rate and the remaining truck resource are summarized in Figure 3.10 while the graphical output from Crystal Ball is captured in Figures 3.11 and 3.12.

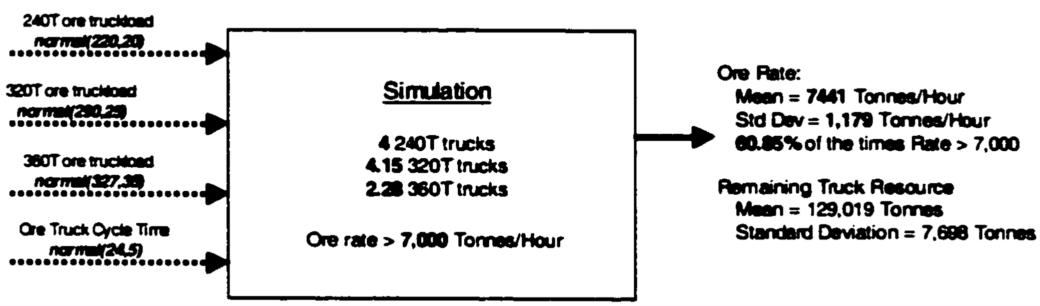


Figure 3.10 – Ore Throughput Using Recourse-Based Solution

[19] Crystal Ball is a user-friendly, graphically oriented forecasting and risk analysis program that takes the uncertainty out of decision-making. Since it is implemented as an Excel Add-on, Crystal Ball can leverage the power of visual calculation offered in Microsoft Excel spreadsheet software. As it relies on the Monte Carlo simulation, Crystal Ball forecasts the entire range of results possible for a given situation as well as provides abundance statistical information accompanying the forecasted results. Further product information is available at http://www.decisioneering.com

(The software used in the thesis work: Crystal Ball 2000 with Microsoft Excel 97 running on a Pentium III personal computer)

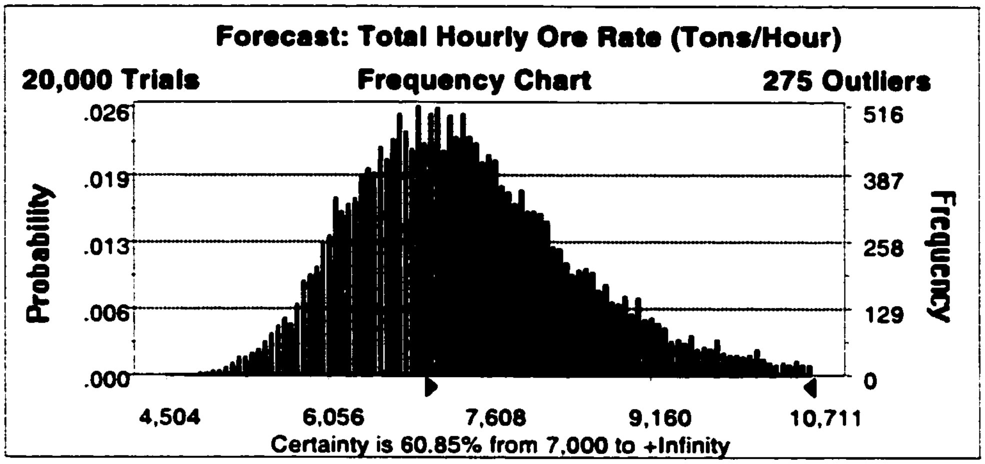


Figure 3.11– Frequency chart of ore rate (with truck solution in Table 3.6)

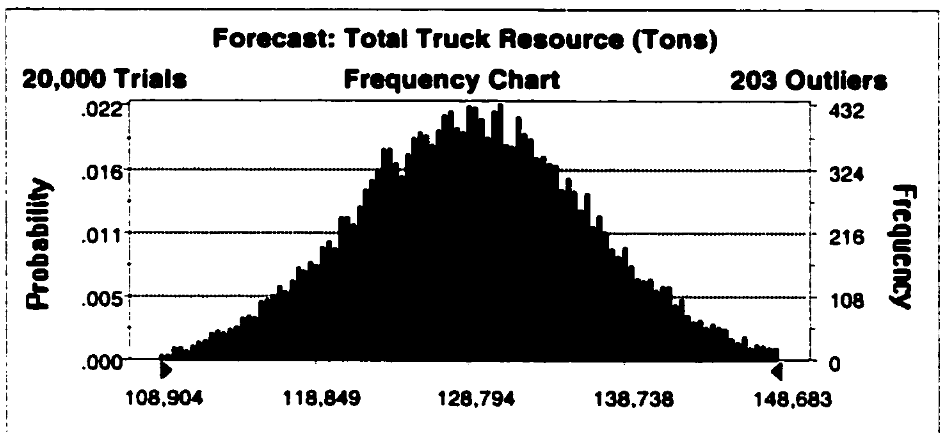


Figure 3.12 – Remaining Truck Resource Simulation (Table 3.6 Solution)

Using the model data in Table 3.3 & 3.4 and Normally distributed data generated according to Table 3.5 data, the distribution of the average rate of ore is obtained as shown in Figure 3.11. It is found that the ore is delivered with a mean rate of 7,441 Tph and with a standard deviation of 1,179 Tph. This rate of ore will only satisfy the ore production requirement 61% of the time. As the result, this implementation relies on the recourse action to offset the shortfall. This source of ore can come from additional trucks that are assigned to the route as time goes by. In the actual operation, it is impractical to assign trucks to deliver the shortfall at the end of any period. In reality trucks are shifted as necessary to maintain the desired throughput. The amount of recourse can simply be thought of as a measure of how much reallocation needs to be done. If we do not reallocate during the shift but retain the original allocation, the two-stage recourse will measure the size of the production shortfall. The second issue is that 7,441 Tph cannot be delivered to the surge facility if only 7000 Tph are withdrawn by Extraction. The dump control will create truck queues that will lower the truck productivity and maintain 7000 Tph delivery rate.

3.7.2. Chance-Constrained Model

Chance-constrained model allows a certain frequency of violation of the model constraint, in this example, the ore production. It can also be interpreted as the degree of confidence in which the ore production is met. For example, a value of $\alpha = 0.95$ means that the ore production constraint will be met 95% of the time.

The production is primarily affected by the ore truck cycle time and ore truckload. Other model parameters such as the waste truck cycle-time, waste truckload, truck fleet sizes are assumed to be constant as they are considered to contribute less to the overall uncertainty of the problem when compared to the primary uncertain parameters discussed above. However, the truck fleet size can also play a major role in the stochastic model as an important uncertain parameter, e.g. uncertain overall truck resource. But the fleet size is kept constant to reduce the problem complexity. (This assumption corresponds to the over-trucking condition in the operation).

The optimization problem is formulated using the chance-constrained method as:

$$Maximize\ \ Z = \sum_{j=1}^{3} \left[R_j - x_j\right] L_{Wj} \frac{60}{\tau_W},\ \text{where j=1, 2, 3} \qquad \text{(Waste Tonnes)}$$

Subject to

Production constraint

$$P\left\{\sum_{j=1}^{3} x_j L_{Oj} \frac{60}{\tau_O} \geq D\right\} \geq \alpha \qquad \text{(Tph)}$$

Truck Resource constraint on truck type *j*:

$$x_j \leq R_j$$

In the chance-constrained programming, it is important to convert the probabilistic constraint into a deterministic equivalent constraint since after this step is complete, a nonlinear solving technique can be applied to solve the problem. The following analysis is based on one of the two probabilistic constraints:

$$P\left\{\sum_{j=1}^{3} x_j L_{Oj} \frac{60}{\tau_O} \geq D\right\} \geq \alpha$$

$$P\left\{\left[60\frac{L_{O1}}{\tau_{O1}}x_1 + 60\frac{L_{O2}}{\tau_{O2}}x_2 + 60\frac{L_{O3}}{\tau_{O3}}x_3\right] \geq D\right\} \geq \alpha$$

New variables, g_j's, are introduced such that, $g_j = \frac{L_j}{\tau_j}$ where $j = 1,2,3$

The probabilistic constraint becomes:

$$P\{[60g_1x_1 + 60g_2x_2 + 60g_3x_3] \geq D\} \geq \alpha$$

In the constraint, the only uncertain variables are g_1, g_2, g_3. These uncertain quantities will vary with Normal distributions because the truckloads and truck cycle-times vary with a Normal distributions, and independently from each other. Their Normal distribution characteristics, i.e. their means and standard deviations can be derived based on those of L and τ. According to Kotz *et al.* [1982] these standard deviations can be approximated using a Taylor expansion. Using only the first term in the expansion,

$$\sigma_{g_j} \cong \frac{L_j}{\tau_j}\sqrt{\frac{\sigma_{L_j}^2}{L_j^2} + \frac{\sigma_{\tau_j}^2}{\tau_j^2}} \quad \text{and} \quad \bar{g}_j = \frac{\bar{L}_j}{\tau_j} \quad \text{where } j = 1,2,3$$

The probabilistic constraint can be rewritten as

$$P\{V \geq D\} \geq \alpha$$

where $V = 60g_1x_1 + 60g_2x_2 + 60g_3x_3$

Since V is linearly dependent upon g_1, g_2, g_3, it is also Normally distributed with the mean $\overline{V}$, and the standard deviation σ_V

$$\overline{V} = 60\overline{g}_1x_1 + 60\overline{g}_2x_2 + 60\overline{g}_3x_3$$
$$\sigma_V = 60\sqrt{x_1^2\sigma_{g_1}^2 + x_2^2\sigma_{g_2}^2 + x_3^2\sigma_{g_3}^2}$$

Following the general development presented earlier, one can derive the equivalent inequality

$$\frac{D-\overline{V}}{\sigma_V} \le F^{-1}(1-\alpha)$$

where $F\left(\frac{D-\overline{V}}{\sigma_V}\right) = P\left[z_V \le \frac{D-\overline{V}}{\sigma_V}\right]$ represents the cumulative distribution function

or

$$\overline{V} \ge D - \sigma_V F^{-1}(1-\alpha)$$

The equivalent deterministic model now becomes

Maximize $z = \sum_{j=1}^{3}[R_j - x_j]L_{Wj}\frac{60}{\tau_W}$, where $j=1, 2, 3$ (Waste Tonnes)

Production constraint:

$$60\overline{g}_1x_1 + 60\overline{g}_2x_2 + 60\overline{g}_3x_3 + 60\sqrt{x_1^2\sigma_{g_1}^2 + x_2^2\sigma_{g_2}^2 + x_3^2\sigma_{g_3}^2}\,F^{-1}(1-\alpha) \ge D$$

(Tph)

where

$$\overline{g}_j = \frac{\overline{L}_j}{\overline{\tau}_j}, \sigma_{g_j} = \frac{L_j}{\tau_j}\sqrt{\frac{\sigma_{L_j}^2}{L_j^2} + \frac{\sigma_{\tau_j}^2}{\tau_j^2}}$$
$$j = 1,2,3$$

Truck resource constraint:

$$x_j \leq R_j$$

where $x_j \geq 0, for\ j = 1,2,3$

Table 3.7 shows the solution obtained from solving the quadratic deterministic truck optimization problem converted from the stochastic problem with 95% confidence on the ore constraint. The result is a total of 111,052 Tonnes of waste material being moved.

Optimization Result (95% confidence level on the ore production constraint)	
Number of Ore trucks to be allocated	
240T trucks (Fleet Size)	5.498 (18)
320T trucks (Fleet Size)	4.239 (9)
360T trucks (Fleet Size)	3.483 (5)
Maximum amount of truck resource remaining for waste movement (Waste Tonnes)	111,052

Table 3.7 – CCP Optimal Results ($\alpha = 0.95$)

While in practice, a truck solution is often sought to guarantee a high degree of confidence, e.g. $\geq$ 90%, it is sometimes important to know what the results would be for lower ranges of confidence level. The chanced-constrained model is then solved with a number of different confident levels and the corresponding results are shown in Table 3.8. Each truck solution is subject to a large number of simulation runs with the results of the realized ore rates being gathered and presented in Table 3.6. The difference between the expected ore rate and the realized ore rates is graphically illustrated in Figure 3.13.

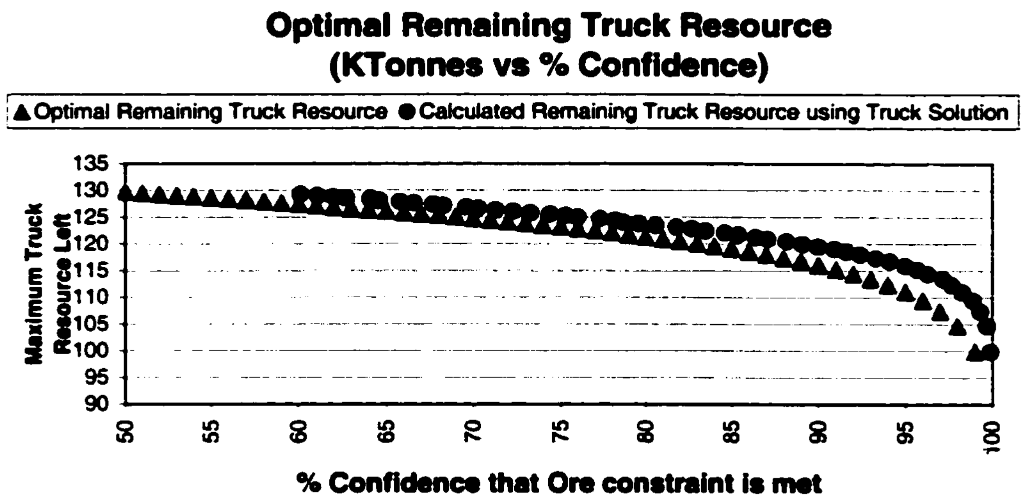


Figure 3.13 – Optimal Truck Remaining Resource vs. % Confidence

Confidence Limit α as input to GAMS	Optimal Truck Solution from GAMS *240T*	*320T*	*360T*	Maximum Truck Resource Remaining (Waste Tonnes)	Simulation Result: Degree of confidence that $OreRate \geq 7000$ (Tph) is satisfied and mean ore rate	
50	11.38	1.03	0.00	129720	51.7	7,354
51	4.32	3.33	2.74	129497	59.4	7,387
52	4.33	3.34	2.74	129,273	60.1	7,400
55	4.38	4.37	3.77	128,591	62.7	7,472
60	4.39	3.39	2.78	128,352	67.7	7,612
65	4.40	3.39	2.79	128,222	72.3	7,751
70	4.41	3.40	2.79	128,120	77.3	7,903
75	4.72	3.64	2.99	123,140	82.2	8,073
80	4.84	3.73	3.07	121,275	86.8	8,278
85	4.98	3.84	3.16	119,073	90.1	8,518
90	5.18	4.00	3.28	115,974	95.3	8,858
95	5.50	4.24	3.48	111,053	98.4	9,397
96	5.60	4.32	3.55	109,429	99.0	9,578
97	5.73	4.42	3.63	107,420	99.4	9,799
98	5.91	4.55	3.74	104,665	99.8	10,101
99	6.21	4.79	3.94	99,913	100	10,623

Table 3.8 – GAMS Allocation Results vs. Independent Calculated Results

The optimal solution found appears conservative. The calculated result suggests that, as compared to the optimal allocation result, fewer trucks can be used to achieve the same degree of confidence while still satisfying the production constraint. The difference may be attributed to the inaccuracy of the deterministic equivalent non-linear constraint (e.g., the approximation of the standard deviation $\sigma_{\left(\frac{Y}{\tau}\right)}$, where Y and τ are Normally distributed). The other reason may be the fact that the combined distribution is not a Normal distribution, especially the distribution of the ratio of the two Normally distributed independent parameters. Yet, the formulation assumes that the overall distribution is Normal with mean and standard deviation that can be approximately derived from the means and standard deviations of the individual uncertain parameters.

3.8. Stochastic Method Conclusions

Table 3.7 summaries the truck solutions found using three different methods: one deterministic solution and two stochastic solutions.

The chance-constrained result appears most conservative and thus corresponds to more truck queuing at either end. The effective ore rate will have to be reduced to avoid the overflow of the surge pile (in the real operation trucks line up in the queue waiting for authorization to dump when the surge pile is full, making overall truck cycle time longer). The chance-constrained method generates solution that guarantees a given degree of confidence that the ore demand will be met while the recourse solution is the expected optimal value given that constraint satisfaction relies on the recourse actions.

The recourse method provides the results, which tend to satisfy the ore production requirement at the mean (7,000 Tph). As previously mentioned, this method relies on the establishment of a good recourse policy, which must be simple to assess and formulate in a mathematical model. Unfortunately, the formulation completed earlier for the single-period truck allocation problem was not realistic and thus not representative of the actual recourse policy. This problem coupled with the heavy requirement for computing resource makes it less suitable for the truck allocation problem.

	Deterministic Formulation	Stochastic Formulation	
		Two-Stage Recourse	Chance Constraint (95%)
Truck Solution: 240T Trucks 320T Trucks 360T Trucks	 *14* *0* *0*	 *2.28* *4.15* *4.0*	 *5.5* *4.24* *3.48*
Maximizing Truck Resource Remaining (Tonnes)	127,831	129,002	111,052
Ore Rate (Tph)	*7,000*	$\mu = 7{,}441$	$\mu = 9{,}397$
EVPI & VSS		EVPI = *8460* Tonnes/shift ~ *29* 320T-truckloads/shift VSS = *840* Tonnes/shift ~ *3* 320T-truckloads/shift	

Table 3.9 – Summary Truck Solutions (Det.,Recourse,CCP)

The deterministic solution is very close to the two-stage recourse solution. Although the EVPI value is large, but since all distributions are considered Normal, e.g., symmetrical about the mean, the expected optimal recourse solution will be similar to the deterministic solution using the mean values for all uncertain parameters. Moreover, the small value of VSS (840 Tonnes/Shift) shows that the stochastic solution is not much of an improvement from the deterministic solution. The EVPI value (8460 Tonnes/Shift) is about 10 times as high as the VSS value, making the truck problem closely resemble the problem case with large EVPI and small VSS (Figure 3.14).

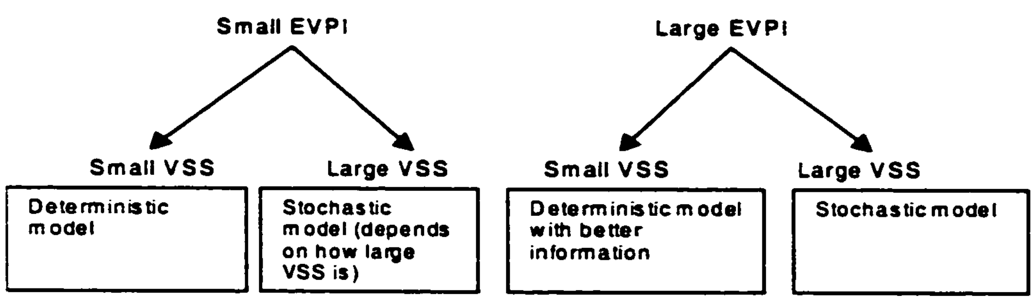


Figure 3.14 – Different Scenarios of EVPI and VSS values

A modest amount of nonnegative VSS value shows that it is still beneficial to implement the truck allocation problem using the stochastic approach. The recourse method is viable only if the recourse action can be accurately modeled; moreover, the method requires an intensive computer resource for its calculation, which is directly dependent on the number of realizations (even at a high number with discrete realization). As a result, the two-stage recourse method was not recommended as the method of choice for the truck allocation problem.

On the other hand, while the chance-constrained method is viable and simple to formulate, it tends to provide results, which are more conservative than they should be, making the truck solution impractical. The conservatism of the truck solution could be attributed to the inaccuracy of the distribution of the uncertain events. Indeed, the Normal distribution will tend to have a long tail at each end of the spectrum, while in reality, the distributions of the uncertain event (truckload and truck cycle time) do not possess a long tail. One way to correct this deficiency is to repeat the allocation process, each time with an improved distribution characteristic. An 'update' process, which will use recent data to determine and provide a more appropriate distribution characteristic to the next allocation run. The body of this work will be presented in Chapter 4.

4. Real-Time Truck Allocation

The key focus of this chapter is to investigate parameter updates (Figure 4.1). Updates to the parameters used in the model such as truck cycle time and truckload are important because they allow the allocation process to adapt to their changes with respect to time. Both models, chance-constrained and deterministic, will be implemented in a real-time system that incorporates the parameter update approach. The comparison of the two results will be made to illustrate the important difference between two models. In Chapter 4, the truck allocation model is enhanced with the addition of the surge pile, which acts as a buffer and helps regulate the ore stream flowing to Extraction.

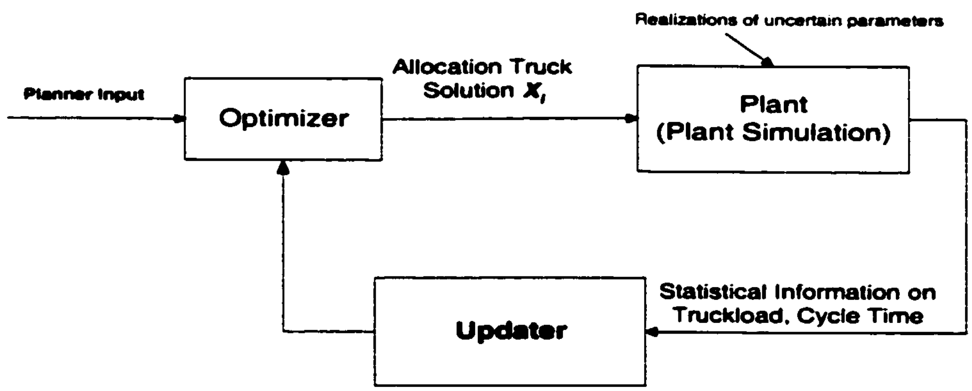


Figure 4.1 – Optimization and Plant Simulation with Updates

In Figure 4.1, the plant input consists of information such as the required ore rate to Extraction, the truck resource limitation, the number of shovels, the number of dumps, etc. These pieces of information are considered as static parameters in the truck model while other pieces of data on truck cycle time and truck payload, provided by the updater, change from one period to the next period.

The main issue in the real-time truck allocation is to investigate the potential benefit of using the parameter update approach in the real-time implementation. The parameter update not only allows the allocation process to adapt to the changes of the parameters, but also helps reduce any crude approximations of the characteristics of the distributions of the uncertain parameters.

In this chapter, trucks are allocated based on route. They are deployed in a simulated[20] hauling operation over a 3-hour period during which statistical data on truckloads and truck cycle times is gathered. This data is then used in the next optimization calculation and the allocation is modified to adapt to any changes. Figure 4.2 shows how successive optimal truck solutions are obtained every 3-hour period.

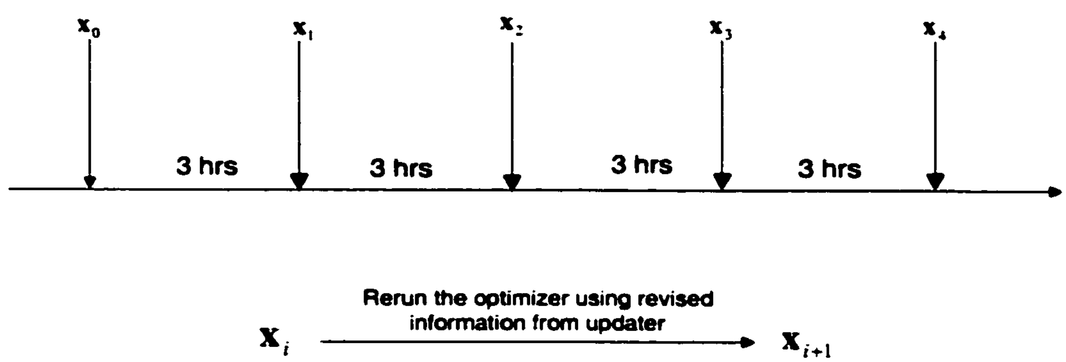


Figure 4.2 – Consecutive Simulation Periods

The 'updater' module relies on a simple rule to update the mean and standard deviation for truckload and cycle time. A simple averaging rule, essentially a simple form of low-pass filter, is used. The primary goal of the

---

[20] Refer to Appendix D for logical details of the simulator

low-pass filter is to eliminate high-frequency components, which correspond to short-term changes or pikes of the parameters such as truck cycle time and truckload. Let $\mathbf{P}_t$ represent the parameter vector used in the optimization process at time t and $\mathbf{P}_r$ represent the parameter vector that is used during the operation between t and $t+1$, the updated parameter vector for the optimization process at $t+1$ is defined as $\mathbf{P}_{t+1} = \lambda\,\mathbf{P}_t + (1-\lambda)\,\mathbf{P}_r$, where $0 \le \lambda \le 1$ (for our case study, $\lambda = 0.5$). The value of λ governs how quickly the system reacts to changes. For $\lambda \succ 0.5$, the system is designed to respond more slowly to the changes while for $\lambda \prec 0.5$, the system reacts more quickly to uncertain events occurring. $\mathbf{P} = [\mu_L \quad \sigma_L \quad \mu_\tau \quad \sigma_\tau]^T$ is the parameter vector that contains the four statistical model parameters: mean truckload, standard deviation of truckload, mean cycle time, standard deviation of cycle time respectively.

