THE UNIVERSITY OF MANITOBA

A SLAM II COMPUTER SIMULATION MODEL OF UNDERGROUND PRODUCTION MINING

by

Alan B. Litke

A Thesis
Submitted to the Faculty of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree of Master of Science

DEPARTMENT OF MECHANICAL AND INDUSTRIAL ENGINEERING

Winnipeg, Manitoba

ISBN 0-315-92252-4

A SLAM II COMPUTER SIMULATION MODEL OF UNDERGROUND PRODUCTION MINING

BY

ALAN B. LITKE

A Thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

ABSTRACT

This work presents the development of a prototype SLAM II computer simulation model of underground production mining based upon a specific mining complex found at Inco Limited, Manitoba Division. Mining processes included in the model consist of production drilling and blasting, mucking and hauling, train tramming, and backfilling. Specific work shifts and chimney capacities are fixed at the start of the first simulation run, while activity times, hauling and tramming capacities, and equipment uptimes and downtimes are randomly sampled from their respective distributions.

Logic diagrams and time charts are used as communication tools to present some of the operational characteristics of the model. As a case study, the model was used to determine activity and system bottlenecks, as well as, the expected average daily production rate based on four typical mining blocks found at the complex.

Inputs to the model are easily changed to examine various other mining blocks and types of system configurations. Model outputs include the averages and confidence intervals for various availability ratios, production rates, activity times, as well as, a production schedule table. In addition, similar information is available on a per run basis, along with the daily and cumulative production rates, as well as, preliminary cost information.

The results from the study showed that computer simulation is an excellent tool for evaluating complex systems such as underground production mining.

ACKNOWLEDGMENTS

I would like to take this opportunity to express my thanks to several individuals for their various contributions throughout my studies. First, my thanks to Mines Research of Inco Limited, Manitoba Division, for their financial contribution and commitment to this project. I am very grateful for the technical guidance, support and camaraderie provided by Donald Laroche.

Next, my sincere thanks to my advisor, Dr. Doug Strong, for his advice, guidance and encouragement throughout my studies. I feel very fortunate to have been able to participate in this project, and without Dr. Strong, in all likelihood this project would not have materialized.

I would also like to thank various other individuals at Inco for their cooperation in this project, in particular the people in the Birchtree Mine Engineering and Mines Research groups, who always provided me with information and explanations when I needed it.

Finally, to my wife Daryl, I thank you for your patience and understanding over our long distance separations during my studies. I also thank you for the support and encouragement you have given me in completing my work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

This thesis is based upon research work performed for the Manitoba Division of Inco Limited, which is located approximately 750 kilometers north of Winnipeg in Thompson, Manitoba.

In general, Inco Limited is an international corporation whose primary business is nickel; however, it is also a substantial producer of copper, cobalt, sulphur products and gold. In the Primary Metals unit of the company, production operations are located in Sudbury, Ontario, and Thompson, as well as in Indonesia. In the Alloys unit of the corporation, nickel alloys are produced in the United Kingdom, United States and Canada. In addition, Inco sales offices are located around the world. In total, Inco has approximately 19,000 employees worldwide: 12,000 employed in Canada, of which more than 2,000 are with the Manitoba Division in Thompson.

The Manitoba Division origins date back as early as 1946; however, it was not until 1956 that Inco's exploration teams discovered an orebody that justified construction of a nickel mine. Within five years, a mining and processing complex was developed, along with a new community now known as Thompson (in honor of the Inco Chairman John F. Thompson). By 1961 the Manitoba Division was shipping pure electrolytic nickel to markets worldwide.

Today, mining within the Manitoba Division takes place in both open pit and underground mines. Mining of the Thompson Open Pit is nearly completed, whereas, underground mining operations have reported ore reserves that should last another twenty years. Presently, underground mining is performed at Thompson Mine located adjacent to the city of Thompson, as well as Birchtree Mine which is located five kilometers southwest of the city.

Ore delivered from the Thompson and Birchtree Mines is processed through the Mill, Smelter, then Refinery to produce 99.9 percent electrolytic nickel in the form of slabs, as well as, small button-shaped ROUNDS (a trademark of the Inco family of companies). In addition to nickel, the Manitoba Division produces cobalt oxide powder which is shipped directly to markets worldwide, as well as, copper and precious metals which are sent to other Inco locations for further processing.

The research work presented in this thesis focuses on the underground production mining operations, in particular, those found at an orebody within the Birchtree Mine called the "83 Orebody".

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO COMPUTER SIMULATION

Computer simulation, a tool which has been around for approximately thirty years, has traditionally been viewed as a "method of last resort". In recent years, however, it has gained in popularity due to the increased use of personal computers and accompanying software programs, as well as, the increased size and complexity of system problems. As these trends continue, the expanded use of computer simulation is sure to follow.

What is computer simulation? Pritsker (1986) defines computer simulation as the process of designing a mathematical-logical model of a real system, and experimenting with this model on a computer. Thus, simulation is both the model building process, as well as, the design and implementation of an appropriate experiment. It should also be noted that the term "real" system, as it is used above, refers to both existing and non-existing systems.

Depending on the problem being studied, there can be many different dimensions to a computer simulation model. Three dimensions as provided by Law and Kelton (1991) are:

1. *Static vs. Dynamic Simulation Models:* A static simulation model is a representation of a system at a particular point in time, or one that may be used to represent a system

in which time plays no role. A dynamic simulation model, on the other hand, represents a system as it evolves over time.

2. *Deterministic vs. Stochastic Simulation Models:* A deterministic simulation model does not contain any probabilistic or random components. A stochastic simulation model, on the other hand, has random components.

3. *Continuous vs. Discrete Simulation Models:* A continuous simulation model is one in which the state variables (i.e., variables that describe the state of the system) change continuously with respect to time. A discrete simulation model is one in which the state variables change instantaneously at separate points in time.

Potential uses of computer simulation models are numerous. From a global perspective, according to Pritsker (1986), simulation models can be employed at four levels:

1. As *explanatory devices* to define a system or problem,
2. As *analysis vehicles* to determine critical elements, components and issues,
3. As *design assessors* to synthesize and evaluate proposed solutions, and
4. As *predictors* to forecast and aid in planning future developments.

In the past, computer simulation has been applied in many areas of industry and service. For example, Law and Kelton (1991) note simulation has been useful in designing and analyzing manufacturing systems; evaluating hardware and software requirements for a computer system; evaluating new military weapon systems or tactics; determining ordering policies for an inventory system; designing communications systems; designing operating transportation facilities such as freeways, airports, subways, or ports; evaluating designs for service organizations such as hospitals, post offices, or fast-food restaurants; and analyzing financial or economic systems. In the future, computer simulation will continue to be used in these areas, as well as others.

Perhaps the *flexibility* of computer simulation as a problem solving tool is one of the reasons for its wide use. In addition to flexibility, other advantages of using simulation are:

- *Realism.* Simulation models can be realistic since they can capture the actual characteristics of the system being modelled (Schriber, 1991).
- *Nonexistent systems.* The system whose behavior is to be investigated using simulation need not actually exist (Schriber, 1991).
- *Experimental control.* In simulation, all variables can be held constant except the one whose influence is being studied (Schriber, 1991).
- *Time compression or expansion.* Simulation can be used to study a system with a long time frame in compressed time, or alternatively to study the detailed workings of a system in expanded time (Law and Kelton, 1991).
- *Comparing alternative system configurations.* Alternative proposed system designs, or alternative operating policies for a single system, can be compared using simulation to see which best meets a specified requirement (Law and Kelton, 1991).
- *Only available tool.* Most complex, real-world systems with stochastic elements cannot be accurately described by a mathematical model that is evaluated analytically, thus, simulation is often the only type of investigation possible (Law and Kelton, 1991).

Conversely, some disadvantages of using simulation are:

- *Failure to produce exact results* (Schriber, 1991). Each run of a stochastic simulation model produces only estimates of a model's true characteristics for a particular set of input parameters. Thus, several independent runs of the model are usually required for each set of input parameters to be studied (Law and Kelton, 1991).
- *Cost for providing a simulation capability* (Schriber, 1991). Simulation models are often expensive and time-consuming to develop (Law and Kelton, 1991).

- *Misuse of simulation* (Schriber, 1991). The large volume of numbers produced by a simulation study, or the persuasive impact of a realistic animation, often creates a tendency to place greater confidence in a study's results than is justified. If a model is not a "valid" representation of a system under study, the simulation results will provide little useful information about the actual system (Law and Kelton, 1991).
- *Long lead times.* A simulation study cannot be conducted over a weekend. Months of effort can be required to gather data; build, verify, and validate models; design experiments, and evaluate and interpret the results (Schriber, 1991).

As a final note to this general introduction to computer simulation, six major steps in performing a simulation study are now given (based upon material presented in Pritsker, Sigal and Hammesfahr, 1989):

1. *Formulate the problem.* This involves understanding the problem context, identifying project goals, specifying system performance measures, setting specific modelling objectives, and defining the system to be modelled.
2. *Specify the model.* This pertains to conceptualizing the model. The crucial questions in model specification focus on what simplifying assumptions are reasonable to make, what components should be included in the model, and what interactions occur among the components.
3. *Build the model.* There are three substeps to building the model: develop the simulation model, collect data, and define experimental controls. Develop the simulation model is concerned with building or programming the model. Data collection may involve performing detailed time studies, getting information from equipment manufactures, and talking to system operators. In defining experimental controls, information such as beginning and ending times, number of runs, report types, initial values for status variables are specified. All three substeps to building the model are performed concurrently.

4. *Simulate the model.* There are three substeps to simulate the model: run (or execute) the model, verify the model, and validate the model. Running the model involves executing the simulation program using a dry run only. The verification substep consists of determining whether or not the simulation model behaves as intended. The validation substep seeks to establish that the simulation model is a reasonable representation of the system.
5. *Use the model.* The use of the model involves performing production runs, and the interpretation and presentation of the outputs from these runs.
6. *Support decision making.* The final step in the modelling and simulation process is to support decision making.

1.2 SCOPE OF RESEARCH

The main objective of the research work presented in this thesis was to develop a computer simulation model of underground production mining operations found at the 83 orebody of the Birchtree Mine, Inco Limited, Manitoba Division. A dynamic stochastic discrete computer simulation model was constructed to simulate the working mine as it was, and then used to answer various "what if?" questions of interest. A general description of the mining system modelled, as shown in Figure 1.1, is now presented.

The simulation model was built assuming all necessary orebody infrastructure was developed. That is, the four chimneys (three orepasses and one rockpass), drifts, ramps, and other mine complex elements between the 1700 and 2100 foot levels, inclusive, were already in place. Within the existing orebody infrastructure, the major activities and pieces of equipment required for underground production mining are modelled. As shown in Figure 1.1, the major activities modelled have been divided into four categories:

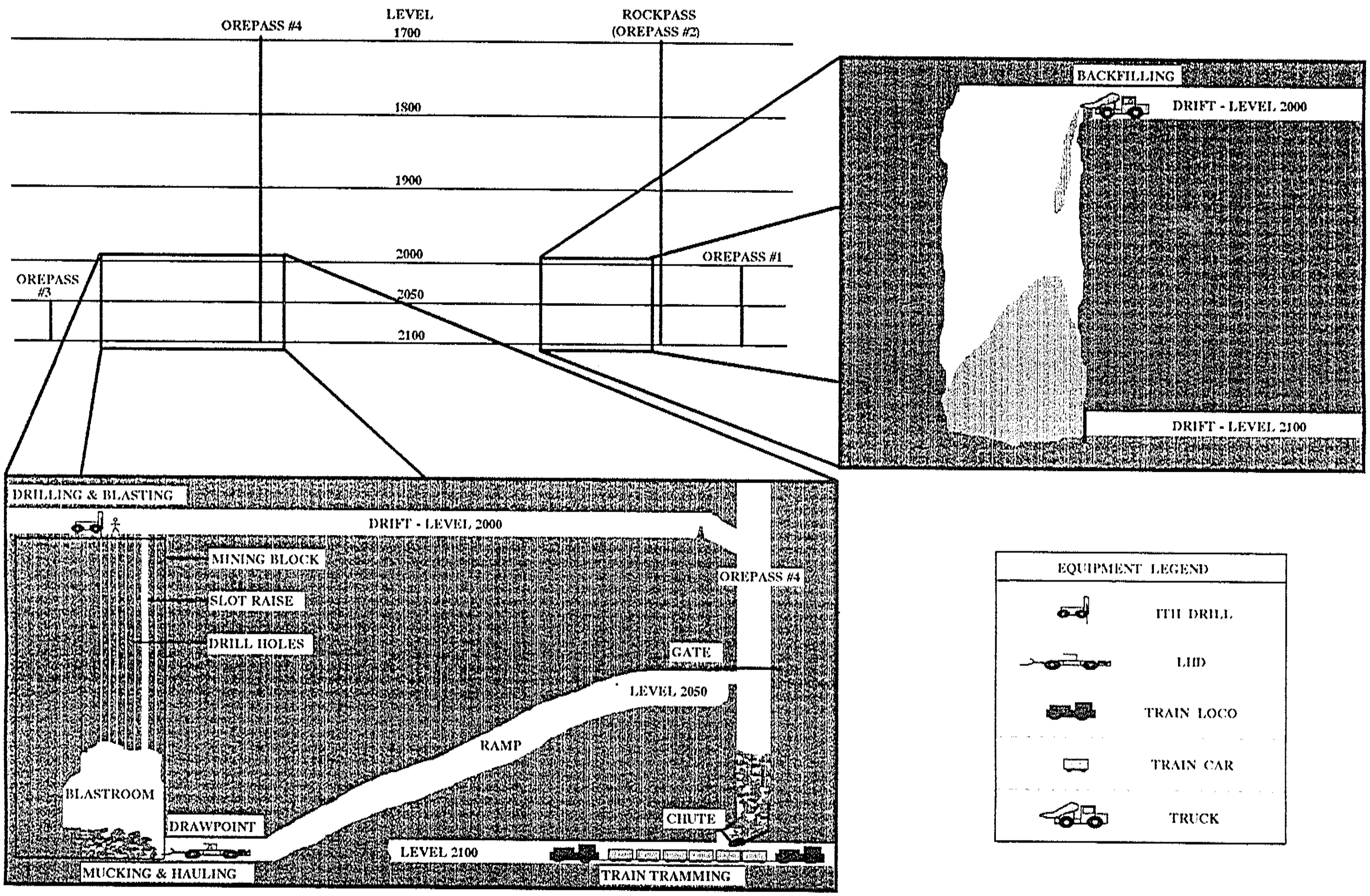


Figure 1.1: Pictorial representation of the mining system modelled.

1. Drilling and blasting, which includes relocating the drill, drilling holes, cleaning holes, measuring holes, loading explosives and guarding the blast,
2. Mucking and hauling,
3. Train tramming, and
4. Backfilling.

The drilling method used at the 83 orebody is called Vertical Block Mining (VBM). Basically, in the VBM method, holes are drilled using an in-the-hole (ITH) drill from the top of a section of a mining block within the orebody, to a depth from 30 to 110 feet. Following drilling, the holes are then cleaned, measured, and loaded with explosives. Near the end of the work shift, the area is secured, or guarded, before the explosives are detonated.

Upon detonation, the blast section of ore, that is, the bottom section or sometimes side section of the mining block, is fragmented and falls to an area called the drawpoint. At the drawpoint the broken ore is loaded, or mucked, by a front-end loader commonly referred to as a scooptram or Load-Haul-Dump (LHD) vehicle. Once loaded, the bucket of ore is then hauled with the LHD to an appropriate orepass. Ore from the orepass is then loaded onto a train and transported, or trammed, to another area underground where it is dumped into a holding area, or dump station, at the shaft. The ore is then crushed and skipped to the surface (crushing and skipping are not included in the model).

The above process of drilling and blasting, mucking and hauling and train tramming is repeated until the entire mining block has been mined. Once this has occurred, the cavity is backfilled with an appropriate material. Further details on the system modelled are provided later in this thesis.

1.3 THESIS ORGANIZATION

Chapter 2 begins with a review of the literature focusing on underground mining simulation studies. Then, a brief discussion on computer simulation model building languages is presented, followed by a comparison between general-purpose languages and simulation languages. Finally, reasons for selecting SLAM II as the simulation language for this project are given.

Chapter 3 presents an introduction to the SLAM II simulation language. This chapter focuses on the following: the SLAM II network symbols used, the methods of linking user written FORTRAN code employed, and the SLAM II control statements used in the underground production mining simulation model. A simplified open pit mining system is modelled using SLAM II for demonstration purposes.

Chapter 4 focuses on the underground production mining simulation model. The chapter begins with a detailed description of the system, followed by a discussion on the SLAM II model logic used. Next, model verification and validation are discussed, followed by model inputs and outputs, and finally the number of runs and run length.

Chapter 5 uses the underground production mining simulation model on four typical mining blocks found at the 83 orebody of the Birchtree Mine. Time charts from the basecase scenario are used to demonstrate some of the operational characteristics of the model. Next, results from various scenarios are presented to determine activity and system bottlenecks, as well as, the expected average daily production. Finally, other scenarios that were investigated with an earlier version of the model are discussed, such as, alternative LHD configurations, increasing the average daily production rate, and alternative mining block mucking rules.

Chapter 6 presents concluding remarks on the research work accomplished and directions for future work.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

The work presented in this thesis is a prototype model of a particular mining complex, and does not build upon previously published material. Consequently, the literature review has been conducted only to determine whether other similar or related studies have been published. Specifically, the search focuses on relevant publications on underground mining simulation studies. Papers from this search have been divided into three categories:

1. Similar Studies,
2. Other Related Studies, and
3. Coal Studies.

The papers within the Similar Studies category are critiqued since they are similar to the work presented in this thesis. On the other hand, only synopses are provided for the papers within the Other Related Studies and Coal Studies categories since their similarities to the work presented in this thesis are limited. The papers within these categories (i.e., Other Related Studies and Coal Studies) have been included only to exemplify some of the simulation projects that have been performed in underground mining. Furthermore, if the literature review was limited to publications that are only similar to the work presented in this thesis, then very few papers would have been included in this review, as will become obvious.

Following the review of the various publications is a brief discussion on simulation model building languages. A comparison is then made between general-purpose languages and simulation languages, followed by reasons for selecting the SLAM II simulation language for this project.

2.2 UNDERGROUND MINING SIMULATION STUDIES

2.2.1 Similar Studies

Touwen and Joughin (1973) discuss two independent FORTRAN IV programs designed to simulate the principal underground production processes in gold mining. One program is used to simulate the stoping (stope) system, while a second program is used to simulate the transport system. The stoping model simulates the rockbreaking (i.e., drilling and blasting operations), cleaning (i.e., face and gully scraping operations), and support operations for one stope at a time. The transport model simulates the functions and movements of all vehicles running on rails on one underground level. Both programs consider random generation for the duration of an operation, the probability of failure or interruption while the operation is in progress, the duration of the interruption, and the outcome of the operation. A shortcoming of these programs is the dynamic interaction between the stoping and transport models are limited. For example, waiting within each system (i.e., either the stoping or transportation system) adversely affects the other system that can only be observed by one coherent program. Feeding the outputs of the stoping program as inputs to the transportation program does not properly model the dynamic interaction of the two systems. Whether or not the model was used at a specific mine is not mentioned; however, acknowledgment is given to several mines for their assistance.

Potter, Richardson, Hoare and Koenig (1988) discuss the use of two SLAM II simulation programs developed to study stoping methods at Z.C. Mines located at Broken Hill, New South Wales, Australia. One model is used to simulate the longhole open stoping method, and another to simulate the loader cut and fill stoping method for particular stopes.

The longhole open stoping model, as noted above, first considers development of the stope (i.e., slot drilling, ring drilling and slot blasting/removal), then loader size blocks of ore (i.e., buckets) are generated to simulate the removal of ore by load-haul-dump vehicles. Another section of the model is used to simulate the orepass operations, as well as the trucking or tramming operations from the orepass to the chute. A major shortcoming of this model is it fails to simulate the fragmentation module (i.e., production drilling and blasting), and goes from development of the slot raise to generation of buckets of ore.

The loader cut and fill stoping model, as noted above, considers each slice mined one at a time. The model considers some mine development, as well as block drilling, blasting, securing ground, multiple loader selection, loader operation and tramming, multiple face operations, rockbolting, cable dowelling (bolting) and stope filling. The loader cut and fill method of mining is not similar to the one modelled in this thesis; therefore, further comment is reserved.

Hoare and Willis (1992) present a case study of animated computer simulation of underground mining at a lead/zinc mine in Elura, New South Wales, Australia. The initial underground mining model was built using SLAM II, and consisted of six concurrently producing stopes, two concurrent level development areas, an orepass system, a crusher system, a hoisting system, as well as other associated components. The SLAM II model was subsequently translated to run under SIMAN, and then animated using the CINEMA software. Unfortunately, details of the model are not provided within the publication.

Litke, Laroche, Strong and Szymanski (1993) discuss a SLAM II simulation model of underground production mining based upon current methods employed at an existing mine at Inco Limited, Manitoba Division. Mining processes included in the model consist of production drilling and blasting, mucking and hauling, train tramming and backfilling. Specific work shifts and orepass/rockpass capacities are fixed within the model, while activity times, and equipment uptimes and downtimes are randomly sampled from their respective distributions. The model produces various levels of output reports and can simulate up to ten mining blocks. This publication is based upon some of the work presented in this thesis.

2.2.2 Other Related Studies

Suboleski and Lucas (1969), as reported by Martin and Sabuda (1982), developed a simulation program for underground room and pillar face mining. The room and pillar mining system contains three major components: the loading machine, the loading and hauling vehicle, and the support operations. The loading machine, a continuous miner or a shovel, loads ore onto loading and hauling vehicles. The loading and hauling vehicles, which consist of up to three cable-reel or eight self-powered cars or trucks, remove the ore from the mine. Support operations include blasting, drilling, undercutting and roof bolting. Various outputs from the model are discussed. It is reported that the program has been used by various companies for evaluation of mining systems.

Hayashi and Robinson (1981), as reported by Martin and Sabuda (1982), developed a simulation model of a single track underground mine rail haulage system with waiting areas to allow trains to pass. Major elements of the model include loading of the mine cars, train movement through a rail network, dumping of the mine cars, and train assignment. Various levels of outputs from the model are presented. Whether or not the model was used at a specific mine is not mentioned.

Jardine and Evans (1988) present a mine scheduling software package called MINESTAR, a system developed at Mincom USA Inc., which can be used for mine scheduling on a range of mines - coal and metalliferous, open pit and underground. They view mine scheduling as the process of simulating the extraction of a deposit over time. The process comprises of defining the deposit as a group of mining blocks, establishing rates of removal and the sequence in which blocks are extracted, simulating the extraction sequence, and reporting the results in a schedule. The major features of MINESTAR reported are: a flexible mining block database, dynamic removal rate simulations, interactive graphics, user tailorable reports and graphs, plus several other features. Details on the workings of this package, in particular the dynamic removal rate simulations, are not provided.

Sturgul and Singhal (1988) present arguments in favor of using the GPSS simulation language for modelling mining operations. Four examples of GPSS mining simulation studies in Australia are briefly described within the paper for illustrative purposes. The first example is on a small coal mine that used simulation in justifying the purchase of a dispatching system for a two shovel - three dump location complex. The second example is on a medium sized surface mine that used simulation to predict the height of a dam and pump size for tailings from the mill. A third example is on a major utility company that used simulation in considering the size of bunkers for storage of coal that would be needed for a larger power plant. The fourth example deals with the problem of determining the optimum number of spare parts to have for a mining vehicle.

Bruno, Massacci and Raspa (1989) present a simulation model of an underground rock salt mine in Sicily, Italy. The model is based on room and pillar mining methods which are performed on several levels. The demand for three types of salt is also included in the model. Results from the specific application are provided, along with a general outline of the program.

Lavrencic (1989) presents an underground and open pit mine simulator called MINSIM. MINSIM is a general model comprised of a network of working places which are connected with transport paths of various qualities and slopes. Working groups are used to represent individuals or crews that use the equipment units to perform specialized operations. The general structure of MINSIM, which is an event driven system, is discussed. Whether or not the simulator was used at a specific mine is not mentioned.

Szymanski, Laroche, Murchie and Maynard (1992) present two computer simulation models developed for Inco Limited, Manitoba Division. One model, written in Pascal, is used to simulate the skipping system and includes hoisting, crushing, pass capacity, waste rock and ore production from various areas, and tramming. A second model, built using the SLAM II simulation language, was used to simulate the cage-hoisting system, which includes the transportation of men and materials between the surface and various levels underground. Results from both models are presented for the specific application at Inco.

2.2.3 Coal Studies

Lee and Mutmansky (1985) present a simulation model of a longwall mining system for underground coal mining. Basically, the longwall mining system is composed of the shearer subsystem, the face conveyor subsystem, the roof support subsystem, and the external conveyor subsystem. The shearer subsystem extracts coal from the face and drops it onto a face conveyor, which in turn carries the coal to the end of the face onto a belt conveyor. The roof support subsystem provides temporary support over the immediate face area created by the extraction of the coal. The external conveyor subsystem transports the coal out of the mine. The program structure of the various subsystems are discussed in some detail. Whether or not the model was used at a specific mine is not mentioned.

Topuz and Nasuf (1985) present a FORTRAN 77 computer simulation program for continuous mining systems used in underground coal mining. The model, called CONSIM, considers the following events: cutting and loading, haulage, roof-bolting, equipment breakdown, completion of cut, and end of job. The general program structure is discussed along with various input screens and output results. Whether or not the model was used at a specific mine is not mentioned.

Topuz, Nasuf and Michalopoulos (1985) present a FORTRAN 77 program for longwall underground coal mining. The longwall mining system was divided up into four main events: coal extraction (i.e., cutting the coal from the face, loading the coal onto the face conveyor, and conveying the broken coal from the face), support advancement (i.e., supporting newly exposed roofs and controlling exposed roofs behind the advancing supports), equipment failures, and end of shift or end of a simulation run. The results from a case study on a coal mine in West Virginia are presented.

Zaikang (1985) presents a simulation model that is mainly concerned with the transportation system from longwall faces to surface bunkers in underground coal mining in China. The model consists of various subroutines representing working areas, the locomotive train, the main belt haulage conveyor, the bunkers, and the skipping system. The general program outline is discussed along with output results from the specific application. The model has been used at several mines in China, and some results from these studies are provided.

Kelly (1986) discusses the use of two computer simulations programs in analyzing the haulage requirements at the Loveridge Mine, located near Fairmont, West Virginia.

The first program, a modified version of the Virginia Polytechnical Institute's rail simulation program, was used to simulate approximately half of the production from the track haulage system. A second program, written in FORTRAN, was used to simulate the remaining production from the belt system. Results from various scenarios are presented and discussed for the specific case study.

Nan and Wilke (1986) present a FORTRAN IV simulation model for the analysis of underground coal mine production systems in China. The general structure of the computer program is discussed, which includes input/output subroutines, the face production subsystem, the rail haulage subsystem, the belt conveyor subsystem, and other subroutines for auxiliary mathematical operations such as generating normally distributed random numbers and renewing the time schedule. The model was first used to analyze a rail haulage system at an existing mine located in Shanxi Province of China. Later, the model was used by Yue, Xilan and Yongzun (1987) to analyze a belt transportation system at an existing mine also located in Shanxi Province of China.

Chatterjee, Johnston and Holguin (1987) present a FORTRAN 77 simulation program for computer-aided underground coal mine planning (i.e., layout, scheduling and cost evaluation), incorporating existing geological information. The user can design a layout and perform a deterministic simulation to calculate the production rate for various extraction strategies. The economic evaluations of different layouts by the simulation are used to select the optimum layout. The program has been used at a mine in Ohio.

Sturgal and Harrison (1987) outline reasons for using a special, non-procedural programming language for simulation studies of coal mines rather than a procedural language such as FORTRAN. In particular, benefits of using the GPSS simulation language are presented, and several theoretical examples as related to mining are discussed.

Zaikang (1987) presents a simulation model of a belt conveyor system for an underground coal mine in China. The program, which was written using ALGOL, is used to design the capacities of the main belt conveyor in conjunction with the bunker capacities. Specific results using the model at a mine in eastern China are discussed, along with some of the general program logic.

Sinha, Pal and Mitra (1988) discuss the use of two FORTRAN IV simulation models developed to study underground coal mining. The first model is used to analyze a conveyor network, along with existing bunkers. The second model is used to evaluate additional bunkers as proposed by the first model. Thus, the two models are used collectively to evaluate the functioning of the underground evacuation network. Both models are described along with the results from an underground mining complex study in India.

Edwards, Mort and Hollands (1992) discuss the use of animation in the simulation of an underground coal mine transportation system using the SIMAN/CINEMA package. The transportation system model includes extraction from several faces by means of coal cutting shearers and an assortment of interconnected underground belt conveyors which carry the coal from the faces to several bunkers. The coal is then conveyed from the bunkers to the single shaft. A hierarchical structure is used in the animation, allowing for three different levels of viewing: the top level which provides an overview of the entire mine; the intermediate level which provides viewing of one subsystem of the mine; and a detailed level that concentrates on one coal face. Samples of the various animation levels are provided along with other model outputs.