A number of simulated scenarios were carried out to study the behavior of the parameter update model. In each scenario, the ore hauling operation was simulated based on the truck allocation, which existed at the beginning of the 3-hour period. Scenario 1 corresponds to the base case, where trucks are driven at regular speeds, and without unexpected events. Scenario 2 involves the simulation with a reduction of the mean truck speed due to worsen driving condition caused by rain, fog, etc. The simulation of Scenario 3 corresponds to an increase in the variance of the truck speed and as well as the average truck speed. An increase in the variance of the truck speed can result from widespread difference of truck operators' responses to the worsening driving condition.

While various truck allocation models were already formulated in previous chapters, a slightly different model is formulated for the purpose of this study. It is believed that the model in this chapter reflects the truck allocation problem more closely. The two key additions are the inclusion of the surge pile and the integer requirement for the number of allocated trucks. Figure 4.3 illustrates of trucks hauling ore from three mine locations to two ore crushers, which is the basis for the truck allocation model in this chapter.

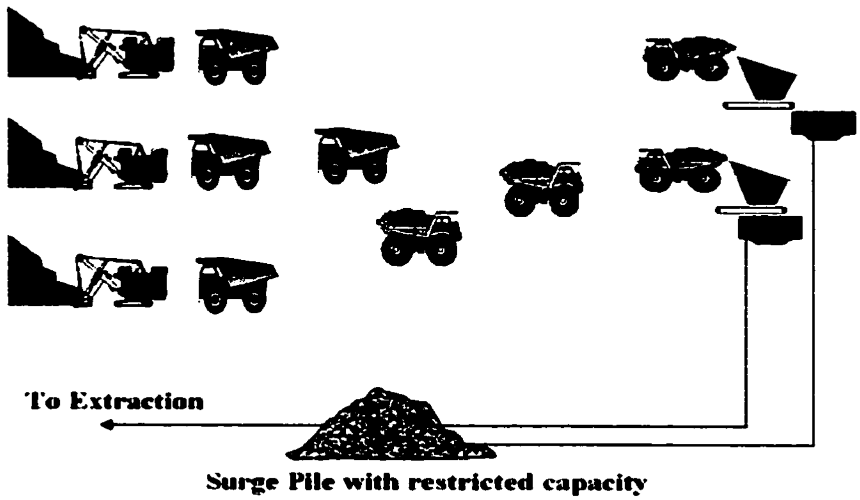


Figure 4.3 – Ore-Hauling Operation in The Mine

4.1. Chance-Constrained Truck Allocation

The truck allocation problem depends on a number of parameters, some of which are not perfectly known or constant, and is stochastic rather than deterministic. Uncertainty and unexpected upsets are part of life in the mining operation and cause difficulty in determining the correct number of trucks. For this reason, extra trucks are allocated at the planning stage and all truck allocation is left to the dispatcher. To get a more accurate, realistic allocation at the planning stage, it becomes necessary to work with the truck allocation problem as a stochastic optimization problem. In this section, a stochastic optimization model is formulated and solved using the chance-constrained method. The critical constraint is to maintain the flow of ore to Extraction. The surge pile is the demarcation between ore delivery and processing and the level is used by the dispatcher to monitor compliance with their production constraint. The probabilistic constraint is developed for the level of the surge pile. The allocation process determines the optimal number of trucks to ensure that the surge pile is higher than a prescribed level with a 95% confidence level. This constraint is probabilistic due to random parameters affecting ore delivery, such as the loading on individual trucks and truck cycle times.

Previously truck models were developed to satisfy the rate of ore delivered by trucks at the crushers, the current truck model is extended to the surge pile. In reality, truck dispatchers do not steward to the ore rate delivered by the haul trucks, instead they try to maintain the volume of the surge pile at an acceptable level (for example, 80% full). There are also automatic controls in place to prevent both overflow and emptying of the surge pile (these controls are also implemented in the simulator in this study).

Moreover, truck allocation can only be implemented using a discrete number of trucks. Therefore, the truck allocation optimization model must be solved using a mixed-integer solver. The stochastic truck allocation problem was solved via the chance-constrained method; however, the equivalent nonlinear mixed-integer model is very difficult to solve directly because the problem is not guaranteed to converge. The approach taken in this study is to work with the problem as a two-step model. In the first step a chance-constrained problem with number of trucks of each type taken as continuous variables was formulated. In Step 2, the solution of the 'relaxed' problem (Step 1) is used as input to solve the mixed-integer linear problem for the ultimate truck solution (discrete number of trucks).

The main motivation for the two-step approach is to guarantee a convergence of the discrete truck solution. Adopting the single-step approach would require one to work with a linear mixed-integer chanced-constrained problem, which is difficult to solve. In the two-step approach, the uncertainty of the parameters is accounted for in the first step, where the problem was relaxed with continuous number of trucks. However, uncertainty is not included in the second step, which corresponds to a mixed-integer linear deterministic problem. The primary goal in the second step is to determine the 'approximate' optimal solution that is closest to the relaxed solution found in the first step of the problem.

The truck allocation model is formulated based on the following assumptions:

- Trucks are allocated to routes, which link ore shovels to crushers.

- Ore shovels cannot be used as waste shovels and vice versa.
- Trucks are not allowed to switch routes within a time segment (3 hours).
- Every effort is made to minimize the truck waiting time at shovels.
- The average values of truck cycle time and of the truckload of the waste trucks are deterministic and are considered constant for a given truck type. These are used to quantify the amount of waste moved by the remaining trucks.
- The average values of truck cycle times and truckload are assumed to be Normally distributed and independent. While it is possible for trucks with heavy loads to be driven at slower speeds, experience shows that this covariance is negligible and can be ignored in the formulation.

4.1.1. Relaxed Truck Model

In the first step, the chance-constrained model includes the number of trucks of each type as the continuous variables (i.e., in this problem the integer variable condition of truck number has been relaxed).

Objective function

As in previous chapters, the objective function is to maximize the waste moved given a limited equipment resource mine layout and production target. This effectively minimizes the usage of truck resource for the ore movement and avoids having to set a hard constraint on the waste as would otherwise be necessary for maximizing profit or revenue. This also accurately reflects the way both the planners and dispatchers view the problem. That is, to meet the production targets as efficiently as possible and use the remaining trucks to remove waste.

Ore production constraints

Since the model has been extended to include the surge pile, the ore production constraint is now imposed on the level of the surge pile. The dispatcher monitors surge height to determine compliance with production targets from Extraction. While the surge pile is small, it still takes approximately 30 minutes to empty a full pile at full production rates (12000 tph) with no feed. It is large enough that it can offset short interruptions and small shortfalls of truck-delivered ore. As a result, the truck allocation solution does not need to be as conservative as previously reported.

Due to the variability in the truckload and the truck cycle time, the amount of ore delivered by trucks to the crushers also vary. The variability in the truckload and truck cycle time can be modeled by a Normal distribution. Historical data has shown that the truckload distribution characteristic is very close to the Normal distribution, and the truck cycle time close to either Weibull or Lognormal characteristic. However, for the purpose of simplicity, both parameters are assumed to vary with a Normal characteristic. Thus the probabilistic ore production constraint is

$$P\{\textit{Surge Pile Volume after H hours} \geq \textit{Minimum Volume}\} \geq \alpha \text{, where } 0 \leq \alpha \leq 1.$$

The value α is the degree of confidence that the constraint on the surge pile is satisfied and the value, $1-\alpha$, is called the tolerable extent to which the probabilistic constraint can be violated. The study in this report is based on the 95% degree of confidence, or $\alpha = 0.95$ (This convenient, high value of confidence is chosen as it is close to the value that corresponds to twice the standard deviation in the Normal distribution. Also, this value is often used in engineering problems).

This constraint can be rewritten as follows

$$P\{\textit{InitSurgeVol} + H[\textit{TruckVol} - \textit{ExtractionRate}] \geq \textit{MinSurgeVol}\} \geq \alpha$$

The initial surge volume (Tonnes), *InitSurgeVol,* is the amount of ore in the surge pile at the beginning of the allocation. Each allocation time period (.e.g., *H* hours) corresponds to a different initial surge volume. The production ore rate to

Extraction is constant throughout the entire implementation. The volume of ore delivered by trucks, *TruckVol*, depends on many parameters (e.g., truckload, truck cycle time, number of trucks) and is the only term in the constraint that embeds the process uncertainty.

Truck resource constraint

A hard limit is imposed on the total number of trucks that can be deployed. Currently, this truck resource limit is fixed.

Shovel constraint

The amount of ore that can be obtained from a shovel is ultimately limited by the shovel capacity, not the number of trucks assigned,

$$\textit{Total Amount of ore loaded on trucks at shovel} \leq \textit{Shovel Capacity} \quad \text{(tph)}$$

The overall mathematical form is

$$\text{Maximize } H\sum_{g}\left[R(g)-\sum_{s,d}X(s,d,g)\right]\frac{60}{\tau_W(g)}L_W(g) \qquad \text{(Tonnes)}$$

subject to:

$$P\left\{InitSurgeVol + H\left[TruckVol - ExtractionRate\right] \geq MinSurgeVol\right\} \geq \alpha$$

$$TruckVol = \sum_{s}\sum_{d}\sum_{g}\frac{60}{\tau(s,d,g)}L_O(s,d,g)X(s,d,g) \qquad \text{(tph)}$$

$$\sum_{d}\sum_{g}\frac{60}{\bar{\tau}_O(s,d,g)}\bar{L}_O(s,d,g)X(s,d,g) \leq ShovelCap(s) \qquad \text{(tph)}$$

$$\sum_{s}\sum_{d}X(s,d,g) \leq R(g)$$

$$X(x,d,g) \geq 0$$

where: g represents the truck groups (e.g., 240T, 320T, 360T), s the shovel index and d the dump index. $L_O(s,d,g)$ is the ore truckload and $L_O(s,d,g)$, $L_W(g)$ is the waste truckload. It is noted that, since the problem is focused only on the delivery of ore, it is not necessary to consider the dependence of waste truckload on the routes. *R(g)* represents the truck resource limit of truck type g and *ShovelCap(s)* denotes the capacity of shovel *s* (tph).

4.1.2. Deterministic Mixed-Integer Model

In current practice, when estimating the number of trucks needed for a specific shovel or route, an initial fractional value is calculated. It is then rounded up to an integer number. These trucks are allocated and deployed on routes, and the rounding process is applied to individual routes. For example, if the continuous solution calls for 5.3 and 6.6 trucks to be deployed on two routes, 6 and 7 trucks will be deployed on these routes, respectively. On simple models with few routes such as given in the example, such heuristic methods can be used to select a satisfactory solution that is close to the optimal solution. However, more complicated models (with many different possible combinations of discrete solutions) requires an integer model to be formulated and solved in order to obtain the optimal integer solution. The following discrete model is added as the second step in the optimization process so that the optimal truck solution can be directly deployed in the ore hauling process. In this second model, the truck solution variables are integer values.

Alternately, a single discrete probabilistic model could be formulated such that the discrete truck solution can be obtained in one step. However this approach yields a nonlinear mixed-integer problem that is not guaranteed to converge. In the two-step approach, the 'optimal' discrete solution is to be searched in the vicinity of the continuous solution. To ensure that this discrete solution is an optimal discrete solution, the discrete linear model must be formulated such that the discrete solution is close to the relaxed continuous solution, which is the true optimal solution to the optimization model (convex programming is assumed).

Objective function

The objective of the model is to minimize the difference in the ore amount delivered by a fractional number of trucks and by a discrete number of trucks. In effect, the optimization converges to the discrete truck solution that is closest to the relaxed continuous truck solution.

Minimize

$$Ore\ Difference = \begin{bmatrix} Ore\ delivered\ by\ discrete \\ number\ of\ trucks \end{bmatrix} - \begin{bmatrix} Ore\ delivered\ by\ continuous \\ number\ of\ trucks \end{bmatrix}$$

$$Ore\ Difference \geq 0$$

The non-negative constraint on the objective function value is imposed to ensure that the discrete truck solution still belongs in the space of the feasible (continuous) solutions.

Shovel constraint

This constraint places an upper limit on the total number of trucks, which is equal to the rounding-up total number of the continuous truck numbers for a given shovel.

On every shovel:

$$\sum_{Routes} \#\ of\ discrete\ trucks\ on\ the\ shovel \leq ceiling\left[\sum_{Routes} \#\ of\ continous\ trucks\ on\ the\ shovel\right]$$

This constraint helps reduce the feasible space of the continuous solutions in which the discrete solution is to be determined.

Truck limitation constraint

This resource constraint is the same as that in the first-stage model. It is included to provide the upper bound on the truck resource.

The overall second-step model is mathematically expressed as

Minimize

$$OreDifference = \sum_s \sum_d \sum_g \frac{60}{\tau(s,d,g)} L_O(s,d,g)[Y(s,d,g) - X(s,d,g)]$$

OreDifference ≥ 0

Subject to

On each shovel s:

$$\sum_d \sum_g \frac{60}{\bar{\tau}_O(s,d,g)} \bar{L}_O(s,d,g) Y(s,d,g) \geq ShovelCap(s)$$

On each shovel s, truck type g: $\sum_d Y(s,d,g) \leq ceil\left[\sum_d X(s,d,g)\right]$

On each truck type g: $\sum_s \sum_d X(s,d,g) \leq R(g)$

$Y(s,d,g) \geq 0$

Refer Appendix G5 for the GAMS program for the overall chance-constrained model.

Also, further information about the solution method that can be used to solve linear integer optimization problems can found in Appendix D.

4.1.3. Sensitivity Analysis

As discussed in chapter 3, the sensitivity analysis involves the study of how sensitive an optimal solution is to model parameters. The parameters on which the sensitivity of the truck solution is to be found are the means and the standard deviations of the truck cycle time and truckload.

The sensitivity analysis is mostly limited to linear programming and its data can be readily obtained together with the optimal solution for linear problem. However, nonlinear problems often pose difficulty in finding such data. In this chapter, the degree of the sensitivity of the solution with respect to various parameter changes is carried out using the perturbation method.

4.2. Deterministic Truck Allocation with the Surge Pile

In an attempt to evaluate the appropriateness of the deterministic truck allocation model in the real-time allocation context, a deterministic truck allocation model is formulated based on the same constraints used in the chance-constrained case. This deterministic model is an enhanced version compared to the allocation model in Chapter 2. The primary difference is the addition of the surge pile. Mathematically, the deterministic model can be represented as

$$\text{Maximize } H\sum_{g}\left[R(g)-\sum_{s,d}Y(s,d,g)\right]\frac{60}{\tau_w(g)}L_w(g)$$

(Tonnes)

Subject to:

$$InitSurgeVol + H\left[TruckVol - ExtractionRate\right] \geq MinSurgeVol$$

(Tonnes)

$$TruckVol = \sum_{s}\sum_{d}\sum_{g}\frac{60}{\tau(s,d,g)}L_O(s,d,g)Y(s,d,g) \qquad \text{(Tph)}$$

$$\sum_{d}\sum_{g}\frac{60}{\bar{\tau}_O(s,d,g)}\bar{L}_O(s,d,g)Y(s,d,g) \leq ShovelCap(s) \qquad \text{(Tph)}$$

$$\sum_{s}\sum_{d}Y(s,d,g) \leq R(g)$$

where $Y(x,d,g)$ is the non-negative integer number of trucks

In comparison to the formulation of the chance-constrained model, two key differences are noted. First, the deterministic model is formulated in a single stage where the solution variables are integer numbers of trucks. Second, the probabilistic constraint on the surge level is now deterministic in the current model. The resulting model is a simple linear deterministic model that can be easily solved with any standard linear solver.

4.3. Results and Discussions

Table 4.1 lists the data used for the model parameters.

Minimum level of surge pile	7000 Tonnes
Ore Rate to Extraction	6000 tph
Degree of confidence to maintain the surge above the minimum level	95%
Number of shovels	3
Number of dumps	2
240T truckload	$\mu = 220, \sigma = 20$ Tonnes
240T truck cycle time	$\mu = 24, \sigma = 5$ minutes
Shovel capacity (Tonnes/h) (same for 3 shovels)	6000 tph
Maximum truck resources	$\{240T, 320T, 360T\} = \{50, 0, 0\}$

Table 4.1 – Model Parameters

4.3.1. Sensitivity Results of the Chance-Constrained Model

Sensitivity data is obtained via the perturbation method. Truck solutions are gathered and accumulated from repeated optimization runs changing one parameter at a time. Since the study is related to the stochastic truck allocation model, the parameters involved in the sensitivity work are all related to the first two moments[21] for the distribution of the two key parameters: truck cycle times and truckload. The charts in Figure 4.4 show the sensitivity result of the truck solution with respect to changes in model parameters. The results are collected from repetitive optimization runs based on the overall model with discrete truck variables.

[21] Mean is the first moment and standard deviation is the second moment

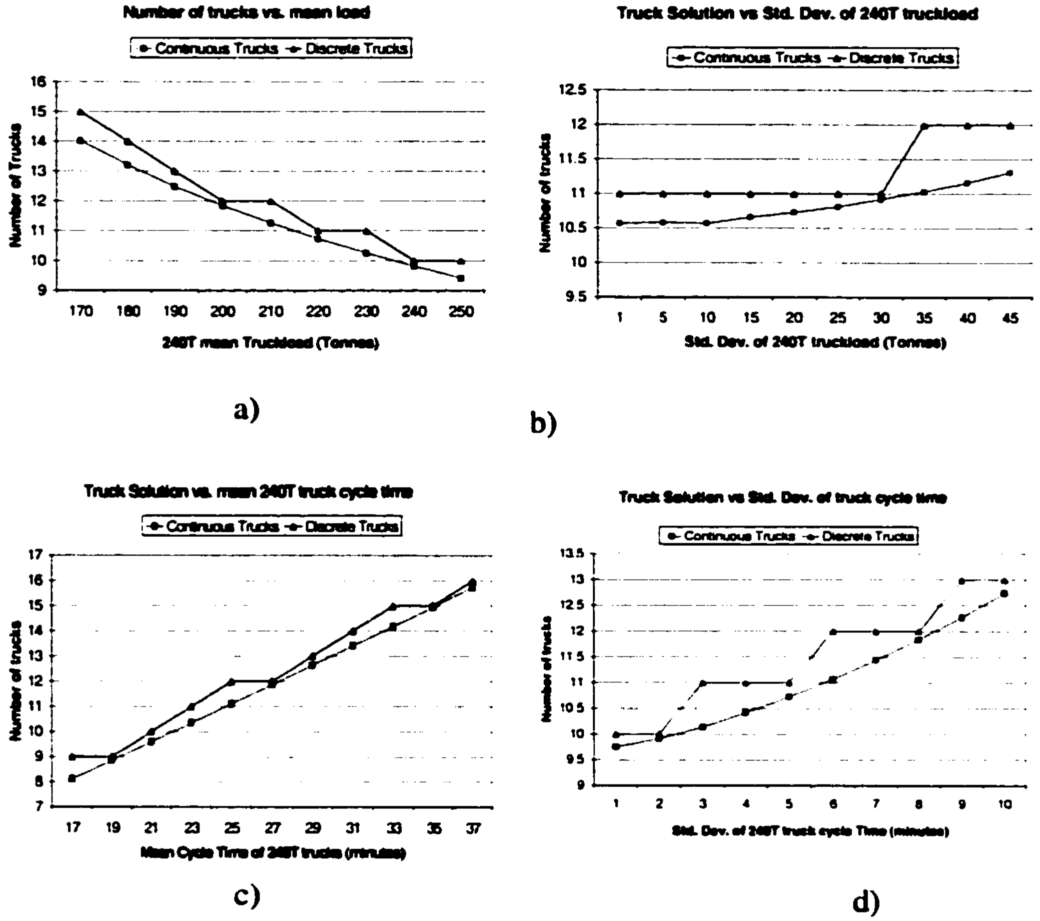


Figure 4.4 – Sensitivity Results (CCP model)

Figure 4.4a shows that as the truckload increases, the optimal number of trucks required decreases. This is an expected result. It also shows that as the average truckload increases, the effect of the discrete solution is more evident. Also as expected, at higher truckloads the change is proportionally smaller and thus may not influence the solution. While the standard deviation of the 240T truckload is small, Figure 4.4b illustrates a high degree of insensitivity of the truck solution to this standard deviation. However, larger changes in the standard deviations than in the mean of truckloads are necessary to affect the result. Figure 4.4c and 4.4d show similar behavior. The discrete solution tracks the continuous truck solution more closely than in 4.4a and 4.4b

The remaining results presented in this section pertain to a mining operation consisting of 3 shovels and 2 dumps. These three shovels are designated to load ore into haul trucks at three pit locations. Despite travelling at different speeds, each truck is considered to travel with two average speeds, one on the haul segment and one on the empty segment. A simulator is used in place of the actual ore hauling process in the mine. The simulator relies on two separate groups of data. Common data such as the distances between ore shovels and dumps, ore rate requirement, loading information, etc. are shown in Table 4.2. Data concerning changes in road conditions is scenario-specific and is thus presented in the appropriate sections.

Parameters	Values	Comments
Upper limit of the surge pile volume (Tonnes)	11.500	When the surge pile reaches this limit, dumping is disallowed. Trucks loaded with ore have to queue and wait at the dump site until dumping is allowed.
Low limit of the surge pile Limit (Tonnes)	4000	When the surge pile is below this limit, the process forces a reallocation. This can happen before the end of the 3-hour period.
Ore Dumps (2)	NB6. NB7	Trucks are allocated to individual routes and are not allowed to switch routes between the period.
Ore Shovels (3)	0078. 0079. 0080	
Loading time for 240T trucks (minutes)	$N(3.0.2^2)$ [22]	The loading time at the shovel is considered to vary with a Normal distribution of a mean of 3 minutes and standard deviation of 0.2 minutes.
Route (0078.NB6)	2 km	
Route (0078.NB7)	2 km	
Route (0079.NB6)	3 km	
Route (0079.NB7)	3 km	
Route (0080.NB6)	3 km	
Route (0080.NB7)	2 km	

Table 4.2 – Layout Data in The Simulator

In the real-time allocation framework, the truck problem is formulated using the chance-constrained approach as well as the deterministic approach. The results from both models will allow planners to recognize the fundamental difference between using the two methods so that they can decide on which is suitable for the problem at hand.

[22] $N(\mu.\sigma^2)$: normal distribution with specified mean and standard deviation

The simulation runs are carried out in two main sets. The first set involves only one type of truck for simplicity. The second set of simulations represents a more realistic allocation problem as it involves all three types of trucks.

4.3.2. Simulation with Single Truck Type

Under a given production constraint, 240T trucks would be required in higher number than the 320T trucks or 360T trucks. Therefore, 240T trucks are most appropriate for the study as compared to trucks with larger capacity since high number of truck units is very useful in building a statistically representative data set to be used in the update procedure.

Scenario 1

This scenario corresponds to a base case where the 240T trucks are driven with speeds that vary according to a specified Normal distribution (Table 4.3). Truckloads are Normally distributed with constant means and standard deviations throughout the scenario.

	240T Hauling speed (km/hour)	240T Empty speed (km/hour)
Period 1	Norm(15,3)	Norm(20,3)
Period 2	Norm(15,3)	Norm(20,3)
Period 3	Norm(15,3)	Norm(20,3)
Period 4	Norm(15,3)	Norm(20,3)
Period 5	Norm(15,3)	Norm(20,3)
Period 6	Norm(15,3)	Norm(20,3)

Table 4.3 – Speed Data in Base Case (Scenario 1)

The simulated result in Figure 4.5 clearly illustrates how the surge pile level changes with time as variation in truck speeds affects the amount delivered. The graph corresponds to the ore hauling operation over twelve consecutive 3-hour periods. With the initial surge level of 8,000 Tonnes, the surge level remains stable between 8000 Tonnes and 10000 Tonnes during the entire period. Despite obvious increases in the sixth and the seventh period, the surge level can still be considered as stable because the graph represents only one instance of the run. The allocation algorithm will always attempt to minimize the number of trucks to haul ore as long as the production constraint is satisfied. When the surge level is higher than the required amount, fewer trucks are required. In the eighth period, one less truck is required to haul ore since the surge is at an all-time high level. This corresponds to the truck-delivered ore rate below the required 6000 tph.

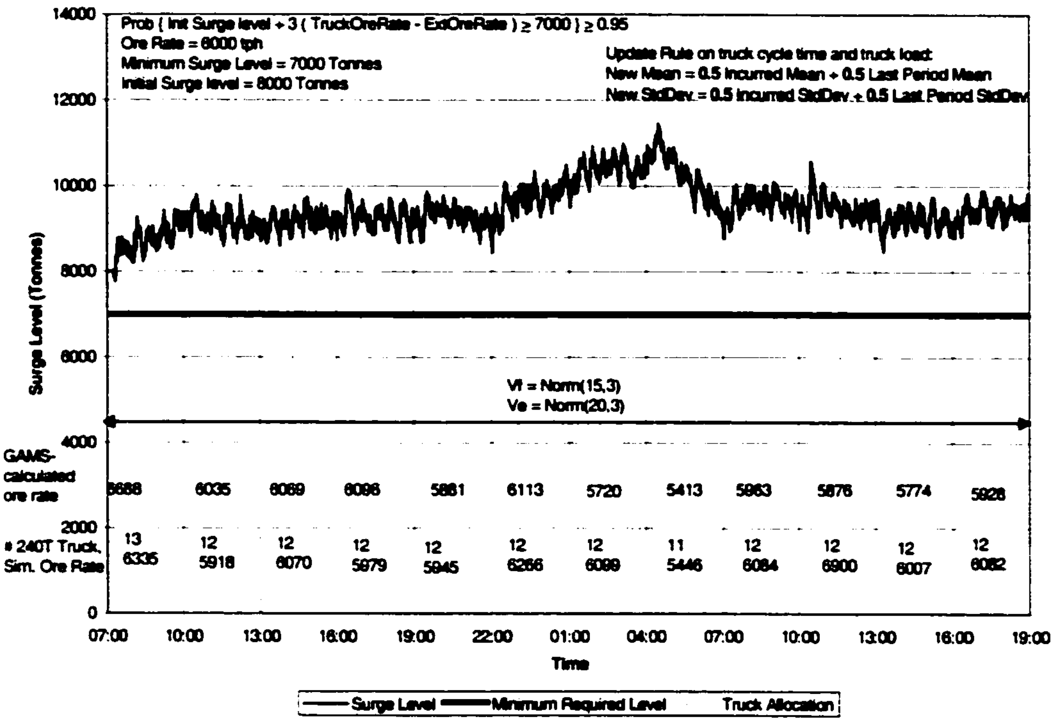


Figure 4.5 – Surge Level vs. Time (Scenario 1, CCP, 240T Trucks)

A short rise of the surge level at the beginning of the run in the first period is due to a short delay of the ore extraction from the surge pile. This artificial delay is introduced to compensate the inaccuracy created by the simulation concerning the initial condition as compared to the real operation. In practice, this behavior does not occur because trucks are not brought to a central location at the end of a shift, instead drivers are brought to the trucks. Further, newly allocated trucks are added to the fleet and trucks are taken away from the fleet gradually, resulting in smooth fluctuations in the surge pile. While the behavior in the study is not realistic, it is a good trade-off for the simple implementation of the simulation. It only affects the very beginning of the simulation.

Figure 4.6 shows the same diagram for the surge level versus time under the deterministic truck model. Since the deterministic model is formulated based on the mean values, given a large enough sample size, the surge level will vary around its corresponding mean level. However, since trucks are allocated as integer numbers, the result will be more conservative as compared to the continuous truck numbers. It means that, strictly speaking, the surge level will vary slightly above the mean surge level. However, this effect is not easily recognized in the chart in Figure 4.6.

Even though the model parameters are updated every 3 hours, the system is still an open-loop model within the 3-hour period and this applies regardless of the method being used in the model formulation. Therefore, it is inevitable that the system will exhibit a certain degree of fluctuation in the surge level. This effect can be magnified if trucks are allocated based on the deterministic model, which is formulated using the historic mean values of the parameters and if there are large swings in the uncertain parameters.

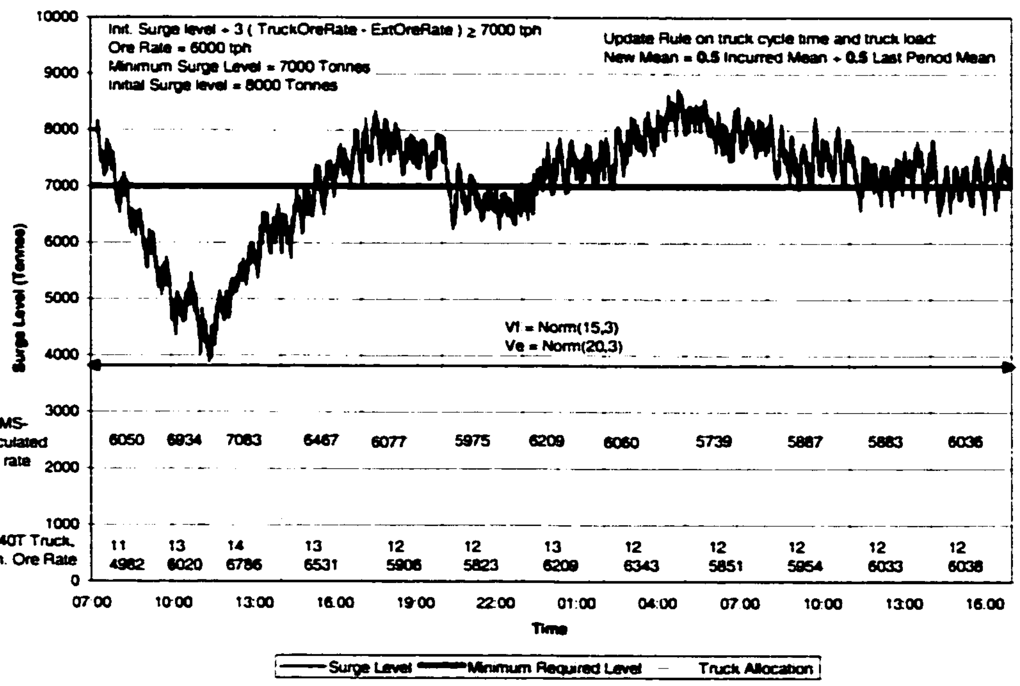


Figure 4.6 – Surge Level vs. Time (Scenario 1, Det., 240T Trucks)

A quick drop of the surge level in the first period in Figure 4.6 is evidence of the lack of trucks being allocated for the task. This is caused by the incompatible data of the model parameters used in the initial truck allocation. This problem is seen throughout the study with varying extents. One way to compensate for it is to start the process with a near-full surge pile such that any shortfall of ore can be offset by the amount of ore in the surge. In this study, another measure being implemented in the simulator is to initiate an early truck allocation when the surge pile is below a minimum threshold (4000 Tonnes as the current parameter).

Scenario 2

In this scenario, a change in the mean cycle time is modeled. Due to the poor driving condition caused by weather conditions such as fog or rain, trucks are driven at lower speeds, causing a drop of the mean cycle time. The simulation scenario involves 7 simulation periods (3-hour each). Table 4.5 shows the parameter data that is used to simulate the changes of truck speeds due to the road condition being considered.

	240T Haul speed V_f (km/hour)	240T Empty speed V_e (km/hour)	Comments
Period 1 to 4	Norm(15,3)	Norm(20,3)	Normal condition.
Period 5 to 12	Norm(10,3)	Norm(15,3)	Driving conditions worsen causing trucks to be driven more slowly. However, the change only happens on the mean value and the speed variance is the same as before.

Table 4.4 – Speed Data in Scenario 2

Figure 4.7 illustrates the behavior of the system when a drop in truck speed is experienced. In the first 4 periods, the operation is similar to Scenario 1 and trucks are allocated to ensure that the surge pile constraint is met 95% of the times. Since the surge level is above the minimum required level, minimum number of trucks are allocated. Starting with Period 5, the surge pile becomes quickly depleted when truck speeds are reduced unexpectedly. At the end of the period the update recognizes that conditions have changed and more accurate information is used in the optimization and the surge pile level recovers over the subsequent periods.

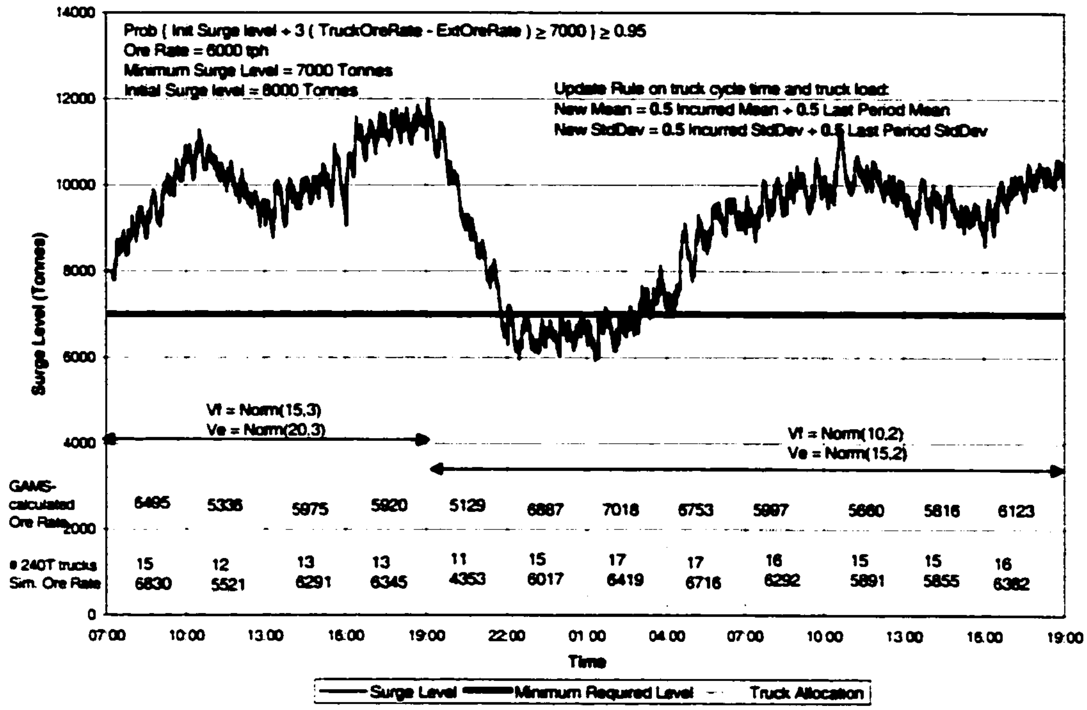


Figure 4.7 – Surge Level vs. Time (Scenario 2, CCP, 240T Trucks)

Although trucks are allocated to ensure that the constraint on the surge pile level is met 95% of the time, when an unexpected change occurs, it is possible that this probabilistic constraint is not satisfied. The main reason for this behavior is due to the difference between the distribution characteristics of the parameters used in the allocation algorithm and those, which are experienced. Even though uncertain information becomes known as time progresses, the slow reaction in the current update rule contributes to this discrepancy. This reasoning can be used to explain what happens to the run instance shown in Figure 4.7, especially in the sixth period. In the fifth period, a sharp reduction of the ore amount in the surge pile is realized due to the effect of the poor road condition. However, the effect of the sharp decrease is not fully accounted for in the next allocation process (for Period 6) because of the effect of the low-pass filter in the current update rule. As a result, not quite enough trucks are allocated to offset the low level of the surge pile, and therefore, the level of the surge pile does not fully recover to the level that was prescribed in the probabilistic constraint.

One of the important issues in choosing an updating algorithm is to decide on the balance between using new information and discarding the old. Putting a strong weight on the newest information allows the fastest response but may cause over compensation if the upset is large and of a short duration. Conversely, if not enough weight is given to current information the process may drift too far from the desired values before the updater takes action. In this study the update is based equally on the past two 3-hour periods. This is a reasonable compromise for the upsets considered here, since they last several time periods.

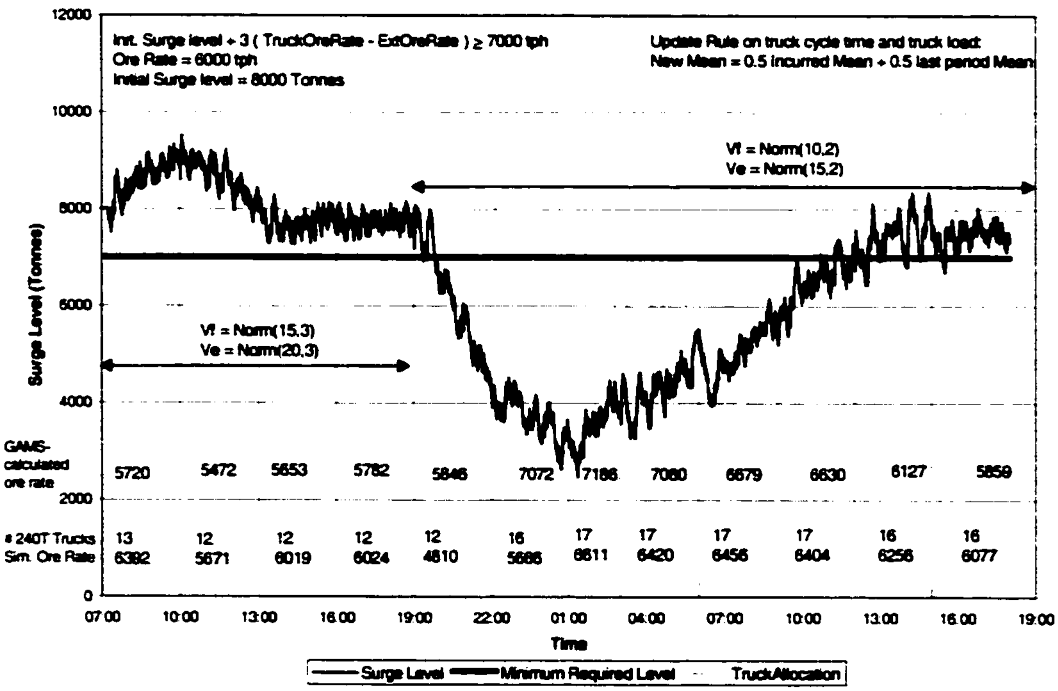


Figure 4.8 – Surge Level vs. Time (Scenario 2, Det., 240T trucks)

Correspondingly, Figure 4.8 shows the same diagram for the deterministic truck model. Once again, the surge quickly depletes in Period 5 where the road condition worsens and similar behavior of the surge pile is shown. However, it is clearly shown that, in the case of deterministic model, it takes longer for the surge level to recover to desired level. This is due to the fundamental difference between the two models. The chance-constrained based truck model is more conservative in the allocation of trucks such that the surge level is always 95% higher than the required minimum level. Thus, it allows for fast recovery after any upset.

Scenario 3

This scenario simulates changes in both the mean and the standard deviation of the truck speed. When the road condition worsens due to weather conditions drivers will not all slow down by the same amount, they will operate trucks differently. Cautious drivers tend to drive more slowly. This results in a larger variation of the truck speeds. In this scenario, the road condition affects both the average value and the dispersion of truck speeds. Table 4.6 summarizes the parameter data that is used to simulate Scenario 3 while Figure 4.9 shows the simulation results.

	240T Haul speed V_f (km/hour)	240T Empty speed V_e (km/hour)	Comments
Period 1 to 4	$N(15,3^2)$	$N(20,3^2)$	Normal driving condition
Period 5 to 12	$N(12,4^2)$	$N(14,4^2)$	Wider variance of speed due to the effect of the weather causing different drivers to operate trucks at different speeds.

Table 4.5 – Speed Data in Scenario 3

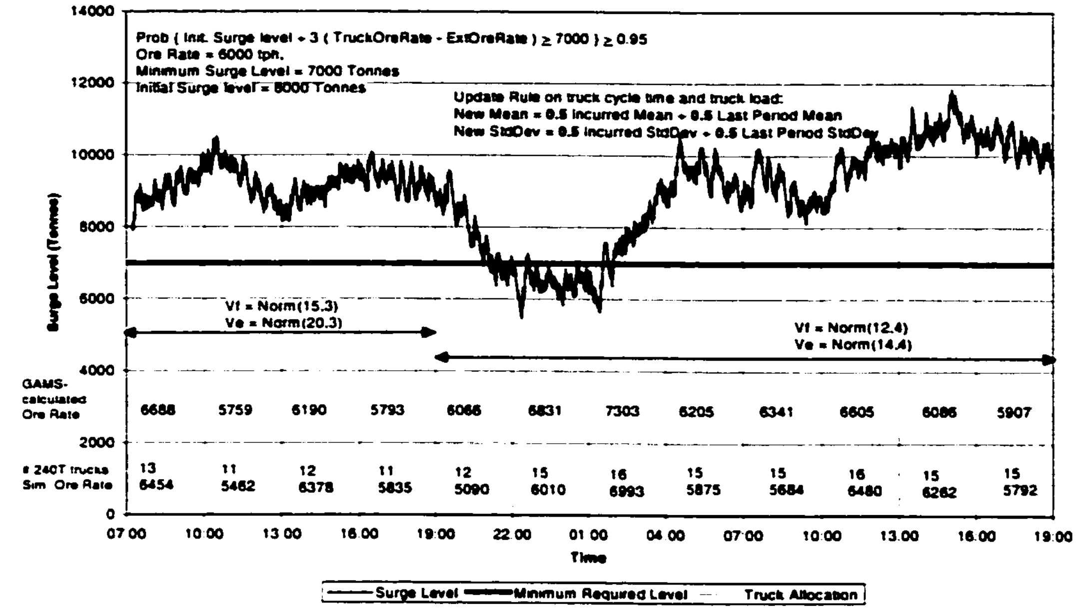


Figure 4.9 – Surge Level vs. Time (Scenario 3, CCP, 240T Trucks)

The effect of the greater variance in the truck cycle time and the slower speeds is felt during the third period. During this time, long cycle times are incurred, resulting in a sharp reduction of the surge pile. The effect of the poor road conditions is taken into account in the next truck allocation (based on the average update rule). Higher number of trucks in Period 4 helps offset the shortfall in the ore supply and the surge pile starts to recover to the desired level (above 7000 Tonnes).

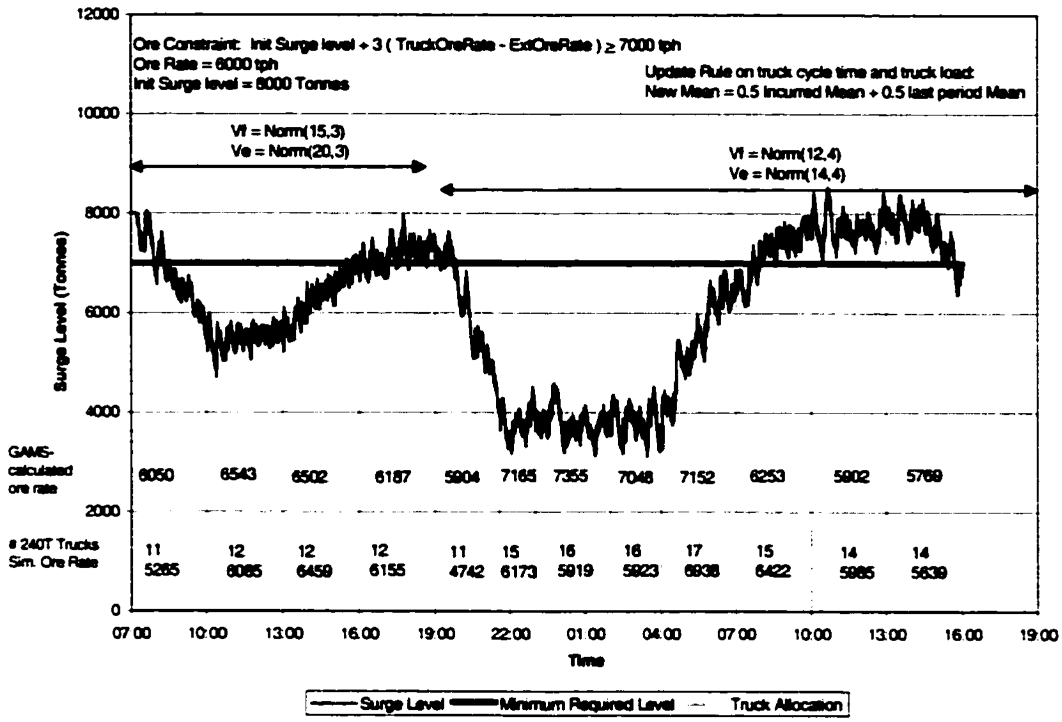


Figure 4.10 – Surge Level vs. Time (Scenario 3, Det., 240T Trucks)

Figure 4.10 shows the result corresponding to the deterministic model. The slow recovery of the surge pile causes the surge level to remain at a very low level, often triggering the truck allocation process before the end of the 3-hour period. In this case, as soon as the surge pile dips below 4000 Tonnes (the value is set in the simulator), the truck allocation process is triggered. But frequent activation of the truck allocation is undesirable since it often results in operational inefficiencies (a requirement of truck resource could be temporary and adding truck resource in such condition can result in truck inefficiency, e.g. increasing waiting time, after the demand disappears).

4.3.3. Simulation with Three Truck Types

Trucks are categorized according to their loading capacity. Three main types of trucks are considered in this thesis: 240T, 320T and 360T. While it is easier for a shovel operator to work with one type of truck consistently during the loading work, it is common to see different types of trucks allocated to haul ore from the same shovel. Therefore, it is practical to include all three types of trucks in the operation. The following simulation is again carried out based on two different truck allocation models: linear chance-constrained model and linear deterministic model.

Results Based on the Chance-Constrained Truck Model

These simulation runs are identical to the first set of the simulation where only 240T trucks were used. The only difference is that different types of trucks are used in the allocation. These trucks are categorized based on their loading capacity. Since trucks are different in size, their loading times are also different correspondingly. Figure 4.11, 4.12, and 4.13 show the simulation results under 3 scenarios as discussed above.

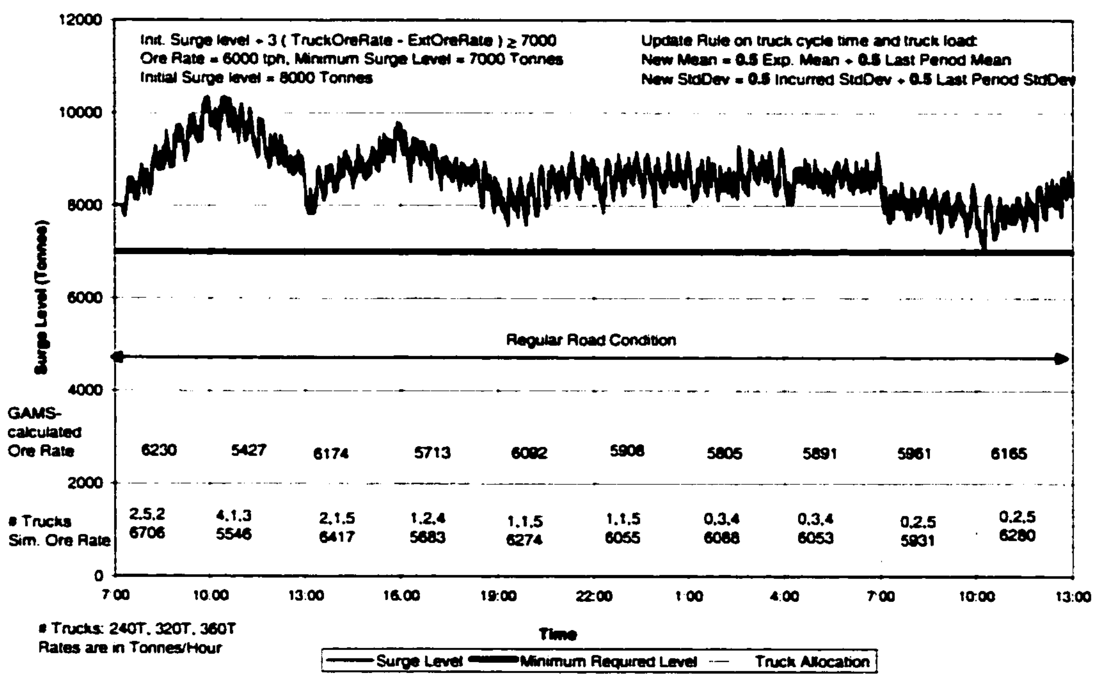


Figure 4.11 – Surge Level vs. Time (Scenario 1, CCP, 3 Truck Types)

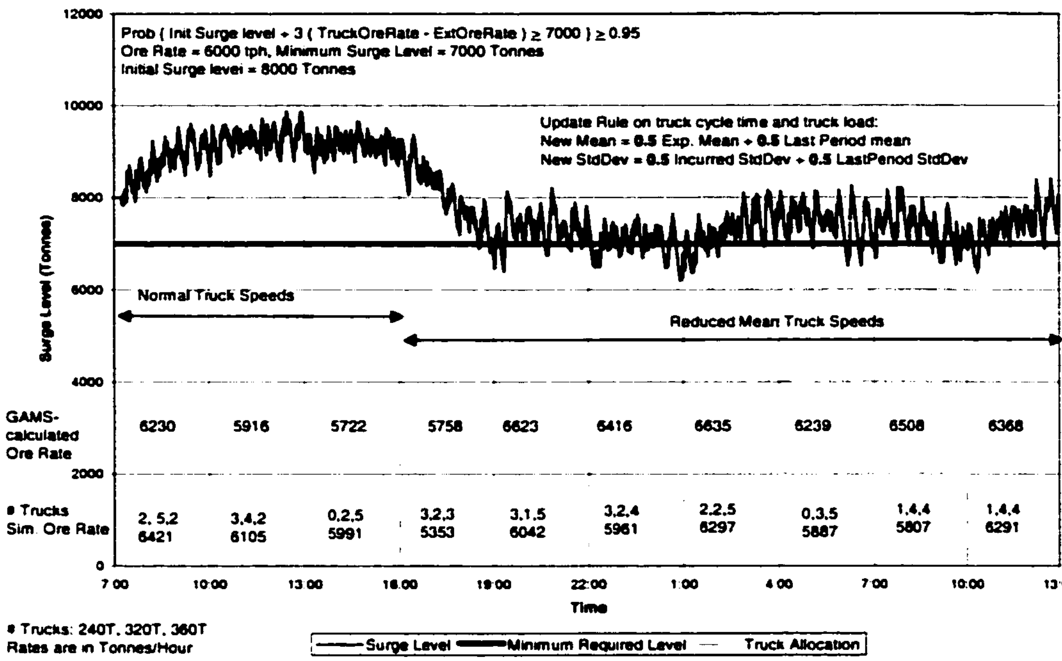


Figure 4.12 – Surge Level vs. Time (Scenario 2, CCP, 3 Truck Types)