2.3 SIMULATION MODEL BUILDING LANGUAGES

2.3.1 Classification of Languages

One of the most important decisions a modeler must make in performing a simulation study is the choice of a language (Law and Kelton, 1991). Basically, there are four classes of simulation model building languages:

1. Spreadsheets,
2. Simulators,
3. Simulation languages, and
4. General-purpose languages.

Spreadsheets, such as LOTUS 1-2-3, QUATTRO PRO and Excel, can all be used as a simulation model building language; however, their use in such a capacity is limited. In particular, they can be used for static simulations; however, they cannot be used for dynamic simulations. In addition, common simulation functions, such as queuing and resource allocation, cannot be performed. It is possible, however, to perform stochastic simulations using spreadsheets that have a random number generator, such as the RAND() function in Excel.

Simulators are computer packages that allow one to simulate a system contained in a specific class of systems with little or no programming (Law and Kelton, 1991). The model must be within the framework of the software, as it is very difficult to model beyond the scope provided (Banks, 1991). Examples of simulators include SIMFACTORY II.5, ProModelPC and AutoMod (Banks, 1992).

Simulation languages are computer packages that are general in nature but may have special features for certain types of applications (Law and Kelton, 1991). A model is developed in a simulation language by writing a program using the language's modelling constructs (Law and Kelton, 1991). Examples of simulation languages include GPSS, SIMAN, SIMSCRIPT II.5 and SLAM II.

General-purpose languages such as FORTRAN, C, Pascal and BASIC can also be used as a simulation model building language. The next section compares general-purpose languages to simulation languages, since these are the most common classes of simulation model building languages used in the past.

2.3.2 General-Purpose Languages Vs. Simulation Languages

Both general-purpose and simulation languages have features that are advantageous to use in building simulation models. According to Law and Kelton (1991), some advantages of building a simulation model using a general-purpose language, as compared to a simulation language, include:

- Most modelers already know a general-purpose language, but this is often not the case with a simulation language.
- General-purpose languages, such as FORTRAN and BASIC, are available on virtually every computer, but a particular simulation language may not be accessible on the computer the analyst wants to use.
- An efficiently written FORTRAN or C program may require less execution time than a corresponding program written in a simulation language.
- General-purpose languages may allow greater programming flexibility than certain simulation languages.
- Software cost may be lower.

On the other hand, according to Law and Kelton (1991), some advantages of building a simulation model using a simulation language, as compared to a general-purpose language, include:

- Simulation languages automatically provide most of the features needed in programming a simulation model which results in a significant decrease in programming time.
- Simulation languages provide a natural framework for modelling.
- Simulation models are generally easier to change when written in a simulation language.
- Most simulation languages provide dynamic storage allocation during execution.
- Simulation languages provide better error detection since many potential types of errors have been identified and are checked for automatically.

In closing, Law and Kelton (1991) conclude that there are clear advantages to using both types of languages; however, they believe that a modeler would be prudent to give serious consideration to using a simulation language.

2.3.3 Language Selection

Of the classes available, it was determined that a simulation language was the most appropriate for the underground mining simulation project presented in this thesis. Firstly, a spreadsheet was not chosen since they do not allow for dynamic simulations, nor can they perform common simulation functions. Secondly, a simulator was not chosen due to their lack of flexibility, nor was one available. Thirdly, a general-purpose language was not chosen since they lack many features that are available in a simulation language. Specifically, various tasks, such as keeping track of time, event scheduling, and random number generation, are easily performed within a simulation language, as compared to writing general-purpose code.

Once the decision to use a simulation language was made, then the next decision was to select a suitable simulation language. By the process of elimination, potential simulation language candidates were identified and contacted for relevant literature. Upon examination of the literature it was determined that SLAM II was a suitable language for this project for the following reasons:

- Since there are no simulation languages specifically designed for mining, a general-purpose simulation language, such as SLAM II, would have to be used.
- SLAM II is easily linked with user written FORTRAN code, providing a considerable degree of flexibility.
- SLAM II was readily available at both Inco and the University of Manitoba.

CHAPTER 3

INTRODUCTION TO SLAM II

3.1 OVERVIEW

SLAM II (Simulation Language for Alternative Modelling), initially released in 1981, is a general-purpose FORTRAN-based simulation language that can be used to build discrete, continuous, or combined discrete-continuous simulation models (see Pritsker, 1986). Discrete simulation occurs when the dependent variables of the model change discretely at specified points in simulated time. Continuous simulation occurs when the dependent variables of the model change continuously over simulated time (as defined by equations and the incremental step size, or time increment). Discrete-continuous simulation occurs when the dependent variables of the model may change both discretely and continuously.

A SLAM II simulation model normally begins with a flow diagram, or network, which graphically portrays the flow of entities (e.g., machines, parts, people, etc.) through the system (see O'Reilly and Ryan, 1992). The network is basically made up of a standard set of symbols called nodes and branches. Nodes are used for such functions as entering or exiting the system, seizing or freeing a resource, changing variable values, collecting statistics, and starting or stopping entity flow based on system conditions (O'Reilly, 1991).

Branches are used to connect the nodes and define the routing of entities through the system, as well as to model activities. Time delays on activities can be used to represent processing times, travel times, or waiting times (O'Reilly, 1991). Entities which proceed from node to node over activities may have unique characteristics, called attributes, which can be used to control their processing (O'Reilly and Ryan, 1992).

When aspects of a model cannot be easily represented within the network, then user written FORTRAN or C inserts can be used. From the user written code, the programmer may also call upon a library of subprograms which perform common simulation functions, including accessing network operations and altering its status (O'Reilly and Ryan, 1992).

Once the network has been constructed, it is then translated into SLAM II program statements, in conjunction with control statements, which are used to execute the SLAM II simulation program. When model construction is performed within SLAMSYSTEM, translation of the network is performed automatically. SLAMSYSTEM, introduced in 1988, is an integrated simulation system based on the Microsoft Windows interface (under MS-DOS) or the OS/2 Presentation Manager (O'Reilly and Ryan, 1992). Basically, SLAMSYSTEM provides an environment, with pull-down menus and dialog boxes, that facilitates model development using the SLAM II language.

3.2 SLAM II NETWORK

3.2.1 Introduction

This section provides a general description of the SLAM II network symbols used in the underground simulation model described in the next chapter. (For details on symbols other than those provided see Pritsker, 1986.) To begin, general descriptions are provided for the CREATE, TERMINATE, GOON, AWAIT, PREEMPT, FREE, ALTER, ASSIGN, BATCH and EVENT network nodes. Next, the three types of network branching used (i.e., deterministic, probabilistic and conditional upon system variables) are described. Finally, a description of the network RESOURCE block is provided.

3.2.2 Network Nodes Used

CREATE Node

The CREATE node, as shown in Figure 3.1, is a method for creating entities for arrival or insertion into the network (see Pritsker, 1986). The value of TF specifies the time for the first entity to be created. The variable TBC is used to specify the time between creations of entities after the first. Entities are created until the value of MC, the maximum number of creations for the node, is reached. The time at which the entity is created can be assigned to an attribute number specified by MA. The maximum number of branches that can be selected by the entity is specified by the value of M.

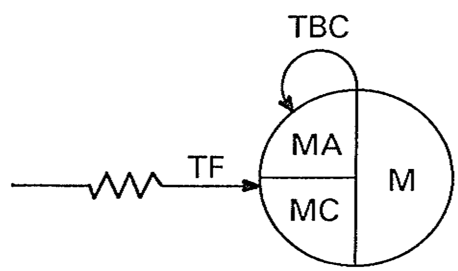


Figure 3.1: The CREATE node.

TERMINATE Node

The TERMINATE node, as shown in Figure 3.2, is used to destroy or delete entities from the network (see Pritsker, 1986). It can also be used to specify the number of entities to be processed on a simulation run, referred to as the termination count or TC value.

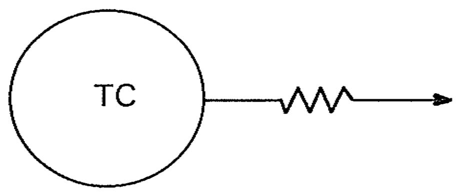


Figure 3.2: The TERMINATE node.

GOON Node

The GOON node, as shown in Figure 3.3, is a continue type node and is used in modelling sequential activities (see Pritsker, 1986). As before, the value of M specifies the maximum number of branches that can be selected by the entity.

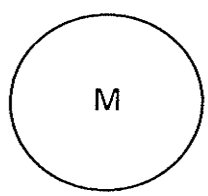


Figure 3.3: The GOON node.

AWAIT Node

The AWAIT node, as shown in Figure 3.4, is used to store entities waiting for UR units of resource RES (or waiting for gate GATE to open - see Pritsker, 1986). When an entity arrives to an AWAIT node, and the required units of the resource are available (or the GATE is opened), then the entity passes directly through the node and is routed according to the M number. If the required units of the resource are not available then the entity must wait at the node and is placed in file IFL in accordance with the priority assigned to that file. The maximum number of entities that can reside at the node is specified by the value of QC.

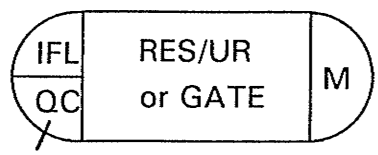


Figure 3.4: The AWAIT node.

PREEMPT Node

The PREEMPT node, as shown in Figure 3.5, is a type of AWAIT node in which the entity can preempt one unit of a resource that has been allocated to some other entity (see Pritsker, 1986). Preemption will always be attempted if the entity using the resource came from an AWAIT node, or if the priority assigned to the PREEMPT node is greater than the priority of the PREEMPT node from which the entity currently using the resource came. As before, IFL identifies the file number for waiting entities at the node in accordance with the priority assigned to that file, and RES identifies the resource requested at the PREEMPT node. The priority for the PREEMPT node, PR, can either be specified as LOW(K) or HIGH(K) where K is an attribute number, and is used to determine if the incoming entity will attempt to preempt another entity which has the required resource. A preempted entity is routed to a node as specified by the send node label SNLBL. Attribute NATR can be used to store the time remaining to process the entity when it is preempted. If a send node label is not specified, then the preempted entity is routed to the AWAIT or PREEMPT node at which it was allocated the resource.

At that node, it is established as the first entity waiting for the resource. When the resource is reassigned to the preempted entity, its remaining processing time will be used. As before, the maximum number of branches that can be selected by the entity is specified by the value of M.

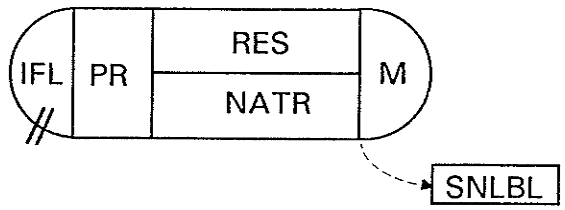


Figure 3.5: The PREEMPT node.

FREE Node

The FREE node, as shown in Figure 3.6, is used to release a resource when an entity arrives to the node (see Pritsker, 1986). That is, every entity arriving to a FREE node releases UF units of resource type RES before being routed in accordance with the M number.

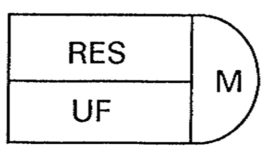


Figure 3.6: The FREE node.

ALTER Node

The ALTER node, as shown in Figure 3.7, is used to change the capacity of resource type RES by CC units (see Pritsker, 1986). If CC is positive, the number of available units is increased. Conversely, if CC is negative, the capacity is decreased. As previously stated, the maximum number of branches that can be selected by the entity is specified by the M value.

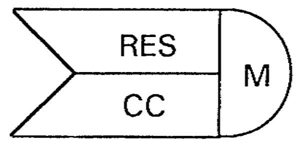


Figure 3.7: The ALTER node.

ASSIGN Node

The ASSIGN node, as shown in Figure 3.8, is used to assign values to either the attributes of an entity passing through the node or to system variables that pertain to the network in general (see Pritsker, 1986). As before, the maximum number of branches that can be selected by the entity is specified by M.

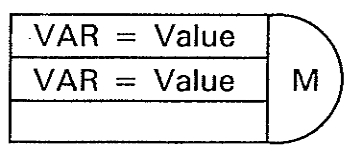


Figure 3.8: The ASSIGN node.

BATCH Node

The BATCH node, as shown in Figure 3.9, is used to combine entities to a specified threshold level and then to release a single entity which represents the batch (see Pritsker, 1986). NBATCH is the total number of batches that can be accumulated concurrently at the BATCH node. NATRB is the attribute number that is common for all entities in the batch. NATRS is the number of the attribute which contains the value to be summed and tested against the threshold value THRESH. When this sum is greater than or equal to THRESH, a batched entity is formed and released from the BATCH node. If NATRS is not specified, then THRESH defines the number of entities to be included in a batch. SAVE is used to specify the criterion for defining the attributes of the batched entity, which are: FIRST entity included in the batch; LAST entity included in the batch; the entity with the lowest value of attribute I, LOW(I); and the entity with the highest value of attribute I, HIGH(I). RETAIN indicates whether the individual entities included in the batch should be maintained for future use. The M value, as previously defined, specifies the maximum number of branches that can be selected by the entity.

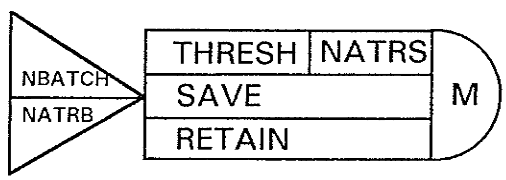


Figure 3.9: The BATCH node.

EVENT Node

The EVENT node, as shown in Figure 3.10, is used to interface the network portion of a model with event code written by the modeler (see Pritsker, 1986). The EVENT node causes subroutine EVENT(JEVNT) to be called every time an entity arrives to the node, where the value of JEVNT specifies the event code to be executed. As before, M specifies the maximum number of emanating activities to be taken by the entity.

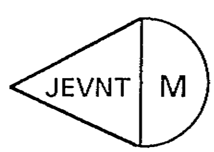


Figure 3.10: The EVENT node.

3.2.3 Network Branching Used

Branches are used to connect the nodes and define the routing of entities through the system, as well as to model activities. Basically, SLAM II has two types of activities available: REGULAR and SERVICE (see Pritsker, 1986). A REGULAR activity is any activity emanating from a node other than a QUEUE or SELECT node. A SERVICE activity, on the other hand, is any activity emanating from a QUEUE or SELECT node. Since only the REGULAR activity was used in the simulation model discussed in Chapter 4, it is the only one described here.

The REGULAR activity is used to delay entities and perform probabilistic or conditional testing and branching. The symbol for a branch representing a REGULAR activity is shown in Figure 3.11. The activity number is given by the value of A, while DUR specifies the duration for the activity. Either a probability specification for selecting the activity, PROB, or a condition for selecting the activity, COND, can be specified. NLBL can be used as a GOTO function within the network when a branch does not connect to a node.

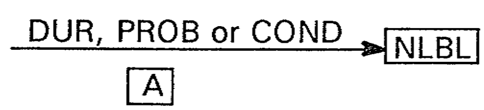


Figure 3.11: The REGULAR activity.

In addition to probabilistic and conditional branching, deterministic branching was also used in the model. Deterministic branching causes an entity to be duplicated and routed over a number of branches as specified by the M value associated with the node.

3.2.4 RESOURCE Block

The RESOURCE block, as shown in Figure 3.12, is used to specify the following: the resource name or label, RLBL; the initial resource capacity, that is, the initial number of resource units available, CAP; and the order in which files (IFLs) associated with AWAIT and PREEMPT nodes are to be polled to allocate freed units of the resource to entities (see Pritsker, 1986). The user may also specify the resource number, RNUM, or allow SLAM II to assign a numeric code to each resource. The RESOURCE block has no inputs or outputs as entities do not flow through it, consequently it is referred to as a block rather than a node.

RNUM	RLBL	CAP	IFL1	IFL2	. . .

Figure 3.12: The RESOURCE block.

3.3 LINKING USER WRITTEN FORTRAN CODE

When aspects of a model cannot be easily represented within the network, then user written inserts can be used. In the simulation model described in Chapter 4, four methods of linking user written FORTRAN code were used. One method, which has already been described above, uses the EVENT node to interface the network portion of the model with user written FORTRAN event code.

A second method uses the function USERF(IFN), where the argument IFN is the user function number. In the model described in the next chapter, USERF(IFN) was inserted in the DUR field for many REGULAR activities within the network in order to interface the network portion of the model with user written FORTRAN code.

A third method uses the SLAM II subroutine INTLC, which is called by the SLAM II processor at the beginning of each simulation run. INTLC, which is written by the user, can be used for start of run processing, such as, setting initial conditions, reading a data file, and prompting the user for information.

A fourth method uses the SLAM II subroutine OTPUT, which is called by the SLAM II processor at the end of each simulation run. OTPUT, which is written by the user, can be used for end of run processing, such as updating arrays, and printing results.

3.4 SLAM II CONTROL STATEMENTS USED

In addition to the SLAM II network , control statements are required to execute a SLAM II simulation program. The following SLAM II control statements were used in the simulation model described in Chapter 4: GEN, LIMITS, INTLC, ARRAY, NETWORK, PRIORITY, INITIALIZE, and FIN. For a detailed explanation, as well as a complete listing of all SLAM II control statements, refer to Pritsker (1986). A brief description of each statement used is provided below:

- The GEN statement provides general information about a simulation run or runs, such as the analyst's name, the project's name, the date, the number of simulation runs, etc.
- The LIMITS statement is used to specify limits on the largest file number used, the largest number of attributes per entity, and the maximum number of entities expected in all files in the model at one time.
- The INTLC statement is used to assign initial values to SLAM II variables.
- The ARRAY statement is used to define one row of a user defined global table.
- The NETWORK statement is used to denote the beginning of a network description.
- The PRIORITY statement is used to specify the criterion for ranking entities within a file.

- The INITIALIZE statement is used to specify the beginning time and the ending time for a simulation, as well as, initialization options for clearing statistics, initializing variables, and initializing files.
- The FIN statement denotes the end of all SLAM II input statements.

3.5 A SLAM II OPEN PIT MINING SYSTEM EXAMPLE

A simplified open pit mining system is modelled in this section to exemplify the SLAM II model building process. The system, as depicted in Figure 3.13, consists of a shovel, a crusher and four trucks. Loading of the trucks with the shovel is assumed to be normally distributed with a mean of 6 and a standard deviation of 2 minutes. Truck travel time full from the shovel to the crusher is assumed to be normally distributed with a mean of 12 and a standard deviation of 4 minutes. Unloading or dumping of the trucks at the crusher is assumed to be normally distributed with a mean of 3 and a standard deviation of 1 minute. Truck travel time empty from the crusher to the shovel is assumed to be normally distributed with a mean of 9 and a standard deviation of 3 minutes.

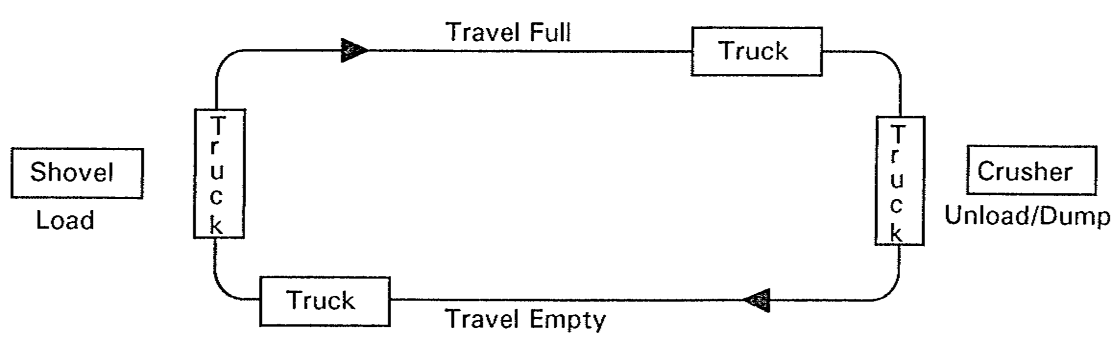


Figure 3.13: Simplified open pit mining system.

The SLAM II control statements for the above system is shown in Figure 3.14, and the network in Figure 3.15. As shown in Figure 3.15, both the shovel and the crusher are modelled as resources using the RESOURCE block. The trucks, on the other hand, are modelled as entities using the CREATE node. A brief explanation of the network diagram is now provided. For explanations on specific network symbols not provided here, the reader can refer to earlier sections in this chapter, or see Pritsker (1986).

Following creation at the CREATE node, the first truck entity proceeds to the AWAIT node labelled START, at which it seizes the SHOVEL resource and proceeds to the activity labelled Load. Similarly, following creation, the remaining truck entities proceed to the AWAIT node labelled START; however, since the SHOVEL resource is not available they must wait in the queue (i.e., file 1) until each is first in line and the SHOVEL resource is available.

When the simulated time for the load activity for the first truck entity has elapsed, it proceeds to the FREE node where the SHOVEL resource is freed. Since the SHOVEL resource is now free, the second truck entity is removed from file 1 and proceeds to seize the SHOVEL resource, and then begins the activity labelled Load. Immediately thereafter, the first truck entity then proceeds to the activity labelled Travel Full.

Following the elapse of the travel full activity time for the first truck entity, it arrives at the AWAIT node for the CRUSHER resource, at which it seizes the resource and proceeds to the activity labelled Unload/Dump. (This assumes that random generation of activity times for the truck entities are such that the first truck entity completes the travel full activity before the other truck entities.) Following the unload/dump activity, the first truck entity frees the CRUSHER resource, proceeds through the activity labelled Travel Empty, then is routed back to the AWAIT node labelled START.

The truck entities are looped, similar to above, until the end time of the simulation run is reached (i.e., 120 simulated minutes, as defined in the control statement

INITIALIZE, has passed). Various statistics are automatically gathered which are available in the Summary Report at the end of the simulation run. For the purpose of this example, the Summary Report is not discussed here.

```
GEN,LITKE,OPEN PIT MINING,12/17/93,1,Y,Y,Y/Y,Y,Y/1,72;
LIMITS,2,1,3;
NETWORK;
INITIALIZE,,120,Y;
FIN;
```

Figure 3.14: SLAM II control statements for the simplified open pit mining system.

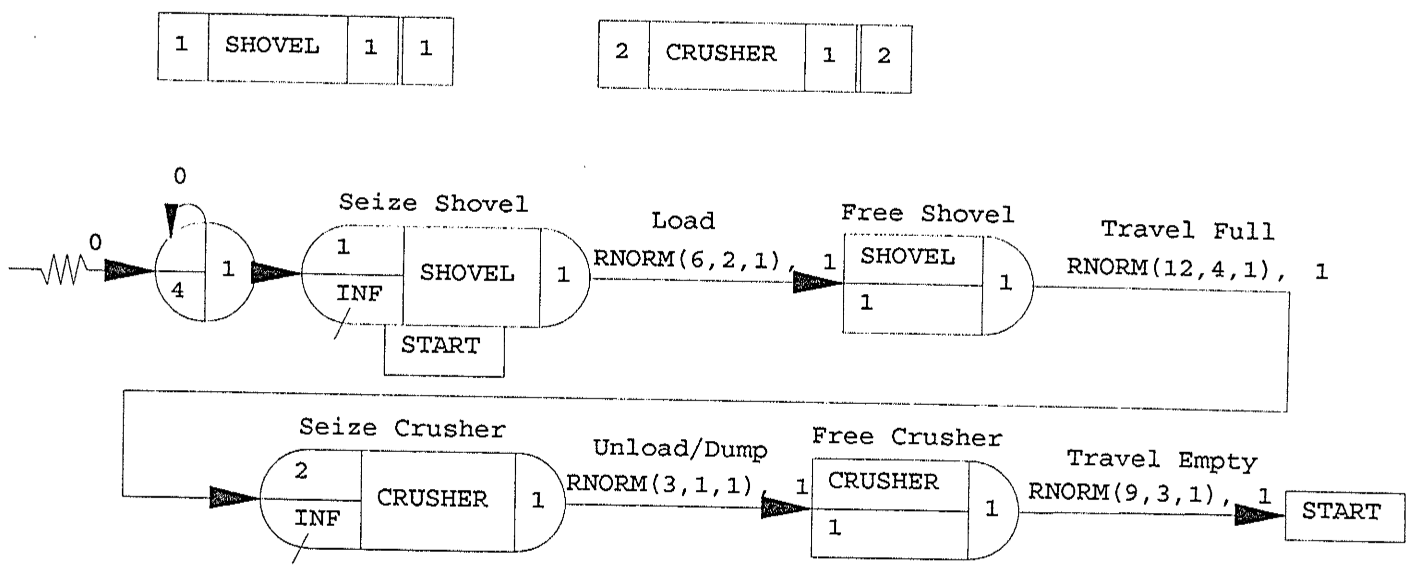


Figure 3.15: SLAM II network for the simplified open pit mining system.

CHAPTER 4

UNDERGROUND PRODUCTION MINING SIMULATION MODEL

4.1 SYSTEM DESCRIPTION

A pictorial representation of the mining system modelled is given in Figure 1.1. As previously mentioned, the underground production mining simulation model was built assuming all necessary orebody infrastructure was already developed. That is, the four chimneys (three orepasses and one rockpass), drifts, ramps and other mine complex elements between the 1700 and 2100 foot levels, inclusive, were already in place. Within the existing orebody infrastructure, the major activities and pieces of equipment required for underground production mining are modelled. Basically, the major activities modelled have been divided into four categories:

1. Drilling and blasting, which includes relocating the drill, drilling holes, cleaning holes, measuring holes, loading explosives, and guarding the blast,
2. Mucking and hauling,
3. Train tramming, and
4. Backfilling.

As previously mentioned, the drilling and blasting method used at the 83 orebody is called Vertical Block Mining (VBM). Basically, in the VBM method, holes are drilled using an in-the-hole (ITH) drill from the top of a section of the mining block within the orebody, to a depth from 30 to 110 feet. Following drilling, the holes are then cleaned, measured, and loaded with explosives. Near the end of the work shift, the area is secured, or guarded, before the explosives are detonated. Upon detonation, the blast section of ore, that is, the bottom section or sometimes side section of the mining block, is fragmented and falls to an area called the drawpoint. At the drawpoint the broken ore is loaded, or mucked, with the Load-Haul-Dump (LHD) vehicle, and hauled to an appropriate orepass. Ore from the orepass is then loaded onto a train and trammed to the dump station at the shaft.

The above process of drilling and blasting, mucking and hauling, and train tramming is repeated until the entire block has been mined. Once this has occurred, the cavity is backfilled with an appropriate material. Depending on the block within the mining complex, this material may be cemented rockfill (i.e., a combination of rock and cement), or just rock itself.

The major pieces of equipment used to perform the above activities include:

- Two in-the-hole (ITH) drills for drilling,
- Two load-haul-dump (LHD) vehicles for mucking and hauling,
- One train with six ore cars for tramming ore and ten rock cars for tramming development rock, and
- One truck for backfilling.

In addition to the above, some important system constraints that had to be incorporated into the simulation model include:

- All major pieces of equipment are subject to random failures dependent upon the number of hours of operation (i.e., the summation of the indirect and direct production times as defined in Appendix C).
- All activity times, hauling speeds, LHD bucket capacities, train car capacities, and the truck capacity, are randomly sampled from their respective distributions.
- Work shifts are scheduled as follows: three shifts per day for drilling and blasting; two shifts per day for mucking and hauling; three shifts per day for train tramming (two of which are used exclusively for tramming ore, and the third shift is used primarily for tramming rock, however, when there is no rock to be trammed on the third shift then ore is trammed); and two shifts per day for backfilling.
- The amount of ore within each orepass is a function of the ore dumped into the orepass, the orepass capacity at the dumping level, and the amount of ore removed by the ore train. When the orepass is full for a particular level, then mucking and hauling is discontinued until room is made available by removing ore from the orepass and loading it onto the train.
- The amount of development rock within the rockpass is a function of the rock dumped into the rockpass, and the amount of rock removed by the rock train.
- The ITH drills, which are moved from mining block to mining block according to the planned sequence, cannot be relocated from a block until the last blast section of ore within the mining block has been blasted. In addition, drilling can occur at two separate locations at the same time since there are two ITH drills available.
- Due to the varying mining block geometry and blast section of ore tonnages, the number of holes drilled, the number of feet drilled, and the number of feet loaded with explosives vary from blast section of ore to blast section of ore within each mining block.

- When a blast is detonated it requires a certain amount of room for expansion (referred to within this thesis as the blastroom), consequently a certain percentage of broken ore from a blast section of ore must be mucked out before the sequential blast section of ore can be blasted for a particular mining block.
- A primary LHD is used for mucking and hauling, with a secondary LHD for backup purposes. The secondary LHD is used only when the primary is not available.
- The LHD hauling distances to and from the orepasses vary from mining block to mining block.
- The train tramming distances to and from the chimneys (i.e., the orepasses and the rockpass) to the dump station at the shaft vary for each chimney.
- The truck hauling distances from the rockfill loading site to the blocks vary for each mining block.