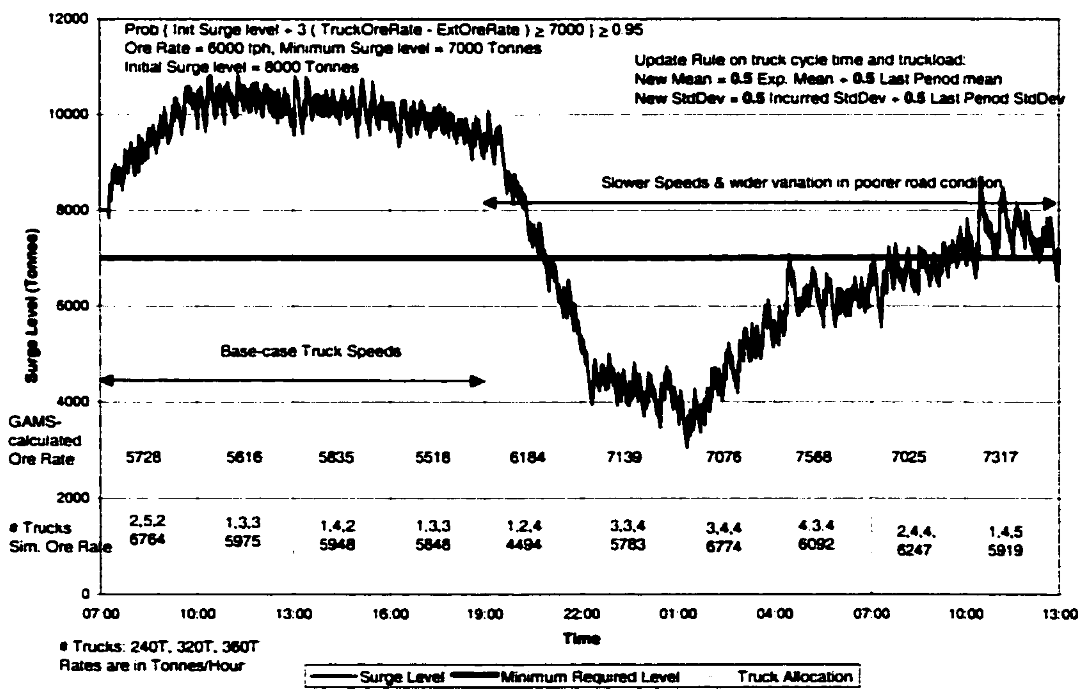


Figure 4.13 – Surge Level vs. Time (Scenario 3, CCP, 3 Truck Types)

Except when there are unpredictable changes in the road condition causing slow truck speeds, the surge is maintained a high, stable level in the case of the chance-constrained model. Since the surge is operating most of the time at high levels, it is capable of absorbing ore shortfalls in the supply stream, making the process suitable for the operation with frequent unpredicted upsets.

Intuitively, the availability of different truck types helps make the truck allocation more flexible, and thus more efficient. However, this benefit of using 3 different truck types versus1 truck type is not quantitatively gathered for the purpose of this comparison. The only concrete conclusion that can be made about this difference is that involving 3 truck types in the calculation increases the level of complexity, causing the final optimal solution longer to converge.

Results Based on the Deterministic Truck Model

Figure 4.14 to 4.16 again show the simulation results corresponding to the deterministic truck model. Figure 4.14 shows a very smooth ore hauling operation with a very stable surge level. The surge level is varying about the specified level of 7000 Tonnes in normal condition. This result shows that, under ideal normal condition, the truck allocation problem based on the deterministic model is equally good as compared to the chance-constrained model. However, when there are unfavorable operating condition as simulated in Scenario 2 and 3, the surge reduces to a low level and is slow to recover to the desired level in subsequent periods (Figure 4.15, 4.16).

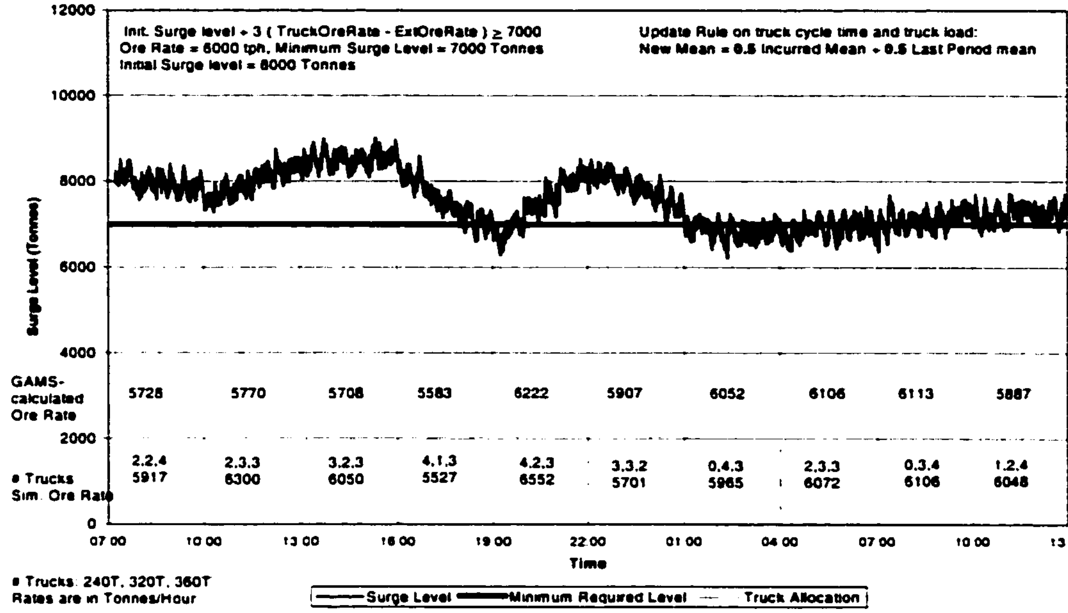


Figure 4.14 – Surge Level vs. Time (Scenario 1, Det., 3 Truck Types)

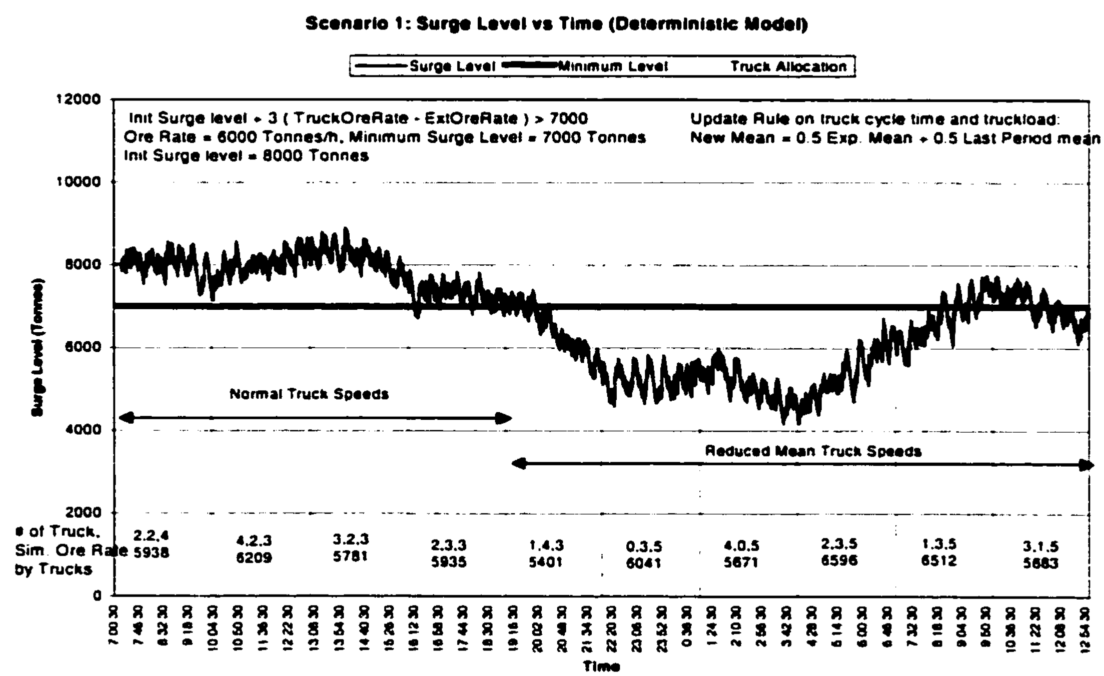


Figure 4.15 – Surge Level vs. Time (Scenario 2, Det., 3 Truck Types)

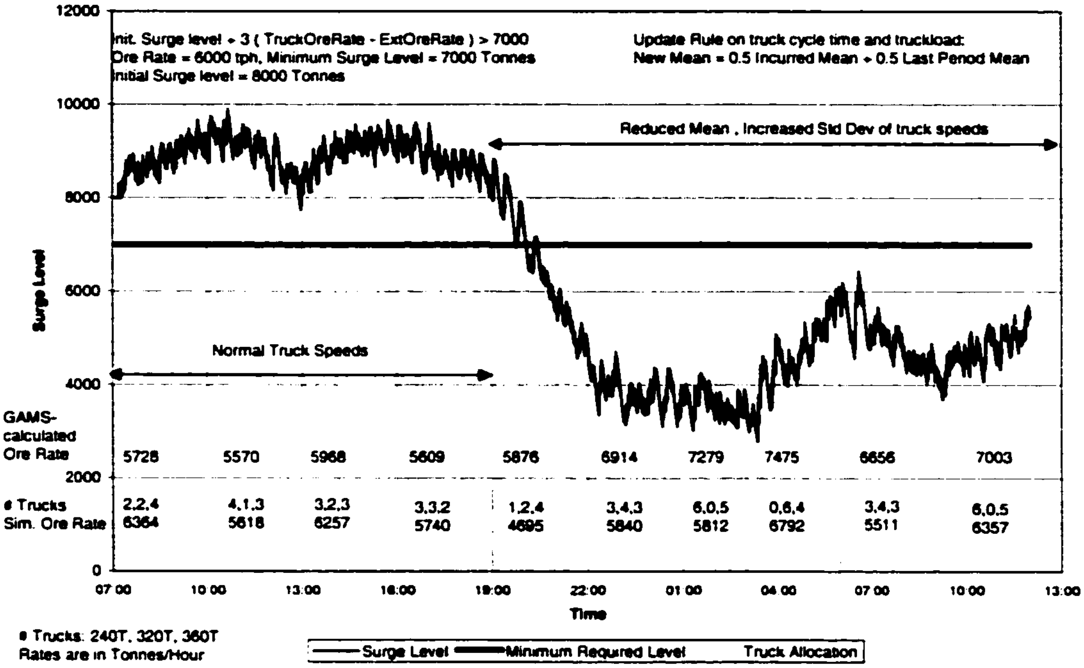


Figure 4.16 – Surge Level vs. Time (Scenario 3, Det., 3 Truck Types)

In general, when truck cycle time increases as driving conditions worsen, the surge pile depletes quickly. The ability to react to these changes greatly depends on the update rule being used. Since the average rule is being implemented in this context, only 50% of the effect of the change will be adapted and reflected in the subsequent reallocation. It is possible to increase the weight of the change in the update rule such that the system reacts more quickly to the change. But if these changes are short-lived, such a rule could cause instability in the level of the surge.

Aside from the fact that the surge level is always at a level higher than the required level more than 95% of the time, the chance-constrained model appears to handle the negative effect of the unfavorable driving condition better than the deterministic model. This fact is easily recognized because, at most times, the chance-constrained based solution is always more conservative than the deterministic counterpart. The surge level is always maintained close to 2 standard deviations higher than the required level. Therefore, the system under the chance-constrained model has a better capability to recover from upsets of ore shortfalls.

The parameter update method is a valuable tool in the process of allocating trucks in the mine. The online approach allows the uncertain process information to be updated and used in the future allocation decision. This implementation allows the allocation process to adjust to the changes of the operation. It can be used as a valuable online tool to aid the operator in making dispatching decisions in the mine.

5. Summary and Conclusions

Successful truck allocation can have a significant impact on the overall performance of the mine. The current allocation method relies largely on a heuristic approach where trucks are allocated based on historical data, as well as the experiences of the planner and dispatcher.

Chapter 2 presented an initial attempt to formally model the truck allocation problem using a linear programming approach. In spite of the variability in truck cycle time and truckload, the model was formulated as a deterministic model using the mean values. The deterministic simplification results in a simple linear model, which can be solved easily for the optimal solution. The benefit of the study in this chapter is not about the actual solution to the problem, but rather about laying a valuable groundwork for subsequent studies in later chapters.

Chapter 3 presented an overview of the two primary methods to stochastic programming: recourse-based and chance-constrained. Recourse-based method is characterized by the fact that violation is associated with penalty. Two main drawbacks of the method are the heavy computing resource required to carry out the calculation and the difficulty in assessing the penalty value associated with the constraint violation. The first drawback is somewhat alleviated by the advancement of computer technology. The second drawback exists on most models where it is difficult to assess the constraint violation. In the truck allocation problem, it is very difficult to quantify the appropriate penalty for the violation of the ore constraint.

Given the variability of the truckload and truck cycle time, it becomes necessary to formulate the truck allocation model using the stochastic programming approach. Motivation for the adoption of the stochastic approach is illustrated in findings of the Expected Value of the Perfect Information (EVPI) and the Value of Stochastic Solution (VSS). Studies have shown that when either of these two values is large, working with stochastic model may provide a benefit. But since calculating these values requires the formulation of the recourse-based model, it can be complicated to determine these values when chance-constrained programming is adopted as the stochastic technique.

The formulation of the chance-constrained model is characterized with the probabilistic constraints. In contrast to the recourse-based mode, chance-constrained model allows a certain degree of the violation in some constraints. This degree of violation is often specified as the confidence level with which the constraint is satisfied. In most industrial applications, where chance-constrained methods are used, this confidence value often lies above 90%. In the truck model in Chapter 3, the probabilistic constraint is to satisfy the required rate of ore 95% of the time. The chance-constrained result appears too conservative to be practical (over-trucked). But this shortcoming is corrected in Chapter 4 with the addition of the surge pile in the model.

Nevertheless, the results found in Chapter 3 point to the adoption of the chance-constrained method as the stochastic method for the truck allocation problem. The recourse-based method was not selected mainly due to fact that the truck allocation problem is appropriate to be broken into distinct stages. The chance-constrained method is simpler to formulate and the probabilistic constraint is easy to comprehend, especially by plant staff. The model is exactly the same as the deterministic model with the exception of the probabilistic constraints. Moreover, the solution method is well-established and simple to use.

The concept of the parameter update model is introduced in Chapter 4. This is an important concept in the implementation, because the overall model allows the truck allocation model to be adapted to changes in operating conditions such as driving conditions on the roads. In this context, the 'updater' model is an important component because it provides the allocation model with information reflecting the real environment. This thesis is limited to a simple update rule and leaves more complex rules to further research.

The combination of the effect of the surge pile inclusion and the parameter update approach has made the deterministic model as viable as the chance-constrained model. This conclusion is based on two key factors. First, since the distribution of the uncertain parameters is symmetrical (i.e. truck cycle time and truckload are Normally distributed), the problem can be formulated as a deterministic model using the mean values of the uncertain parameters. Second, thanks to the parameter update approach, any shortfall of the ore level below a minimum threshold can quickly trigger the reallocation of trucks, which is designed to account for the change in the operating environment that caused the original shortfall. The benefit of the deterministic approach comes from the simplicity of the formulation and a quick conversion to the optimal solution.

However, the chance-constrained method is more suitable when a high degree confidence in the constraint satisfaction is required. Given the same level of the surge pile, the chance-constrained model results in a truck solution that is more conservative than the deterministic model. The deterministic model is formulated using the mean values of the uncertain parameters, which vary according to a symmetrical distribution, its truck solution corresponds to the surge level that varies around the target mean surge level. Equivalently, the constraint on the surge level is violated around 50% of the times. In other words, the chance-constrained method guarantees the lower bound of the surge level while the deterministic method guarantees the mean level of the surge.

The chance-constrained method used in this thesis relies on the assumption that all uncertain parameters are independent. In practice, this assumption may not be true since there is always a certain degree of correlation among the changes of the parameters. An extensive data gathering process is required to fully establish the characteristics of the uncertainty and the correlation among the uncertain parameters. The correlation information is required in the process to convert the probabilistic constraints into their deterministic equivalents. Although some correlation does exist between truckload and truck cycle time in the truck allocation problem, it is felt this assumption does not greatly affect the study, and more importantly does not skew the results of the study extensively.

The size of the surge pile has a direct impact on the truck allocation process as well as the length of the parameter update approach. A large surge pile allows a longer ore shortfall by the truck delivery system, while a small surge makes the process highly sensitive to these shortfalls. Therefore, in the parameter update approach, it is important to select a proper time period such that it is long enough to gather sufficient data, and at the same time it is short enough to allow the system to be able to recover from potentially low surge levels. The discussion of the ideal size of the surge pile is beyond the scope of the thesis because the designed size of the surge depends on many practical factors not considered, such as the mixing requirement, or the minimum size required due to potential breakdowns of crushers, and other equipment.

Another complexity in obtaining the solution arises due to the integer requirement on the truck solution. This requirement does not cause as many problems to the deterministic model as it does to the chance-constrained model. It is commonly found in many industrial applications that a nonlinear discrete model has difficulty converging to an optimal solution. While this convergence problem does not occur on the chance-constrained based truck model, the integer requirement has some effect on the relative accuracy of the integer solution. That is, the problem does not converge to an integer solution within a reasonable number of calculation iterations.

This integer requirement on the truck solution also causes difficulty in obtaining the sensitivity of the truck solution. It is found that the degree of the sensitivity of the truck solution to various model parameters is a non-continuous function with respect to the location of the optimal solution within the feasible space of the solutions. Therefore it is important to quantitatively determine the model parameter to which the solution is most sensitive and to identify the component in the model to be used to offset the negative effect. In the truck allocation problem, in order to negate the effect of the ore shortfall due to the high sensitivity of the truck solution to parameters i.e. the mean of the ore truckload and truck cycle times, the process should start with a full surge and consistently maintain it at a high level.

The main contribution of this thesis is that formal stochastic methods can be used to deal with process uncertainty in the allocation of haul truck resource in mining industry where the truck-and-shovel technology is implemented. Particularly in the oil sand industry where trucks haul ore to the surge pile, the chance-constrained method is more suitable than the recourse-based method.

It was also found that when the distribution of the uncertain parameters is symmetrical and the cost of the constraint violation is not high, it is more appropriate to use the deterministic method because its formulation is simpler. Moreover, the solution to the resulting linear mixed-integer model converges very quickly.

However, in a general truck allocation problem, it is recommended that the chance-constrained method be used in the model formulation so that uncertainty can be handled in a structured manner. The resulting nonlinear model is not overly complicated and small enough that an integer solution can be obtained without extensive computer resource. While the deterministic approach can be considered an equally efficient method in the context of a parameter update approach, especially when the uncertain distribution is symmetric, the chance-constrained approach is still preferable since it does provide an additional level of comfort in meeting the ore production constraint as compared to the deterministic counterpart.

Many opportunities exist to improve the allocation model. One of the possible improvements is to expand the probabilistic production constraint from one constraint into many constraints, one for every hour during the time period. This modification imposes additional constraints in the model, thereby reducing the feasible space of the solutions. The problem is more difficult to solve since the model involves joint-probabilistic constraints. However, since more constraints are added to the model, it is possible that it takes less time for the solution to converge, if a solution exists.

Another critical addition to the truck allocation model is to include the grade requirement as a constraint. While being seen and used as the buffer to help regulate the flow of ore to Extraction, the surge pile also has to fulfil the role of a mixer. Since ore comes from different pits in the mine with different ore grades, it is important that proper mixing be done in the surge pile to guarantee a certain level of grade in the ore flowing to Extraction. This requirement leads to an additional constraint pertaining to the material grade in the truck model. This constraint is best expressed in a probabilistic form since there is also variability in the grade.

While not as critical as the study of the formulation methods and their associated solution methods, the simulator plays an important role to show the feasibility of the methods being proposed. The current simulator can be improved to model more closely the actual ore hauling operation regarding the initial state of the simulating period. Different types of distributions such as triangle, binomial, or uniform can be used to investigate the appropriateness of the optimization model and also to study the sensitivity of the truck model with respect to these distribution characteristics.

Bibliography

[1] Beale E.M.L., *"On minimizing a convex function subject to linear inequalities"*, J. Royal Statistical Society, Series B 17 (1955) p173-184.

[2] Benders, J; *"Partitioning Procedures for Solving Mixed-Variables Programming Problems,"* Numerische Mathematik 4, 238-252 (1962)

[3] Birge J.R, Wets, R.J., *"Computing Bounds for Stochastic Programming Problems by Means of a Generalized Moment Problem"*, Mathematics of Operations Resarch 12, p149-162, 1987

[4] Birge J.R., Louveaux F.,*"Introduction to Stochastic Programming"*, Springer, 1997.

[5] Charnes A., Cooper W.W., *"Chance-Constrained Programming"*, Operations Research 6, 73-79 (1959)

[6] Charnes A., Cooper W.W., *"Deterministic Equivalents For Optimizing And Satisficing Under Chance Constraints"*, Operations Research 11, 18-39 (1963)

[7] Charnes A., Cooper W.W.,*"Cost Horizons and Certainty Equivalents: An Approach to Stochastic Programming of Heating Oil"*, Management Science 4, 1958, p235-263

[8] Coward, J., *"Optimizing Truck Haul Fleets using Linear programming"*, Syncrude Monthly Progress Report, Vol 20(9) 1991.

[9] Dantzig G. B., Infanger G., "Multi-stage stochastic linear programs for portfolio optimization", Department of Operations Research, Stanford University, 1993.

[10] Dantzig G. B., Infanger G., *"Approaches To Stochastic Programming With Application To Electric Power Systems"*, Department of Operations Research, Stanford University, 1994.

[11] Dantzig G.B., Infanger G., *"Multi-stage stochastic linear programs for portfolio optimization"*, Department of Operations Research, Stanford University, 1993.

[12] Dantzig G.B., *"Linear programming under uncertainty"* Management Science 1 (1955) p197-206

[13] Dantzig, G., D. Fulkerson, and S. Johnson; *"Solution of a Large Scale Travelling Salesman Problem,"* Operations Research 2(4), 393-410 (1954)

[14] Eppen, G. D.; Martin, R. K.; Schrage, L.; *"A scenario approach to capacity planning"*, Operations Research, Vol. 37, No. 4, July-August 1989

[15] Fózwiak S.F., *"Optimization of Structures with Random Parameters"*, Engineering Software IV, Procedure of the 4th International Conference, 1985, p. 11-21

[16] Glover, F.: "A New Foundation for a Simplified Primal Integer Programming Algorithm," Operations Research 16(4), 727-748 (1968)

[17] Gomory, R.E. *"An Algorithm for Integer Solutions to Linear Programs,"* in Recent Advances in Mathematical Programming (eds.: Graves and Wolfe), New York, McGrawHill, 1963

[18] Gomory, R.E. *"An Algorithm for the Mixed Integer Problem."* The Rand Corporation, RM-2597, 1960

[19] Gomory, R.E.;*"Faces of an Integer Polyhedron,"* Proceedings of the National Academy of Science 57(1), 16-18 (1967)

[20] Gomory, R.E.;*"On the Relation Between Integer and Noninteger Solutions to Linear Programs,"* Proceedings of the National Academy of Science 53(2), 260-265 (1965)

[21] Hogan A.J. Morris J.G., Thompson H.E. *"Decision problems under risk and chance constrained programming: Dilemmas in the transition"*, Management Science Vol 27, No. 6, June 1981 p698-716

[22] Huang, C. C., Vertinsky, I. and Ziemba, W.T., *"Sharp Bounds on the Value of Perfect Information"*, Operation Research, Vol. 25 (1977), p. 128-139

[23] IBM General Information Manual, *"An Introduction to Modeling Using Mixed Integer Programming,"* Amsterdam, 1972

[24] Infanger, G., *"GAMS/DECIS User's Guide"*, Dr. Gerd Infanger, 1590 Escondido Way, Belmont, CA 94002.

[25] Jagannathan, R.; *"Linear programming with stochastic process as parameters as applied to prodcution planning"*, Annals of Operations Research 30 (1991), pp 107-114.

[26] Kall, P., Ruszczynski A., Frauendorfer K. *"Approximation Techniques in Stochastic Programming"*, Numerical Techniques for Stochastic Optimization", Springer Verlag, Berlin, 1988.

[27] Kataoka, S. *"A Stochastic Programming Model"*, Econometrica, Vol. 31., No.1-2 (1963), pp. 181-196

[28] Kotz S., Normal L. J., *"Encyclopedia of Statistical Sciences"*, Vol 2, John Wiley & Sons Publisher, 1982, p549

[29] Little, J. D.; Murty, K.G.; Sweeney, D.W.; and Karel, C. *"An Algorithm for the Traveling Salesman Problems,"* Operations Research, Vol. 14, 1963, pp. 972-989

[30] Madansky A. *"Inequalities for stochastic linear programming problems"*, Management Science 6 (1960) pp. 197-204

[31] Markowitz, H., and A. Manne; *"On the Solution of Discrete Programming Problems,"* Econometrica 25(1), 84-110 (1957)

[32] Prékopa A., *"Stochastic Programming"*, Kluwer Academic Publishers, 1995

[33] Raiffa H, Schlaifer R., *"Applied statistical Decision Theory"*, Harvard University, Boston, MA, 1961

[34] Rao, S.S, *"Structural Optimization By Chance Constrained Programming Techniques"*, Computers & Structures, Vol. 12, 1980, p777-781

[35] Ravindran, A.; Phillips D. T.; Solberg J.J *"Operations Research: Principles and Practice"*, John Wiley & Sons, 2nd Edition, 1987

[36] Salkin H. M., *"Integer Programming"*, Addison-Wesley, 1975

[37] Symonds G.H.,*"Deterministic Solutions for a Class of Chance-Constrained Programming Problems"*, Operations Research ?, p495-512.

[38] Ta, C.; Kresta, J. *"Syncrude Research Monthly Progress Report, Feb 2001"*

[39] Ta, C.; Kresta, J. *"Syncrude Research Monthly Progress Report, Apr 2001"*

[40] Wets R. J-B, *"Solving stochastic programs with simple recourse"*, WP # 69, Department of Mathematics, University of Kentucky, 1979.

[41] Young, R., *"A Simplified Primal (All-Integer) Integer Programming Algorithm,"* Operations Research 16(4), 750-782 (1971)

Appendix A – Linear programming

A linear programming model involves the optimization of a linear function subject to linear constraints on the variables. Since linear functions arise frequently in economics, networks, scheduling, etc. and are very easy to work with, linear models are often constructed to represent the real world. A linear model has the form

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

$$s.t.\ \mathbf{Ax} \geq \mathbf{b}\text{, where } \mathbf{x} \geq 0, \mathbf{x} \in R^n, \mathbf{A} \in R^{m \times n}, \mathbf{b} \in R^m, \mathbf{c} \in R^n$$

In a deterministic model, the parameters **A, c,** and **b** are known and constant. Such an optimization model can be solved easily via the Simplex method [Dantzig, 1947]. The solution always lies at a corner of the feasible region, which is defined by the linear constraints. Figure A1 illustrates a simple example of a 2-variable optimization linear model with one degree of freedom. The optimal solution can be easily seen at the corner A in the graph.

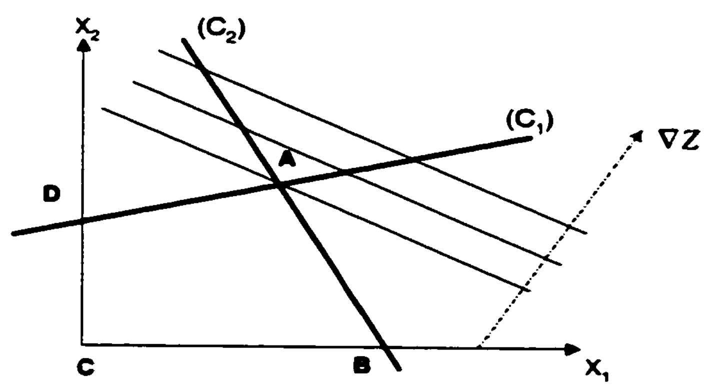


Figure A 1 – Graphical Solution in Linear Model

The feasible region in this case is the area bounded by the four segment AB, BC, CD, and DA (segment AB and AD correspond to constraints C2 and C1 respectively, the other two segments BC and CD correspond to the non-negativity constraints on the variables x_1 and x_2).

The simplex method has been an effective solution method for solving linear programs and enhancements to this method have made it the method of choice for solving linear problems for many years. Examples of these enhancements can be found in the work of Lemke [1954] and Beal [1954] on the Dual Simplex Method where the concept of duality is being appled. Other enhancements include using the Product Form of the inverse by Dantzig and Orchard-Hays [1954], the Column Generation [Eisemann, 1957], and the Decomposition principle [Dantzig & Wolfe, 1960].

Another important method, namely the interior-point method, has been proposed to solve linear problems by Karmarkar [1984]. Karmarmar's new algorithm was later thought to be closely related to a group of algorithms known as barrier methods [Fiacco & McCormick, 1968]. A key feature of this new method is the fact that the iterates are strictly feasible. In contrast to the simplex algorithm where the movement is along the boundary of the feasible region, the points generated in this new approach lie in the interior of the feasible region. Interior-point remains one of the most active research areas in optimization.

While both techniques can be used to solve the linear problems effectively, the interior-point method is more effective with very large linear models (many variables and constraints) and the Simplex method is simple and effective with small to medium models. Another key feature of the Simplex method is the fact that the optimal solution is obtained along with the sensitivity information. Since the model parameters in reality are not always constant, but can fluctuate with time or with different scenarios, it is extremely important to know the extent of the effect of these changes on the optimal solution.

Sensitivity in Linear Programming

In the field of operation research, post-optimality study is equally important as the determination of the optimal solution. The post-optimality study mainly involves the study of the sensitivity of the optimal solution to the model parameters. The decision to implement the optimal solution in practical problems rests on the validity of the solution in light of possible changes that often happen to many model parameters. A small change in a parameter can make the original optimal solution no longer optimal or in the worse case, the optimal solution is no longer feasible. Therefore, it is important to study both the stability and the sensitivity of the optimal solution to the model parameter.

The important goals of a sensitivity study are to identify the *sensitive* parameters (i.e. the parameters whose values cannot be changed without changing the optimal solution) and, for the less sensitive parameters, to determine the range of values over which the optimal solution will remain unchanged. This range of value can also be called the *allowable range* to stay feasible. The worst scenario corresponds to the change that makes the optimization problem infeasible. Knowing the sensitive parameters is important because special care must be taken in estimating or selecting these values to obtain the most accurate optimal solution.

The sensitivity study can also help predict the effect of a small change to the parameter to the objective function of the model. It is beneficial to be able to predict the effect of the change because the problem does not need to be resolved for a new objective. However, this ability is subject to two conditions: a) the model must be linear, and b) the change is considered small enough that it does not cause other non-active constraints to become active. When these conditions are not met, it is required that the problem be resolved for a new optimal solution.

In practice, since model parameters always represent actual physical quantities or real-world events, it is inevitable that they change and ultimately affect the final optimal solution. Therefore, it is important to account for these changes after the optimal solution is found. Figure A2 shows a graphical solution to a general linear optimization problem (optimal solution lies on point ***D***). The feasible region is bounded by the two axes, line ***C1*** and line ***C2*** (the area with 4 corners ***A, B, C, D***). However, lines ***C1*** and ***C2*** are moved to other locations shown as the dotted lines due to the changes of some parameters such that the optimal solution now moves from ***D*** to ***E***. Therefore, it is important to know how sensitive the solution is to each parameter in the model.

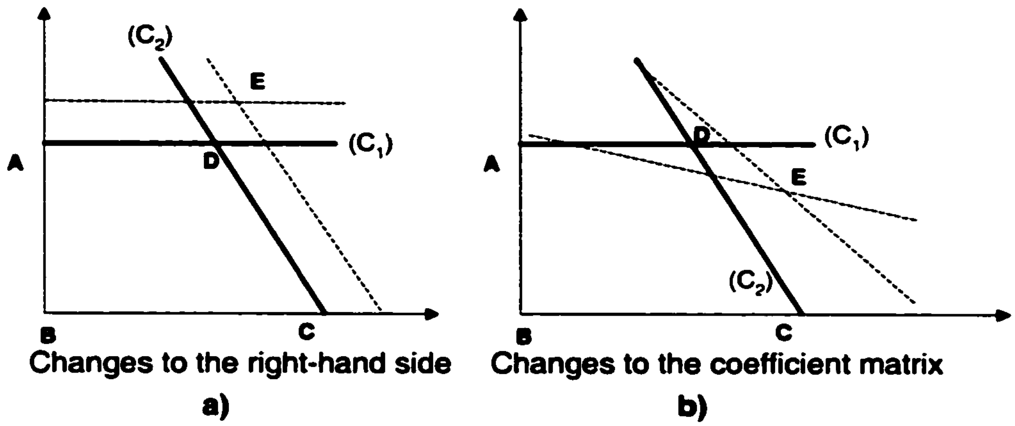


Figure A 2 – Effect of Parameters Changes in Linear Model

Parameters can appear at three different locations: the *right-hand-side* matrix **b**, the *coefficient matrix* **A**, the *reduced cost* **c** in a linear optimization model, such as optimize $z = \mathbf{c}^T\mathbf{x}$ subject to $\mathbf{Ax} \geq \mathbf{b}, \underline{\mathbf{x}} \geq \mathbf{0}$. Figure A2-a corresponds to the changes to the *right-hand-side* parameters in the linear model while Figure A2-b corresponds to the changes that happen in the coefficient matrix **A**. Either of these two cases will result in a different optimal solution. It is necessary to perform the sensitivity study on all parameters. Most solvers can readily provide the optimal solution along with the sensitivity data in the form of marginal values of *the right-hand-side* vector **b**. The sensitivity with respect to the *reduced cost*, or the objective function coefficients, can also be obtained easily by solving the corresponding dual problem. However, changes to the coefficient matrix **A** present the most challenging task in the sensitivity work since it is not possible to obtain the marginal values of the parameters in this coefficient matrix easily from the solver output. The other alternative is to obtain these sensitivity data by numerical method that is to perform the calculation after each step change.

The key benefit of the sensitivity analysis is the ability to predict the new optimal solution without resolving the optimization problem. This benefit is clearly recognized with the large-scale problem that involves thousands of variables and thousands of constraints. However, it is also important to understand the limitation in the interpretation of the sensitivity data. For example, the correctness in the predicted optimal solution in linear models is guaranteed only when the original optimal basis[23] remains optimal after the changes, that is when new variables enter the basis or existing variables leave basis, a full recalculation is typically required.

[23] Being associated with the Simplex method as a linear solver *[Dantzig, 1963]*. *Optimal basis* includes a set of non-zero decision variables, which is said to leave the basis when it becomes zero and it is said to enter the basis when it becomes non-zero.

Appendix B – Stochastic Programming: Two-Stage Recourse

Many researchers have studied recourse problems and have developed valuable formulations for two-stage recourse problems and have recommended various solution techniques. Dantzig [1955] and Beale [1955] among other operation research scientists have built a good theoretical foundation for stochastic models for many industrial applications. Many algorithms were suggested [Huang *et al.*, 1977] to solve for approximate solutions to large-scale problems. The formulation of a generic two-stage recourse problem, developed below and shown in Figure B1, generally involves a large number of realizations. In this problem, the main objective is to determine an optimal first-stage decision **x** which leads to the lowest overall cost. The first-stage cost is directly proportional to the first-stage decision vector (constant cost vector $\mathbf{c}^{\mathrm{T}}$). The second-stage cost is the expectation of the product of the second-stage decision vector **y** and the second-stage cost coefficient vector $\mathbf{q}(\omega)^{\mathrm{T}}$.

$$\begin{aligned}
&\textit{minimize } z = \mathbf{c}^{\mathrm{T}}\mathbf{x} + E_{\xi}\left\{\min \mathbf{q}(\omega)^{\mathrm{T}}\mathbf{y}(\omega)\right\} \\
&\textit{subject to } \mathbf{A}\mathbf{x} = \mathbf{b}, \\
&\qquad \mathbf{T}(\omega)\mathbf{x} + \mathbf{W}\mathbf{y}(\omega) = \mathbf{h}(\omega) \\
&\textit{where } \quad \mathbf{x} \geq 0, \mathbf{y} \geq 0, \sum_{i} P\{\omega_i\} = 1
\end{aligned} \tag{B.1}$$

$E_{\xi}\{a(\omega)\}$ represents the *expectation* of a over the random vector $\boldsymbol{\xi}$ while ω denotes individual realization. Familiar parameters often encountered in the linear model include **x**, $\mathbf{c}^{\mathrm{T}}, \mathbf{A}, \mathbf{b}$. These parameters belong to the group of first-stage parameters whereas the remaining parameters fall under the group of second-stage parameters, all of which are dependent on ω with the exception of W, which denotes the coefficient matrix. These second-stage ω-dependent parameters include $\mathbf{y}(\omega), \mathbf{q}(\omega)^T, \mathbf{T}(\omega), \mathbf{h}(\omega)$. $\mathbf{y}(\omega)$ represents the second-stage decision vector; $\mathbf{q}(\omega)^T$ the cost vector and $\mathbf{T}(\omega), \mathbf{h}(\omega)$ both represent the coefficient matrices.

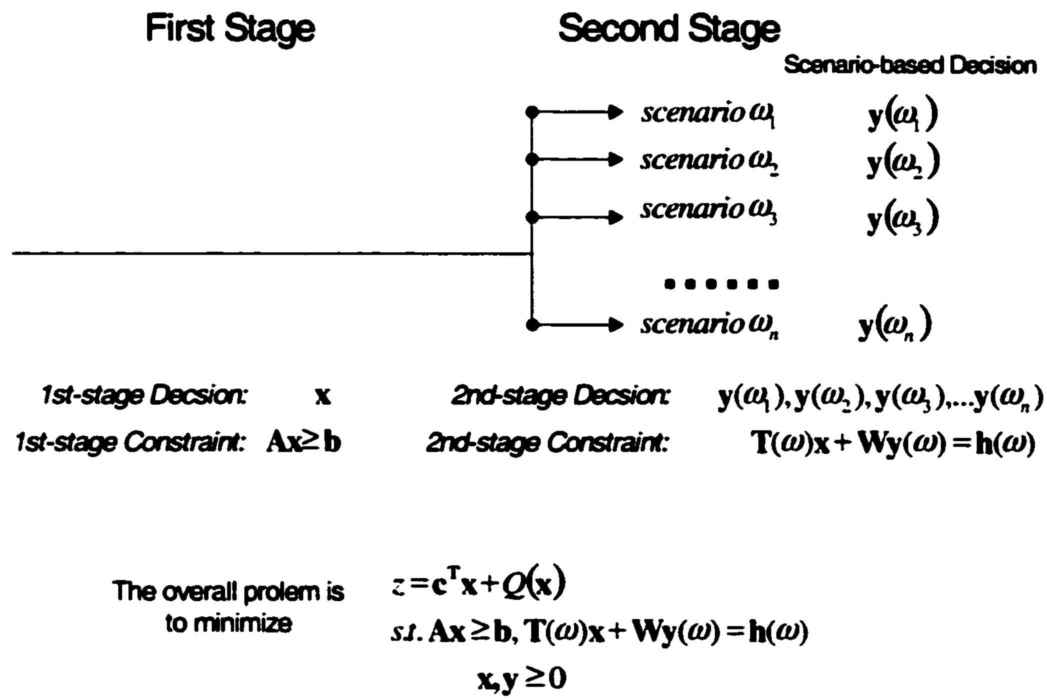


Figure B 1 – Two-Stage Problem with Fixed Recourse

If the second term, $E_\xi\{\min \mathbf{q}(\omega)^T \mathbf{y}(\omega)\}$, can be written or expressed in a deterministic manner. The stochastic problem can be reformulated as a deterministic problem. The second stage decision vector $\mathbf{y}(\omega)$ depends on ω and $\mathbf{x}$ because a unique vector $\mathbf{y}(\omega)$ can be found for a given pair $(\omega, \mathbf{x})$ by solving the following sub-optimization problem

$$\min_{y} Q(\mathbf{x}, \xi(\omega)) = \mathbf{q}(\omega)^T \mathbf{y}(\omega)$$
$$s.t.\ \mathbf{T}(\omega)\mathbf{x} + \mathbf{W}\mathbf{y}(\omega) = \mathbf{h}(\omega),\ \mathbf{y} \geq 0$$

Let $\Psi(\mathbf{x}) = E_\xi\{Q(\mathbf{x}, \xi(\omega))\}$ and substituting $\Psi(\mathbf{x})$ into the general two-stage recourse model yields the following deterministic equivalent problem

$$\begin{aligned}&\textit{minimize } z = \mathbf{c}^{\mathrm{T}}\mathbf{x} + \Psi(\mathbf{x})\\&s.t.\ \mathbf{A}\mathbf{x} = \mathbf{b},\ \ \mathbf{x} \geq 0\end{aligned}$$

The difficulty of the recourse-based program lies in the computation burden of calculating $\Psi(\mathbf{x})$ for all $\mathbf{x}$. If $\Psi(\mathbf{x})$ were convex and differentiable, classical nonlinear programming techniques could be applied to solve for the optimal solution. In the farmer's case, the corresponding second-stage function, $\Psi(\mathbf{x})$, is constructed analytically and its optimal solution can be determined easily.

For most practical problems with small random vectors, one can determine $\Psi(\mathbf{x})$ by numerically integrating $Q(\mathbf{x}, \xi(\omega))$ for a given value of $\mathbf{x}$. In addition, most nonlinear optimization techniques would also require the calculation of the gradient of $\Psi(\mathbf{x})$, which in turn relies on numerical integration. Numerical integration has become an effective computation technique for problems with small number of realizations.

The distribution of the uncertainty plays an important role in the feasibility of the solution to the recourse-based problems. Let K_1 denote the set of solutions that are determined by the fixed constraints and K_2 denote the second-stage feasibility set, i.e. $K_1 = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\}$ and $K_2 = \{\mathbf{x} \mid \Psi(x) \prec \infty\}$, the above deterministic equivalent model can be rewritten as

$$\begin{aligned}&\textit{minimize } z = \mathbf{c}^{\mathrm{T}}\mathbf{x} + \Psi(\mathbf{x})\\&s.t.\ \mathbf{x} \in K_1 \cap K_2\end{aligned}$$

When ξ has finite second moments, Birge and Louveaux [1997] proved that K_2 will be closed and convex and that $\Psi(\mathbf{x})$ will be *Lipschitz* convex, and is finite on K_2. The convexity requirement is essential because it can guarantee a conversion to a global optimal solution. Complete analysis on the properties and characteristics of the function $\Psi(\mathbf{x})$ is beyond the scope of this study.

Intensive computation requirement is the main characteristic of the recourse-based program due to the complexity of determining the equivalent deterministic function, $\Psi(\mathbf{x})$. In deed, the size of the optimization problem (number of constraints and number of unknown second-stage variables) increases when the number of realizations increases. The complexity of the problem combined with the heavy computing demand has been the main obstacle for the adoption of the method in the industry.

Fortunately, in recent years new solution techniques coupled with the advancement in computer technology have reduced the computational burden. When distribution characteristics of the uncertain parameters are known, powerful sampling techniques can be used in conjunction with an efficient calculation algorithm to solve for the stochastic solution with modest computer resources. Powerful stochastic-based commercial software packages have appeared within the past 10 years, not only on time-shared based computers but also on personal computers, and are successfully used to solve many industrial problems.

The concept of multistage recourse problems is a natural expansion of the two-stage recourse problem. The resulting recourse model with three or more stages will be more complex and requires higher computer resources. In some applications, it is better to work on models that involve three or more stages, especially when meaningful corresponding recourse actions can be modeled. However, since adding more stages increases the problem complexity and the degree of difficulty, detailed assessment is required to ensure that the benefit outweighs the added complexity. In the farmer problem, if the farmer can react to certain crop conditions due to the uncertain effect of the weather (e.g. to add water or fertilizer) he can potentially reduce the negative effect of the weather condition to the final yield. Figure B2 illustrates the three-stage recourse model that the farmer would use to formulate his crop allocation problem. The first stage involves the decision on the crop size, the second stage involves possible correction, as needed in the case of the low yield, and the last stage involves the amount of extra crops purchased due to crop shortages and the amount of crops sold. However, the addition of the extra stage in the farmer problem only makes sense if the farmer can quantify the effect of the recourse and whether by taking this recourse action he can maintain the yield at the target level.

It is important to recognize the difference between stage and period of time in recourse-based stochastic problems. It is common to formulate the multi-period stochastic applications using the multi-stage recourse method, while in most cases, the two-stage recourse method is sufficient. The context of the actual problem will control the pertinent relationship between stage and time period. In the farmer case, with a 4-month crop, the outcome of the crop condition after the first two months and the farmer's experience are used to determine the appropriate recourse actions in Stage 2 of the 3-stage problem. It is also possible to map a stage to a 1-month period instead of the 2-month period as long as the stage is long enough for the recourse action to influence the value of the objective function. The length of the time in each stage is clearly problem specific; in this case, the farmer should know when to start the second stage.

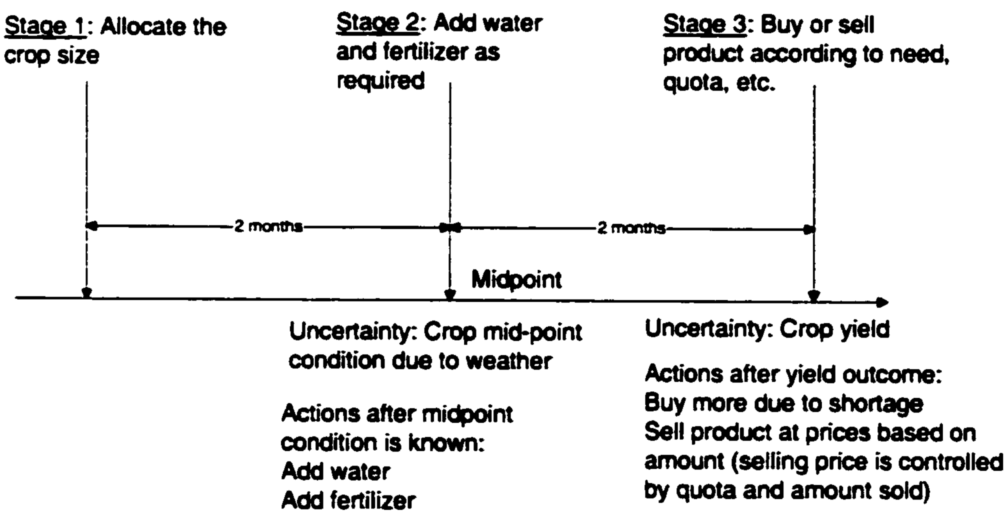


Figure B 2 – Farmer Problem with 3-stage Recourse

Appendix C – Stochastic Programming: Chance-Constrained

In the following section, deterministic equivalent constraints are derived under different cases of the location of the uncertainty. The analysis is focused on the study of linear probabilistic model having the general form as

$$\begin{aligned}&\textit{minimize } h(\mathbf{x})\\&\textit{subject to } P\{\mathbf{A}\mathbf{x} \geq \mathbf{b}\} \geq \boldsymbol{\alpha},\\&\qquad 0 \leq \boldsymbol{\alpha} \leq 1, \mathbf{x} \geq 0\\&\textit{where } \mathbf{A} \in R^{m \times n}, \mathbf{b} \in R^{m}, \mathbf{x} \in R^{n}, \boldsymbol{\alpha} \in R^{m}\end{aligned}$$

The approach to solve such a model relies mainly on the conversion of the probabilistic constraint into a deterministic counterpart.

<u>Uncertainty on the right-hand side:</u> **b**

The task is to convert the probabilistic constraint into the deterministic equivalent. Consider the general probabilistic constraint

$$P[\mathbf{A}\mathbf{x} \geq \mathbf{b}] \geq \boldsymbol{\alpha}, \quad 0 \leq \boldsymbol{\alpha} \leq 1, \mathbf{x} \geq 0$$

where **b** randomly varies according to a known Normal distribution.