Further details, assumptions made, equations and rates on the above described system are given in Appendix D. Briefly, the activity time equations used in the model are based upon Inco's incentive standards hourly rates, or incentive rates, adjusted as deemed appropriate. The equipment uptime and downtime equations used in the model are based upon Inco's past records for the equipment modelled, also adjusted as deemed appropriate. Additional model inputs used are discussed later in this chapter.

4.2 SLAM II MODEL LOGIC

4.2.1 Introduction

As previously noted, the underground production mining process was modelled using the SLAM II simulation language within the SLAMSYSTEM environment. Model construction was primarily performed within the graphical network and linked with user written FORTRAN subroutines and functions as necessary.

Basically, the graphical network was used for sequencing major activities. The complete network is given in Appendix A.1, for which logic diagrams and explanations are provided later in this chapter.

The main entity used within the network is a blast section of ore which passes through the drilling and blasting network, proceeds through the mucking and hauling network, then through the backfilling network, and finally terminated to control the run length of the simulation. In addition, other entities are used for the train tramming, the time of day blasts can occur, the equipment uptime and downtime scheduling, and the shift scheduling of equipment/workers, as well as other logic.

Resources and activities, as noted in Figure 4.1, were used as necessary in order to achieve the desired logic and sequencing. For all major activities (i.e., drilling and blasting, mucking and hauling, train tramming, and backfilling) the equipment and crew were modelled as one resource since both are required for the respective activities to be performed, except for the drilling and blasting activity. For the drilling and blasting activity, the crew and the drill are still modelled as one resource; however, the direct and indirect production times for the drill, as defined in Appendix C, were collected based upon whether the drill was used or not. For example, when the crew uses the drill for cleaning holes, then the direct and indirect production times for the drill are incremented accordingly. On the other hand, when the crew does not use the drill for cleaning holes, then the direct and indirect production times for the drill are not incremented, rather the time would become part of the stand-by time for the drill (see Appendix C for further details).

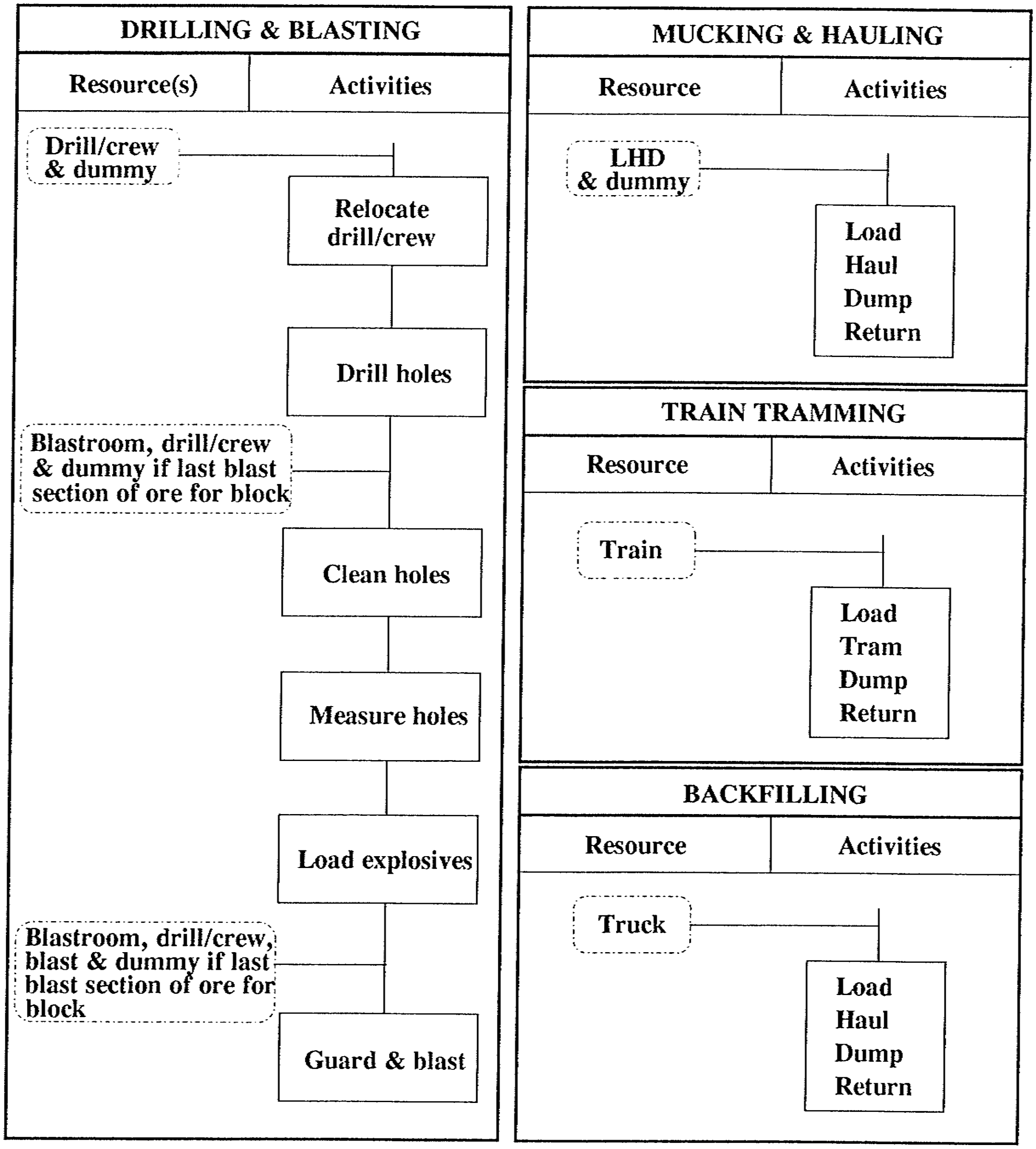


Figure 4.1: Simplified resource and activity block diagram.

FORTRAN subroutines and functions were primarily used for the following reasons: to read input data and initialize variables; to perform complex calculations; to perform logic decisions that could not be easily done within the network; to gather simulation results; and to print output reports. The complete FORTRAN code for the simulation model is given in Appendix A.6.

The model can accommodate up to four chimneys, three of which must be defined as orepasses and one as a rockpass. The existing chimneys are shown by the filled rectangles in Figure 4.2; however, the model has been built to allow for expansions of chimney's #1 and #3 from levels 2000 and 2050 respectively, if required. The I.D. numbers (#10 through #29) represent dump locations for either ore or rock, and are required within the program to distinguish between the various chimneys and levels.

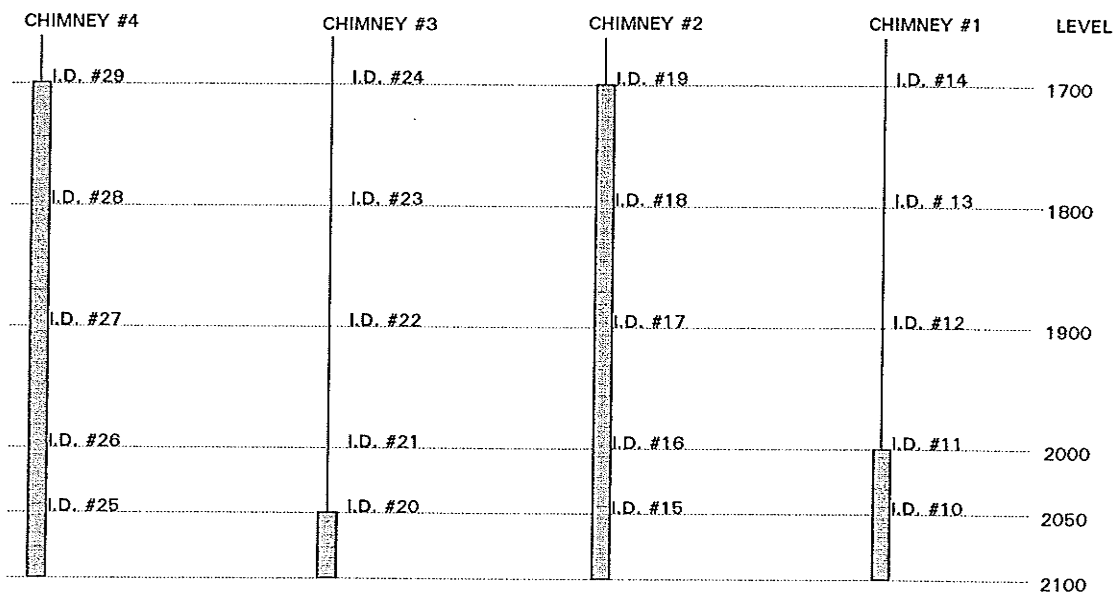


Figure 4.2: Chimneys and dump location I.D. numbers.

Priority levels for various resources and their respective activities are shown in Table 4.1. The smaller the priority level, the higher the priority of the activity. For example, an activity that has a priority level of 2 can be preempted by an activity with a priority level of 1. Activities with the same priority level cannot preempt each other.

Table 4.1 Priority levels for various resources and activities.

Resource	Priority Level	Activity
Drills/crews:		
	1	Two hour shift delay
	2	Drill breakdown
	3	Clean, measure & load holes, & guard & blast
	4	Relocate drill/crew & drill holes
LHDs:		
	1	Two hour shift delay & shift shutdown
	2	LHD breakdown
	2	Blast delay & check vents
	3	Mucking & hauling
	3	Wait for orepass room
Train:		
	1	Two hour shift delay
	2	Train breakdown
	3	Train tramming
	3	Wait for ore to tram
Truck:		
	1	Two hour shift delay & shift shutdown
	2	Truck breakdown
	2	Blast delay & check vents
	3	Hauling rockfill

The next several sections describe the logic used in the SLAM II underground production mining model given in Appendix A. For this purpose, the SLAM II network has been divided into the following segments: drilling and blasting, mucking and hauling, backfilling, train tramming, blast delay and check vents, equipment resource breakdowns, shifts, and miscellaneous and resource segments. The drilling and blasting segment contains a detailed description of the SLAM II symbols used in order to introduce interested readers to the overall program structure (i.e., how the network is used in conjunction with the FORTRAN code). In addition, the drilling and blasting segment also includes a more general logic description that makes use of a logic diagram. For the remaining segments, only general logic descriptions and diagrams are provided. It may be helpful to view these logic diagrams in conjunction with the SLAM II network given in Appendix A.1 and the FORTRAN code given in Appendix A.6.

4.2.2 Drilling and Blasting Segment

Detailed Network Description

The SLAM II drilling and blasting network, as shown in Figure 4.3, is now described in detail. To begin, 23 entities, each representing a specific blast section of ore, are created at the CREATE node labelled START. The time of the first creation, and the time between creations, have both been set to 0. A maximum of 1 emanating activity can be taken for each release of the CREATE node.

Following the CREATE node, an EVENT node is used to interface the network with event FORTRAN code 1 as shown in Appendix A.6.4 (i.e., SUBROUTINE EVENT, CASE (1)). Each time an entity arrives at this EVENT node, the FORTRAN code for CASE (1) is executed, which is used to initialize attribute values 1 through 15 for each

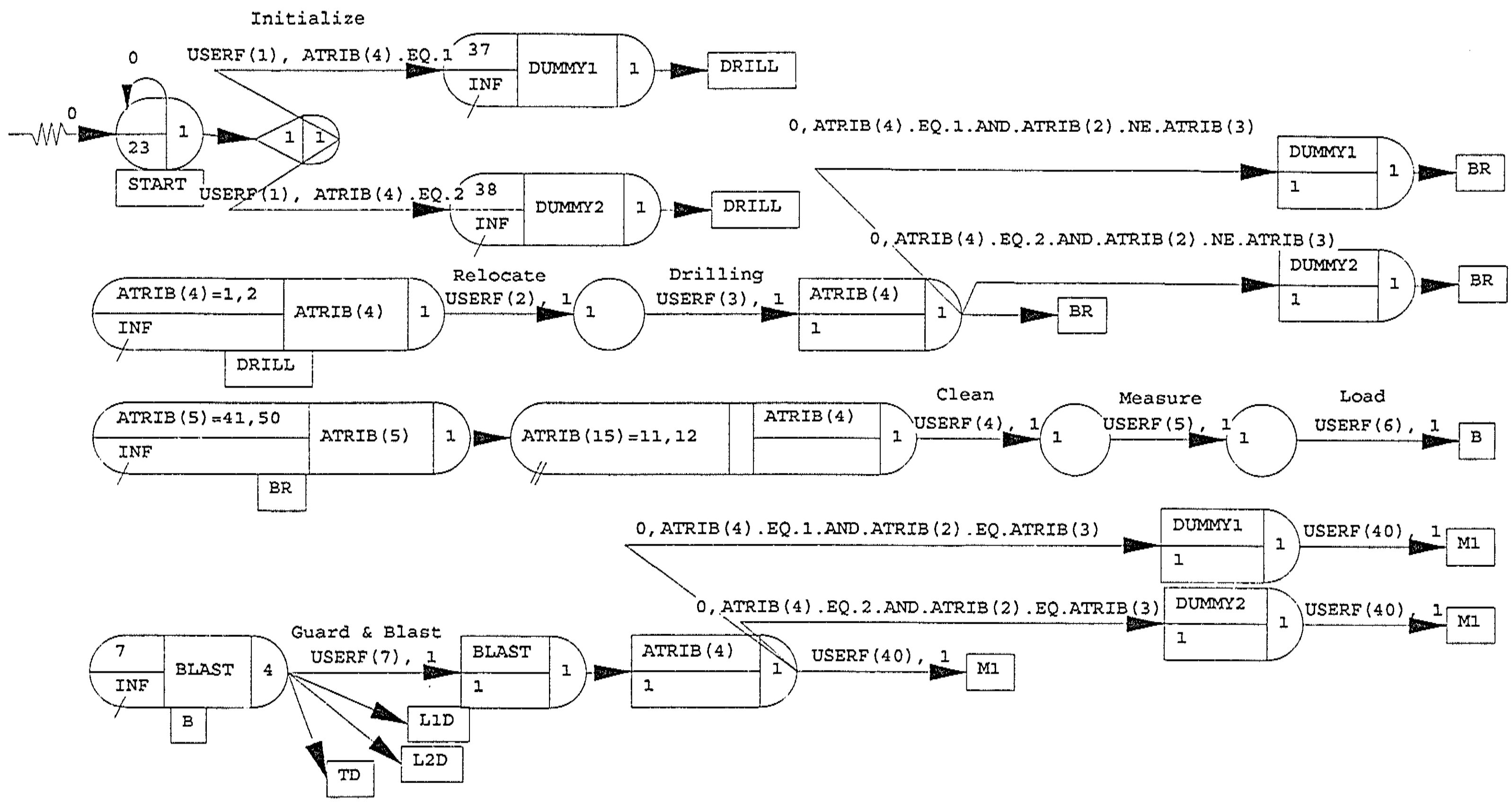


Figure 4.3: SLAM II drilling and blasting network.

entity. The attribute values are characteristics specific for each blast section of ore, and are explained in detail in Appendix A.4.

Following the EVENT node, conditional branching is used to route the entity through the appropriate activity. That is, if attribute 4 of the entity is equal to 1, then the USERF(1) FORTRAN code is executed (see Appendix A.6.3) before the entity proceeds to the AWAIT node for the resource DUMMY1. If attribute 4 of the entity is equal to 2, then USERF(1) FORTRAN code is still executed; however, the entity proceeds to the AWAIT node for the resource DUMMY2.

Assuming that the first arc is taken and the entity arrives at the AWAIT node for resource DUMMY1, then if the resource is available, it seizes the resource and proceeds. Otherwise, the entity waits in file 37 until the resource DUMMY1 is available. The queue capacity for this AWAIT node is assumed to be infinite (INF), and a maximum of 1 emanating activity can be taken by the entity. A similar explanation can be used for the AWAIT node for resource DUMMY2. The reason for using these dummy resources is explained later in this section.

Upon seizing the appropriate dummy resource, the entity then proceeds to the AWAIT node labelled DRILL. Here, attribute 4 is used to distinguish which drill/crew resource is required by the entity. If the required drill/crew resource is available when the entity arrives at the AWAIT node, then it seizes the appropriate resource and proceeds. Otherwise, the entity waits in the appropriate file as specified by attribute 4. The queue capacity is assumed to be infinite, and a maximum of 1 emanating activity can be taken by the entity.

Following the AWAIT node labelled DRILL, the entity proceeds to the activity labelled Relocate, where the FORTRAN code USERF(2) is executed. After relocating the drill activity, the entity proceeds through the GOON node to the next activity labelled

Drilling, where USERF(3) is executed. Next, the entity proceeds to the FREE node and the appropriate drill/crew resource, as specified by attribute 4, is freed.

Following the above FREE node, conditional branching is used to determine which dummy resource, if any, is freed. If attribute 4 is equal to 1, and attribute 2 is not equal to attribute 3, then DUMMY1 is freed. If attribute 4 is equal to 2, and attribute 2 is not equal to attribute 3, then DUMMY2 is freed. If neither of these conditions are met then the entity proceeds without freeing a dummy resource. In other words, if the blast section of ore is not the last one for a particular mining block, then the appropriate dummy resource can be freed to allow drilling to continue on the sequential blast section of ore within the mining block. However, if the blast section of ore is the last one for a particular mining block, then the appropriate dummy resource is not freed since the drill/crew resource should not be relocated until all of the drilling and blasting activity for the entire mining block has been completed.

Next, the entity proceeds to the blastroom resource AWAIT node labelled BR. If the blastroom resource is available, as defined by attribute 5, then the entity proceeds; otherwise, the entity waits in the appropriate file as specified by attribute 5. The blastroom resource is made available once 25 percent of the previous blast section of ore for the mining block has been mucked and hauled by the LHD. Freeing the blastroom resource is performed in the mucking and hauling network which is described in the next section.

Following the AWAIT node labelled BR, the entity proceeds to the PREEMPT node where the appropriate drill/crew resource, as specified by attribute 4, is preempted. If preemption is not accomplished, then the entity waits in the appropriate file, as specified by attribute 15, until the drill/crew resource becomes available. As previously mentioned, the drill/crew resource was freed to allow drilling to continue on the sequential blast section of ore for a particular mining block; however, as soon as 25 percent of the appropriate blast section of ore has been mucked, then the cleaning, measuring and loading activities can be performed on the sequential blast section of ore. That is, following preemption, the entity proceeds and the activity labelled Clean is performed by executing the FORTRAN code in USERF(4). Next, the entity proceeds to the activity labelled Measure and USERF(5) is executed, followed by the activity labelled Load and USERF(6) being executed.

The entity then proceeds to the blast resource AWAIT node labelled B. If the simulated time is within 1.5 hours of the end of the shift, then the resource BLAST is available and the entity can proceed to the Guard and Blast activity; otherwise, the entity waits in file 7. Following this AWAIT node, the entity is split into four identical entities and routed accordingly. One entity is routed to the activity Guard & Blast where the FORTRAN code USERF(7) is executed. A second entity is routed to the node labelled L1D, where it is used to delay mucking and hauling for the blast if LHD1 (i.e., the 6 cubic yard LHD) is being used. A third entity is routed to a node labelled L2D, where it is used to delay mucking and hauling for the blast if LHD2 (i.e., the 9 cubic yard LHD) is being used. A forth entity is routed to a node labelled TD, where it is used to delay the truck for the blast if it is being used.

Following the Guard and Blast activity, the BLAST resource is freed, then the appropriate drill/crew resource is freed, as specified by attribute 4. Next, conditional branching is used to free the appropriate dummy resource if required. That is, for the last blast section of ore for a particular mining block, the appropriate dummy resource is freed. For blast sections of ore other than the last one for the mining block, the dummy resource has already been freed earlier within the network. The entity then proceeds to a node labelled M1, which represents the start of the mucking and hauling network.

General Logic Description

A simplified logic diagram for the drilling and blasting segment is shown in Figure 4.4. To begin, each entity is initialized with attribute values particular to a specific blast section of ore. Next, the entity must seize the required dummy resource before it can proceed to seize the required drill/crew resource.

Once the appropriate drill/crew resource has been seized, the relocate drill/crew resource activity is performed. The holes are then drilled for the blast section of ore before the drill/crew resource is freed. If the entity is not the last blast section of ore for the mining block, then the appropriate dummy resource is freed; otherwise it is not. Freeing the dummy resource allows the relocate and drilling activities to be performed on the sequential blast section of ore within the same mining block. On the other hand, not freeing the dummy resource for the last blast section of ore for the mining block does not allow the sequential blast section of ore, which is the first blast section of ore for the next mining block, to begin its relocate and drilling activities.

Next, the blastroom resource is seized when available. The blastroom resource for the mining block is made available in the mucking and hauling network when 25 percent of the broken ore from the previous blast section of ore has been mucked. Consequently, when the blastroom resource is not available, drilling is continued on sequential blast sections of ore for a particular mining block. However, when the blastroom resource becomes available the drill/crew resource is preempted (i.e., drilling is discontinued on sequential blast sections of ore for a particular mining block), and the holes of the current blast section of ore are cleaned, measured, then loaded with explosives.

The blast resource, which is made available for the last 1.5 hours of each shift, is seized before the guard and blast activity can be performed. At the same time, the entity is split in order to delay the LHDs and the truck resources for the blast. Following the guard and blast activity, the blast and drill/crew resources are freed. If the entity is the last blast

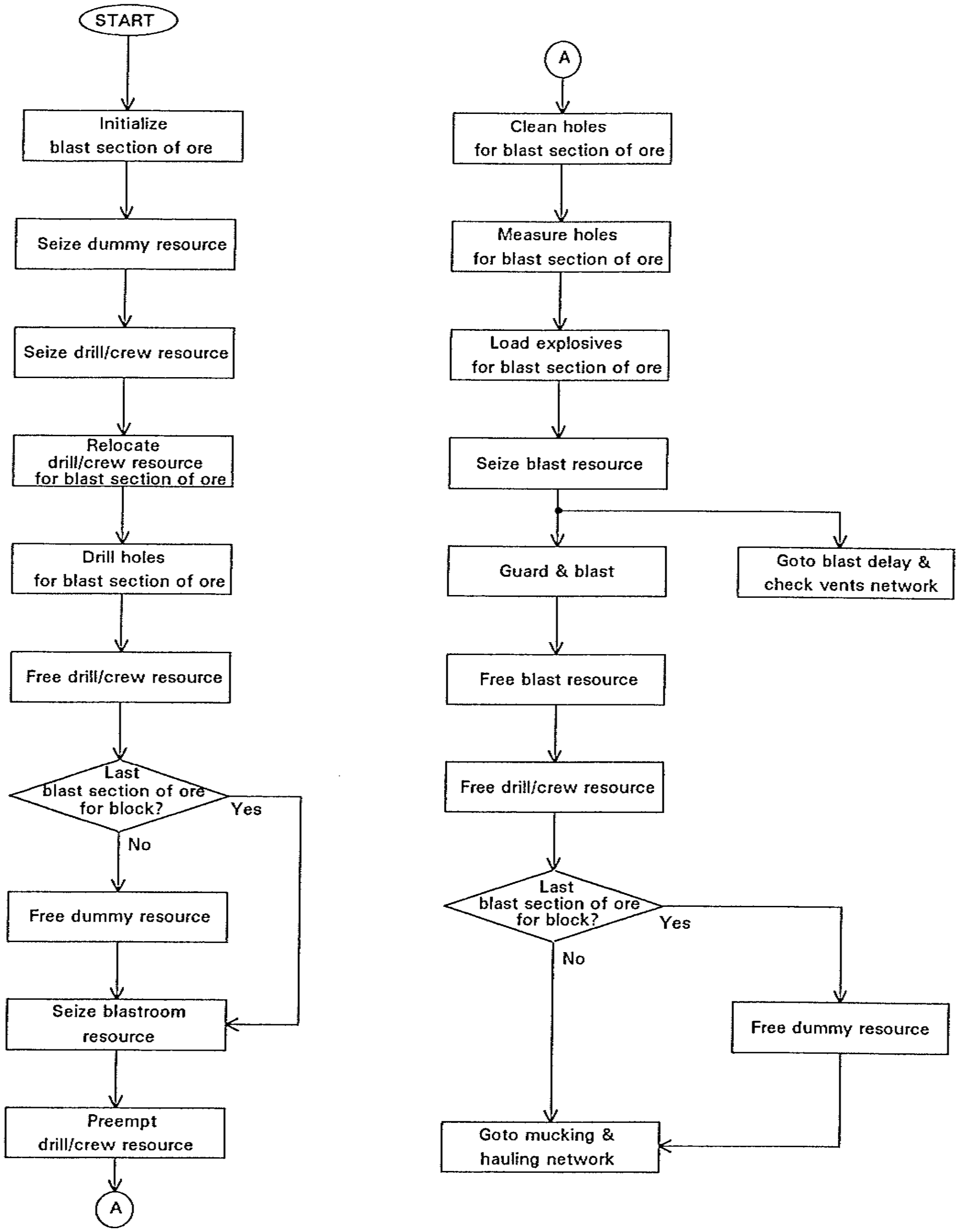


Figure 4.4: Logic diagram for the drilling and blasting segment.

section of ore for the mining block, then the dummy resource is also freed to allow the drill/crew resource to be relocated to the next mining block. (The dummy resource has already been freed for entities that are not the last blast section of ore for the mining block.) The entity is then forwarded to the mucking and hauling network.

4.2.3 Mucking and Hauling Segment

A simplified logic diagram for the mucking and hauling segment is shown in Figure 4.5. (The SLAM II mucking and hauling network can be found in Appendix A.1.) First, the dummy resource is seized, which is used to restrict the number of entities between its seize (AWAIT) and free nodes to one. Next, a check is made to see if the primary LHD is available. If the primary LHD is available, then the entity can proceed as described in the next paragraph. However, if the primary LHD is not available, then a check is made to see if the backup LHD is available. If the backup LHD is available, then the entity can proceed as described in the next paragraph. If neither the primary nor the backup LHDs are available, then the entity waits for 0.1 hours before rechecking. After the wait, the entity is split into two identical entities to ensure it is placed in the dummy resource queue before the dummy resource is freed. This is done to maintain the desired mining block mucking sequencing when two or more mining blocks are available for mucking at the same time.

When a LHD is available (i.e., either the primary or the backup), it is then seized and a check is made to see if there is room available in the orepass. If there is room available in the orepass then a bucket of ore is mucked and hauled to the orepass, and appropriate variables updated. If there is no room available, then the entity is delayed 0.1 hours before rechecking. (As shown in the logic diagram, the entity must first pass through various nodes and activities before rechecking.)

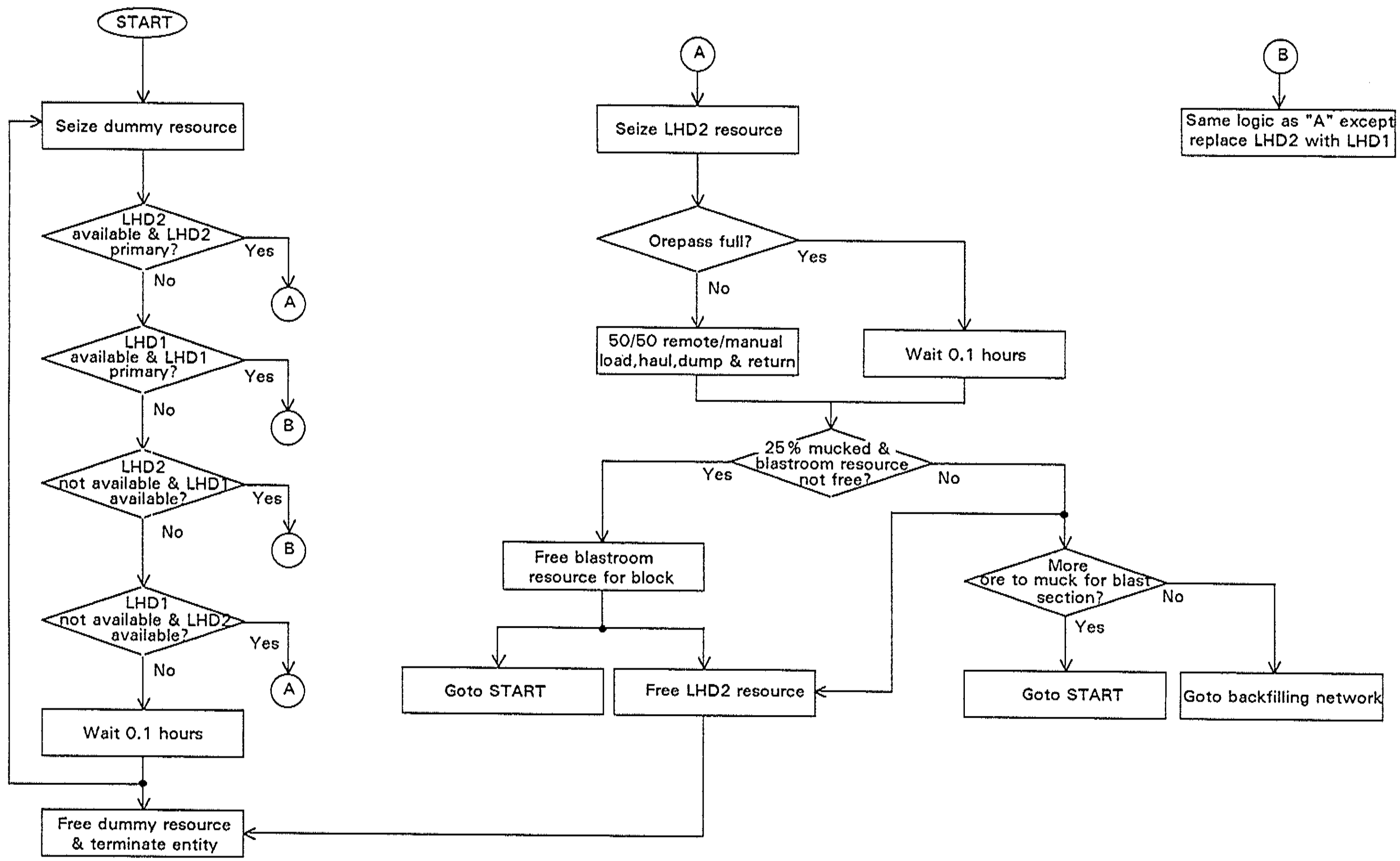


Figure 4.5: Logic diagram for the mucking and hauling segment.