At row i, the constraint is written as

$P[\mathbf{A}_i\mathbf{x} \geq b_i] \geq \alpha_i$, where $\mathbf{x} \geq 0$, and $\mathbf{A}_i$ is the row vector that corresponds to row i

or

$$P[b_i \leq \mathbf{A}_i\mathbf{x}] \geq \alpha_i$$

Let $\bar{b}_i, \sigma_i$ represent the mean value and the standard deviation of the random parameter b_i, subtracting both sides of the constraint by $\bar{b}_i$ and dividing by σ_i yields the following probabilistic constraint

$$P\left[\frac{b_i - \bar{b}_i}{\sigma_{b_i}} \le \frac{\mathbf{A}_i\mathbf{x} - \bar{b}_i}{\sigma_{b_i}}\right] \ge \alpha_i$$

Define $z_i = \dfrac{b_i - \bar{b}_i}{\sigma_{b_i}}$ as a new variable, which is a random variable of Normal distribution *N(0,1)*. The probabilistic constraint can then be rewritten as $P\left[z_i \le \dfrac{\mathbf{A}_i\mathbf{x} - \bar{b}_i}{\sigma_{b_i}}\right] \ge \alpha_i$

By definition, the probability $P\left[z_i \le \dfrac{\mathbf{A}_i\mathbf{x} - \bar{b}_i}{\sigma_{b_i}}\right]$ is the area defined by the distribution function *f(.)* and the interval of z_i where $z_i \in \left[0, \dfrac{\mathbf{A}_i\mathbf{x} - \bar{b}_i}{\sigma_{b_i}}\right]$ (Figure C1).

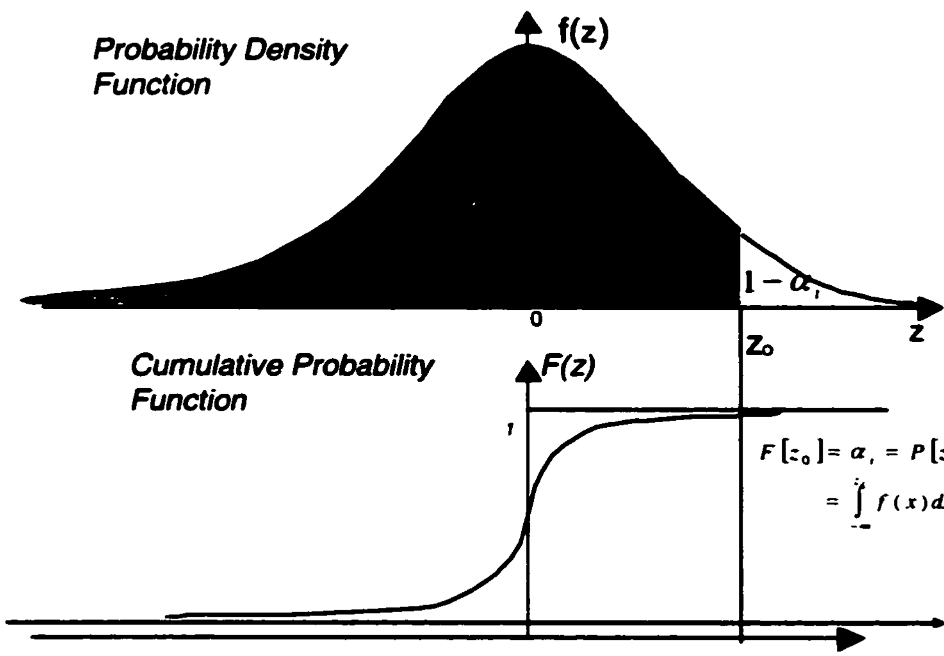


Figure C 1 – Probability Density Function and Cumulative Probability Function

Let $F(z)$ be the cumulative probability function

$F(z) = \int_{-\infty}^{z} f(x)dx \le 1$, where $f(x)$ is the corresponding probability density function

It can be written that

$$P\left[z_i \le \frac{\mathbf{A}_i\mathbf{x} - \bar{b}_i}{\sigma_{b_i}}\right] = F\left[\frac{\mathbf{A}_i\mathbf{x} - \bar{b}_i}{\sigma_{b_i}}\right] \ge \alpha_i$$

Since the cumulative distribution function *F(z)* is *monotonically* increasing with z, the probabilistic constraint can be converted to a deterministic equivalent as

$$\frac{\mathbf{A}_i\mathbf{x} - \bar{b}_i}{\sigma_{b_i}} \ge F^{-1}[\alpha_i]$$

or $\quad \mathbf{A}_i\mathbf{x} - \sigma_{b_i} F^{-1}[\alpha_i] \ge \bar{b}_i$

This deterministic inequality at row *i* can be generalized into a matrix form:

$$\mathbf{Ax} - \boldsymbol{\sigma}_\mathbf{b}\mathbf{F}^{-1}[\boldsymbol{\alpha}] \ge \bar{\mathbf{b}}$$

where $\quad \boldsymbol{\alpha}_\mathbf{b} = \begin{bmatrix} \alpha_{b_1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \alpha_{b_n} \end{bmatrix}, \mathbf{F}^{-1}[\boldsymbol{\alpha}] = \begin{bmatrix} F^{-1}(\alpha_{b_1}) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & F^{-1}(\alpha_{b_n}) \end{bmatrix}, \bar{\mathbf{b}} = \begin{bmatrix} \bar{b}_1 \\ \vdots \\ \bar{b}_n \end{bmatrix}$

In principle, any cumulative probability distribution function can be inverted. When the distribution function is available in an analytical format, it is simple to obtain the corresponding inverted function. But without this analytical function, one would have to rely on computing a table of the inverse function.

The decision variables in **x**, now depend on the distribution characteristic of the uncertain parameters. The original probabilistic constraint $P[\mathbf{Ax} \ge \mathbf{b}] \ge \boldsymbol{\alpha}$ is equivalent to the deterministic constraint $\mathbf{Ax} \ge \bar{\mathbf{b}} + \boldsymbol{\sigma}_\mathbf{b}\mathbf{F}^{-1}[\boldsymbol{\alpha}]$ in which the term in the right-hand side represents an additional demand term that increases the value of the decision variables in **x**, as compared to the constraint $\mathbf{Ax} \ge \bar{\mathbf{b}}$. Large values of $\boldsymbol{\alpha}$ (high degree of confidence) or of $\boldsymbol{\sigma}_\mathbf{b}$ (flat distributions) will require large values for the decision variable (**x**) to satisfy the deterministic equivalent constraint. This implies that when either high confidence is required or there is a substantial lack of good information, one must choose a solution that is often costly and conservative.

Problems with uncertainty in the right-hand side are very common and remain the simplest to solve among all chance-constrained problems. The deterministic equivalent constraint is again a linear constraint, which makes the equivalent deterministic optimization problem very easy to solve.

<u>Uncertainty on the left-hand side</u>: **A**

The uncertainty in the left-hand-side parameters of stochastic linear models is more complicated. Not only is the derivation more involved, the resulting deterministic equivalent is subject to more assumptions and approximations.

Consider the same probabilistic constraint $P[\mathbf{Ax} \geq \mathbf{b}] \geq \boldsymbol{\alpha}$

where $\mathbf{x} = [x_1 \quad x_2 \quad \cdots \quad x_n]^T$, $\mathbf{x} \geq \mathbf{0}$

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix},$$

each(a_{ij})can vary according to distribution Normal($\bar{a}_{ij}, \sigma_{ij}$)

$\mathbf{b}, \boldsymbol{\alpha}$ *are deterministic column vectors with size m*

The probabilistic constraint can also be written as

$$P[\mathbf{Ax} \leq \mathbf{b}] \leq \mathbf{1} - \boldsymbol{\alpha}$$

Let $\mathbf{Y} = \mathbf{Ax}$, where $y_i = \mathbf{A}_i\mathbf{x} = [a_{i1} \quad a_{i2} \quad \cdots \quad a_{in}]\mathbf{x}$ is a linear function of $\mathbf{x}$ and rewriting the constraint yields

$$P[\mathbf{Y} \leq \mathbf{b}] \leq \mathbf{1} - \boldsymbol{\alpha}$$

or at row i,

$$P[y_i \leq b_i] \leq 1 - \alpha_i$$

The following development is based on row i. Since $(a_{ij}, j = 1,2,...,m)$ varies according to a Normal distribution, y_i will also vary with a Normal distribution with mean $\bar{y}_i$ and standard deviation σ_{y_i}

$$\bar{y}_i = \sum_{j=1}^{n} \bar{a}_{ij} x_j$$

$$\sigma_{y_i} = \sqrt{\sum_{j=1}^{n} (\bar{\sigma}_{ij} x_j)^2}$$

Subtracting $\bar{y}_i$ from both sides of the inequality and dividing the result by σ_{y_i} yields following probabilistic constraint

$$P\left[\frac{y_i - \bar{y}_i}{\sigma_{y_i}} \le \frac{b_i - \bar{y}_i}{\sigma_{y_i}}\right] \le 1 - \alpha_i$$

or $P\left[z \le \frac{b_i - \bar{y}_i}{\sigma_{y_i}}\right] \le 1 - \alpha_i$, where z is the uncertain variate with mean *0* and standard deviation *1*

$$P\left[z \le \frac{b_i - \bar{y}_i}{\sigma_{y_i}}\right] = F\left[\frac{b_i - \bar{y}_i}{\sigma_{y_i}}\right] \le 1 - \alpha_i$$

Similar derivation leads to the following equality constraint:

$$\frac{b_i - \bar{y}_i}{\sigma_{y_i}} \le F^{-1}(1 - \alpha_i)$$

or

$$\bar{y}_i + \sigma_{y_i} F^{-1}(1 - \alpha_i) \ge b_i$$

At row i, the constraint $\overline{\mathbf{A}_i}\mathbf{x} \ge b_i$ corresponds to the base deterministic case with $\overline{\mathbf{A}}_i$ as the mean coefficient vector. The second term on the left-hand side, $\sigma_y F^{-1}(1-\alpha)$, accounts for the effect of uncertainty. As developed in the previous case with the uncertain parameter, b_i, this term will cause the solution variables $\mathbf{x}$ to increase ($\sigma_y F^{-1}(1-\alpha) \le 0$). Similarly, when either a high degree of confidence is required or the extent of the uncertainty is significant, the solution will be very conservative and undoubtedly costly.

Uncertainty on both sides of the constraint: **A, b**

In the general case, the uncertainty occurs in both sides of the constraints. Consider the same probabilistic constraint $P\{\mathbf{Ax} \geq \mathbf{b}\} \geq \boldsymbol{\alpha}$ where both **A** and **b** contain parameters that vary with Normal distributions.

The probabilistic constraint can also be written as $P\{\mathbf{Ax} - \mathbf{b} \geq 0\} \geq \boldsymbol{\alpha}$

Let $\mathbf{A'} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & -b_1 \\ a_{21} & a_{22} & & a_{2n} & -b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & -b_m \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ 1 \end{bmatrix}$

The probabilistic constraint becomes $P\{\mathbf{A'y} \geq 0\} \geq \boldsymbol{\alpha}$, where $\mathbf{y} \geq 0$, $y_{n+1} = 1$ ($\mathbf{y} \in \mathbf{R}^{n+1}$), and $\mathbf{A'}$ is the matrix with uncertain coefficients ($\mathbf{A'} \in \mathbf{R}^{m(n+1)}$). The derivation of the deterministic equivalent for this case is omitted as it has been shown earlier.

Since many algorithms and software tools are now available to solve nonlinear deterministic optimization problems, the later step of solving the deterministic model becomes routine and is not as important as the initial step, which involves the determination of the deterministic equivalent constraint. The correctness of the final solution depends on the degree of accuracy with which the deterministic equivalent constraint is derived. For example, consider the probabilistic constraint, $P\{a_1x_1 + a_2x_2 \geq b\} \geq \alpha$ where a_1 and a_2 vary according to some distribution (b, α are constant), if a_1 and a_2 are Normally distributed with known characteristics the dependent function y ($y = a_1x_1 + a_2x_2$) will also be Normally distributed with characteristics that can be easily derived from the characteristics of a_1 and a_2. However, when a_1 and a_2 do not vary according to the same distribution characteristics or they are similar but not necessarily Normal, it will become difficult to determine the distribution of y, making it hard to determine the deterministic equivalent constraint. The derivation becomes more complicated if the random events are not independent.

Appendix D – Mixed-Integer Programming

Many approaches are available to solve mixed-integer programming problems. They include (a) cutting plane techniques, (b) enumerative methods, (c) partitioning algorithms, and (d) group theoretic approaches. Only the first two methods are discussed in this report. Details on partitioning techniques can be found in Benders [1962] and information on the group theoretic algorithms is available in Gomory [1965, 1967].

The main idea behind the cutting plane technique is to deduce supplementary inequalities from the integrality and constraint requirements, which ultimately produces a linear program that can then be solved for the integer values. This constraint generation idea was proposed by Dantzig *et al.* [1954] in their work to solve the travelling salesman problem and then by Markowitz and Manne [1957]. The first cutting plane algorithm was developed in 1958 [Gomory, 1963] to solve integer programs, and then generalized by Gomory [1960] for the mixed-integer case. Contributions to the cutting plane algorithm also came from Glover [1968] and Young [1971] on integer programs that are primal feasible.

Cutting plane technique is very computing efficient as much computing time is saved by adding the new constraint in every program iteration (with a reduced feasible region). The Gomory cut algorithm may not converge, but when it does, the algorithm converges reasonably quickly.

The branch-and-bound technique basically applies an efficient enumeration technique to search for an integer solution in the space of the feasible solutions. The branch-and-bound algorithm was developed by Little *et al.* [1963] to solve the traveling salesman problem. Since then, it has been used widely to solve both pure and mixed-integer programming problems and has been implemented in many commercial software systems. Further details of the algorithm can be found in many textbooks such as those by Salkin [1975], or by Ravindran *et al.* [1987].

It was found that the solution depends greatly on the way the integer problem is formulated. In general, the number of integer variables should be kept as small as possible because problems with more integer variables take longer to solve. In contrast, the addition of new constraints reduces the time required to solve the problem since the feasible solution space is reduced when more constraints are added. In addition, good (tight) lower and upper bounds on the integer variables speeds up the solution process. Further information about guidelines for problem formulation using the branch-and-bound technique can be found in the IBM reference manual [1972].

Between the two methods – cutting plane technique and the branch-and-bound technique – that are available to solve integer or mixed-integer programming problems, the branch-and-bound technique is newer, and is used more widely among practitioners. In this study, the linear discrete truck model is solved using GAMS/BDMLP, which is an LP, MIP[24] solver that is included in the GAMS software system. It is developed based on the branch-and-bound algorithm and can be used to solve reasonably large linear models, which are non-degenerate[25] and well-scaled[26].

---

[24] LP: Linear programming; MIP: Mixed-integer linear programming

[25] When one or more of the basic variables in a basic feasible solution is zero, this solution is said to be located at a ***degenerate*** vertex, the linear program is said to be degenerate. Degeneracy arises when a linear program contains a redundant constraint.

[26] Poorly scaled models can occur when many constraints are measured in inconsistent units leading to poor calculation performance.

Appendix E – Discrete-Time Truck Simulator

The simulation does not include the process of the crusher, which is a major component in the mining loop. As ore must be crushed into smaller sizes before it is sent to the surge pile, any breakdown in the crusher can potentially disrupt the flow of the ore. Although this is a common problem often encountered during the operation, it is omitted to reduce the complexity of the problem.

As this simulator was developed around the operation of the trucks, it is necessary to define possible truck states and the order of the state change. The diagram in Figure E1 illustrates how an ore truck operates with regards to these states. States missing from the diagram, though existing in practice, include the parking state, the breakdown state and lube state. The parking state corresponds to the condition when the operator is on lunch or coffee break. The breakdown state occurs due to an actual mechanical breakdown of the truck and the lube state corresponds to the condition where truck is required to leave for refueling. Table E1 outlines the condition logic for the state changes in a truck cycle.

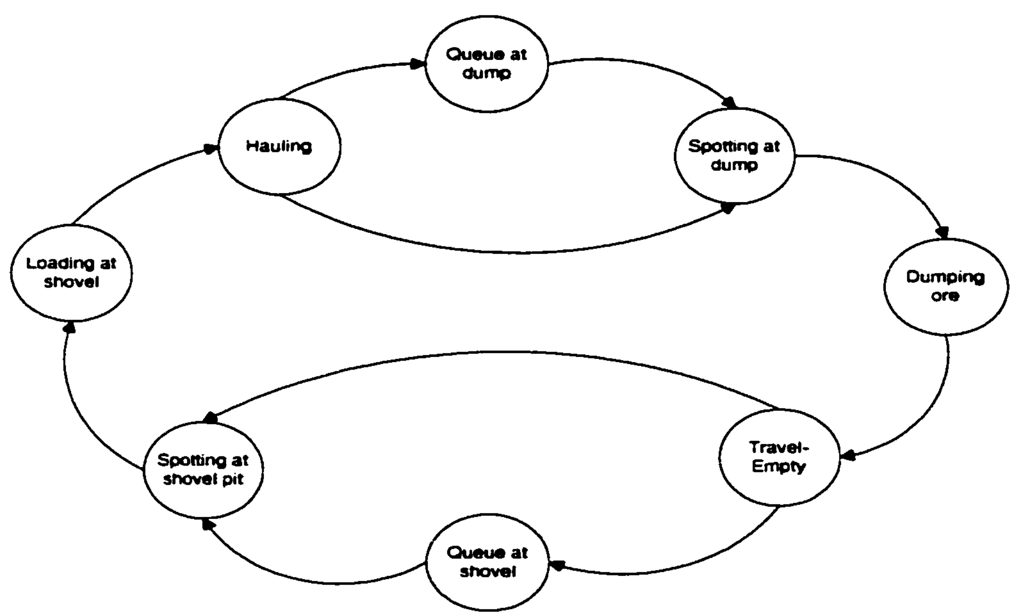


Figure E 1 – Truck State Transitional Diagram

Source State	Destination State	Logic conditions (all conditions must be met)
Hauling	Spotting at Dump	Spent enough time during the hauling state: $\Delta\tau_h \geq \frac{60 * Distance}{v_h}$ (minutes), v_h is Normally distributed with mean and variance $\overline{V}_h, \sigma^2_{Vh}$ Dumping is allowed (the surge pile is not full and the crusher is operating) Less than two trucks are at the dump[27]
Hauling	Queue at Dump	Spent enough time during the hauling state: $\Delta\tau_h \geq \frac{60 * Distance}{v_h}$ (minutes), v_h is Normally distributed with mean and variance $\overline{V}_h, \sigma^2_{Vh}$ Dumping is not allowed (either the surge pile is full or the crusher is down) Two trucks are already at the dump
Queue at Dump	Spotting at Dump	Dumping is allowed Less than two trucks are at the dump.
Spotting at Dump	Dumping	Spent enough time spotting at the dump 1.5 minutes (currently fixed) Dumping is allowed
Dumping	Travel Empty	The truck has already spent longer than the specified dumping time (specified in the parameter file)
Travel Empty	Spotting at Shovel	Spent enough time during travel empty state: $\Delta\tau_e \geq \frac{60 * Distance}{v_e}$ (minutes), v_e is Normally distributed with mean and variance $\overline{V}_e, \sigma^2_{Ve}$ Loading is allowed Less than two trucks are in service at the shovel[28]
Travel Empty	Queue at Shovel	Spent enough time during travel empty state: $\Delta\tau_e \geq \frac{60 * Distance}{v_e}$ (minutes), v_e is Normally distributed with mean and variance $\overline{V}_e, \sigma^2_{Ve}$ Loading is not allowed Two trucks are already in service at the shovel
Spotting at Shovel	Loading	Spent enough time spotting at the shovel (the spotting time is currently fixed at 2 minutes) Loading is allowed Less than two trucks are in service at the shovel

[27] A truck is defined in service at the dump when it is in either one of the two states: Spotting at dump, Dumping

[28] A truck is defined in service at the shovel when it is in either one of the two states: Spotting at Shovel, Loading

Loading	Hauling	Spent enough time loading at the shovel: $\Delta\tau_{gl} \geq normal(\bar{\tau}_{gl}, \sigma_{gl})$, $g \in \{240T, 320T, 360T\}$ The information on the means and the variances is provided in the parameter file

Table E 1 – Transition of Truck State (Simulator)

The simulation uses a random number generator to obtain values for the truck speeds, truckloads, spotting time at shovels and dumps, loading times at the shovels. While the truckload is generated directly using this random number generator, the truck cycle time is determined as a linear function of the time-based parameters associated with various states in a truck cycle. Table E2 shows the quantities that rely on this random number generator.

States	Quantities	Random generator	Input Parameters
Hauling	Truck hauling speeds (for different truck groups: 240T, 320T, 360T)	$V_{hg} \Leftarrow normal(\bar{V}_{hg}, \sigma_{hg})$ km/h $g \in \{240T, 320T, 360T\}$	$\bar{V}_{hg}, \sigma_{hg}$
Traveling-Empty	Truck travel-empty speeds (for different truck groups: 240T, 320T, 360T)	$V_{eg} \Leftarrow normal(\bar{V}_{eg}, \sigma_{eg})$ km/h $g \in \{240T, 320T, 360T\}$	$\bar{V}_{eg}, \sigma_{eg}$
Spotting at Dump	The elapsed time required to spot the truck (get the truck into the dump position)	$\Delta\tau_{spot} \Leftarrow normal(\bar{\tau}_{spot}, \sigma_{spot})$ depending on the dump	$\bar{\tau}_{spot}, \sigma_{spot}$
Loading	The elapsed time during loading (mins) The truckload (Tonnes)	$\Delta\tau_{lg} \Leftarrow normal(\bar{\tau}_{lg}, \sigma_{lg})$ $L_g \Leftarrow normal(\bar{L}_g, \sigma_{Lg})$	$\bar{\tau}_{lg}, \sigma_{lg}$ $\bar{L}_g, \sigma_{Lg}$
Spotting at shovel	The elapsed time required to spot the truck (get the truck into the loading position) (mins)	$\Delta\tau_{spot} \Leftarrow normal(\bar{\tau}_{spot}, \sigma_{spot})$ depending on the shovel	$\bar{\tau}_{spot}, \sigma_{spot}$

Table E 2 – Random-Generated Parameters

The simulator was designed to work with multiple dumps and multiple shovels with each route uniquely connecting a shovel to a dump. The trucks are assigned to individual routes and their number remains fixed during the simulation period, which is measured in hours. The case studies involve the simulation of the ore handling process with 3 ore shovels and 2 ore dumps (Figure 4.3). The two dumps feed the crushed ore in the surge pile, which has a restricted capacity. When this surge pile is full, these loaded trucks have to stay in the queue and wait for the signal to dump.

The simulator can run in two modes: stand-alone mode and GAMS-coupling mode. In the stand-alone mode, the simulator works with a parameter file, which is the only input source. This file contains the parameters that describe the simulated conditions and the simulation time period. When running in GAMS-coupling mode, the simulator uses GAMS result data in addition to the regular parameter data. For common parameter information, GAMS data will take precedent over the data from the regular parameter file.