Following either of the above activities (i.e., either the 0.1 hour delay, or the mucking and hauling of a bucket) a check is made to see if 25 percent of the blast section of ore has been mucked and the blastroom resource is not free. If these conditions are met, then the blastroom resource is freed for the mining block, thus allowing the cleaning, measuring and loading of explosives for the sequential blast section of ore for the mining block to begin in the drilling and blasting network. The entity is then split, one is sent back to the start of the mucking and hauling network, and the other is used to free the LHD and dummy resources before being terminated.

If 25 percent of the blast section of ore has not been mucked, or the blastroom resource for the mining block has already been freed, then the LHD and dummy resources are still freed; however, a check is also made to see if there is additional mucking required for the blast section of ore. If so, the entity is routed back to the start of the mucking and hauling network; otherwise, the entity is routed to the backfilling network.

4.2.4 Backfilling Segment

A simplified logic diagram for the backfilling segment is shown in Figure 4.6. (The SLAM II backfilling network can be found in Appendix A.1.) To begin, all blast sections of ore for a particular mining block are grouped into one batched entity before proceeding to seize the truck resource when available. Next, a truckload cycle is performed (i.e., load, haul, dump and return) and appropriate variables updated. A check is then made to see if additional backfilling is required for the mining block. If so, the entity is duplicated where one is sent to the seize truck resource queue, and the other is used to free the truck resource before being destroyed. The reason for using this logic is to maintain proper backfilling sequencing between mining blocks, that is, to ensure the truck resource is not seized by another entity (i.e., mining block) that may be in the truck resource queue.

If no additional backfilling is required, then the entity is still duplicated, where one entity is used to free the truck resource before being terminated, and the other entity is used to increment the termination counter for the simulation run before being terminated.

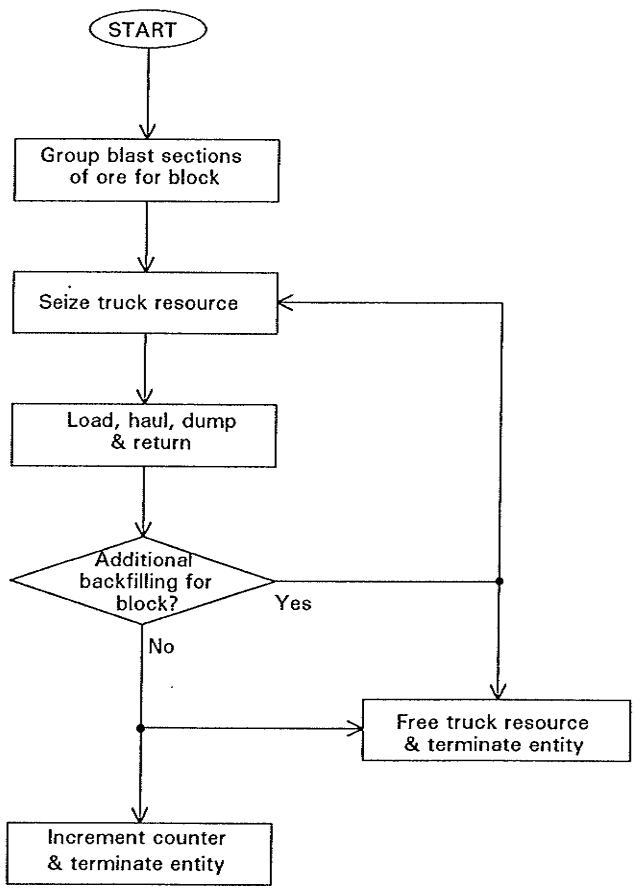


Figure 4.6: Logic diagram for the backfilling segment.

4.2.5 Train Tramming Segment

A simplified logic diagram for the train tramming segment is shown in Figure 4.7. (The SLAM II network for train tramming can be found in Appendix A.1.) Both the ore train and the rock train have been modelled as one resource; however, they are appropriately distinguished within the SLAM II network and the FORTRAN code as required.

As shown, first the (ore) train resource is seized, then the orepass with the greatest amount of ore in it is checked for a trainload. If there is sufficient ore for a trainload then one is trammed from the orepass and relevant variables are updated before the (ore) train resource is freed. If there is not sufficient ore for a trainload then, after a delay of 0.1 hours, the (ore) train resource is freed. A check is then made to see if it is the third shift. If it is not the third shift then ore is trammed; however, if it is the third shift then rock is trammed. When there is no more rock to be trammed on the third, then ore is trammed.

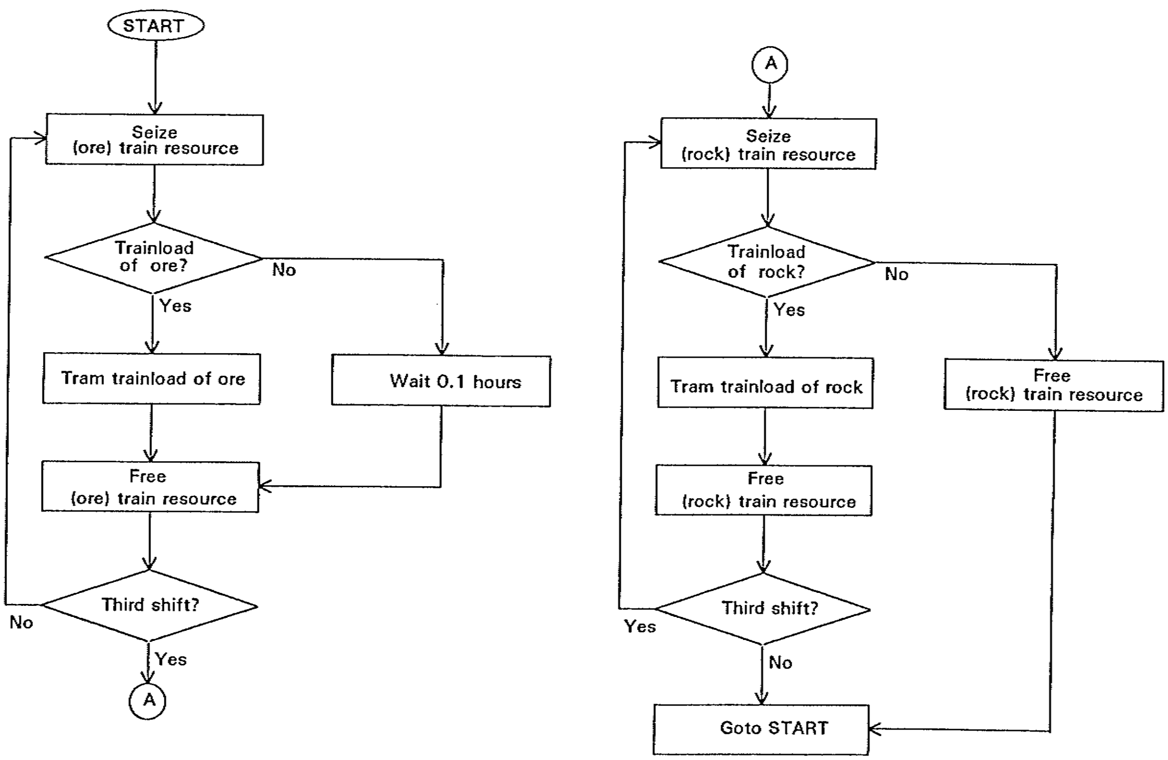


Figure 4.7: Logic diagram for the train tramming segment.

4.2.6 Blast Delay and Check Vents Segment

A simplified logic diagram for the blast delay and check vents segment for the LHDs and truck resources is shown in Figure 4.8. (The SLAM II blast delay and check vents network can be found in Appendix A.1.)

When there is a blast delay and check vents activity, a check is made to verify if the resource (i.e., LHD1, LHD2 or the TRUCK) is being used. If it is being used, the resource is preempted and delayed for the calculated amount of time. Following this, the resource is freed before the entity is destroyed. If the resource is not being used, then a check is made to verify if the required time for the blast delay and check vents activity has elapsed. If so, the entity is terminated; otherwise, the entity waits 0.1 hours before rechecking to see if the resource is being used.

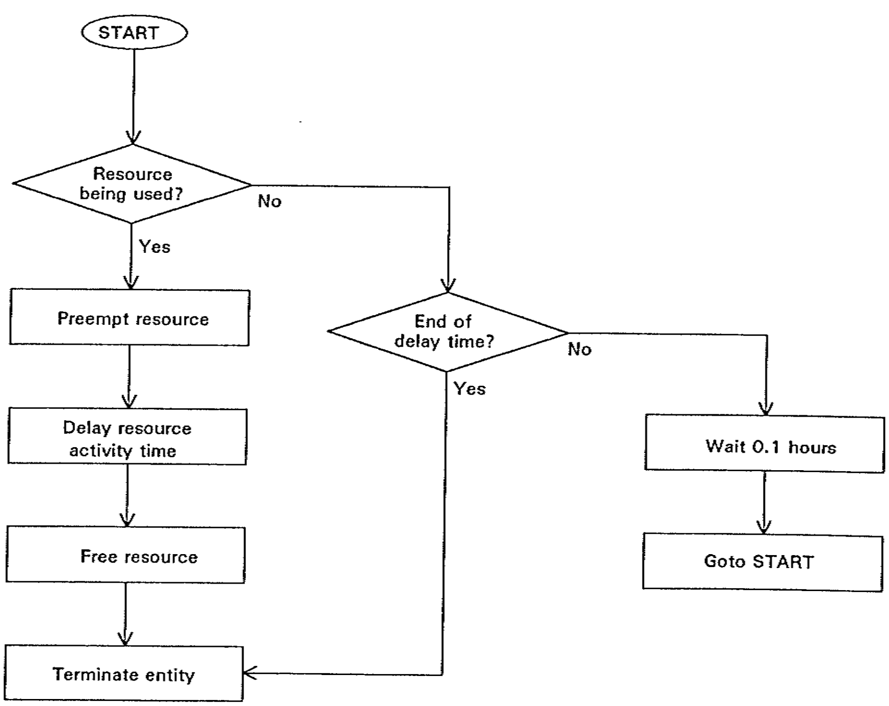


Figure 4.8: Logic diagram for the blast delay and check vents segment.

4.2.7 Equipment Breakdowns Segment

A simplified logic diagram for the equipment breakdowns segment is shown in Figure 4.9. (The SLAM II network for the equipment breakdowns segment can be found in Appendix A.1.) First, an uptime value, which represents how long the equipment resource can operate before it is sent for a breakdown, is calculated. Next, a banked time variable is used to keep track of the operating time of the equipment resource. (The operating time of the equipment resource is the summation of the direct and indirect production times.) Since the simulation model is discrete, changes in the banked time variable are also discrete. For example, if an activity takes 5 hours of indirect and direct production time to perform, then the banked time variable is increased by 5 hours at the start of the activity.

The value of the banked time variable is updated and checked against the uptime value once every 0.1 hours. When the value of the banked time variable is greater than the uptime value, the equipment resource is sent for a breakdown by setting the uptime activity time equal to the uptime value, and allowing it to elapse. When the uptime activity time is elapsing, additional banked time may be accumulating for sequential breakdowns. The banked time variable is then reduced by the amount of the uptime value.

When all, or part of, the uptime activity time elapses on the third shift which is not part of the production schedule time (see Appendix C for an explanation of times), then the appropriate amount of the uptime activity time is extended to the beginning of the first shift. Finally, the equipment resource is preempted, relevant variables are reinitialized and the repair time activity elapses. The equipment resource is then freed and the cycle is repeated again.

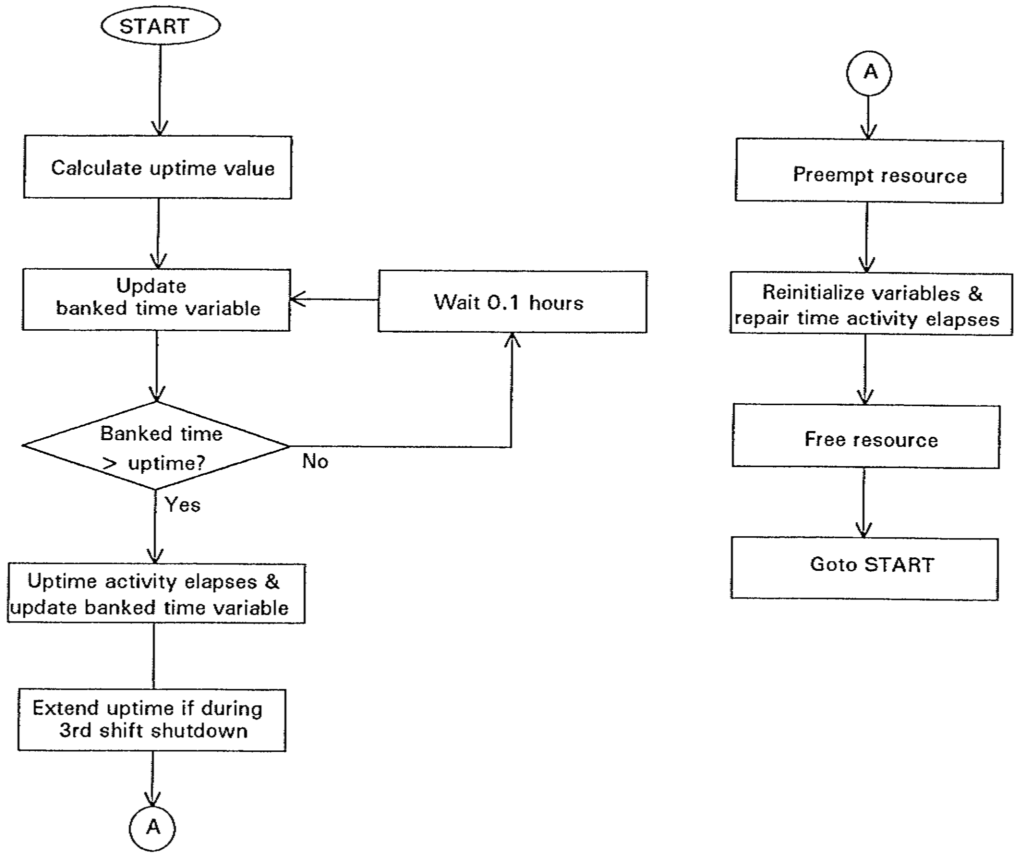


Figure 4.9: Logic diagram for the equipment breakdowns segment.

4.2.8 Shifts Segment

A simplified logic diagram for the shifts segment is shown in Figure 4.10. (The SLAM II network for the shifts segment can be found in Appendix A.1.) First, the resource (i.e., the drills/crews, the LHDs, the train, or the truck) is preempted at the start of the shift for 2 hours of non-productive time for the day shift. (The non-productive time, more appropriately referred to as indirect production time in Appendix C, consists of time for shift changes, lunch, coffee breaks, etc.) Following this, the resource is freed for 6 hours of productive time for the day shift. This does not mean 6 hours of productive time occur, rather the resource is made available for 6 hours of productive activities. Similar logic is used for the afternoon shift.

The logic for the night shift is slightly different than the logic for the day and afternoon shifts, since some of the resources have three shifts per day, whereas, others have only two shifts per day. For resources that have three shifts per day, the logic is the same as that used for the day and afternoon shifts. However, for resources that have only two shifts per day, they are preempted, or made unavailable for 8 hours rather than 2 hours. This is necessary since the model runs on a 24 hour day.

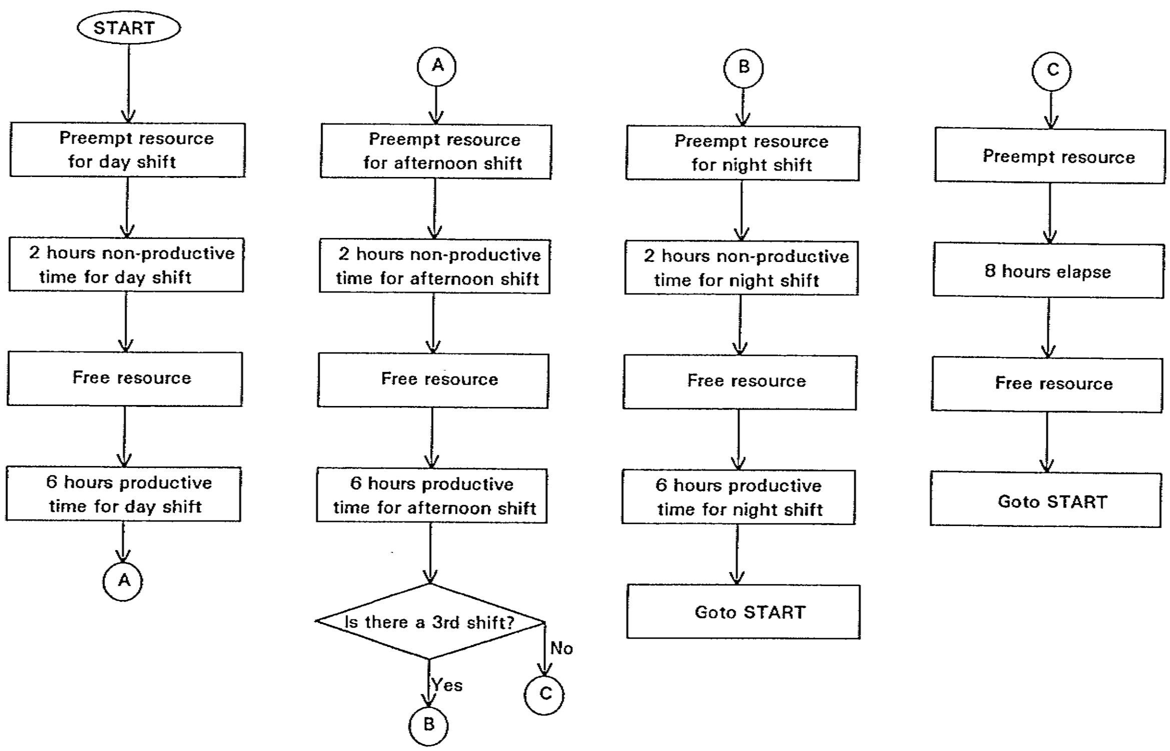


Figure 4.10: Logic diagram for the shifts segment.

4.2.9 Miscellaneous and Resource Segments

Since the logic for the miscellaneous segments is relatively straight forward, a logic diagram has not been constructed. (The SLAM II network for the miscellaneous segment can be found in Appendix A.1.) Basically, in this segment various entities are created and destroyed to perform the following functions:

1. To deposit development ore and rock into the appropriate chimneys once every 2 hours for the first and second shifts.
2. To print the daily values of various variables to appropriate files, as well as, reinitialize variables as necessary.
3. To control the availability of the blast resource so that blasts can only occur at the end of the shifts. That is, the blast resource is made unavailable for the first 6.5 hours and available for the last 1.5 hours of every shift.

In addition a resource segment, as shown in Appendix A.1, is used to define the various resources used within the model.

4.3 MODEL VERIFICATION AND VALIDATION

Model verification is the process of determining whether a computer simulation program performs as intended, that is, it involves debugging the computer program (Law and Kelton, 1991). For the underground production mining simulation model discussed in this chapter, verification was performed as the model evolved by doing detailed and selective traces, as well as, printing various values to output files. Detailed traces involve recording the start time, the event node, the current attribute values and the activity summary for each entity as it moves through the network. When the model became too large and complex to continue doing detailed traces, selective traces were performed, that is, only event nodes and activities of interest were traced (Litke, Laroche, Strong and Szymanski, 1993). In addition, various values were printed to output files to ensure they were consistent with what was expected.

Model validation is concerned with determining whether the conceptual simulation model is an accurate representation of the system under study (Law and Kelton, 1991). Validation for the underground production mining simulation model was broken down into three main steps (see Litke, Laroche, Strong and Szymanski, 1993). The first step involved analysis of the model logic and minimizing differences between the real world system and the conceptual simulation model. This approach was iterated, in conjunction with verification, with each resolution throughout the entire model building process.

The second step of validation focused on finding suitable input distributions for activity times, and equipment uptimes and downtimes. Inco's incentive standards hourly rates, or incentive rates, were used as a basis for the activity times, adjusted as deemed appropriate (see Appendix D.1). Various values based upon the adjusted incentive rates were then entered into UniFit II, a probability distribution calculation software package, to obtain the activity distributions and corresponding SLAM II equations. Beta distributions, as given in Appendix D, were obtained for all activity times. A sample of the SLAM II generated beta distributions for manual and remote control LHD loading is shown in Figure 4.11 (based upon 1000 samples).

With respect to equipment uptimes and downtimes, past records along with expert opinions were used to obtain various values. These values were then entered into UniFit II to obtain the distributions and corresponding SLAM II equations. Gamma distributions, as given in Appendix D, were obtained for all equipment uptimes and downtimes. A sample of the SLAM II generated gamma distributions for LHD uptime and downtime is shown in Figure 4.12 (based upon 1000 samples).

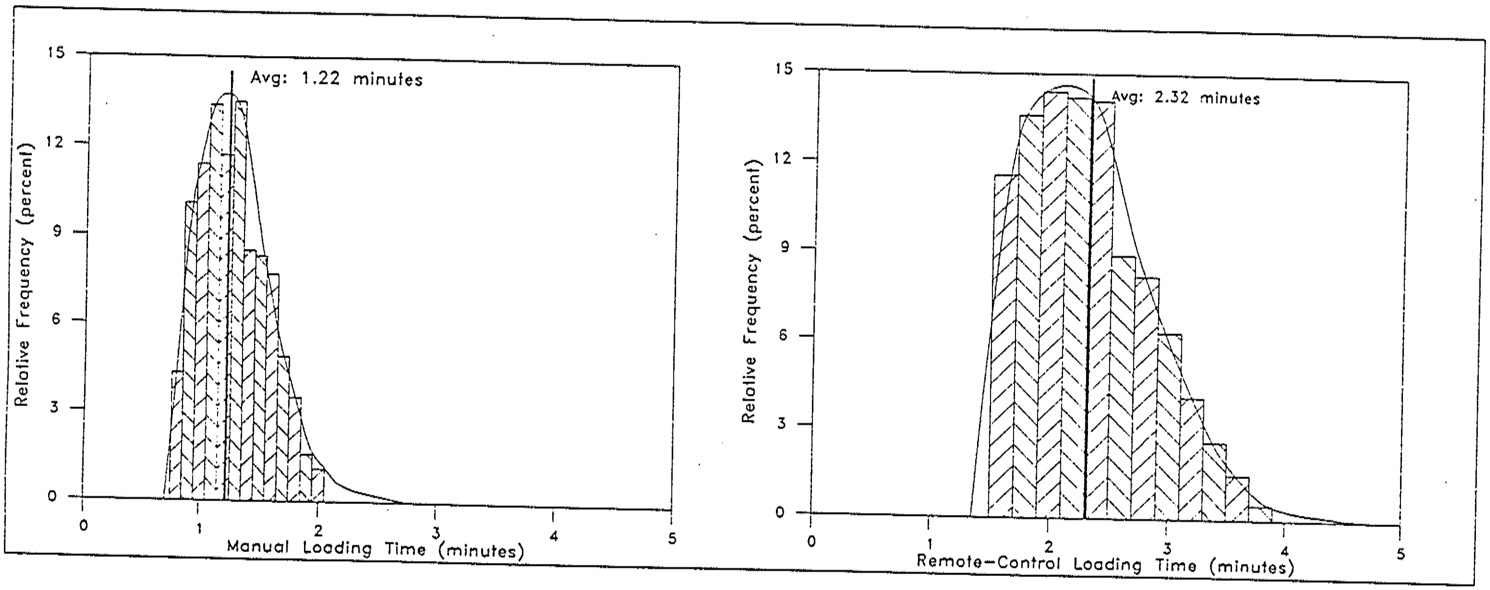


Figure 4.11: SLAM II generated beta distributions for manual and remote control LHD loading (Litke, Laroche, Strong and Szymanski, 1993).

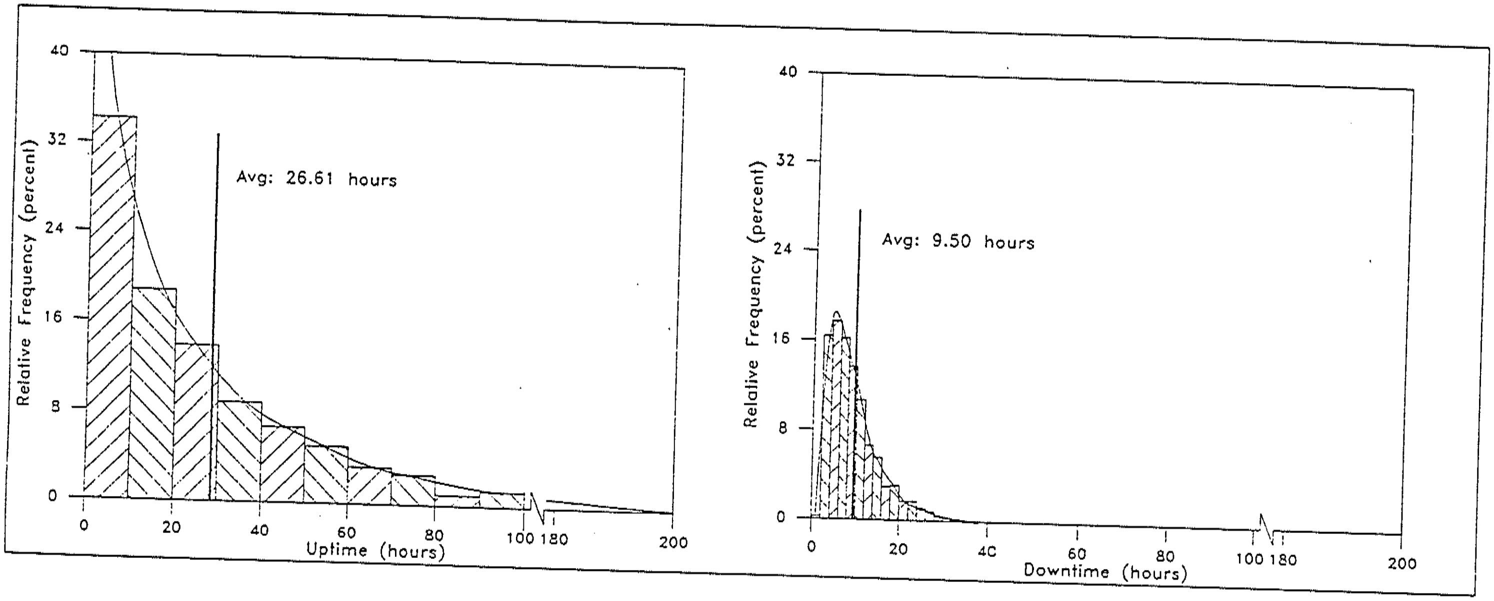


Figure 4.12: SLAM II generated gamma distributions for LHD uptime and downtime (Litke, Laroche, Strong and Szymanski, 1993).

The third step of validation performed focused on comparing output results generated by the simulation model with actual data from the existing mine. Key parameters, such as the production rate, the time to perform various activities and the availability ratios, were found to compare favorably. That is, the output results generated by the simulation model closely resembled actual data from the mine for the selected mining blocks.

4.4 MODEL INPUTS

Model inputs have been divided into internal input data and external input data. Each is described within their respective section, followed by a list of amendments that may have to be made to the model when the data file is changed.

Internal Input Data

Internal input data consists of data that is within the program itself, primarily within file INTLC (see Appendix A.6.2) and file USERF (see Appendix A.6.3). Examples of internal input data are the activity time equations, uptime and downtime equations, LHD bucket capacities, train car capacities, and truck capacities, to name a few. A complete listing of these equations and rates can be found in Appendix D.