The simulator was developed in Visual C++ (Visual Studio 6.0) and operated in the Microsoft Windows NT 4.0 environment. The simulator and the GAMS program were integrated and controlled with a command file. The following illustrates a typical command file that was used for the simulation test.

```
rem -------------------------------------------------------------
-
rem Scenario 1:
rem      To simulate the change of road condition which affects
rem      truck speed.
rem      For example, the fog in the morning reduces the truck
speed
rem      however as the sun rises, trucks can be driven faster
and
rem      the truck speeds gradually increase during the time
period.
Rem
Rem      levelminlp.gms
Rem            This is the Gams program as shown in Appendix G4
Rem      MinePar.dat
Rem            This file contains  parameter data for the
simulator
Rem      The logical flag '1'
Rem            is the indicator for the Gams-coupled mode
Rem      MineSim2.exe
Rem           is the simulator executable program
Rem      Init.cmd
Rem            is the miscellaneous bat file used to initialize
the
```

```
Rem             simulating environment and set up initial data
for
Rem             the Gams program (such as initial surge volume,
etc.)
rem -----------------------------------------------------------
-

call init

gams levelminlp
d:\truck\MIneSim2\Release\MineSim2 MinePar.dat 1
if errorlevel 1 goto end

gams levelminlp
d:\truck\MIneSim2\Release\MineSim2 MinePar.dat 1
if errorlevel 1 goto end

gams levelminlp
d:\truck\MIneSim2\Release\MineSim2 MinePar.dat 1
if errorlevel 1 goto end

gams levelminlp
d:\truck\MIneSim2\Release\MineSim2 MinePar.dat 1
if errorlevel 1 goto end

gams levelminlp
d:\truck\MIneSim2\Release\MineSim2 MinePar.dat 1
if errorlevel 1 goto end

gams levelminlp
d:\truck\MIneSim2\Release\MineSim2 MinePar.dat 1
if errorlevel 1 goto end

gams levelminlp
d:\truck\MIneSim2\Release\MineSim2 MinePar.dat 1
if errorlevel 1 goto end

gams levelminlp
d:\truck\MIneSim2\Release\MineSim2 MinePar.dat 1
if errorlevel 1 goto end

gams levelminlp
d:\truck\MIneSim2\Release\MineSim2 MinePar.dat 1
if errorlevel 1 goto end

gams levelminlp
```

```
d:\truck\MIneSim2\Release\MineSim2 MinePar.dat 1
if errorlevel 1 goto end


del level.dat
type level*.dat > AllLevel%1.dat
rename Summary.csv Summary%1.csv
rename MineSim.out MineSim%1.out
rename SurgeVol.csv SurgeVol%1.csv
:end
```

Appendix F – Chapter-4 Result Data

Period	Total number of 240T trucks	Truck-delivered ore rate (tph)	GAMS-calculated ore rate (tph)
1	13	6335	6688
2	12	5918	6035
3	12	6070	6069
4	12	5979	6096
5	12	5945	5881
6	12	6266	6113
7	12	6099	5720
8	11	5446	5413
9	12	6084	5963
10	12	6900	5876
11	12	6007	5774
12	12	6082	5928

Table F 1 – Results (Scenario 1, CCP, Single Truck Type)

Period	Total number of 240T trucks	Truck-delivered ore rate (tph)	GAMS-calculated ore rate (tph)
1	15	6830	6495
2	12	5521	5336
3	13	6291	5975
4	13	6345	5920
5	11	4353	5129
6	15	6017	6887
7	17	6419	7018
8	17	6716	6753
9	16	6292	5997
10	15	5891	5660
11	15	5855	5816
12	16	6382	6123

Table F 2 – Results (Scenario 2, CCP, 1 Truck Type)

Period	Total number of 240T trucks	Truck-delivered ore rate (tph)	GAMS-calculated ore rate (tph)
1	13	6454	6688
2	11	5462	5759
3	12	6378	6190
4	11	5835	5793
5	12	5090	6066
6	15	6010	6831
7	16	6993	7303
8	15	5875	6205
9	15	5684	6341
10	16	6480	6605
11	15	6262	6086
12	15	5792	5907

Table F 3 – Results (Scenario 3, CCP, Single Truck Type)

Period	Total number of 240T trucks	Truck-delivered ore rate (tph)	GAMS-calculated ore rate (tph)
1	11	4982	6050
2	13	6020	6934
3	14	6786	7083
4	13	6531	6467
5	12	5906	6077
6	12	5823	5975
7	13	6209	6209
8	12	6343	6060
9	12	5851	5739
10	12	5954	5887
11	12	6033	5883
12	12	6038	6036

Table F 4 – Results (Scenario 1, Deterministic, Single Truck type)

Period	Total number of 240T trucks	Truck-delivered ore rate (tph)	GAMS-calculated ore rate (tph)
1	13	6392	5720
2	12	5671	5472
3	12	6019	5653
4	12	6024	5782
5	12	4810	5846
6	16	5666	7072
7	17	6611	7186
8	17	6420	7060
9	17	6456	6679
10	17	6404	6630
11	16	6256	6127
12	16	6077	5859

Table F 5 – Results (Scenario 2, Deterministic, Single Truck Type)

Period	Total number of 240T trucks	Truck-delivered ore rate (tph)	GAMS-calculated ore rate (tph)
1	11	5265	6050
2	12	6085	6543
3	12	6459	6502
4	12	6155	6187
5	11	4742	5904
6	15	6173	7185
7	16	5919	7355
8	16	5923	7048
9	17	6938	7152
10	15	6422	6253
11	14	5985	5902
12	14	5639	5769

Table F 6 – Results (Scenario 3, Deterministic, Single Truck Type)

Period	Total number of trucks required			Truck-delivered ore rate (tph)	GAMS-calculated ore rate (tph)
	240T	320T	360T		
1	2	5	2	6706	6230
2	4	1	3	5546	5427
3	2	1	5	6417	6174
4	1	2	4	5683	5713
5	1	1	5	6274	6092
6	1	1	5	6055	5908
7	0	3	4	6088	5805
8	0	3	4	6053	5891
9	0	2	5	5931	5961
10	0	2	5	6280	6165

Table F 7 – Results (Scenario 1, CCP, 3 Truck Types)

Period	Total number of trucks required			Truck-delivered ore rate (tph)	GAMS-calculated ore rate (tph)
	240T	320T	360T		
1	2	5	2	6421	6230
2	3	4	2	6105	5916
3	0	2	5	5991	5722
4	3	2	3	5353	5758
5	3	1	5	6042	6623
6	3	2	4	5961	6416
7	2	2	5	6297	6635
8	0	3	5	5887	6239
9	1	4	4	5807	6508
10	1	4	4	6291	6368

Table F 8 – Results (Scenario 2, CCP, 3 Truck Types)

Period	Total number of trucks required			Truck-delivered ore rate (tph)	GAMS-calculated ore rate (tph)
	240T	320T	360T		
1	2	5	2	6764	5728
2	1	3	3	5975	5616
3	1	4	2	5948	5835
4	1	3	3	5848	5518
5	1	2	4	4494	6184
6	3	3	4	5783	7139
7	3	4	4	6774	7076
8	4	3	4	6092	7568
9	2	4	4	6247	7025
10	1	4	5	5919	7317

Table F 9 – Results (Scenario 3, CCP, 3 truck types)

Period	Total number of trucks required			Truck-delivered ore rate (tph)	GAMS-calculated ore rate (tph)
	240T	320T	360T		
1	2	2	4	5917	5728
2	2	3	3	6300	5770
3	3	2	3	6050	5708
4	4	1	3	5527	5583
5	4	2	3	6552	6222
6	3	3	2	5701	5907
7	0	4	3	5965	6052
8	2	3	3	6072	6106
9	0	4	3	6106	6113
10	1	2	4	6048	5887

Table F 10 – Results (Scenario 1, Deterministic, 3 Truck Types)

Period	Total number of trucks required			Truck-delivered ore rate (tph)	GAMS-calculated ore rate (tph)
	240T	320T	360T		
1	2	2	4	5938	5728
2	4	2	3	6209	6043
3	3	2	3	5781	5653
4	2	3	3	5935	6100
5	1	4	3	5401	6328
6	0	3	5	6041	6690
7	4	0	5	5671	6473
8	2	3	5	6596	7014
9	1	3	5	6512	6541
10	3	1	5	5683	6184

Table F 11 – Results (Scenario 2, Deterministic, 3 Truck Types)

Period	Total number of trucks required			Truck-delivered ore rate (tph)	GAMS-calculated ore rate (tph)
	240T	320T	360T		
1	2	2	4	6364	5728
2	4	1	3	5618	5570
3	3	2	3	6257	5968
4	3	3	2	5740	5609
5	1	2	4	4695	5876
6	3	4	3	5840	6914
7	6	0	5	5812	7279
8	0	6	4	6792	7475
9	3	4	3	5511	6656
10	6	0	5	6357	7003

Table F 12 – Results (Scenario 3, Deterministic, 3 Truck Types)

Appendix G – GAMS Programs

Appendix G1 – Deterministic Linear Program (Chapter 1)

```
*----------------------------------------------------------------
-----
* Truck Allocation optimization problem
*
* Ore trucks are allocated at the beginning of the 12-hour shift
* to haul ore.
* Deterministic model with one ore shovel and one crusher
* Ore truck cycle time is constant
* Waste truck cycle time is also constant but different for each
* truck type. Truckload is kept constant (Trucks are categorized
based
* on its loading capacity.
* The time period calculated in the model: 12 hours
* (single-peiod problem)
*----------------------------------------------------------------
-----

Sets
     m   'Mine material'/'o', 'w'/
     j   "truck type"    /'360T','320T','240T'/
     ;

Parameter Cap(j)        /'360T' 360,'320T' 320,'240T' 240/;

Table L(m,j) "Truckload (tons)"
         '360T' '320T'  '240T'
     'o'  327  290  220
     'w'  327  290  220;

Parameter R(j) "Resource Truck limit"
         /'360T' 5, '320T' 9, '240T' 18/;

Parameter Tw(j) /'360T' 25, '320T' 30, '240T' 35/;
Scalars
     Hours                                              /12/
     To "Ore Truck Cycle Time based on shovel (minutes)"  /25/
     D "Ore rate extracted by the Extraction plant"      /7000/
```

```
     WasteMin "Minimum amount of waste to be moved"
/60000/
     ;

Variables
  TotalWaste1 "Total amount of waste moved as the maximized
number"
  TotalWaste2 "Total amount of waste moved as the maximized
number"
  Xo(j)      "Number of Ore production trucks required"
  iXo(j)     "Integer Number of Ore production trucks required"
  Xw(j)      "Number of Waste production trucks required"
  iXw(j)      "Number of Waste production trucks required"
  ;
Positive Variable Xo,Xw;
Integer Variable iXo,iXw;

Equations
     eW   "Waste requirement for the whole time"
     eW_int    "Waste requirement for the whole time"
     eTL(j)    "Truck Resource limit on second half of period"
     eTL_int(j)     "Truck Resource limit on second half of
period"
     eDemand
     eDemand_int
     eWasteMin
     eWasteMin_int
     ;
eW   .. TotalWaste1 =e= sum(j,Hours * L('w',j) * Xw(j) *
60/Tw(j)) ;
eW_int    .. TotalWaste2 =e= sum(j,Hours * L('w',j) * iXw(j) *
60/Tw(j)) ;

eDemand .. sum(j, Xo(j) * L('o',j) * 60/To) =g= D;
eDemand_int .. sum(j, iXo(j) * L('o',j) * 60/To) =g= D;

eWasteMin .. Hours*sum(j,Xw(j)   * L('w',j)* 60/Tw(j)) =g=
WasteMin;
eWasteMin_int .. Hours*sum(j,iXw(j)   * L('w',j)* 60/Tw(j)) =g=
WasteMin;

eTL(j) ..  Xo(j)+Xw(j)   =l= R(j);
eTL_int(j) ..  iXo(j)+iXw(j)   =l= R(j);

Model transport /eW,eDemand,eWasteMin,eTL/;
Model transport_int /eW_int,eDemand_int,eWasteMin_int,eTL_int/;
```

```
solve transport using lp maximizing TotalWaste1;

Option OptCR = 0.01;
solve transport_int using mip maximizing TotalWaste2;

scalar ActOreRate;
scalar CalcWaste;

file out /sdetlpw3.dat/;
put out;

put /'Input' /;
put '  Truck fleet
(240T,320T,360T):',R('240T'):4:0,R('320T'):4:0,R('360T'):4:0 //;
put '  Ore Truck cycle Time:   ', To:4:0,' minutes'/;
put '  Waste Truck cycle Time: ',
Tw('360T'):5:0,Tw('320T'):5:0,Tw('240T'):5:0,' minutes'/;
put '  Ore Truckload:    ', L('o','360T'):5:0,
L('o','320T'):5:0,L('o','240T'):5:0 /;
put '  Waste Truckload: ', L('w','360T'):5:0,
L('w','320T'):5:0,L('w','240T'):5:0 //;
put '  Number of hours in the period: ', Hours:3:0/;
put '------------------------------------------------------------
'//;

put "Continuous Result"//;

ActOreRate = sum(j, Xo.l(j) * L('o',j)*60/To);


put 'Deterministic Solution:(solution status =
',transport.modelstat:2:0,')'//;

put '    Ore 240T Trucks: ', Xo.l('240T'):6:2 /;
put '    Ore 320T Trucks: ', Xo.l('320T'):6:2 /;
put '    Ore 360T Trucks: ', Xo.l('360T'):6:2 //;

put '    Waste 240T Trucks: ', Xw.l('240T'):6:2 /;
put '    Waste 320T Trucks: ', Xw.l('320T'):6:2 /;
put '    Waste 360T Trucks: ', Xw.l('360T'):6:2 //;
put '    Marginal Values: '/;
put '            Ore Hourly Demand -> ', eDemand.m:6:3, '
(WTons/OTon)'//;
put '            Truck Resource    -> '; loop(j,put
eTL.m(j):6:0); put ' (WTons/OTon)'//;
put '            (Order: 360T,320T,240T)'//;
```

```
put 'Actual Ore Rate (Tons/hour): ', ActOreRate:12:0,'
(Tons)'//;
put 'Maximum waste moved: ', TotalWaste1.l:12:0, ' (WTons)'//;
put '-----------------------------------------------------------'/;

put "Discrete Result"//;

ActOreRate = sum(j, iXo.l(j) * L('o',j)*60/To);
put 'Deterministic Solution:(solution status =
',transport_int.modelstat:2:0,')'//;
put '     Ore 240T Trucks: ', iXo.l('240T'):6:2 /;
put '     Ore 320T Trucks: ', iXo.l('320T'):6:2 /;
put '     Ore 360T Trucks: ', iXo.l('360T'):6:2 //;

put '     Waste 240T Trucks: ', iXw.l('240T'):6:2 /;
put '     Waste 320T Trucks: ', iXw.l('320T'):6:2 /;
put '     Waste 360T Trucks: ', iXw.l('360T'):6:2 //;
put 'Actual Ore Rate (Tons/hour): ', ActOreRate:12:0,'
(Tons)'//;
put 'Maximum waste moved: ', TotalWaste2.l:12:0, ' (WTons)'//;
put '-----------------------------------------------------------
'//;

put 'Results as being rounded up from the continuous solution'/;
ActOreRate = sum(j, ceil(Xo.l(j)) * L('o',j)*60/To);
CalcWaste = Hours*sum(j, (R(j) -
ceil(Xo.l(j)))*L('w',j)*60/Tw(j));

put '     Ore 240T Trucks: ', ceil(Xo.l('240T')):6:2 /;
put '     Ore 320T Trucks: ', ceil(Xo.l('320T')):6:2 /;
put '     Ore 360T Trucks: ', ceil(Xo.l('360T')):6:2 //;

put 'Actual Ore Rate (Tons/hour): ', ActOreRate:12:0,'
(Tons)'//;
put 'Maximum waste moved: ', CalcWaste:12:0, ' (WTons)'//;

putclose;
```

Appendix G2 – Two-Stage Recourse Program (Chapter 2)

```
$ontext
 Truck allocation optimization problem as a stochastic model via
 2-stage recourse programming (DECIS solver)

 Uncertain parameters (assumed independent to one another)
```

```
     Truckload for 240-Ton truck       normal(220,20)
     Truckload for 320-Ton truck       normal(290,25)
     Truckload for 360-Ton truck       normal(327,30)
     Ore Truck cycle time for 240-Ton truck normal(24,5)
     Ore Truck cycle time for 320-Ton truck normal(24,5)
     Ore Truck cycle time for 360-Ton truck normal(24,5)
$offtext
*---------------------------------------------------------------
-----
Sets
     m    'Mine material'/'o', 'w'/
     j    "truck type"    /'360T','320T','240T'/
     h    "hours in the shift" /1*12/
     stoch "Stochatic set data" /'out','pro'/
     w1   "truckload scenario for 240-Ton truck" /w-1*w-11/
     w2   "truckload scenario for 320-Ton truck" /w-1*w-11/
     w3   "truckload scenario for 360-Ton truck" /w-1*w-11/
     t1 "scenario for cycle time of 240-Ton ore truck" /t-1*t-
11/
     t2 "scenario for cycle time of 320-Ton ore truck" /t-1*t-
11/
     t3 "scenario for cycle time of 360-Ton ore truck" /t-1*t-
11/
;
*---------------------------------------------------------------
-----
* Probability Distribution of demand
*---------------------------------------------------------------
-----
Table LO240(stoch,w1) "Ore Truckload for 240-Ton truck"
    w-1      w-2     w-3     w-4     w-5     w-6     w-7     w-8
w-9     w-10    w-11
out 135.00  152.00  169.00  186.00  203.00  220.00  237.00
254.00  271.00  288.00  305.00
pro 0.0001  0.0015  0.0150  0.0846  0.2339  0.3296  0.2339
0.0846  0.0150  0.0015  0.0001;


Table LO320(stoch,w2) "Ore Truckload for 320-Ton truck"
    w-1      w-2     w-3     w-4     w-5     w-6     w-7     w-8
w-9     w-10    w-11
out 185.00  206.00  227.00  248.00  269.00  290.00  311.00
332.00  353.00  374.00  395.00
pro 0.0002  0.0017  0.0159  0.0862  0.2331  0.3259  0.2331
0.0862  0.0159  0.0017  0.0002;

Table LO360(stoch,w3) "Ore Truckload for 360-Ton truck"
```

```
    w-1       w-2     w-3      w-4      w-5      w-6      w-7      w-8
w-9     w-10     w-11
out 177.00  207.00  237.00  267.00  297.00  327.00  357.00
387.00  417.00  447.00  477.00
pro 0.0001  0.0014  0.0144  0.0834  0.2345  0.3322  0.2345
0.0834  0.0144  0.0014  0.0001;

*----------------------------------------------------------------

Table T240(stoch,t1) "Ore Truck cycle Time for 240-Ton trucks"
    t-1       t-2     t-3      t-4      t-5      t-6      t-7      t-8
t-9     t-10     t-11
out 4.00     8.00    12.00    16.00    20.00    24.00    28.00
32.00    36.00    40.00    44.00
pro 0.0003  0.0025  0.0198  0.0927  0.2291  0.3112  0.2291
0.0927  0.0198  0.0025  0.0003;

Table T320(stoch,t2) "Ore Truck cycle Time for 320-Ton trucks"
    t-1       t-2     t-3      t-4      t-5      t-6      t-7      t-8
t-9     t-10     t-11
out 4.00     8.00    12.00    16.00    20.00    24.00    28.00
32.00    36.00    40.00    44.00
pro 0.0003  0.0025  0.0198  0.0927  0.2291  0.3112  0.2291
0.0927  0.0198  0.0025  0.0003;

Table T360(stoch,t3) "Ore Truck cycle Time for 360-Ton trucks"
    t-1       t-2     t-3      t-4      t-5      t-6      t-7      t-8
t-9     t-10     t-11
out 4.00     8.00    12.00    16.00    20.00    24.00    28.00
32.00    36.00    40.00    44.00
pro 0.0003  0.0025  0.0198  0.0927  0.2291  0.3112  0.2291
0.0927  0.0198  0.0025  0.0003;

*-----------------------------------------------------------------
--

Parameter Cap(j)        /'360T' 360,'320T' 320,'240T' 240/;

Table L(m,j) "Truckload (tons)"
         '360T' '320T'  '240T'
     'o'  327  290  220
     'w'  327  290  220;

Parameter mL(j) /'360T' 327,'320T' 290,'240T' 220/;
Parameter sL(j) /'360T' 35,'320T' 25,'240T' 20/;

Scalar mTo "mean ore truck cycle time" /24/;
Scalar sTo  "standard deviation of ore truck cycle time" /5/;
```

```
Parameter R(j) "Resource Truck limit"
        /'360T' 5, '320T' 9, '240T' 18/;

Scalars
     Hours                              /12/
     OTime "Ore Truck Cycle Time based on shovel (minutes)"
/24/
     Tw "Waste Truck Cycle Time based on shovel (minutes)" /30/
     D "Ore rate extracted by the Extraction plant"      /7000/
     WasteMin "Minimum amount of waste to be moved"
/60000/
     ;

Variables
  TotalWaste "Total amount of waste moved as the maximized
number"
  Xo(j)     "Number of Ore production trucks required"
  Q
  ;
Positive Variable Xo, Q;


Equations
     eW   "Waste requirement for the whole time"
     eTL(j)    "Truck Resource limit on second half of period"
     eDemand "Hourly Ore demand specified by Extraction"
     eWasteMin "Minimum Waste amount to be moved"
     ;
eW .. TotalWaste =e= Hours*sum(j, L('w',j)* (R(j)-Xo(j)) *
(60/Tw)) - 2*Q;
eDemand .. Hours*sum(j, Xo(j)*(60/OTime) * L('o',j)) + Q =g=
Hours*D;
eWasteMin .. Hours*sum(j,(R(j)-Xo(j))*(60/Tw) * L('w',j))  =g=
WasteMin;
eTL(j) ..  Xo(j)  =l= R(j);

Model transport /all/;

*---------------------------------------------
* Decision stages
*---------------------------------------------
Xo.stage(j) = 1;
Q.stage = 2;
```

```
eTL.stage(j) = 2;
eWasteMin.stage = 2;
eDemand.stage = 2;

parameter LTO240(stoch,t1,w1);
parameter LTO320(stoch,t2,w2);
parameter LTO360(stoch,t3,w3);

loop(t1,
  loop(w1,
    LTO240('out',t1,w1) = Hours*60 * LO240('out',w1) /
T240('out',t1);
    LTO240('pro',t1,w1) = LO240('pro',w1) * T240('pro',t1);
  );
);

loop(t2,
  loop(w2,
    LTO320('out',t2,w2) = Hours*60 * LO320('out',w2) /
T320('out',t2);
    LTO320('pro',t2,w2) = LO320('pro',w2) * T320('pro',t2);
  );
);

loop(t3,
  loop(w2,
    LTO360('out',t3,w3) = Hours*60 * LO360('out',w3) /
T360('out',t3);
    LTO360('pro',t3,w3) = LO360('pro',w3) * T360('pro',t3);
  );
);

*--------------------------------------------
file stg /MODEL.STG/
put stg;
scalar temp;
put "INDEP DISCRETE" /;

loop((t1,w1),
  put 'Xo  240T eDemand  ',LTO240('out',t1,w1), ' period2 ',
LTO240('pro',t1,w1) /;
);
put '*'/;

loop((t2,w2),
```

```
  put 'Xo 320T eDemand  ',LTO320('out',t2,w2), ' period2 ',
LTO320('pro',t2,w2) /;
);
put '*'/;

loop((t3,w3),
  put 'Xo 360T eDemand ',LTO360('out',t3,w3), ' period2 ',
LTO360('pro',t3,w3) /;
);
put '*'/;


put "*"/;
putclose stg;

*--------------------------------------------
* Output a MINOS option file
*--------------------------------------------
file mopt /MINOS.SPC/;
put mopt;
put "begin"/;
put "rows 250"/;
put "columns 250"/;
put "elements 10000"/;
put "end"/;
putclose mopt;

option lp=decism;
solve transport using lp maximizing TotalWaste;

file out /reclpw4.dat/;
put out;

put /'Output' /;
put '  Recourse-based Stochastic Solution:(solution status =
',transport.modelstat:2:0,')'//;
put '  Ore trucks:'/;
loop(j, put '    ',Cap(j):3:0, 'T: ', Xo.l(j):5:2;);put /;
scalar wastemoved;
wastemoved = Hours*sum(j,(R(j)-Xo.l(j))*L('w',j)*60/Tw);
put '  Actual Truck Resource Remaining: ', wastemoved:12:0, '
(WTons)'//;
*put '  Objective function value: ', TotalWaste.l:12:0/;
put '-------------------------------------------------------------
'/;
```

```
putclose;
```