External Input Data

External input data is entered into the model by the user through either the data file or questions answered by the user at the start of the first simulation run. An example of a data file is given in Appendix A.3. As shown, each row of numbers within the data file (except for the last four numbers) represent characteristics, or attributes, specific to a particular blast section of ore (see Appendix A.4 for an explanation of attribute numbers).

The last four numbers in the data file respectively represent the following: the daily amount of development rock deposited into the rockpass; the I.D. number/location where the rock is deposited; the daily amount of development ore deposited into the orepass; and the I.D. number/location where the ore is deposited.

Questions answered by the user at the start of the first simulation run include:

1. The desired number of simulation runs (between 1 and 10), and
2. The desired confidence level (90, 95, or 98) for the output results.

Data File Changes

When changes to the data file are made, corresponding amendments to the model as listed below, may have to be made.

1. The network (see Appendix A.1) may have to be changed to ensure the following:

   (a) The maximum creations in the CREATE node labelled START within the drilling and blasting network must equal the total number of blast sections of ore within the data file.

   (b) The termination count in the TERMINATE node labelled END within the backfilling network must equal the total number of mining blocks within the data file.

2. The control file (see Appendix A.2) may have to be changed to ensure the following:

   (a) The number of ARRAYs must correspond to the number of blast sections of ore within the data file.

   (b) The ending time in the INITIALIZE statement must be large enough to simulate mining of all blast sections of ore within the data file.

3. The FORTRAN code for file INTLC (see Appendix A.6.2) may have to be changed to ensure the following:

(a) The data file opened within INTLC corresponds to the name of the data file to be opened.

(b) The variable NBLASTS must equal the total number of blast sections of ore within the data file.

4.5 MODEL OUTPUTS

The model contains user written FORTRAN code used to generate the following four output files: 1) SUMMARY, 2) RUNS, 3) AMOUNTS, and 4) COSTS. The first three output files are used to collect simulation results on all runs performed (i.e., from 1 to 10 runs). The COSTS output file is based only upon the last simulation run performed. An example of all four output files can be found in Appendix B.

The SUMMARY output file provides information on the averages and confidence intervals for the following:

- The mechanical availability, physical availability, use of availability, and effective utilization for each piece of equipment, along with the hours and shifts used to calculate these values. (For an explanation of these terms refer to Appendix C.)
- The daily tons of ore and pounds of nickel mucked and hauled by the LHDs.
- The daily tons of ore and rock trammed.
- The mucking tons per day on a per mining block basis (up to 10 mining blocks).
- The production schedule table for various events for each mining block (up to 10 mining blocks).
- The activity times for drilling and blasting, mucking and hauling, and backfilling for each mining block (up to 10 mining blocks).

The RUNS output file provides similar information as the SUMMARY file, except it is for each simulation run performed, rather than the averages and confidence intervals.

The AMOUNTS file provides the following information for each simulation run:

- The daily and cumulative amounts of ore and nickel mucked and hauled by the LHDs.
- The daily and cumulative amounts of ore trammed by the ore train.
- The daily and cumulative amounts of rock trammed by the rock train.
- The tons of rock in the rockpass at the end of the day.

The COSTS file provides preliminary cost information on drilling, blasting, and mucking and hauling for the last simulation run only. The COSTS file is in preliminary stages and requires further developmental work beyond this thesis.

4.6 NUMBER OF RUNS AND RUN LENGTH

The model was built to accommodate 1 to 10 simulation runs as specified by the user at the start of the first simulation run. If more than 10 runs is desired, then various arrays within the FORTRAN code would have to be adjusted accordingly.

The length of each simulation run is determined by the completion of the backfilling activity for all mining blocks considered. Since the beginning and ending conditions of each run affect various variables, warming-up and cooling-down the model was performed as needed. Warming-up the model refers to not collecting data before certain points in time near the beginning of the run (i.e., the start-time), or deleting some of the observations from the beginning of the run. Cooling-down the model refers to not collecting data beyond a certain point in time near the end of the run (i.e., the end-time).

Warming-up and cooling-down the model for the hours used to calculate the availability ratios was performed as follows:

- When a drill/crew resource is initially used on the first blast section of ore of the first block, then the time is marked as the start-time for the drills. The end-time for the drills is the time when the last blast section of ore for all mining blocks considered completes the drilling and blasting activity. (Note that the same start-time and end-time is used for both drills/crews.)

- When a LHD resource is initially used on the first blast section of ore of the first block, then the time is marked as the start-time for the LHDs, as well as the start-time for the train. The end-time for the LHDs and the train is the same as that used for the drills.
- When the truck resource is initially used on the first blast section of ore of the first block, then the time is marked as the start-time for the truck. The end-time for the truck is the same as that used for the drills.

The average daily tons of ore and pounds of nickel mucked and hauled by the LHDs, as well as the average daily tons of ore trammed by the train, is calculated based upon the start-time and end-time for the LHDs and train. On the other hand, the average daily tons of rock trammed is calculated based upon the start of the run to the end-time for the train.

Warming-up and cooling-down was not necessary for various other variables, such as, the mucking tons per day on a per block basis, the production schedule table, and the activity times.

CHAPTER 5

SCENARIOS INVESTIGATED

5.1 INTRODUCTION

The simulation model described in Chapter 4 has been used to model four typical mining blocks found at the 83 orebody in the Birchtree Mine. These mining blocks have been numbered within this thesis as blocks 1 through 4, inclusive, and correspond to Birchtree's mining blocks identified below:

- Block 1 within this thesis corresponds to Birchtree's block 5 (also called 21-810.1),
- Block 2 within this thesis is based upon Birchtree's block 6 (also called 20-785.1), however, the data was modified slightly since it was not a typical mining block,
- Block 3 within this thesis corresponds to Birchtree's block 7 (also called 21-800.3), and
- Block 4 within this thesis corresponds to Birchtree's block 8 (also called 21-810.2).

The basecase scenario discussed in the next section presents various time charts to demonstrate some of the operational characteristics of the current version of the simulation model (Version 1.2). Next, results from various scenarios are presented to determine activity and system bottlenecks, as well as, the expected average daily production of ore.

The final section in this chapter provides a discussion on other scenarios that were investigated, such as, alternative LHD configurations, increasing the average daily production rate, and alternative mining block mucking rules. The simulation results from these scenarios are not presented since they are based upon an earlier version of the simulation model (Version 1.1), and the real world system has changed.

For the purpose of this thesis, ten simulation runs were performed for each scenario considered. When performance measures were compared, the paired t-test, or the paired-difference test, was used at a 95 percent confidence level.

5.2 BASECASE SCENARIO TIME CHARTS

The basecase scenario's input data is given in Appendix A.3, while the assumptions, equations and rates are listed in Appendix D. Output files from the basecase scenario can be found in Appendix B.

The average values for various activities from the production schedule table given in Appendix B.1 are presented in Figure 5.1. As shown, Drill/Crew 1 is used for mining blocks 1 and 3, and Drill/Crew 2 is used for mining blocks 2 and 4. Referring to Figure 5.1, the following observations may aid in understanding some of the operational characteristics of the model:

- Drill/Crew 1 is relocated from block 1 to block 3 as soon as the drilling and blasting activity at block 1 has been completed (similarly for Drill/Crew 2 and blocks 2 and 4).
- Mucking and hauling alternates between blocks 1 and 2, then between blocks 3 and 4. The alternation between mining blocks is based upon the mining block mucking rule (i.e., the lowest mining block number is mucked first). Although a solid rectangle is shown, mucking and hauling only occurs from one mining block at any particular point in time since only one LHD operator is assumed available.

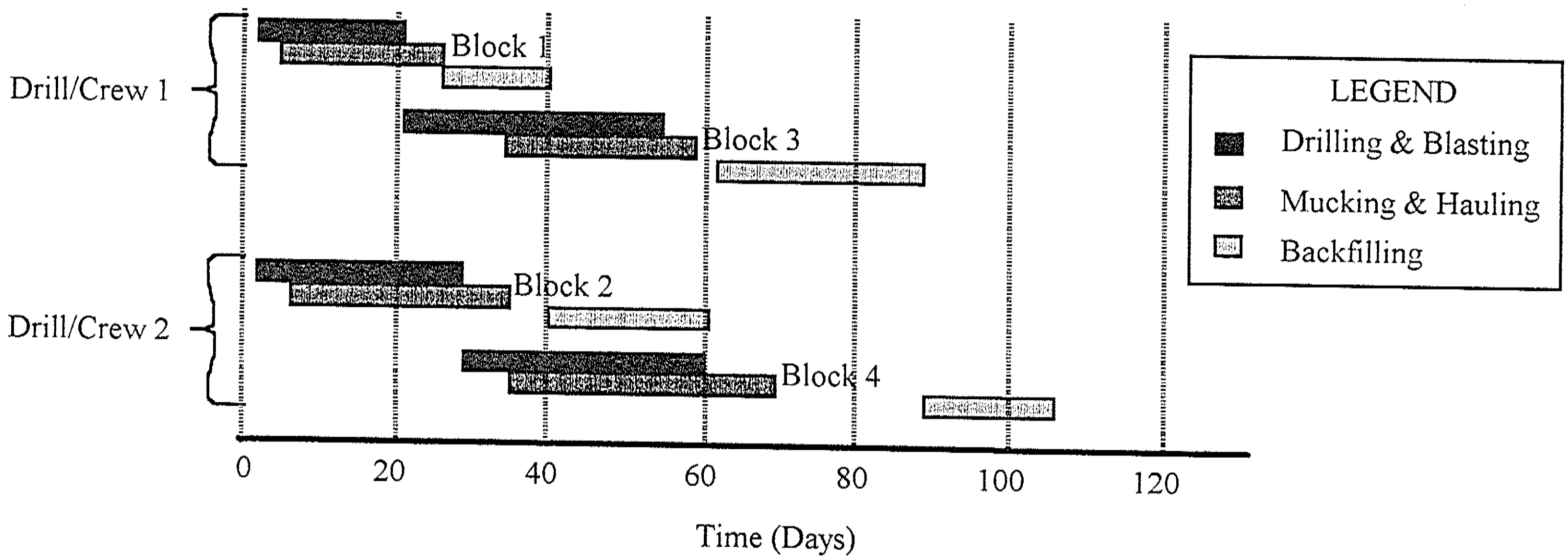


Figure 5.1: General activity time chart.

- As soon as the mucking and hauling activity for block 1 has been completed, the backfilling activity begins. As soon as block 1 has been backfilled, then block 2 is backfilled, followed by block 3, and finally block 4. The order in which the mining blocks are backfilled is based upon the backfilling rule (i.e., the mining block which is finished mucking and hauling first is backfilled first).

To present additional operational characteristics of the model, a five day period on the first simulation run was arbitrarily chosen for selective tracing. Specifically, a section of time from 400 to 520 hours on the first simulation run was examined in detail for the drilling and blasting, mucking and hauling, and train tramming activities. The results from these traces have been presented in graphical form in Figures 5.2 to 5.5, and are now discussed.

Figure 5.2 presents the drilling and blasting activity time chart for both Drill/Crew 1 and Drill/Crew 2. As shown, the drills/crews are either working (i.e., the drilling and blasting activity), being repaired, on miscellaneous activities (i.e., two hours of every shift have been assigned for miscellaneous activities or non-productive time as described in Appendix D), or waiting (i.e., it is not being used for any of the aforementioned activities). Referring to Figure 5.2, the following observations may aid in understanding some of the operational characteristics of the model:

- Every eight hour shift has two hours of miscellaneous activities at the start of the shift, which preempts both the repair and working activities. Furthermore, the repair activity preempts the working activity.
- Drill/Crew 1's wait time between hours 402 and 411 was incurred because the blastroom resource was not available (i.e., 25 percent of the mucking and hauling was not completed on the previous blast section of ore), and the blast section of ore was the last one for the mining block (i.e., the fifth blast section of ore for block 1). The

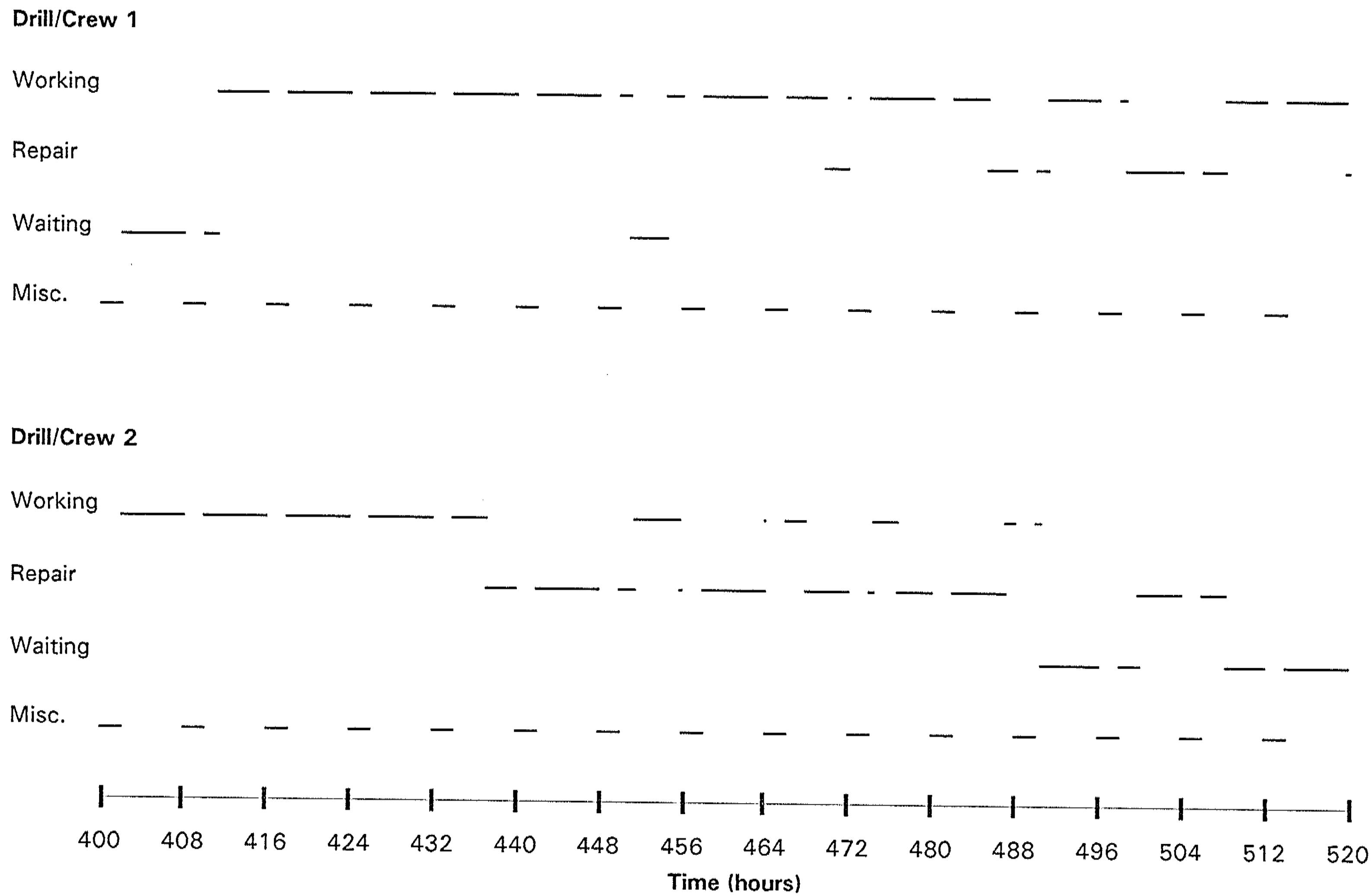


Figure 5.2: Drilling and blasting activity time chart.

waiting between hours 451 and 454.5 was due to the blast resource not being available (i.e., only the last 1.5 hours of each shift are available for the blast and guard activity).

- Drill/Crew 2's waiting after hour 490 was due to the blastroom resource not being available, and the blast section of ore being the last one for the mining block (i.e., the sixth blast section of ore for block 2).

To provide greater detail of the working activity for Drill/Crew 1 shown in Figure 5.2, Figure 5.3 has been divided into the following: drilling holes, relocating the drill, guarding and blasting, loading explosives, measuring holes, cleaning holes, repair and miscellaneous activities, and waiting. The activities performed between hours 400 and 456 are for block 1, blast section of ore 5; and from 456 onwards for block 3, blast section of ore 1. As previously stated, the waiting from approximately 451 to 454.5 hours due to the blast resource not being available, is more obvious in Figure 5.3 than in Figure 5.2.

The mucking and hauling activity time chart for the same time period is shown in Figure 5.4. LHD 9 (i.e., the 9 cubic yard, referred to as LHD2 in the simulation program in Appendix A) is used as the primary LHD, and LHD 6 (i.e., the 6 cubic yard, referred to as LHD1 in the simulation program in Appendix A) is used as the secondary or backup LHD. Recall that the simulation model assumes only one LHD operator at any particular point in time, consequently, the time for LHD 6 when LHD 9 is available is not shown as waiting in Figure 5.4. (However, this time is accounted for in the simulation program as standby time for LHD 6.) Referring to Figure 5.4, the following observations may aid in understanding some of the operational characteristics of the model:

- Every eight hour shift has two hours of miscellaneous activities at the start of the shift, which preempts both the repair and working activities. Furthermore, the repair activity preempts the working activity. Since mucking and hauling is not performed on the third shift, this time is shown in Figure 5.4 as part of the miscellaneous time.

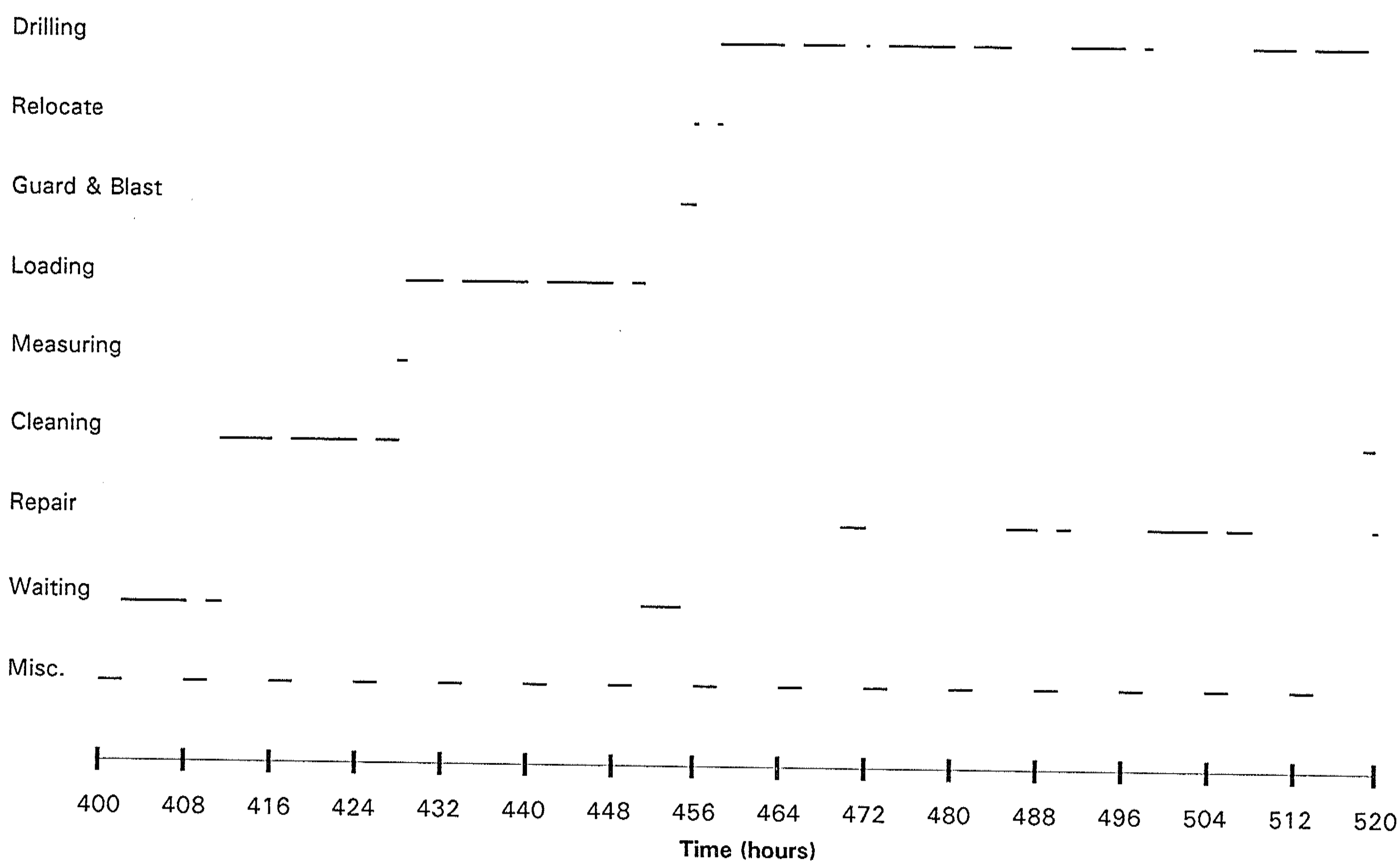


Figure 5.3: Drill/Crew 1 detailed activity time chart.

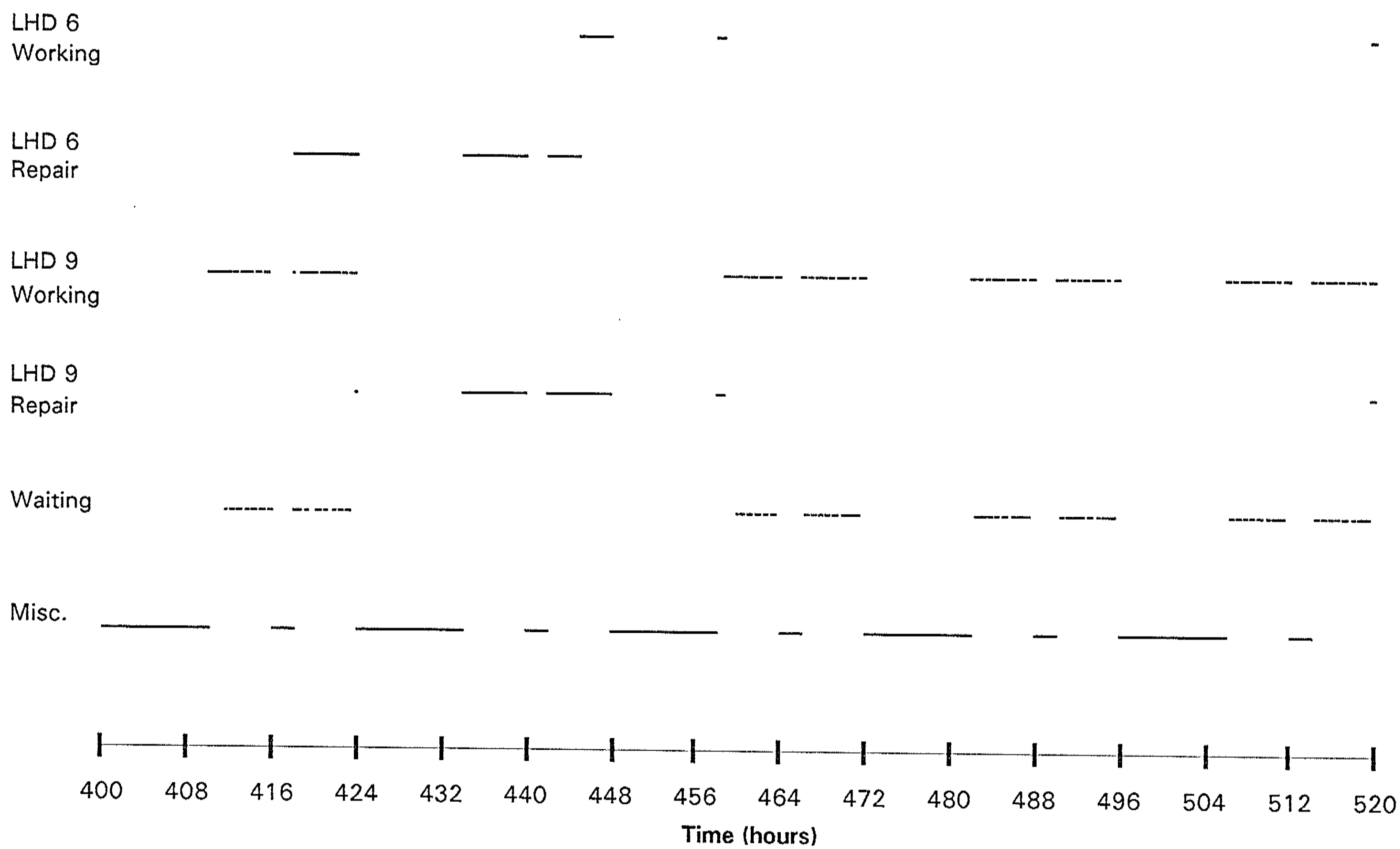


Figure 5.4: Mucking and hauling activity time chart.

- For the time period considered, when LHD 9 was used there was considerable waiting because the orepass was full (in fact, about half the time is spent waiting for orepass storage capacity); however, whenever LHD 6 was used there was no observed waiting for orepass storage capacity.
- Between hours 424 and 445, both LHDs were being repaired. At approximately hour 445, the repair on LHD 6 was complete, therefore, it began working (i.e., the mucking and hauling activity). As soon as the repair on LHD 9 was complete, it replaced LHD 6.

The train tramming activity time chart for the same time period is shown in Figure 5.5. Referring to Figure 5.5, the following observations may aid in understanding some of the operational characteristics of the model:

- Every eight hour shift has two hours of miscellaneous activities at the start of the shift, which preempts both the repair and tramming activities. (Note for the time period considered there was no repair activity).
- Hours 400 to 408, 424 to 432, 448 to 456, 472 to 480, and 496 to 504 are the third shifts, during which rock is trammed first, then ore when there is no rock to be trammed. As shown, there is some waiting on the third shift which implies all the rock and ore within the chimneys was trammed.
- There are insufficient amounts of ore within the chimneys for a trainload between hours 430 and 445 because both LHDs were being repaired, as previously stated.

When Figures 5.2 to 5.5 are viewed collectively, it appears that if a greater daily production rate is desired for these mining blocks, then the orepass storage capacity should be increased. For example, approximately 50 percent of the time is spent waiting for sufficient orepass storage capacity when LHD 9 is being used. The LHD 9 waiting in turn results in drilling and blasting activity waiting for the last blast sections of ore for the mining blocks. In addition, LHD 9 waiting results in train tramming waiting on the third

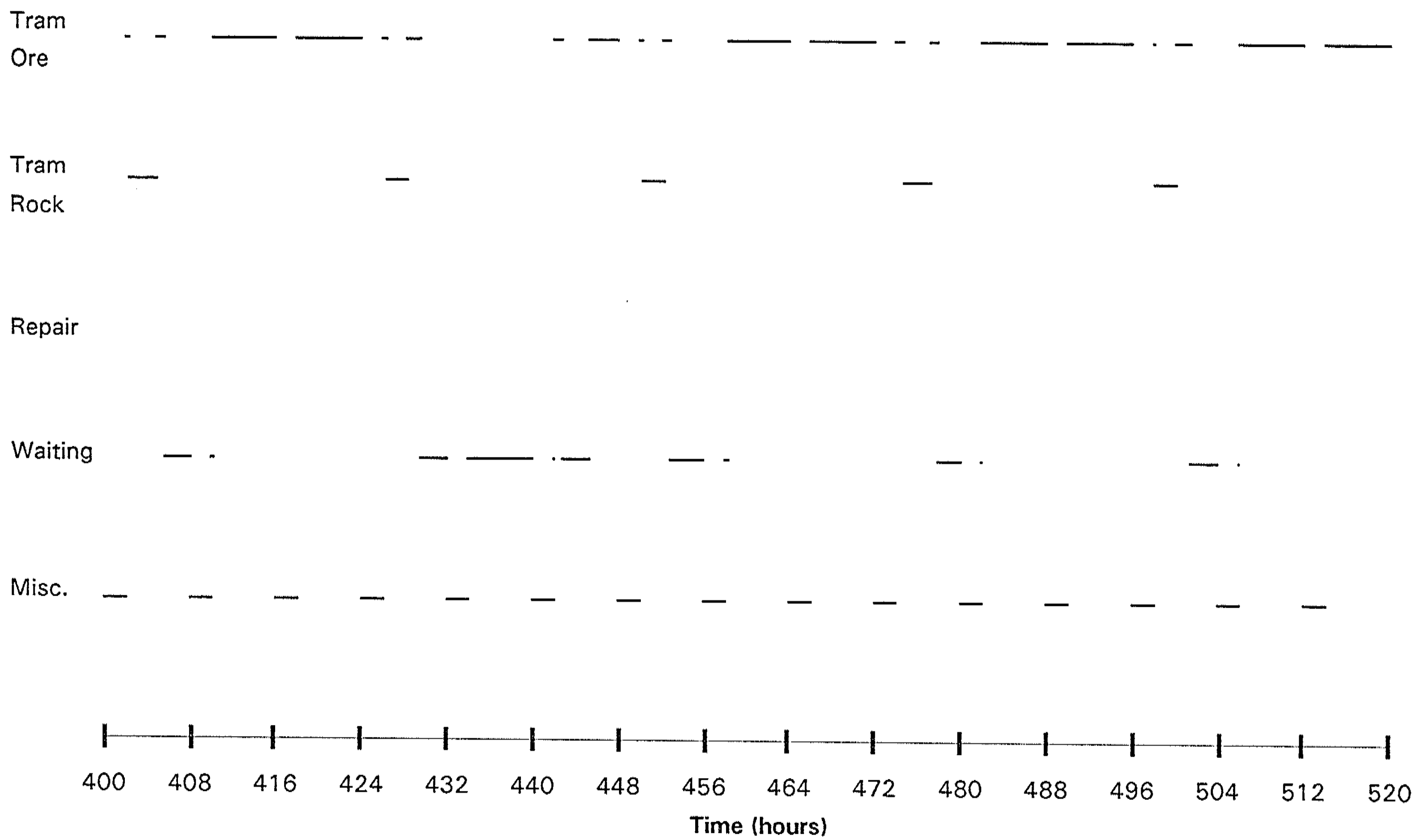


Figure 5.5: Train tramming activity time chart.

shift. This analysis, of course, is based upon specific mining blocks and orepass storage capacities, as well as, only part of one simulation run.

The above method of activity time chart analysis could be performed for additional runs and perhaps extended for greater periods of time; unfortunately, it is very time consuming. Perhaps others may want to automate the process since it appears to be very informative. However, it should be noted that some of the information obtained above is also in numerical form in Appendix B.1. For example, the waiting due to insufficient orepass storage capacity when LHD 9 is being used can be observed in the high value of indirect production time (IPT) for LHD2. (This leads one to question whether the waiting due to the orepass being full should be assigned to standby time rather than indirect production time as described in Appendix C, since it affects the availability ratios.)

5.3 ACTIVITY AND SYSTEM BOTTLENECKS SCENARIOS

5.3.1 Introduction

The activity and system bottlenecks scenarios focused on drilling and blasting, mucking and hauling, and train tramming activities (as well as the related system), since they are directly linked to each other. The activity bottleneck refers to the bottleneck for the activities considered, whereas, the system bottleneck refers to the bottleneck for the system considered. (Note the system contains the activities, as well as, other components such as the orepass storage capacity.) For example, the activity bottleneck may be the drilling and blasting activity; however, the system bottleneck may be the orepass storage capacity.

As previously mentioned, ten simulation runs were performed for each scenario considered. When the performance measure between scenarios was compared, the paired t-test, or the paired-difference test, was used at a 95 percent confidence level (i.e., the probability of rejecting the null hypothesis, H_O, when it is true is 5 percent). The performance measure used was the average ore throughput in tons per day, or the average daily production rate, for the activity of interest.

The activity and system bottlenecks scenarios have been divided into independent activity scenarios and interactive activity scenarios, each will be described within their respective sections. Both categories of scenarios were based upon the simulation model given in Appendix A and the assumptions given in Appendix D, modified as noted.

5.3.2 Independent Activity Scenarios

The independent activity scenarios focused on each activity independent of the others in order to minimize influence on each other. For example, in order to obtain the independent average daily production rate for drilling and blasting, the mucking and hauling, and train tramming activity times were set to zero. That is, the drilling and blasting activity times were as specified in Appendix D, while the activity times for mucking and hauling, as well as, train tramming were set to zero. Similar scenarios were performed in order to obtain the independent average daily production rates for mucking and hauling, as well as, train tramming.

Note that the independent activity scenarios do not necessarily indicate the maximum average daily production rate for the activity of interest since mucking and hauling is on two shifts, while the others are on three shifts. For example, the train tramming activity may have waiting on the third shift, since the amount that can be trammed on the third shift is limited to the amount available in the chimneys.

The results from these scenarios are shown in Table 5.1. In order to broaden the conclusions somewhat, scenarios were performed for both a 1 and 2 drill/crew system, as well as, various LHD return hauling distances. (Note the real world system considered in this case study has 2 drills/crews and LHD return hauling distances of 470 to 600 feet.)

Table 5.1 Average daily production rate with activities independent of each other.

Scenario	Average Daily Ore Mucking & Hauling (tons/day)	Average Daily Ore Train Tramming (tons/day)
Drilling and blasting (3 shifts):		
1 drill/crew	*686 ± 18	N/A
2 drills/crews	*1372 ± 79	N/A
Mucking and hauling (2 shifts):		
470' to 600' return hauling distances	1669 ± 140	N/A
940' to 1200' return hauling distances	1229 ± 74	N/A
1410' to 1800' return hauling distances	1043 ± 52	N/A
Train tramming ore (see Note)	N/A	†1678 ± 39

* The average daily amount of ore that the drilling and blasting activity can supply to the mucking and hauling activity (based upon mucking and hauling on two shifts per day).

N/A Not applicable since the average daily production rate is based upon an activity time of zero.

† In addition to the ore available from drilling and blasting and mucking and hauling (based upon mucking and hauling on two shifts per day), there are 150 tons of development ore per day directly deposited into the orepass.

Note: There are 3 shifts per day for tramming, of which 2 are used for tramming ore and 1 is used for tramming rock; however, if there is no rock to tram on the third shift, then ore is trammed if available.

By comparing the average daily production rates in Table 5.1, the activity bottleneck can be determined. That is, when 1 drill/crew is used, then drilling and blasting is the activity bottleneck for all LHD return hauling distances considered. When 2 drills/crews are used simultaneously at separate locations and the LHD return hauling distances are 470 to 600 feet, then drilling and blasting remains the bottleneck. On the other hand, as the LHD return hauling distances increase, then mucking and hauling could become the activity bottleneck; however, this would not be the case for normal LHD return hauling distances (the return hauling distances of 1410 to 1800 feet are considered to be extreme) and assuming a third shift is available if needed. To continue, note that train tramming for the mining blocks considered was from orepasses #3 and #4, which have the highest return tramming time (see Appendix D.3). Therefore, for the activities considered and the reasons stated above, drilling and blasting is considered to be the activity bottleneck.

5.3.3 Interactive Activity Scenarios

The interactive activity scenarios focused on grouping drilling and blasting, mucking and hauling, and train tramming activities as listed below:

1. Grouping drilling and blasting with mucking and hauling, that is, drilling and blasting and mucking and hauling activity times are as outlined in Appendix D, while the activity times for train tramming are set to zero.
2. Grouping mucking and hauling with train tramming, that is, drilling and blasting activity times are set to zero, while mucking and hauling and train tramming activity times are as outlined in Appendix D.
3. Grouping all activities, that is, drilling and blasting, mucking and hauling, and train tramming activity times are as outlined in Appendix D (i.e., the basecase scenario discussed in Section 5.2).

The results from these scenarios are shown in Table 5.2. (Note that the values presented are for a 2 drill/crew system and 470 to 600 feet LHD return hauling distances.)

Table 5.2 Average daily production rate with activity interaction.

Scenario	Average Daily Ore Mucking & Hauling (tons/day)	Average Daily Ore Train Tramming (tons/day)
Group drilling & blasting with mucking & hauling (train tramming set to 0)	1372 ± 50	N/A
Group mucking & hauling with train tramming (drilling & blasting set to 0)	1238 ± 69	†1391 ± 69
Group all activities	1105 ± 46	†1266 ± 47
Group all activities with unlimited orepass storage capacity	1337 ± 43	†1449 ± 44

N/A Not applicable since the average daily production rate is based upon an activity time of zero.

† In addition to the ore available from drilling and blasting and mucking and hauling (based upon mucking and hauling on two shifts per day), there are 150 tons of development ore per day directly deposited into the orepass.

To begin, note there is reason to believe that grouping the drilling and blasting activity with the mucking and hauling activity did not lower the average daily production rate for drilling and blasting (based upon the 1372 ± 79 value given in Table 5.1 and the 1372 ± 50 value given in Table 5.2). On the other hand, there is reason to believe that grouping the drilling and blasting activity with the mucking and hauling activity did reduce the average daily production rate for mucking and hauling (based upon the 1669 ± 140 value given in Table 5.1 and the 1372 ± 50 value given in Table 5.2). This is consistent with the earlier observation that drilling and blasting is the activity bottleneck.

To continue, note there is reason to believe that grouping the drilling and blasting activity with the mucking and hauling activity has a higher average daily production rate than grouping mucking and hauling with train tramming (based upon the 1372 ± 50 and

1238 ± 69 values given in Table 5.2). Similarly, note there is reason to believe that grouping the drilling and blasting activity with the mucking and hauling activity has a higher average daily production rate than grouping all activities (based upon the 1372 ± 50 and 1105 ± 46 values given in Table 5.2). This leads one to believe that the train tramming activity is holding back production; however, as shown in Table 5.1, there is considerable train tramming capacity (i.e., 1678 ± 39). Consequently, a further scenario was performed that grouped all activities with unlimited orepass storage capacity. When this scenario is considered, there is reason to believe that grouping the drilling and blasting activity with the mucking and hauling activity does not have a different average daily production rate than grouping all activities with unlimited orepass storage capacity (based upon the 1372 ± 50 and the 1337 ± 43 values given in Table 5.2). Therefore, as suspected, train tramming is not holding back production; however, the limited orepass storage capacity for the mining blocks considered is holding back production (i.e., the system bottleneck is the orepass storage capacity). However, when the orepass storage capacity is sufficiently increased, then drilling and blasting becomes the system bottleneck.

Finally, as shown in Table 5.2, an average daily mucking and hauling production rate of 1337 ± 43 (roughly 1300 to 1400 tons per day) can be expected when there is sufficient orepass storage capacity. If there is an additional 150 tons per day of development ore, then an average daily ore train tramming production rate of 1449 ± 44 (roughly 1400 to 1500 tons per day) can be expected.

5.4 OTHER SCENARIOS

Various other scenarios were performed using an earlier version of the simulation model (Version 1.1), such as, alternative LHD configurations, increasing the average daily production rate, and alternative mining block mucking rules. The real world system has changed since these scenarios were performed, thus resulting in Version 1.2. The primary difference between Version 1.1 and Version 1.2 is the overall train tramming capacity. (That is, Version 1.1 had six ore cars 80 percent of the time, and four ore cars 20 percent of the time, without the option of ore train tramming on the third shift. Version 1.2 has six ore cars 100 percent of the time, with ore train tramming on the third shift if there is no rock to tram. In addition, a backfilling priority was added to Version 1.2.) At the present time, there is no reason to rerun the scenarios discussed in this section with Version 1.2.

Alternative LHD Configurations

The alternative LHD configurations scenarios compared a 9 (primary) and 6 (backup) cubic yard LHD system with a 6 (primary) and 6 (backup) cubic yard LHD system. These scenarios were performed for both a 1 and 2 drill/crew system for various LHD return hauling distances. The results from this study showed a greater decrease in the average daily production rate as the LHD return hauling distances increased for the 6 and 6 cubic yard LHD system as compared to the 9 and 6 cubic yard system. Conversely, for lower LHD return hauling distances, there was no statistical difference in the average daily production rate between the systems.

Increasing the Average Daily Production Rate

This study looked at ways to increase the average daily production rate by focusing on drilling and blasting since it is the activity bottleneck. The following scenarios were examined:

1. Basecase, which is used as a benchmark against the others,
2. Increasing the mechanical availability of the drills from 69 to 85 percent,

3. Increasing the penetration rate of the drills from 40 to 50 feet per hour, and
4. Increasing the drill factor by reducing the number of feet drilled per blast section of ore by 10 percent, the number of holes drilled per blast section of ore by 10 percent, and the number of feet loaded per blast section of ore by 5 percent.

The results from these scenarios showed increasing either the penetration rate or the drill factor, did not result in a statistical difference in the average daily production rate. On the other hand, increasing the mechanical availability of the drills did result in a statistical increase in the average daily production rate.

Alternative Mining Block Mucking Rules

The alternative mining block mucking rules scenarios investigated the effect of altering the priority order in which mining blocks are mucked. For example, in a 2 drill/crew system, if there are two mining blocks that can be mucked at the same time, what is the preferred order to muck them. The potential mucking orders, or rules, considered were:

1. First blasted priority, that is, the mining block, or blast section of ore, blasted first is mucked first.
2. Block priority, that is, the mining blocks, or blast sections of ore, are mucked in a predetermined set sequence (i.e., the order in which they are entered in the data file).
3. Smallest blast priority, that is, the blast section of ore with the smallest amount of ore is mucked first.
4. Largest blast priority, that is, the blast section of ore with the largest amount of ore is mucked first.

The results from this study showed no statistical difference in the average daily production rate for the mining block mucking rules considered.

CHAPTER 6

CONCLUDING REMARKS

6.1 RESEARCH WORK ACCOMPLISHED

A prototype computer simulation model of underground production mining has been developed for a specific mining complex found at Inco Limited, Manitoba Division. The model can simulate up to ten different mining blocks, each of which can be divided into as many blast sections of ore as necessary. The simulation model was built assuming all necessary orebody infrastructure was already developed. Within the existing orebody infrastructure, the major activities and pieces of equipment required for underground production mining are modelled. Basically, the major activities modelled have been divided into four categories: 1) drilling and blasting, which includes relocating the drill, drilling holes, cleaning holes, measuring holes, loading explosives, and guarding the blast, 2) mucking and hauling, 3) train tramming, and 4) backfilling. The model can accommodate up to four chimneys, three of which must be defined as orepasses and one as a rockpass.

The model was built using SLAM II and contains user written FORTRAN code used to generate the following four output files: 1) SUMMARY, 2) RUNS, 3) AMOUNTS and 4) COSTS. The SUMMARY output file provides information on the averages and confidence intervals for the following: the mechanical availability, physical availability, use of availability, and effective utilization for each piece of equipment, along with the hours and shifts used to calculate these values; the daily tons and pounds of nickel mucked and hauled by the LHDs; the daily tons of ore and rock trammed; the mucking tons per day on a per mining block basis (up to 10 mining blocks); the production schedule table for various events for each mining block (up to 10 mining blocks); and the activity times for drilling and blasting, mucking and hauling, and backfilling for each mining block (up to 10 mining blocks). The RUNS output file provides similar information as the SUMMARY file, except it is for each simulation run performed, rather than averages and the confidence intervals. The AMOUNTS file provides the following information for each simulation run: the daily and cumulative amounts of ore and nickel mucked and hauled by the LHDs; the daily and cumulative amounts of ore trammed by the ore train; the daily and cumulative amounts of rock trammed by the rock train; and the tons of rock in the rockpass at the end of the day. The COSTS file provides preliminary cost information on drilling, blasting, and mucking and hauling for the last simulation run only.

Logic diagrams are provided for the following segments: drilling and blasting; mucking and hauling; backfilling; train tramming; blast delay and check vents; equipment breakdowns; and shifts. These logic diagrams serve as a type of flow diagram to facilitate explanations of model logic used.

The model was used on four typical mining blocks found at the 83 orebody in the Birchtree Mine. A five day period on the first simulation run for the basecase scenario was chosen for selective tracing of the drilling and blasting, mucking and hauling, and train tramming activities. Graphical time charts were constructed from the traces for these activities, which served as a communication tool to present some of the operational characteristics of the model.

In addition, various other scenarios were performed on the same four mining blocks, modified as noted, to determine activity and system bottlenecks with the average daily production rate used as the performance measure. Based upon these scenarios, the following conclusions were made: drilling and blasting is the activity bottleneck; orepass storage capacity is the system bottleneck for the mining blocks considered; when the orepass storage capacity is sufficiently increased, then drilling and blasting becomes the system bottleneck; roughly an average daily mucking and hauling production rate between 1300 and 1400 tons per day can be expected when there is sufficient orepass storage capacity; and when there is an additional 150 tons per day of development ore, then roughly an average daily ore train tramming production rate between 1400 and 1500 tons per day can be expected.

The above conclusions are based upon specific mining blocks and model assumptions, of course, both internal and external inputs to the model can be changed to examine various other mining blocks and types of system configurations. In addition, the model structure can be altered as necessary in order to accommodate changes in the real world system, as was the case from Version 1.1 to Version 1.2. The degree of effort for such modifications, of course, depends on the magnitude of the change and the familiarity with the model.

6.2 DIRECTIONS FOR FUTURE WORK

As with most computer programming projects, there are usually a variety of refinements and/or additions that could be made to better the "final product"; however, insufficient time is a limiting factor. Consequently, a list of suggestions for future development are listed below:

- The recommended program refinements identified in Appendix E are incorporated into the model.
- Graphical production schedules are generated using the model. Currently, the model produces a numerical production schedule table for selected activity events. Perhaps the model could be amended for the purpose of producing production schedules in both graphical, as well as tabular form. These production schedules could be based upon the selected number of simulation runs, perhaps similar to that given in Figure 5.1.
- The FORTRAN code for the production operating costs is updated to provide results based upon 1 to 10 simulation runs. Presently the model contains code that produces very rough numbers based only upon the last simulation run.
- Automate the activity time charts presented in Chapter 5 (i.e., Figures 5.2 to 5.5). This could be accomplished by writing the time of various events to specific files, then later reading this information into a spreadsheet package or perhaps generating the graphs with user written code in the OTPUT file. These activity time charts could only be based upon individual simulation runs.
- External input data to the model is linked to a spreadsheet. Perhaps setting up a spreadsheet in this manner would simplify the external input data process and make it more user friendly. Presently, the external input data is primarily entered in as a matrix of numbers using the data file.
- Animation is added to the model. Animations can be created within SLAMSYSTEM using the Facility Builder for the design of the background and symbols, and the Script Builder to specify which actions should occur when a particular simulation event happens.
- Dynamic scheduling of productive time within the model. That is, scheduling of weekends and the third shift as productive time would be done within the model only

as necessary. Presently, all activities have their shifts set at the start of the first simulation run.

- Preventive maintenance and equipment breakdowns are modelled in greater detail and as separate occurrences. Presently, the minor maintenance done daily on-site by the operator has been grouped into the two hour per shift shutdown time, and the major preventive maintenance performed at the garage by a mechanic has been grouped into the equipment breakdown time.
- The percentage mucked and hauled by the LHD resource before the blastroom resource is released is made a function of the room, or cavity, available. Presently, the blastroom resource is made available once 25 percent of the previous blast section of ore for the mining block has been mucked.
- The drill penetration rate is made a function of rock hardness on a per blast section of ore basis. This could be done by adding an additional attribute number for rock hardness and appropriately adjusting the code. Presently, a fixed penetration rate of 40 feet per hour is used for all blast sections of ore.
- Optimal block mining sequences are determined using the model. Of course, geological considerations greatly determine the sequence in which mining blocks are mined; however, for situations where the geological considerations are not a determining factor, then the optimal sequencing based upon the selected performance measure(s) can be investigated. Presently, alternative sequencing of mining blocks can be achieved by changing the order in which they appear in the input data file.
- The model is expanded to include additional activities, such as skipping ore to the surface.

Since computer simulation is an excellent tool for evaluating complex systems such as underground production mining, it is highly recommended that further research work be performed in this area.

REFERENCES

Banks, J., "Selecting Simulation Software", *In Proceedings of the 1991 Winter Simulation Conference*, pp. 15-20, 1991.

Banks, J., "Simulation Languages and Simulators", *In Proceedings of the 1992 Winter Simulation Conference*, pp. 88-96, 1992.

Bruno, R., P. Massacci, and G. Raspa, "Test of Different Production Plans and Production Scheduling in a Rock Salt Deposit", *21st Application of Computers and Operations Research in the Mineral Industry*, Littleton, CO, USA, pp. 397-406, 1989.

Chatterjee, P. K., G. Johnston, and S. Holguin, "Computer-aided schedule optimization in room-and-pillar and longwall operations", *Mining Engineering (Littleton, Colorado)*, V. 39, No. 6, pp. 433-436, 1987.

Edwards, J. B., N. Mort, and R. J. Hollands, "The Use of Animation in the Simulation of Coalmine Production Systems", *In the Proceedings of the 3rd International Conference on Competitive Performance Through Advanced Technology - FACTORY 2000*, IEEE Conference Publication n. 359, pp. 278-284, 1992.

Elbrond, J., Workshop notes, University of Queensland, Brisbane, Australia, 1977.

Hayashi, F., and D. Robinson, "Computer Simulation of Mine Rail Haulage System", *In the Proceedings of the 1981 Winter Simulation Conference*, pp. 121-127, 1981.

Hoare, R. T., and R. J. Willis, "A Case Study of Animated Computer Simulation in the Australian Mining Industry", *Journal of the Operational Research Society*, Vol. 43, No. 12, pp. 1113-1120, 1992.

Jardine, G. H., and B. J. Evans, "Mine scheduling using interactive graphics", *Computer Applications in the Mineral Industry*, In Proceedings of the First Canadian Conference, pp. 19-21, 1988.

Kelly, C. M., "Use of Underground Haulage Simulators - A Case Study of the Loveridge Mine", *19th Application of Computers and Operations Research in the Mineral Industry*, University Park, Pennsylvania, USA, pp. 203-207, 1986.

Lavrencic, J. B., "MINSIM - the MINe SIMulator", *21st Application of Computers and Operations Research in the Mineral Industry*, Littleton, CO, USA, pp. 653-661, 1989.

Law, A. M., and W. D. Kelton, *Simulation Modelling and Analysis*, Second Edition, McGraw-Hill, Inc., New York, 1991.

Law, A. M., and S. G Vincent, *UNIFIT II User's Guide*, Averill M. Law and Associates, 1991.

Lee, C., and J. M. Mutmansky, "A Simulation Model for Longwall Systems Utilizing Shearers", *In the Proceedings of the Second Conference on the Use of Computers in the Coal Industry*, Tuscaloosa, AL, USA, pp. 165-175, 1985.

Lilegdon, W. R., "Introduction to SLAMSYSTEM®", *In the Proceedings of the 1991 Winter Simulation Conference*, pp. 118-121, 1991.

Litke, A. B., D. Laroche, D. Strong, and J. Szymanski, "A Computer Simulation Model of Underground Production Mining: A Tool for Benchmarking", APCOM XXIV -

International Symposium on the Application of Computers and Operations Research in the Mineral Industries, Montreal, Que., Canada, Vol. 3, pp. 402-409, 1993.

Martin, D. L., J. D. Sabuda, "Increasing Productivity in Mining Operations through Simulation", Presented at the 88th Annual Northwest Mining Association Convention, Spokane, WA, 1982.

Nan, Y., and Wilke, F. L. "Computer Simulation Analysis of Underground Coal-Mine Main Production Systems in the Peoples Republic of China", *19th Application of Computers and Operations Research in the Mineral Industry*, University Park, Pennsylvania, USA, pp. 196-202, 1986.

O'Reilly, J. J., "SLAM II® Tutorial", *In Proceedings of the 1991 Winter Simulation Conference*, pp. 112-117, 1991.

O'Reilly, J. J., and N. K. Ryan, "Introduction to SLAM II® and SLAMSYSTEM", *In Proceedings of the 1992 Winter Simulation Conference*, pp. 352-356, 1992.

Potter, M., M. Richardson, R. Hoare, and R. Koenig, "Computer Simulation of Stoping Methods at the Z.C. Mines Broken Hill Operations", *Proceedings of the AusIMM Conference*, Sydney, NSW, pp. 117-121, 1988.

Pritsker, A. A. B., *Introduction to Simulation and SLAM II*, Third Edition, Halsted Press, John Wiley & Sons Inc., New York, 1986.

Pritsker, A. A. B., C. E. Sigal, and R. D. J. Hammesfahr, *SLAM II Network Models for Decision Support*, Prentice-Hall Inc., A Division of Simon & Schrester, Englewood Cliffs, New Jersey, 1989.

Schriber, T. J., *An Introduction to Simulation*, John Wiley & Sons, New York, 1991.

Sense, J. J., "Scheduling Equipment - Including Utilization and Availability", *Surface Mining*, Editor AIME, 1972.

Sinha, D. K., A. K. Pal, and S. Mitra, "Planning of Underground Coal Clearance System by System Simulation", *Journal of the Institution of Engineers (India)*, Part MI: Mining Engineering Division, V. 68, Pt. 2, pp. 43-46, 1988.

Sturgul, J. R., and R. K. Singhal, "Using the personal computer to simulate mining operations", *Computer Applications in the Mineral Industry*, In Proceedings of the First Canadian Conference, pp. 439-442, 1988.

Sturgul, J. R., and J. F. Harrison, "Using a Special Computer Language for Simulating Coal Mines", *The Coal Journal*, No. 18, pp. 21-28, 1987.

Suboleski, S. C., and J. R. Lucas, "Simulation of Room and Pillar Face Mining Systems", *A Decade of Digital Computing in the Mineral Industry - A Review of the State of the Art*, Society of Mining Engineers of AIME, pp. 385-400, 1969.

Swain, J., "Simulation Software Survey", *OR/MS Today*, October, pp. 81-102, 1991.

Szymanski, J., D. Laroche, A. Murchie, and B. Maynard, "A Simulation-Optimization Model of Skip and Cage Operation", *MASSMIN 92*, Johannesburg, The South African Institute of Mining and Metallurgy, pp. 323-332, 1992.

Topuz, E., and E. Nasuf, "CONSIM - An Interactive Micro-Computer Program for Continuous Mining Systems", *In the Proceedings of the Second Conference on the Use of Computers in the Coal Industry*, Tuscaloosa, AL, USA, pp. 183-192, 1985.

Topuz, E., E. Nasuf, and N. Michalopoulos, "An Interactive Micro-Computer Simulator for Longwall Operations", *In the Proceedings of the Second Conference on the Use of Computers in the Coal Industry*, Tuscaloosa, AL, USA, pp. 176-182, 1985.

Touwen, F. H., and N.C. Joughin, "The Simulation of Underground Stoping and Transport Operations in Gold Mining", *Application of Computer Methods in the Mineral Industry*, In Proceedings of the Tenth International Symposium, Johannesburg, Republic of South Africa, pp. 231-236, 1973.

Valenzuela, C. A., and Dr. G. E. Whitehouse, "The Application of Q-GERT to Materials Handling Problems in Open Pit Mining", *Modelling and Simulation*, Proceedings of the Ninth Annual Pittsburgh Conference, pp. 273-273, 1978.

Vincent, S. G., and A. M. Law, "UniFit II: Total Support for Simulation Input Modeling", *In the Proceedings of the 1991 Winter Simulation Conference*, pp. 136-142, 1991.

Yue, N., H. Xilan, and L. Yongzun, "Computer Simulation Study of a Belt Transportation System in an Underground Coal Mine", *Mining Science and Technology*, Proceedings of an International Symposium organized by The China Institute of Mining and Technology, Beijing, People's Republic of China, pp. 218-223, 1987.

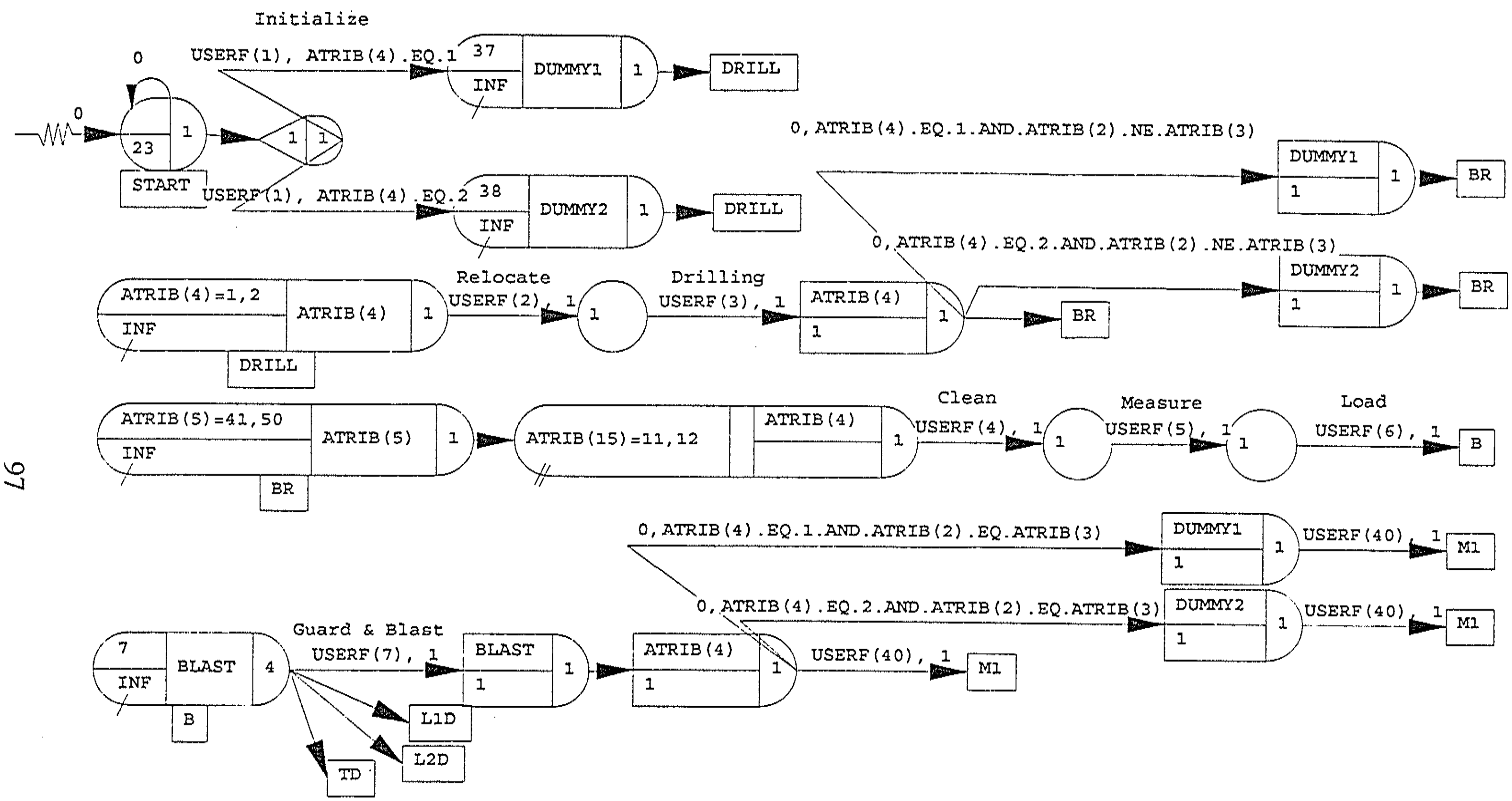
Zaikang, L., "Computer Simulation and Its Application in the Extraction, Conveyance and Hoisting System of Coal Mines", *In the Proceedings of the Second Conference on the Use of Computers in the Coal Industry*, Tuscaloosa, AL, USA, pp. 214-222, 1985.