Appendix G3 – Programs to Determine WS, RP, EEV Values (Chapter 3)

```
*-----------------------------------------------------------------
-----
* Program to determine the WS value
*
* Assume that the planner can afford to wait for the uncertainty
to
* reveal itself before trucks are assigned to the hauling tasks.
* Therefore, it is possible that exact number of trucks can be
* assigned
* to transport ore. The resulting effect is that the hauling
process
* will be done with the highest effiency possible.
*
* The model is run many times. The number of running times is
equal
* to the number of realizations being considered as part of the
* uncertainty. The available resource will be averaged over this
* domain of results and it will be the best possible truck
resource  * remaining.
*-----------------------------------------------------------------
-----
*
$offlisting
$offsymlist
$offsymxref

Sets
  s1 /1*100/
  s2 /1*100/
  ;
*----------------------------------------
Scalars
     LoSTD      "Standard deviation of ore truckload"  /25/
     ToSTD      "Standard deviation of ore cycle time" /4/
     WasteL     "Waste Truckload in tonnes"           /290/
     OreL       "Ore Truckload in tonnes"             /290/
     R          "Total number of 320T trucks"         /30/
     WTime      "Waste cycle time in minutes"         /30/
```

```
     OTime      "Waste cycle time in minutes"          /30/
     H          "Number of hours in a shift"           /12/
     D          "Houre ore demand tph"                 /7000/
     ;
*---------------------------------------
* Data for distribution of truckload
*---------------------------------------
Parameter To(s1), Lo(s2);

To(s1) = normal(OTime,ToSTD);
Lo(s2) = normal(OreL,LoSTD);
*-------------------------------------------------------
Variables AvailRes, X;

Positive variable X;

Equations
  Obj
  OreProd
  TruckLm
  ;

Obj          ..  AvailRes =e= (R-X) * WasteL * (60/WTime) ;
OreProd      ..  X * OreL * 60/OTime =g= D;
TruckLm      ..  X =l= R;

Model Alloc /all/;

Scalar RunTotal, AveAvailRes;;

file out /ws.dat/;
put out;

put ' To(s1) = normal(',OTime:2:0,',',ToSTD:2:0,')
(',card(s1):4:0,' samples)'/;
put ' Lo(s2) = normal(',OreL:3:0,',',LoSTD:2:0,')
(',card(s2):4:0,' samples)'/;

RunTotal = 0;
loop((s1,s2),

    OTime = To(s1);
    OreL = Lo(s2);

    Solve Alloc using lp maximizing AvailRes;
```

```
    RunTotal = RunTotal + AvailRes.l;
);
AveAvailRes = RunTotal/( card(s1)*card(s2));
put '   WS = ', AveAvailRes::0//;

putclose;


*-----------------------------------------------------------------
-----
* Gams Program to determine the Recourse solution value (RP)
*
* The stochastic model is solved as a recourse-based method.
* Due to the variations of the truck cycle time and the ore
truck
* load
* the hourly ore demand is not guaranteed to be met. Any
shortfall
* will be
* made up by adding an amount of ore, which is assumed to come
from
* fictitious recourse supply.
* However, penalty will be added to the objective function to
* account for the addition of this fictituous source of ore
supply.
* The fictituous amount of ore will be determined also as part
of the
* decision variables.
*-----------------------------------------------------------------
-----
$offlisting
$offsymlist
$offsymxref

option iterlim = 100000
       reslim = 10000;

Sets
  s1 /1*100/
  s2 /1*100/
  ;
*----------------------------------------
Scalars
     LoSTD     "Standard deviation of Lo"   /25/
     ToSTD     "Standard deviation for To"  /4/
     WasteL    "Waste Truckload in tonnes"  /290/
```

```
     OreL      "Ore Truckload in tonnes"     /290/
     R    "Total number of 320T trucks"      /30/
     WTime     "Waste cycle time in minutes" /30/
     OTime     "Waste cycle time in minutes" /30/
     H    "Number of hours in a shift"       /12/
     D    "Houre ore demand tph"             /7000/
     ;
*----------------------------------------
* Data for distribution of truckload
*----------------------------------------

Parameter To(s1), Lo(s2);

To(s1) = normal(OTime,ToSTD);
Lo(s2) = normal(OreL,LoSTD);
*--------------------------------------------------------
Variables AvailRes, X, ExtraOre(s1,s2);

Positive variable X, ExtraOre;

Equations
  Obj
  OreProd(s1,s2)
  TruckLm
  ;

Obj            ..  AvailRes =e= (R-X) * WasteL * (60/WTime) -
2*sum((s1,s2),ExtraOre(s1,s2))/(card(s1)*card(s2));
OreProd(s1,s2)      ..  X * Lo(s2) * (60/To(s1)) +
ExtraOre(s1,s2) =g= D;
TruckLm        ..  X =l= R;

Model Alloc /all/;

file out /RP.dat/;
put out;

put ' To(s1) = normal(',OTime:2:0,',',ToSTD:2:0,')
(',card(s1):4:0,' samples)'/;
put ' Lo(s2) = normal(',OreL:3:0,',',LoSTD:2:0,')
(',card(s2):4:0,' samples)'/;

Solve Alloc using lp maximizing AvailRes;

display AvailRes.l;
```

```
put '   RP = ', AvailRes.l::0;
put '    (Solution status: ', Alloc.modelstat:3:0,')'//;

putclose;


*-----------------------------------------------------------------
-----
* This pgoram determines the EEV value
*
* The number of trucks is first allocated to satisfy the
expected ore
* production (that is >= 7000 tons/hour).
* However, as the ore truckload and truck cycle time vary
according
* to Normal distributions (assumed to be independent
distributions)
*
* Once the # of trucks is allocated, due to the change in the
ore
* demand the amount of truck resource remaining will also
fluctuate.
* The fluctuation is measured and averaged as the EEV value.
*-----------------------------------------------------------------
-----
$offlisting
$offsymlist
$offsymxref

Sets
  s1 /1*100/
  s2 /1*100/
  ;
*----------------------------------------
Scalars
     LoSTD     "Standard Deviation of Truckload"
     /25/
     ToSTD     "Standard Deviation of Ore Truck cycle time"
     /4/
     WasteL    "Waste Truckload in tonnes"
     /290/
     OreL      "Ore Truckload in tonnes"
     /290/
     R    "Total number of 320T trucks"
     /30/
     WTime     "Waste cycle time in minutes"
     /30/
```

```
     OTime     "Waste cycle time in minutes"
     /30/
     H    "Number of hours in a shift"
     /12/
     D    "Houre ore demand tph"
     /7000/
     ;
*---------------------------------------
* Data for distribution of truckload
*---------------------------------------

Parameter To(s1), Lo(s2);
To(s1) = normal(OTime,ToSTD);
Lo(s2) = normal(OreL,LoSTD);
*-------------------------------------------------------
Variables MeanAvailRes, X;

Positive variable X;

Equations
  Obj
  OreProd
  TruckLm
  ;

Obj          ..  MeanAvailRes =e= (R-X) * WasteL * (60/WTime) ;
OreProd      ..  X * OreL * 60/OTime =g= D;
TruckLm      ..  X =l= R;

Model Alloc /all/;

Solve Alloc using lp maximizing MeanAvailRes;

Scalar OreMoved, RunTotal, ReqExtraOre;
scalar EEV, count;

file out /EEV.dat/;
put out;

put ' To(s1) = normal(',OTime:2:0,',',ToSTD:2:0,')
(',card(s1):4:0,' samples)'/;
put ' Lo(s2) = normal(',OreL:3:0,',',LoSTD:2:0,')
(',card(s2):4:0,' samples)'/;

RunTotal = 0;
count = 0;
```

```
loop((s1,s2),

* Determine the actual ore moved using the number of trucks
assigned
* while taking into account fluctuation of the ore truckload and
* ore cycle time

    OreMoved = X.l * Lo(s2) * 60/To(s1);

    if (OreMoved > D,
       ReqExtraOre = 0;
    else
       ReqExtraOre = D - OreMoved;
    );

    RunTotal = RunTotal + ( MeanAvailRes.l - 2*ReqExtraOre);
    count = count + 1;
  );
EEV = RunTotal /count;

Display EEV;

put '   EEV = ', EEV::0//;

putclose;
```

Appendix G4 – Chance-Constrained Program (Chapter 4)

```
*-----------------------------------------------------------------
-
* Chance Constrained based Truck Allocation model
*
*    Model 1 is a quadratic deterministic equivalent model
*    that is solved for the relaxed truck solution (continuous)
*
*    Model 2 uses the solution from Model 1 to solve for the
*    the discrete solution which lies in the proximity of the
*    the relaxed optimal solution
*
* Chung Ta
* Jul 2001
*-----------------------------------------------------------------
-
```

```
Sets
     t    "Domain for all the hours" /1*3/
Sets
     l    "Domain for the first two moments" / mean, std /
     g    "Truck group" /240T, 320T, 360T/
     s    "Shovel" /s1*s3/
     d    "Crusher" /d1*d2/
     ;

Table Load(s,d,g,l)
$include 'Truckload.dat'
;
Table CycTime(s,d,g,l)
$include 'CycTime.dat'
;

scalar ORate /
$include 'OreRate.dat'
/;
scalar InitSurgeVol /
$include 'SurgeVol.dat'
/;

scalar MinSurgeVol /
$include 'MinSurgeVol.dat'
/;

Scalars
     Kmin "Coefficient corresponds to 95% confidence"      /-
1.645/
     Tw   "Cycle time of waste truck"                      /25/
     MaxSurgeVol "Maximum allowed surge level"             /12000/
     MinTrucksPerShovel "Minimum number of trucks pershovel"
     /3/
     AvgLoadTime "Average Loadtime in minutes"
     /3/
     ;


Parameter Capacity(s) /s1 6000, s2 6000, s3 6000/;

Parameter TruckLm(g);

TruckLm('240T') = 25;
TruckLm('320T') = 10;
TruckLm('360T') = 5;
```

```
Parameter WasteLoad(g)  "Truckload of waste truck"
              /'240T' 220, '320T' 290, '360T' 327/;

*-----------------------------------------------------------------
-----
Variables
     TotalResLeft "Total truck resource left for waste movement
in Tonnes"
     X(s,d,g)  "Number of 240T trucks allocated for the segment"
     iX(s,d,g) "Number of 240T trucks allocated for the segment"
     TV        "The amount of ore delivered every hour by trucks"
     stdTV          "Std. Dev. of TV"
     varLT(s,d,g)   "Variance of the ration of load and cycle
time for each truck type"
     Diff
     ;

Positive Variable X;
Integer variable iX;
*-------------------------------------------------
*Initialization
*-------------------------------------------------
X.l(s,d,'240T') = 4;
X.l(s,d,'320T') = 0;
X.l(s,d,'360T') = 0;

* Initialize the mean hourly ore amount delivered
TV.l =
60*sum((s,d,g),X.l(s,d,g)*Load(s,d,g,'mean')/CycTime(s,d,g,'mean
'));

* Initialize the std. dev. of the ratio of load over cycle time
varLT.l(s,d,g)  = sqr( Load(s,d,g,'mean')/CycTime(s,d,g,'mean'))
*
          ( sqr( Load(s,d,g,'std')/Load(s,d,g,'mean')) +
            sqr(CycTime(s,d,g,'std')/CycTime(s,d,g,'mean')) );

* Initialize the std. dev. of the hourly ore amount delivered
stdTV.l = 60*sqrt( sum((s,d,g), varLT.l(s,d,g) * sqr(X.l(s,d,g))
) );

*-------------------------------------------------
* Define the equations
*-------------------------------------------------
Equations
```

```
     RemTruckRes    "Maximizing remaining truck resource"
     LowerSurgeCons "Ore production constraint"
     ResCons(g)     "Truck Resource constraint"
     eTV      "Mean Ore Volume delivered by trucks"
     estdTV         "Std. Dev. of the mean ore volume delivered
by trucks"
     eVarLT(s,d,g)   "Variance of the quotient of truckload over
cycle time"
     ShovelCons(s) "Upper bound constraint of the shovel"
     MinTrucks(s)   "lower constraint on number of trucks on each
shovel"
     MaxTrucks(s)   "upper constraint on number of trucks on each
shovel"
     ;


*----------------------------------------
* Constraints for the quadratic model
*----------------------------------------
  RemTruckRes .. TotalResLeft =e= sum(g, (TruckLm(g) -
sum((s,d),X(s,d,g)))
                              * WasteLoad(g) * (60/Tw));


  eVarLT(s,d,g) .. VarLT(s,d,g)  =e= sqr(
Load(s,d,g,'mean')/CycTime(s,d,g,'mean')) *
                    ( sqr( Load(s,d,g,'std')/Load(s,d,g,'mean'))
+

sqr(CycTime(s,d,g,'std')/CycTime(s,d,g,'mean')) );


  eTV .. TV =e=
60*sum((s,d,g),X(s,d,g)*Load(s,d,g,'mean')/CycTime(s,d,g,'mean')
);


  estdTV .. stdTV =e= 60*sqrt( sum((s,d,g), VarLT(s,d,g) *
sqr(X(s,d,g))));


  LowerSurgeCons .. InitSurgeVol + (TV - ORate)*3 + Kmin * 3 *
stdTV =g= MinSurgeVol;


  ResCons(g) .. sum((s,d),X(s,d,g)) =l= TruckLm(g);


  ShovelCons(s) ..
sum((d,g),60*X(s,d,g)*Load(s,d,g,'mean')/CycTime(s,d,g,'mean'))
                          =l= Capacity(s);


  MinTrucks(s) .. sum((d,g), X(s,d,g)) =g= MinTrucksPerShovel;
```

```
  MaxTrucks(s) .. sum((d,g), X(s,d,g) * 60 /
CycTime(s,d,g,'mean')) =l= 60 / AvgLoadTime;

*--------------------------------------------------
model TruckAlloc /all/;
*--------------------------------------------------

Solve TruckAlloc using nlp maximizing TotalResLeft;

*****************************************************************
* Mixed-integer truck model
*****************************************************************

loop((s,d,g),
   iX.lo(s,d,g) = floor(X.l(s,d,g));
);

Diff.lo = 0;

Equations
   IntObj  "Minimizing objective"
   IntTruckLm(g) "Resource constraint on each truck group"
   DistCons(s,g) "Constraint to narrow the interval of the
variable space"
   ;

IntObj .. Diff =e= sum((s,d,g), (iX(s,d,g) - X.l(s,d,g)) *
Load(s,d,g,'mean')*60/CycTime(s,d,g,'mean'));

IntTruckLm(g) .. sum((s,d),iX(s,d,g)) =l= TruckLm(g);

DistCons(s,g) .. sum(d, iX(s,d,g)) =l= ceil(sum(d,X.l(s,d,g)));

model IntTruckAlloc / IntObj, IntTruckLm, DistCons/;

option iterlim = 800;

Solve IntTruckAlloc using Mip minimizing Diff;

Display X.l, iX.l;

parameter TotalContTrucks(g), TotalIntTrucks(g);

loop(g,
  TotalContTrucks(g) = sum((s,d), X.l(s,d,g));
  TotalIntTrucks(g) = sum((s,d), iX.l(s,d,g));
```

```
);

parameter cTrucks(s,g), iTrucks(s,g);

file out /level.dat/;
put out;
put '*'/;
put '* Rate delivered by trucks: '; put TV.l::0 /;
loop(s,
 put '* Shovel, ', ord(s):6:0;
 loop(g,
   cTrucks(s,g) = sum(d,X.l(s,d,g));
   iTrucks(s,g) = sum(d,iX.l(s,d,g));
   put iTrucks(s,g):6:0, ' (',cTrucks(s,g):5:2,')';
 );
 put /;
);
put '*'/;
put 'Hourly Ore Rate, ', ORate::0/;
put 'Initial surge Volume,', InitSurgeVol::0/;
put 'Truck-delivered rate,', TV.l::0/;
put 'Hours of simulation, 3' /;
put 'NLP Model Status, ',TruckAlloc.modelstat:2:0 /;
put 'MIP Model Status, ',IntTruckAlloc.modelstat:2:0 /;

loop((s,d),
 put 'Route, ', ord(s):2:0,',',ord(d):2:0;
 put         ',240T,', iX.l(s,d,'240T');
 put         ',320T,', iX.l(s,d,'320T');
 put         ',360T,', iX.l(s,d,'360T') /;
);
putclose;
```

Appendix G5 - Deterministic Program (Chapter 4)

```
*-----------------------------------------------------------------
-
* Deterministic linear Truck Allocation model
*
* Chung Ta
* Jul 2001
*-----------------------------------------------------------------
-
Sets
```

```
     t    "domain for all the hours" /1*3/
Sets
     l    "domain for the first two moments" / mean, std /
     g    "truck group" /240T, 320T, 360T/
     s    "Shovel" /s1*s3/
     d    "Crusher" /d1*d2/
     ;

Table Load(s,d,g,l)
$include 'Truckload.dat'
;
Table CycTime(s,d,g,l)
$include 'CycTime.dat'
;

scalar ORate /
$include 'OreRate.dat'
/;

scalar InitSurgeVol /
$include 'SurgeVol.dat'
/;

scalar MinSurgeVol /
$include 'MinSurgeVol.dat'
/;

Parameter Capacity(s) /s1 3000, s2 3000, s3 3000/;

Parameter TruckLm(g);

TruckLm('240T') = 25;
TruckLm('320T') = 10;
TruckLm('360T') = 5;

Parameter WasteLoad(g)  "Truckload of waste truck"
              /'240T' 220, '320T' 290, '360T' 327/;

*-----------------------------------------------------------------
-----
Scalars
 Tw       "Waste Truck cycle time"                   /30/
 LoadTime "Longest loading time (for 360T) in minutes"    /3/
 MinTrucksPerShovel "Minimum number of trucks per shovel" /2/
 AvgLoadTime   "Average Load time per truck"              /3/
```

```
 ;
*------------------------------------------------
Variables
     TotalResLeft "Total truck resource left for waste movement
in Tonnes"
     X(s,d,g)  "Number of 240T trucks allocated for the segment"
     TV
     ;
integer variable X;
*------------------------------------------------
*Initialization
*------------------------------------------------
X.l(s,d,'240T') = 4;
X.l(s,d,'320T') = 1;
X.l(s,d,'360T') = 1;

*------------------------------------------------
* Define the equations
*------------------------------------------------
Equations
 RemTruckRes   "Maximizing remaining truck resource"
 SurgeCons     "Ore production constraint"
 ResCons(g)    "Truck Resource constraint"
 eTV           "Mean Ore Volume delivered by trucks per hour"
 ShovelCons(s) "Shovel constraint"
 MaxTrucks(s)  "Upper constraint on number of trucks on each
shovel"
 MinTrucks(s)  "lower constraint on number of trucks on each
shovel"
 ;

 RemTruckRes .. TotalResLeft =e= sum(g,(TruckLm(g) -
sum((s,d),X(s,d,g)))* WasteLoad(g) * (60/Tw));

 eTV .. TV =e=
sum((s,d,g),X(s,d,g)*60*Load(s,d,g,'mean')/Cyctime(s,d,g,'mean')
);

 SurgeCons .. InitSurgeVol + (TV - ORate)* 3 =g= MinSurgeVol;

 ResCons(g) .. sum((s,d),X(s,d,g)) =l= TruckLm(g);

 ShovelCons(s) ..
sum((d,g),60*X(s,d,g)*Load(s,d,g,'mean')/CycTime(s,d,g,'mean'))
=l=
              Capacity(s);
```

```
 MaxTrucks(s) .. sum((d,g), X(s,d,g) * 60 /
CycTime(s,d,g,'mean')) =l= ceil(60 / AvgLoadTime);

 MinTrucks(s) .. sum((d,g), X(s,d,g)) =g= MinTrucksPerShovel;
*------------------------------------------------
* Define the model
*------------------------------------------------
model TruckAlloc /all/;
*------------------------------------------------

Solve TruckAlloc using mip maximizing TotalResLeft;

file out /level.dat/;
put out;
put '*
=============================================================='/
;
put '* Minimum required surge level at the end: ',
MinSurgeVol::0/;
put '* Truck delivered rate: ', TV.l::0/;
put '* MIP Solver Status = ', TruckAlloc.solvestat:2:0 /;
put '*'/;
put 'Hourly Ore Rate, ', ORate::0/;
put 'Initial surge Volume,', InitSurgeVol::0/;
put 'Hours of simulation, 3' /;
put 'MIP Model Status, ',TruckAlloc.modelstat:2:0 /;

loop((s,d),
 put 'Route, ', ord(s):2:0,',',ord(d):2:0;
 put          ',240T,', X.l(s,d,'240T');
 put          ',320T,', X.l(s,d,'320T');
 put          ',360T,', X.l(s,d,'360T') /;
);
putclose;
```

Appendix H - CycleTime & Truckload Data

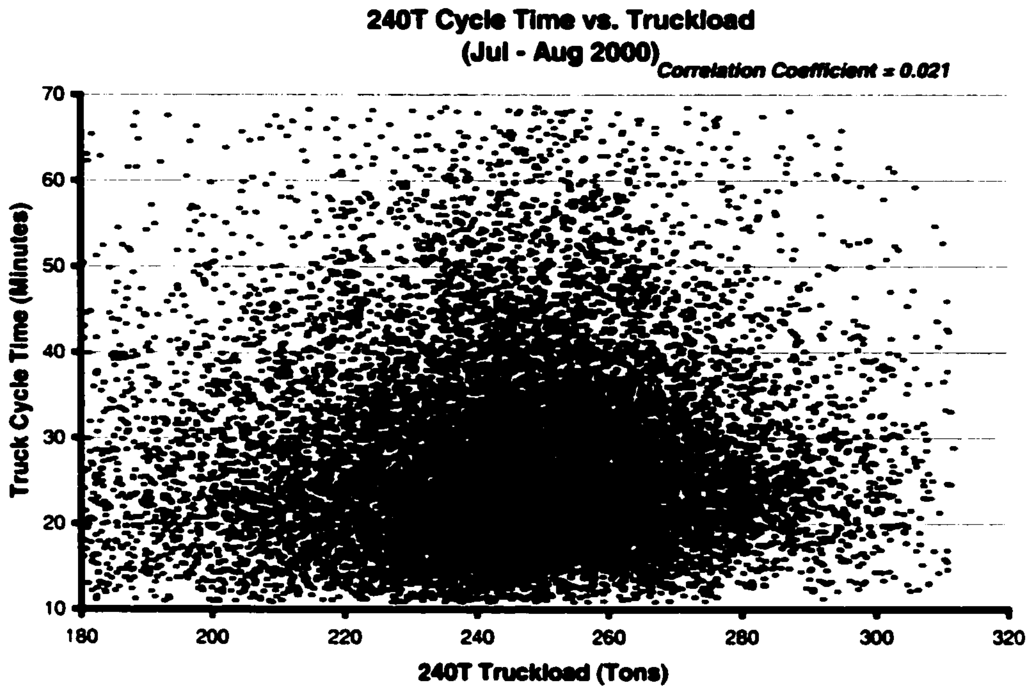


Figure H 1 – Truckload and Cycle Time Correlation (240T)

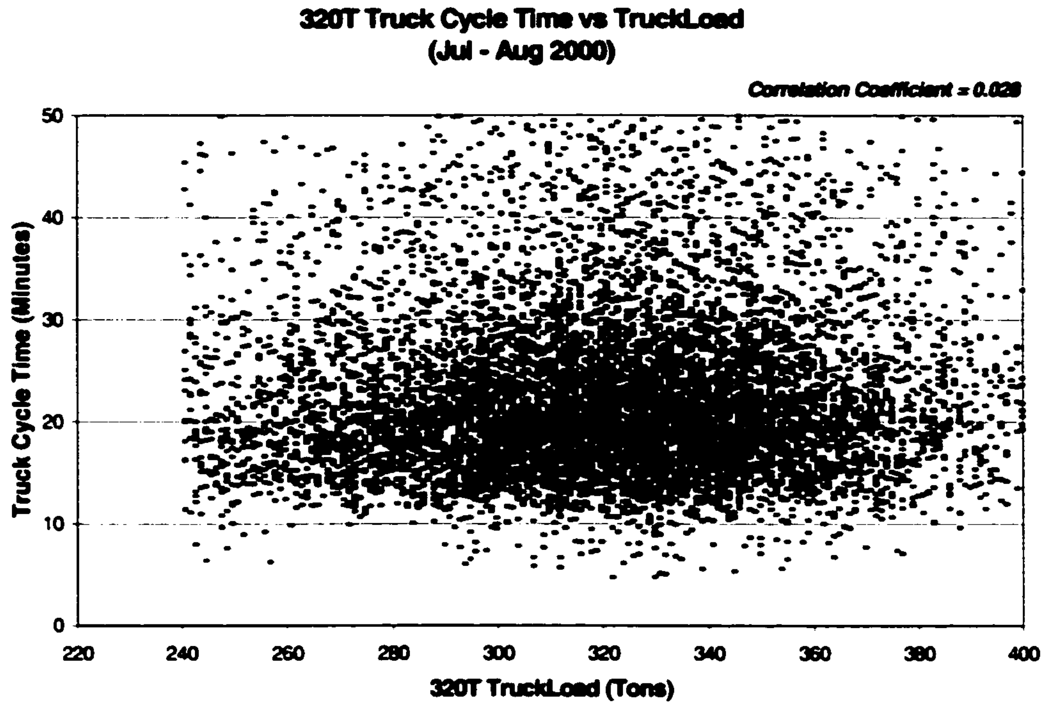


Figure H 2 – Truckload and Cycle Time Correlation (320T)

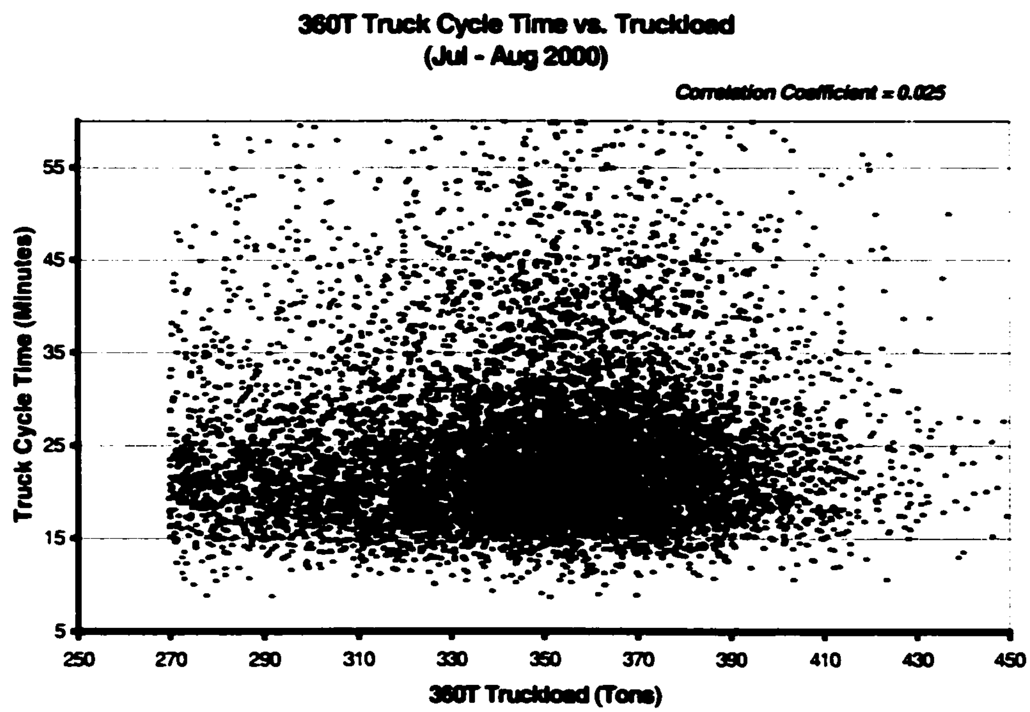


Figure H 3 – Truckload and Cycle Time Correlation (360T)