Zaikang, L., "The Design of the Main Belt and Bunker Capacities Using Computer Simulation", *Mining Science and Technology*, Proceedings of an International Symposium organized by The China Institute of Mining and Technology, Beijing, People's Republic of China, pp. 224-229, 1987.

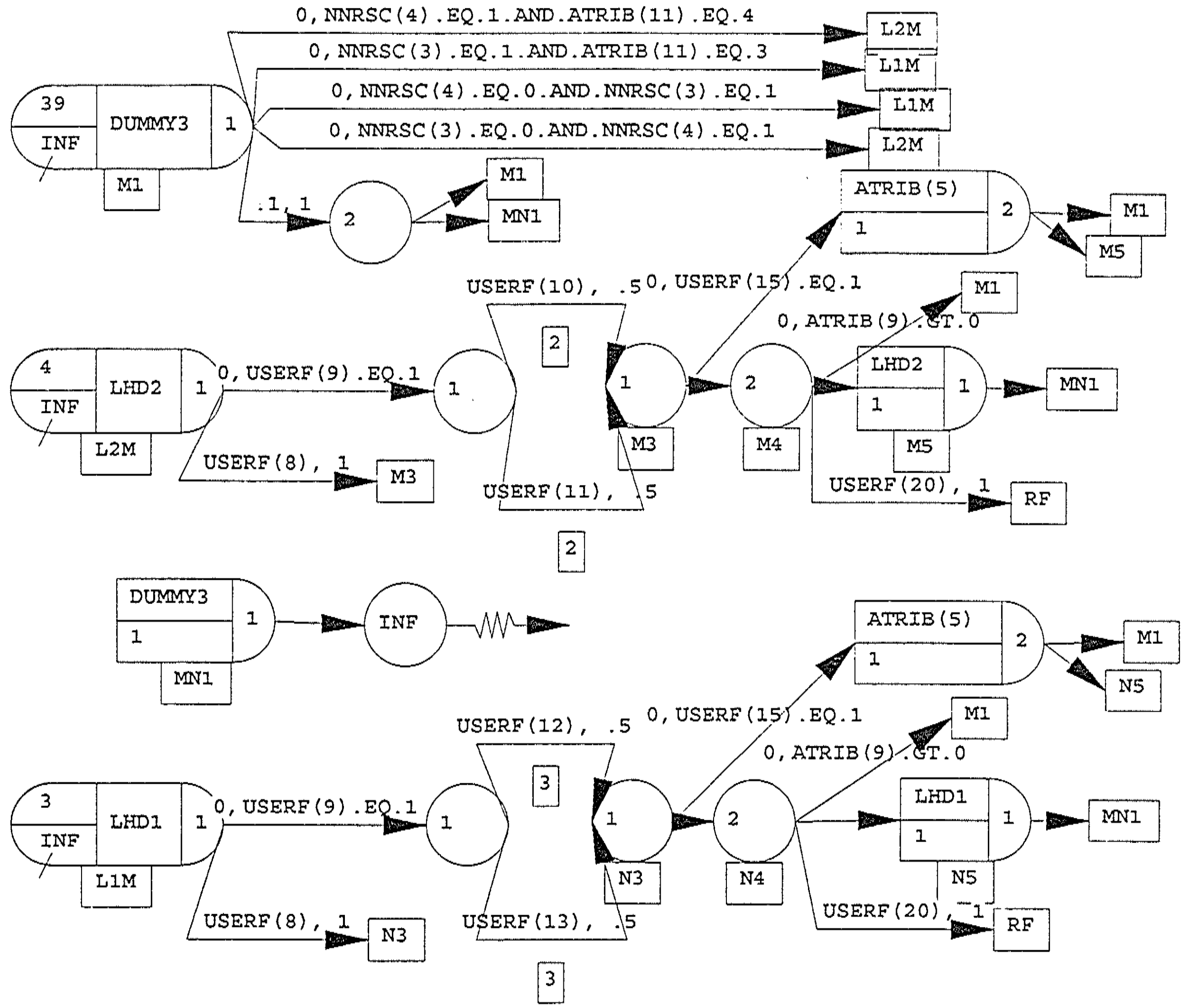
APPENDIX A: SIMULATION PROGRAM

Appendix A contains the simulation program, as well as explanations for the attribute numbers and XX( ) variables. It has been divided as follows:

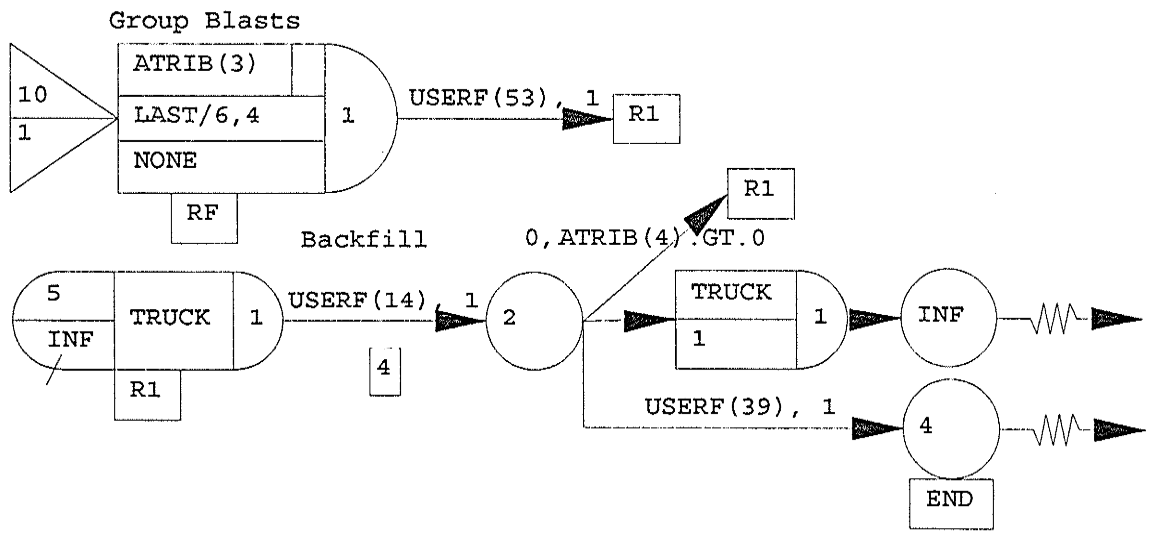
Appendix A.1: Network



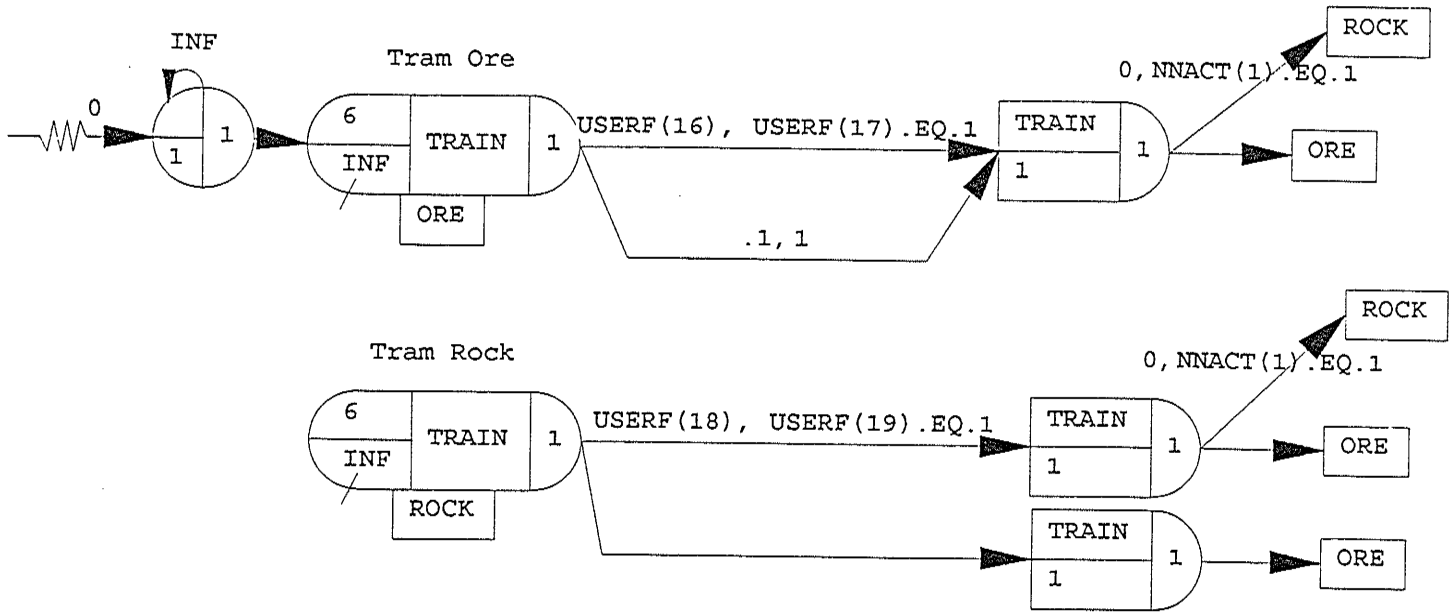
Drilling and Blasting Segment



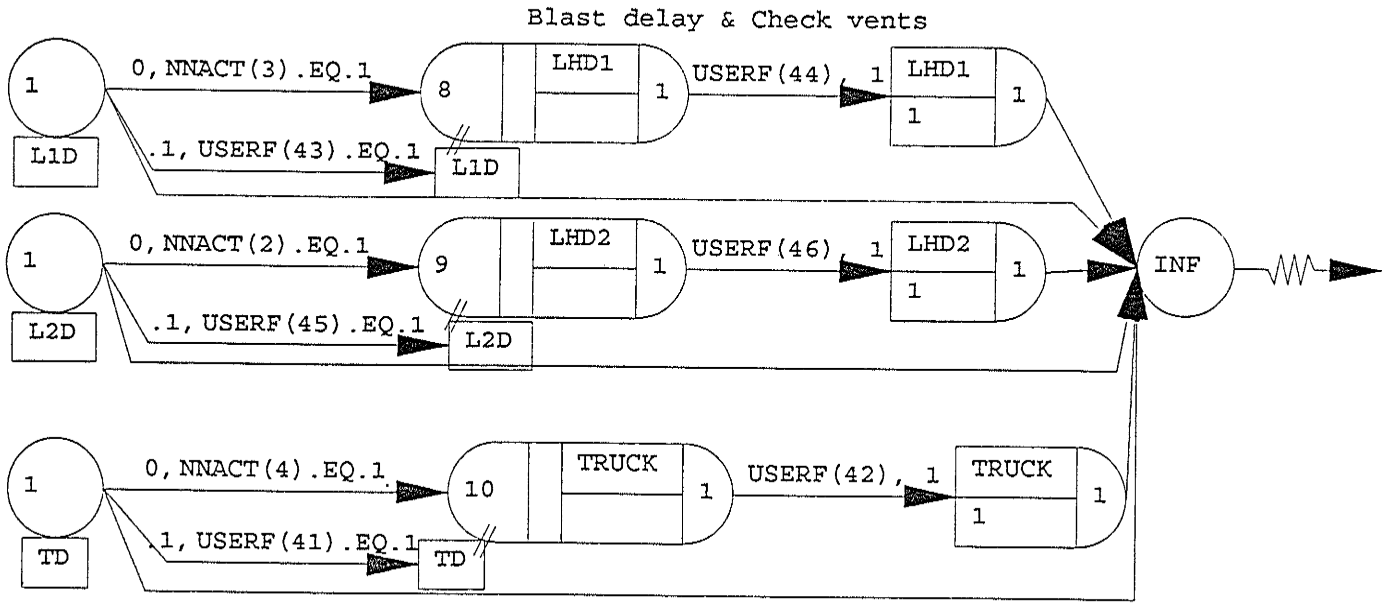
Mucking and Hauling Segment



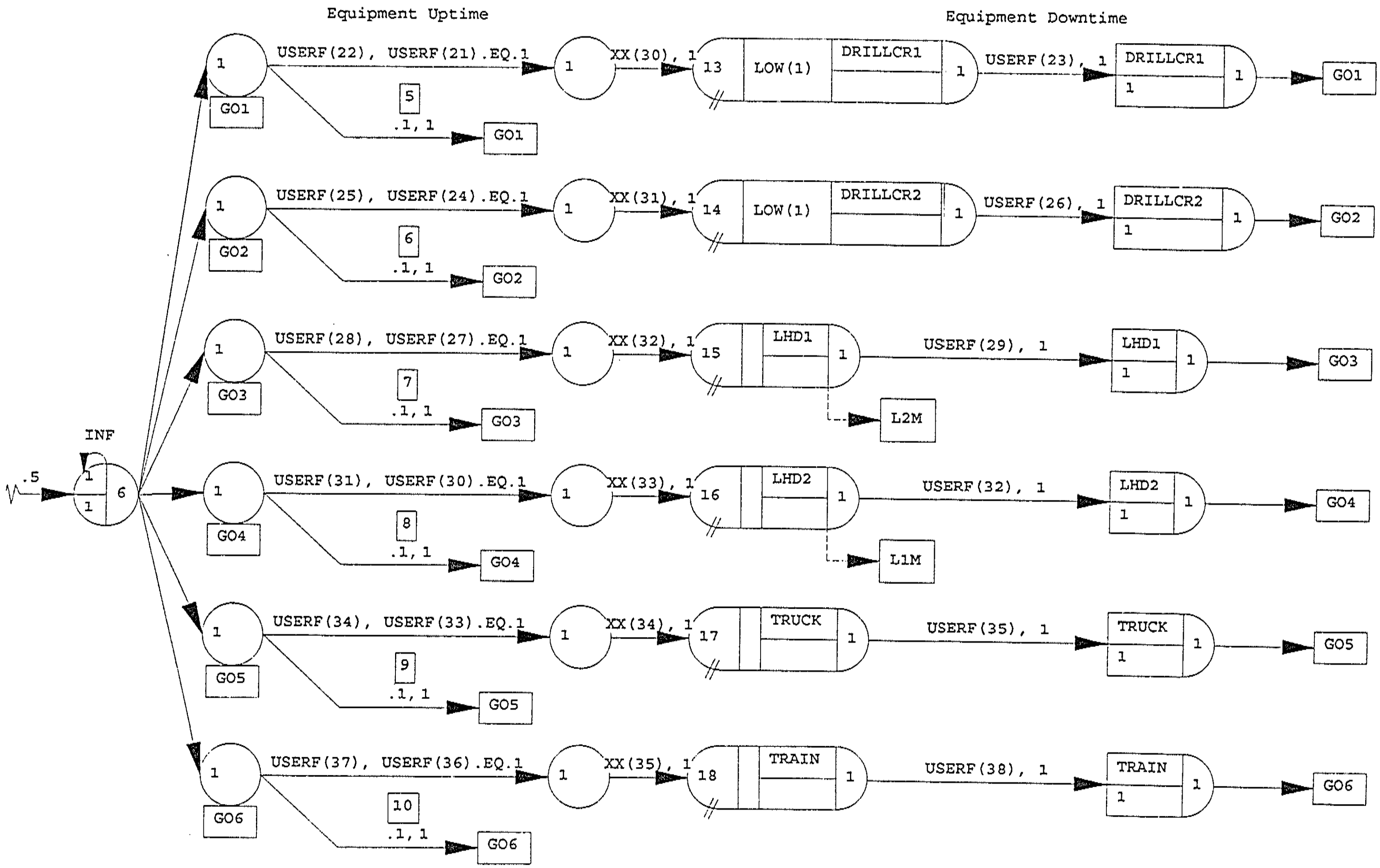
Backfilling Segment



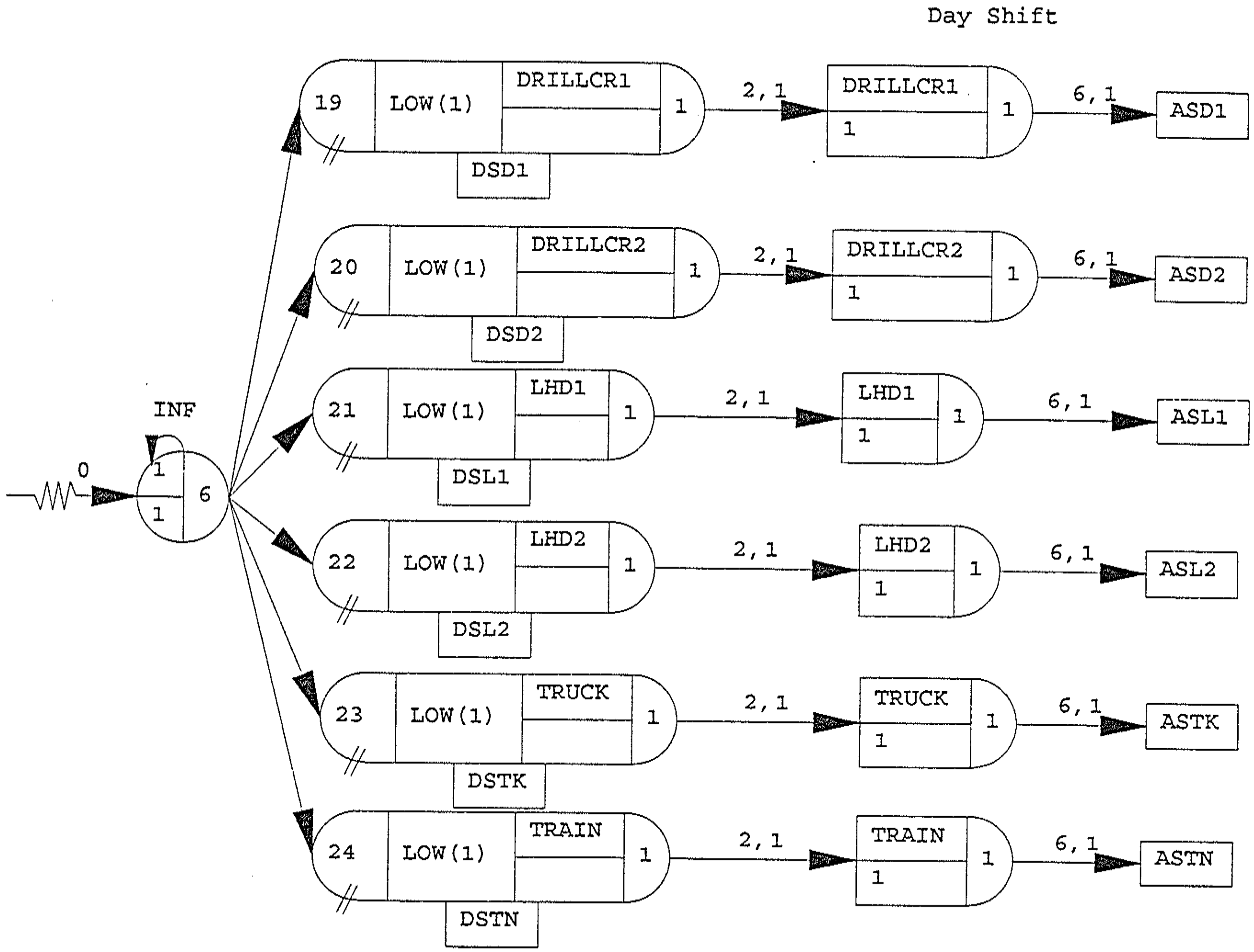
Train Tramming Segment



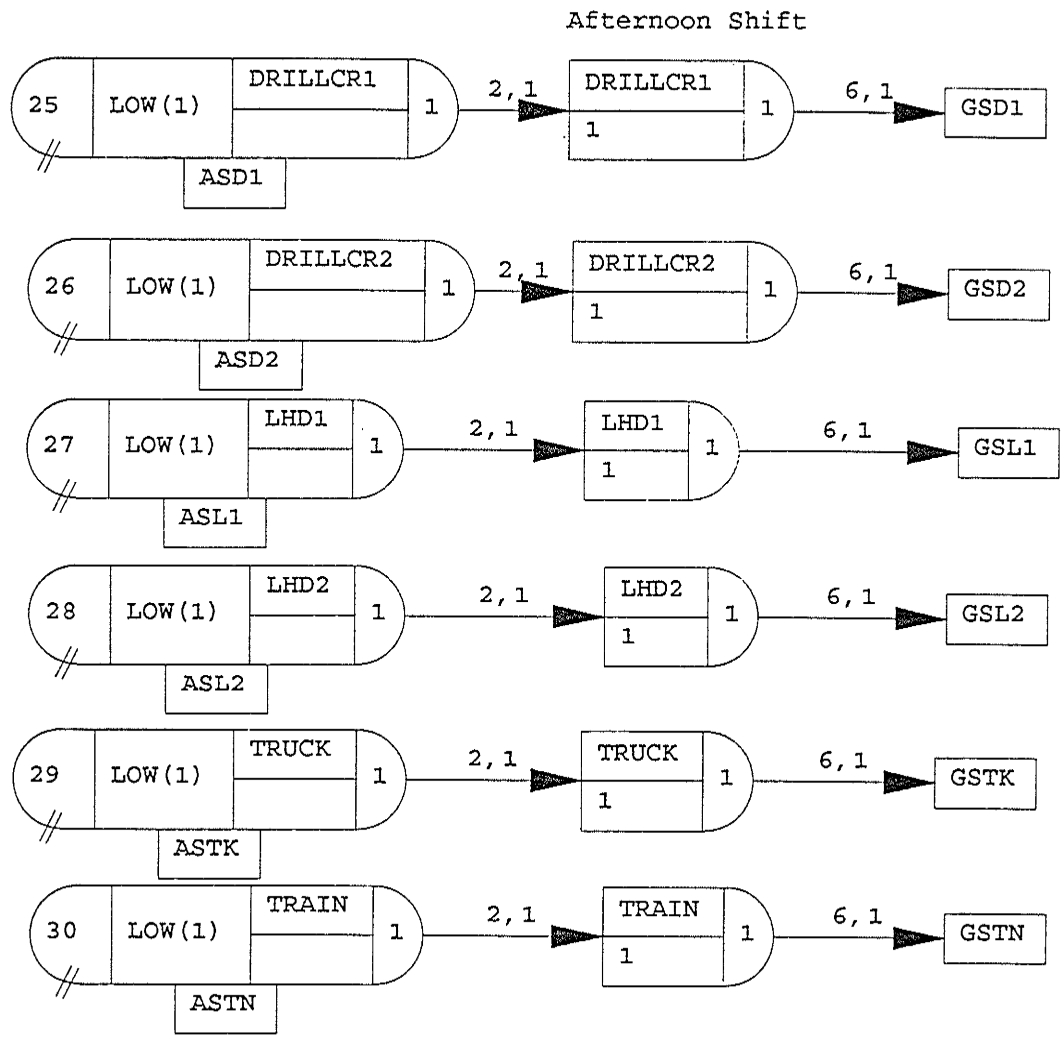
Blast Delay and Check Vents Segment



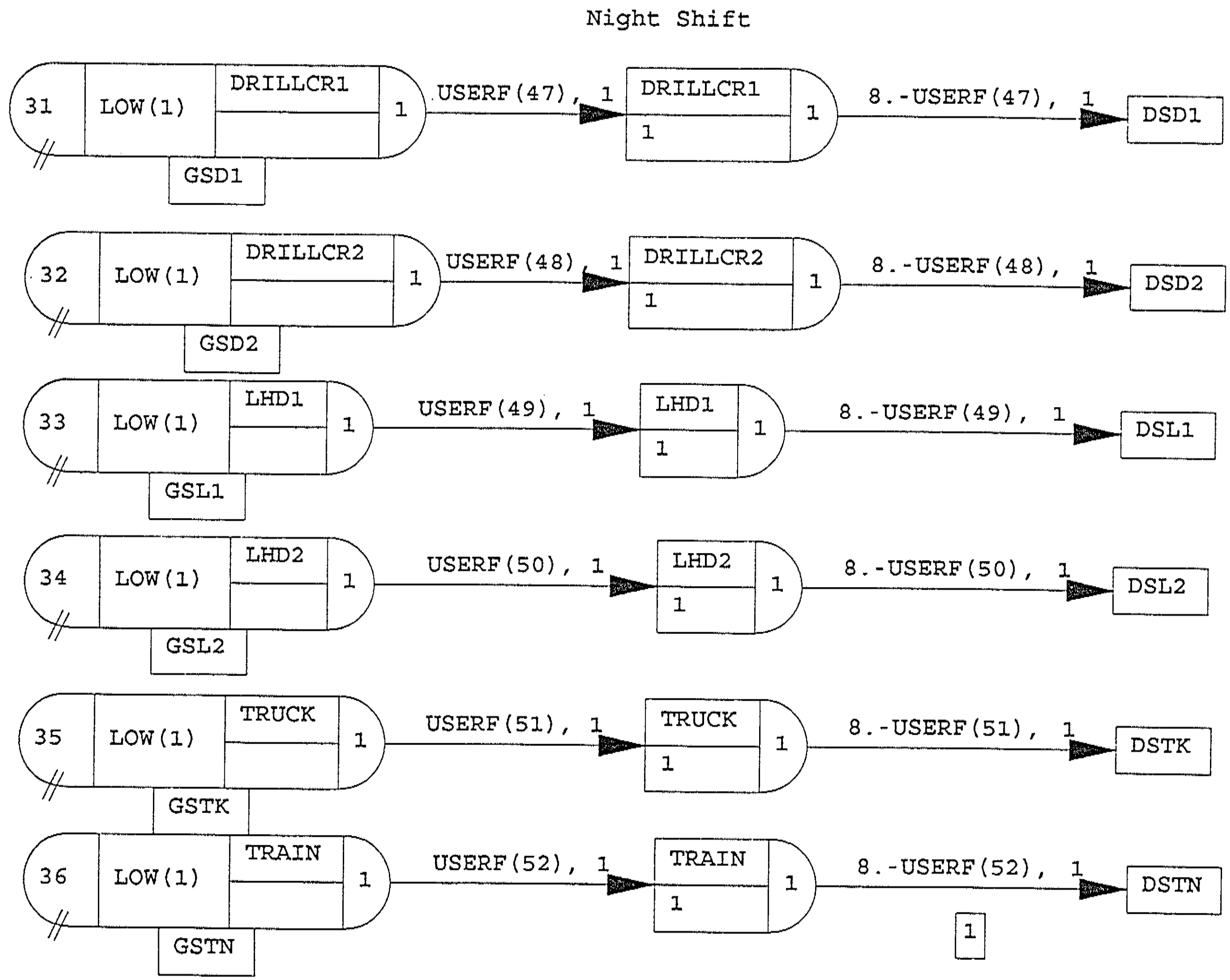
Equipment Breakdowns Segment



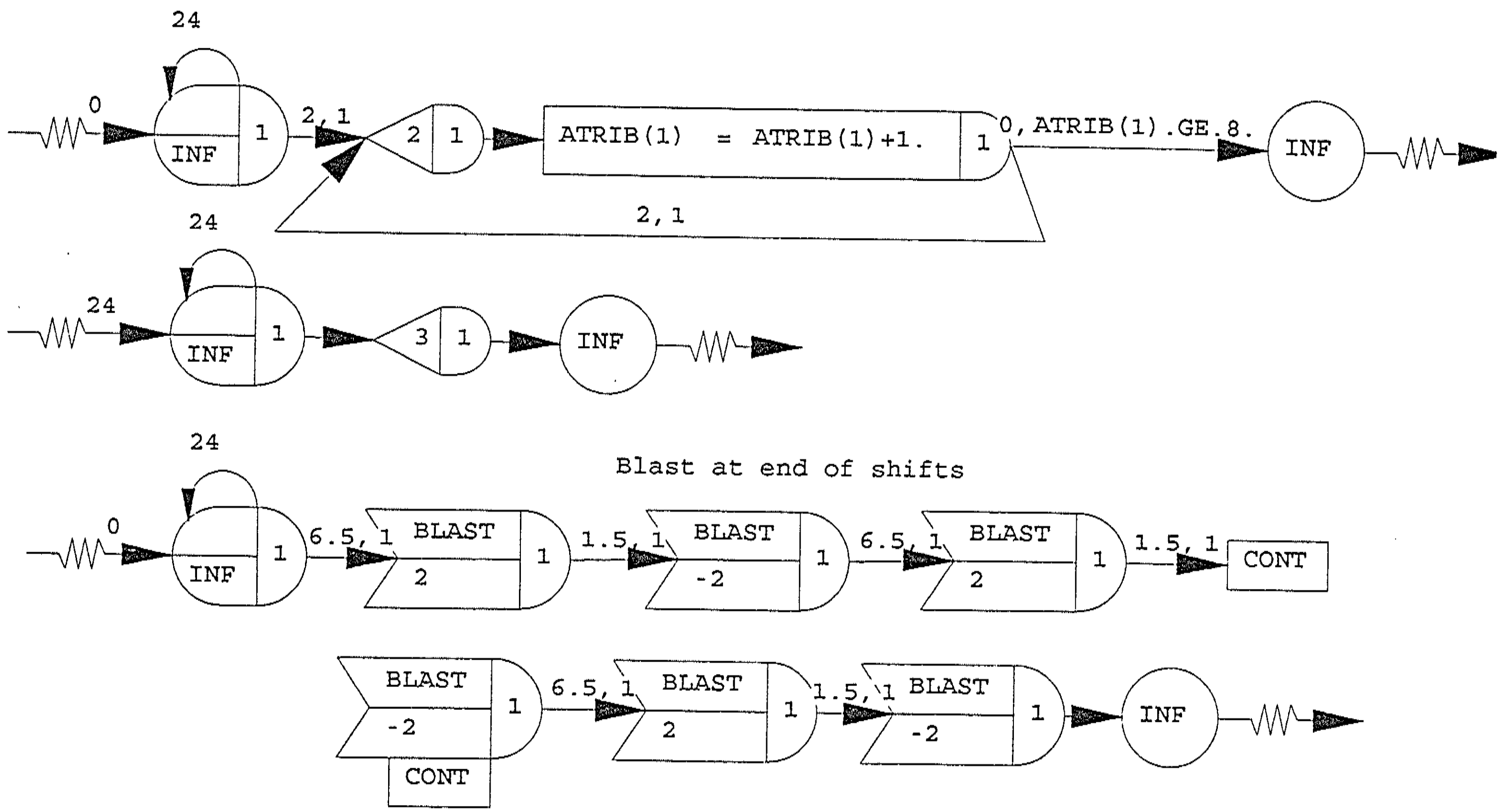
Shifts Segment (1 of 3)



Shifts Segment (2 of 3)



Shifts Segment (3 of 3)



Miscellaneous Segment

1	DRILLCR1	1	31	25	19	13	11	1
2	DRILLCR2	1	32	26	20	14	12	2
3	LHD1	1	33	27	21	15	8	3
4	LHD2	1	34	28	22	16	9	4
5	TRUCK	1	35	29	23	17	10	5
6	TRAIN	1	36	30	24	18	6	
7	BLAST	0	7					
8	DUMMY1	1	37					
9	DUMMY2	1	38					
10	DUMMY3	1	39					
41	BRB1	1	41					
42	BRB2	1	42					
43	BRB3	1	43					
44	BRB4	1	44					
45	BRB5	1	45					
46	BRB6	1	46					
47	BRB7	1	47					
48	BRB8	1	48					
49	BRB9	1	49					
50	BRB10	1	50					

Resource Segment

Appendix A.2: Control File

```
GEN,LITKE,PRODUCTION MINING,12/15/93,10,Y,Y,Y/Y,N,N/1,72;
LIMITS,52,15,62;
INTLC,XX(1)=1;
ARRAY(1,15);
ARRAY(2,15);
ARRAY(3,15);
ARRAY(4,15);
ARRAY(5,15);
ARRAY(6,15);
ARRAY(7,15);
ARRAY(8,15);
ARRAY(9,15);
ARRAY(10,15);
ARRAY(11,15);
ARRAY(12,15);
ARRAY(13,15);
ARRAY(14,15);
ARRAY(15,15);
ARRAY(16,15);
ARRAY(17,15);
ARRAY(18,15);
ARRAY(19,15);
ARRAY(20,15);
ARRAY(21,15);
ARRAY(22,15);
ARRAY(23,15);
NETWORK;
PRIORITY/39,LVF(1)/5,LVF(8);
INITIALIZE,,6000,Y/1;
FIN;
```

Appendix A.3: Input Data File

```
1 1 5 1 1500  614  8  100  142 1.42 4 25 600 1400  50
1 2 5 1  100  734 10  293 1124 1.42 4 25 600 1400  50
1 3 5 1  100  360  5  432 2660 1.42 4 25 600 1400  50
1 4 5 1  100 1102 13  646 3660 1.42 4 25 600 1400  50
1 5 5 1  100    0  0  810 5397 1.42 4 25 600 1400  50
2 1 6 2 1500  720 10  108  180 1.97 4 26 470 1400  50
2 2 6 2  100  900 12  360  900 1.97 4 26 470 1400  50
2 3 6 2  100  432  6  440 3600 1.97 4 26 470 1400  50
2 4 6 2  100  540  8  440 3600 1.97 4 26 470 1400  50
2 5 6 2  100    0  0  612 4320 1.97 4 26 470 1400  50
2 6 6 2  100  972 16  720 5400 1.97 4 26 470 1400  50
3 1 7 1 1500  811 10  131  162 1.34 4 25 600 1400 100
3 2 7 1  100 1324 16  477 1172 1.34 4 25 600 1400 100
3 3 7 1  100  664  8  725 3288 1.34 4 25 600 1400 100
3 4 7 1  100  511  7  363 4623 1.34 4 25 600 1400 100
3 5 7 1  100    0  0 1177 7446 1.34 4 25 600 1400 100
3 6 7 1  100  907 14  690 4631 1.34 4 25 600 1400 100
3 7 7 1  100 1195 25  672 3169 1.34 4 25 600 1400 100
4 1 5 2 1500  755  9  176  150 1.55 4 20 550 1400 100
4 2 5 2  100  871 15  342  800 1.55 4 20 550 1400 100
4 3 5 2  100  400  7  655 3580 1.55 4 20 550 1400 100
4 4 5 2  100    0  0  834 3120 1.55 4 20 550 1400 100
4 5 5 2  100 1320 26 1049 7430 1.55 4 20 550 1400 100
150
11
150
26
```

Appendix A.4: Explanation of Attribute Numbers

The attribute numbers defined by the user in the data file are given in Table A.1, and those defined internally are given in Table A.2. As shown, depending on the location of an entity within the SLAM II network, more than one description may be used.

Table A.1 Attribute numbers defined by the user in the data file.

Description	Attribute
Mining block identifier (sequential from 1)	1
Blast section of ore identifier (sequential from 1 for each mining block)	2
Total number of blast sections of ore for each mining block	3
Drill/crew (enter code "1" for DRILLCR1, "2" for DRILLCR2)	4
Feet drill relocated for blast section of ore	5
Feet (additional) drilled for blast section of ore	6
Number of (additional) holes drilled for blast section of ore	7
Feet loaded for blast section of ore	8
Tons of ore from blast section of ore	9
Grade of ore from blast section of ore (percent)	10
Primary LHD (enter code "3" for 6 yard, code "4" for 9 yard)	11
Orepass/ore dump location I.D. number	12
Feet LHD travels to and from orepass/ore dump	13
Feet truck travels to and from chute	14
Offset/start time in hours (same for all blast sections of ore for mining block)	15

Table A.2 Attribute numbers defined internally within the program.

Description	Attribute
Tons of rockfill required (defined after USERF (40))	4
Mining block blastroom resource number (defined after USERF(2))	5
Priority value for backfilling (defined after USERF(40))	8
Preempt file number for drill/crew (defined after USERF(2))	15
Variable for 25% of blast section of ore mucked (defined after USERF(40))	15

Appendix A.5: Explanation of XX( ) Variables

The XX( ) variables used within the SLAM II program are given in Table A.3.

Table A.3 XX( ) Variables.

Description	Variable
Counter for reading input data file	XX(1)
Tons of ore mucked per day	XX(2)
Total tons of ore mucked	XX(3)
Pounds of nickel mucked per day	XX(4)
Total pounds of nickel mucked	XX(5)
Tons of ore trammed per day	XX(6)
Total tons of ore trammed	XX(7)
Tons of rock trammed per day	XX(8)
Total tons of rock trammed	XX(9)
Quantity in dump location I.D. #10	XX(10)
Quantity in dump location I.D. #11	XX(11)
Quantity in dump location I.D. #12	XX(12)
Quantity in dump location I.D. #13	XX(13)
Quantity in dump location I.D. #14	XX(14)
Quantity in dump location I.D. #15	XX(15)
Quantity in dump location I.D. #16	XX(16)
Quantity in dump location I.D. #17	XX(17)
Quantity in dump location I.D. #18	XX(18)
Quantity in dump location I.D. #19	XX(19)
Quantity in dump location I.D. #20	XX(20)
Quantity in dump location I.D. #21	XX(21)
Quantity in dump location I.D. #22	XX(22)
Quantity in dump location I.D. #23	XX(23)
Quantity in dump location I.D. #24	XX(24)
Quantity in dump location I.D. #25	XX(25)
Quantity in dump location I.D. #26	XX(26)
Quantity in dump location I.D. #27	XX(27)
Quantity in dump location I.D. #28	XX(28)
Quantity in dump location I.D. #29	XX(29)
Drill/Crew 1 third shift breakdown variable	XX(30)
Drill/Crew 2 third shift breakdown variable	XX(31)
LHD 1 third shift breakdown variable	XX(32)
LHD 2 third shift breakdown variable	XX(33)
Truck third shift breakdown variable	XX(34)
Train third shift breakdown variable	XX(35)

Appendix A.6: FORTRAN Code

The FORTRAN code has been divided as shown below:

Note that file VARIOUS contains the FORTRAN code for subroutines EVENT, LARGEST, MUCKHAUL and TRANSFER.

Appendix A.6.1 File COMVAR

```
C       **************************************************************
C       * COMVAR                                                     *
C       **************************************************************
C       *                                                            *
C       * COMVAR, SHORT FOR COMMON VARIABLES, IS USED TO ALLOW VARIOUS*
C       * SUBROUTINES TO DIRECTLY SHARE VARIABLES, WITHOUT HAVING TO *
C       * PASS THEM AS ARGUEMENTS.                                   *
C       *                                                            *
C       * SEE APPENDIX A FOR AN EXPLANATION OF XX() VARIABLES AND    *
C       * ATTRIBUTE NUMBERS.                                         *
C       *                                                            *
C       **************************************************************
C
C       BLKTONS()  : TONS OF ORE FOR BLOCK
C       CAPACITY(): CAPACITY OF OREPASS/ROCKPASS SECTION.
C       FTDRILL()  : NUMBER OF FEET DRILLED
C       PSC()      : PRODUCTION SCHEDULE CHART (COL,ROW,RUN)
C       TIME()     : TIME/HOURS FOR ACTIVITIES (COL,ROW,RUN)
C
C       ADODD      : AMOUNT OF DEVELOPMENT ORE DEPOSITED DAILY
C       ARDD       : AMOUNT OF ROCK DEPOSITED DAILY
C       BOT        : BLASTING OPERATION TIME
C       BT_D1      : BANKED TIME FOR DRILL 1
C       BT_D2      : BANKED TIME FOR DRILL 2
C       BT_LHD1    : BANKED TIME FOR LHD 1
C       BT_LHD2    : BANKED TIME FOR LHD 2
C       BT_TRUC    : BANKED TIME FOR TRUCK
C       BT_TRAM    : BANKED TIME FOR TRAMMING
C       DPT_D1     : DIRECT PRODUCTION TIME FOR DRILL 1
C       DPT_D2     : DIRECT PRODUCTION TIME FOR DRILL 2
C       DPT_LHD1   : DIRECT PRODUCTION TIME FOR LHD 1
C       DPT_LHD2   : DIRECT PRODUCTION TIME FOR LHD 2
C       DPT_TRUC   : DIRECT PRODUCTION TIME FOR TRUCK
C       DPT_TRAM   : DIRECT PRODUCTION TIME FOR TRAMMING
C       DDOID      : DEPOSIT DEVELOPMENT ORE I.D.
C       DRID       : DEPOSIT ROCK I.D.
C       EBD_LHD1   : END TIME FOR BLAST DELAY FOR LHD 1
C       EBD_LHD2   : END TIME FOR BLAST DELAY FOR LHD 2
C       EBD_TRUC   : END TIME FOR BLAST DELAY FOR TRUCK
C       IPT_D1     : INDIRECT PRODUCTION TIME FOR DRILL 1
C       IPT_D2     : INDIRECT PRODUCTION TIME FOR DRILL 2
C       IPT_LHD1   : INDIRECT PRODUCTION TIME FOR LHD 1
C       IPT_LHD2   : INDIRECT PRODUCTION TIME FOR LHD 2
C       IPT_TRUC   : INDIRECT PRODUCTION TIME FOR TRUCK
C       IPT_TRAM   : INDIRECT PRODUCTION TIME FOR TRAMMING
C       LEVEL      : CONFIDENCE LEVEL
C       CBLAST     : CURRENT BLAST
C       NBLASTS    : NUMBER OF BLASTS IN DATA FILE
C       NOCPT      : NUMBER OF ORE CARS PER TRAIN
C       NRCPT      : NUMBER OF ROCK CARS PER TRAIN
C       NRUNS      : NUMBER OF SIMULATION RUNS
C       PD_D1      : PREVIOUS DIRECT PRODUCTION TIME FOR DRILL 1
C       PD_D2      : PREVIOUS DIRECT PRODUCTION TIME FOR DRILL 2
C       PD_LHD1    : PREVIOUS DIRECT PRODUCTION TIME FOR LHD 1
```

```
C      PD_LHD2    : PREVIOUS DIRECT PRODUCTION TIME FOR LHD 2
C      PD_TRUC    : PREVIOUS DIRECT PRODUCTION TIME FOR TRUCK
C      PD_TRAM    : PREVIOUS DIRECT PRODUCTION TIME FOR TRAMMING
C      PI_D1      : PREVIOUS INDIRECT PRODUCTION TIME FOR DRILL 1
C      PI_D2      : PREVIOUS INDIRECT PRODUCTION TIME FOR DRILL 2
C      PI_LHD1    : PREVIOUS INDIRECT PRODUCTION TIME FOR LHD 1
C      PI_LHD2    : PREVIOUS INDIRECT PRODUCTION TIME FOR LHD 2
C      PI_TRUC    : PREVIOUS INDIRECT PRODUCTION TIME FOR TRUCK
C      PI_TRAM    : PREVIOUS INDIRECT PRODUCTION TIME FOR TRAMMING
C      RT_D1      : REPAIR TIME FOR DRILL 1
C      RT_D2      : REPAIR TIME FOR DRILL 2
C      RT_LHD1    : REPAIR TIME FOR LHD 1
C      RT_LHD2    : REPAIR TIME FOR LHD 2
C      RT_TRUC    : REPAIR TIME FOR TRUCK
C      RT_TRAM    : REPAIR TIME FOR TRAMMING
C      S_DRIL     : START TIME FOR DRILLING
C      S_MUCK     : START TIME FOR MUCKING AND HAULING
C      S_BACK     : START TIME FOR BACKFILLING
C      S3_D1      : 3RD SHIFT HOURS FOR DRILL 1
C      S3_D2      : 3RD SHIFT HOURS FOR DRILL 2
C      S3_LHD1    : 3RD SHIFT HOURS FOR LHD 1
C      S3_LHD2    : 3RD SHIFT HOURS FOR LHD 2
C      S3_TRUC    : 3RD SHIFT HOURS FOR TRUCK
C      S3_TRAM    : 3RD SHIFT HOURS FOR TRAMMMING
C      TLHD1      : TONS OF ORE HAULED BY LHD 1
C      TLHD2      : TONS OF ORE HAULED BY LHD 2
C      TRID       : TRAM ROCK I.D.
C      UT_D1      : UPTIME FOR DRILL 1
C      UT_D2      : UPTIME FOR DRILL 2
C      UT_LHD1    : UPTIME FOR LHD 1
C      UT_LHD2    : UPTIME FOR LHD 2
C      UT_TRUC    : UPTIME FOR TRUCK
C      UT_TRAM    : UPTIME FOR TRAMMING
C
       REAL PSC(10,7,10),TIME(5,6,10),BLKTONS(10),FTDRILL(10),NOCPT,
      +NRCPT,IPT_D1,IPT_D2,IPT_LHD1,IPT_LHD2,IPT_TRUC,IPT_TRAM
       INTEGER DRID,TRID,DDOID,CBLAST
C
       COMMON /UCOM1/ CAPACITY(29),PSC,TIME,BLKTONS,FTDRILL,
      +ARDD,ADODD,DRID,DDOID,TRID,NOCPT,NRCPT,NRUNS,NBLASTS,CBLAST,
      +DPT_D1,DPT_D2,DPT_LHD1,DPT_LHD2,DPT_TRUC,DPT_TRAM,
      +IPT_D1,IPT_D2,IPT_LHD1,IPT_LHD2,IPT_TRUC,IPT_TRAM,
      +RT_D1,RT_D2,RT_LHD1,RT_LHD2,RT_TRUC,RT_TRAM,
      +S3_D1,S3_D2,S3_LHD1,S3_LHD2,S3_TRUC,S3_TRAM,
      +BT_D1,BT_D2,BT_LHD1,BT_LHD2,BT_TRUC,BT_TRAM,
      +PD_D1,PD_D2,PD_LHD1,PD_LHD2,PD_TRUC,PD_TRAM,
      +PI_D1,PI_D2,PI_LHD1,PI_LHD2,PI_TRUC,PI_TRAM,
      +UT_D1,UT_D2,UT_LHD1,UT_LHD2,UT_TRUC,UT_TRAM,
      +EBD_LHD1,EBD_LHD2,EBD_TRUC,BOT,TLHD1,TLHD2,
      +S_DRIL,S_MUCK,S_BACK,LEVEL
```

Appendix A.6.2 File INTLC

```
C     **************************************************************
C     * SUBROUTINE:  INTLC                                         *
C     **************************************************************
C     *                                                            *
C     * INTLC IS CALLED BY THE SLAM II MAIN PROGRAM AT THE START OF *
C     * EACH SIMULATION RUN.  INTLC IS USED TO PERFORM VARIOUS     *
C     * FUNCTIONS AS OUTLINED BELOW.                               *
C     *                                                            *
C     * SEE "COMVAR" FOR VARIABLES NOT DEFINED BELOW.              *
C     *                                                            *
C     **************************************************************
C
C     BLADAT() : BLAST SECTION DATA
C
      SUBROUTINE INTLC
      REAL BLADAT(15)
      INTEGER ROW,COL
$INCLUDE:'PARAM.INC'
$INCLUDE:'SCOM1.COM'
$INCLUDE:'COMVAR.FOR'
      DATA CAPACITY/0,0,0,0,0,0,0,0,0,
     +213,213,0,0,0,
     +213,213,426,426,426,
     +213,0,0,0,0,
     +213,213,426,426,426/
      NBLASTS = 23
      NOCPT = 6.
      NRCPT = 10.
C*****THIRD SHIFT: ENTER 2 IF THERE IS A 3RD SHIFT, 8 OTHERWISE.
      S3_D1   = 2.
      S3_D2   = 2.
      S3_LHD1 = 8.
      S3_LHD2 = 8.
      S3_TRUC = 8.
      S3_TRAM = 2.
C*****READ BLAST DATA FILE INTO GLOBAL ARRAY.
      OPEN (UNIT=37,FILE='BLK5D78.DAT')
      DO 200 ROW = 1,NBLASTS
         READ (37,*) (BLADAT(COL),COL=1,15)
         CALL SETARY(ROW,BLADAT)
200   CONTINUE
C*****READ VARIABLES FROM DATA FILE.
      READ (37,*) ARDD
      READ (37,*) DRID
      READ (37,*) ADODD
      READ (37,*) DDOID
C*****CLOSE DATA FILE.
      CLOSE(37)
C*****INITIALIZE VARIABLES.
      EBD_LHD1 = 0.
      EBD_LHD2 = 0.
      EBD_TRUC = 0.
      UT_LHD1 = 0.
      BT_LHD1 = 0
```

```
      PD_LHD1 = 0.
      PI_LHD1 = 0.
      UT_LHD2 = 0.
      BT_LHD2 = 0.
      PD_LHD2 = 0.
      PI_LHD2 = 0.
      UT_D1 = 0.
      BT_D1 = 0.
      PD_D1 = 0.
      PI_D1 = 0.
      UT_D2 = 0.
      BT_D2 = 0.
      PD_D2 = 0.
      PI_D2 = 0.
      UT_TRUC = 0.
      BT_TRUC = 0.
      PD_TRUC = 0.
      PI_TRUC = 0.
      UT_TRAM = 0.
      BT_TRAM = 0.
      PD_TRAM = 0.
      PI_TRAM = 0.
      BOT    = 0.
      TLHD1 = 0.
      TLHD2 = 0.
      CBLAST = 0
C*****DO THE FOLLOWING FOR FIRST RUN.
      IF (NNRUN.EQ.1) THEN
C
         OPEN(UNIT=38,FILE='AMOUNTS')
         OPEN(UNIT=40,FILE='RUNS')
         OPEN(UNIT=41,FILE='SUMMARY')
         OPEN(UNIT=42,FILE='COSTS')
C
         WRITE(38,202)
         WRITE(40,204)
         WRITE(41,206)
         WRITE(42,208)
C
         WRITE(*,*) '  '
         WRITE(*,*) '*******83 OREBODY SIMULATION PROJECT*******'
         WRITE(*,*) '                 Version 1.2'
         WRITE(*,*) '                December 1993'
         WRITE(*,*) 'Please enter number of runs (1 to 10):'
         READ (*,*) NRUNS
         IF (NRUNS.EQ.0) THEN
            STOP
         END IF
         WRITE(*,*) 'Please enter confidence level (90,95,or 98):'
         READ (*,*) LEVEL
         CALL GETTIM (IHR,IMIN,ISEC,I100TH)
         WRITE(*,209) IHR,IMIN,ISEC,I100TH
      END IF
C*****IDENTIFY TRID.
      IF (DRID.GE.(10).AND.DRID.LE.(14)) THEN
          TRID = 10
      ELSE IF(DRID.GE.(15).AND.DRID.LE.(19)) THEN
          TRID = 15
      ELSE IF(DRID.GE.(20).AND.DRID.LE.(24)) THEN
```

```
          TRID = 20
      ELSE IF(DRID.GE.(25).AND.DRID.LE.(29)) THEN
          TRID = 25
      ELSE
          CALL ERROR(1)
      END IF
C*****PRINT HEADINGS FOR FILE AMOUNTS.
      WRITE (38,210)
      WRITE (38,212)NNRUN
      WRITE (38,214)
      WRITE (38,216)
      WRITE (38,218)
      WRITE (38,220)
      WRITE (38,222)
      WRITE (38,224)
      WRITE (38,226)
C
202   FORMAT(65X,'FILE:  AMOUNTS')
204   FORMAT(65X,'FILE:  RUNS')
206   FORMAT(65X,'FILE:  SUMMARY')
208   FORMAT(65X,'FILE:  COSTS')
C
209   FORMAT(  1X,'Start time of simulation: ',I2.2,':',I2.2,':',
     +                                         I2.2,':',I2.2)
C
210   FORMAT(/'********************************************')
212   FORMAT('RESULTS FOR RUN:',I3)
214   FORMAT (/12X,'MUCKING AND HAULING',18X,'TRAMMING')
216   FORMAT (7X,'-----------------------------',2X,
     +           '-----------------------------')
218   FORMAT (13X,'ORE',10X,'NICKEL',12X,'ORE',11X,'ROCK',
     +7X,'ROCKPASS')
220   FORMAT (7X,'--------------',1X,'--------------',
     +        2X,'--------------',1X,'--------------',2X,'---------')
222   FORMAT (2X,'TIME',2X,'DAILY',3X,'TOTAL',2X,'DAILY',3X,'TOTAL',
     +3X,'DAILY',3X,'TOTAL',2X,'DAILY',3X,'TOTAL',4X,'TOTAL')
224   FORMAT ('(Days)',1X,'(Tons)',2X,'(Tons)',1X,'(lbs.)',2X,'(lbs.)',
     +2X,'(Tons)',2X,'(Tons)',1X,'(Tons)',2X,'(Tons)',3X,'(Tons)')
226   FORMAT ('------',1X,'------',2X,'------',1X,'------',2X,'------',
     +2X,'------',2X,'------',1X,'------',2X,'------',3X,'------')
C
      RETURN
      END
```

Appendix A.6.3 File USERF

```
C      ***************************************************************
C      * FUNCTION: USERF                                             *
C      ***************************************************************
C      *                                                             *
C      * USERF IS PRIMARILY USED TO DEFINE ACTIVITY DURATIONS, CHECK *
C      * THE STATUS OF VARIABLES, AND TO UPDATE STATISTICS.  SEE     *
C      * INDIVIDUAL USERF NUMBERS FOR FURTHER DETAILS.               *
C      *                                                             *
C      * ARGUMENTS: IFN - INTEGER USER FUNCTION NUMBER               *
C      *                                                             *
C      * SEE "COMVAR" FOR VARIABLES NOT DEFINED BELOW.               *
C      *                                                             *
C      ***************************************************************
C
C      BKT     : TONS OF ORE PER LHD BUCKET
C      CARLOAD: TONS OF ORE/ROCK PER TRAIN CAR
C      CREW    : TIME CREW SPENDS PERFORMING ACTIVITY
C      DELAY   : DELAY TIME
C      DPT     : DIRECT PRODUCTION TIME
C      IBLK    : INTEGER BLOCK NUMBER
C      ID      : IDENTIFICATION NUMBER FOR OREPASS/ROCKPASS SECTION
C      IPT     : INDIRECT PRODUCTION TIME
C      PBRN    : PREVIOUS BLAST SECTION ROW NUMBER
C      PREP    : PREPARATION TIME
C      SPEED   : SPEED OF VEHICLE
C      TIME?   : PARTIAL TIME OF AN ACTIVITY
C
       FUNCTION USERF(IFN)
       INTEGER PBRN,IBLK
       REAL IPT
$INCLUDE:'PARAM.INC'
$INCLUDE:'SCOM1.COM'
$INCLUDE:'COMVAR.FOR'
       PBRN = ATRIB(2) - 1
       GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,
     +21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,
     +40,41,42,43,44,45,46,47,48,49,50,51,52,53),IFN
C*****INITIALIZE AND BLOCK PREPARATION TIME.
 1     XX(1) = XX(1)+1
       IF (ATRIB(2).EQ.ATRIB(3)) THEN
          PSC(ATRIB(1),1,NNRUN) = ATRIB(15)/24.
       END IF
       USERF = ATRIB(15)
       RETURN
C*****RELOCATE DRILL.
 2     IF (ATRIB(2).EQ.1.) THEN
          PSC(ATRIB(1),2,NNRUN) = TNOW/24.
          IF (ATRIB(1).EQ.1.) THEN
             S_DRIL = TNOW
          END IF
       END IF
       SPEED = 50. + 150.*BETA(2.,3.,1)
       PREP =  10. + 20.*BETA(1.333,2.,1)
       IPT = ATRIB(5)/(SPEED*60.) + PREP/60.
```

```
        IF (ATRIB(4).EQ.1.) THEN
           TIME(2,1,NNRUN) = TIME(2,1,NNRUN) + 4.*IPT/3.
        ELSE IF (ATRIB(4).EQ.2.) THEN
           TIME(2,2,NNRUN) = TIME(2,2,NNRUN) + 4.*IPT/3.
        ELSE
           CALL ERROR(1)
        END IF
        ATRIB(5) = ATRIB(1)+40.
        ATRIB(15) = ATRIB(4)+10.
        USERF = IPT
        RETURN
C*****DRILLING.
 3      TIME1 = 45. + 35.*BETA(1.56261,4.37564,1)
        TIME2 = 1.75 + 1.25*BETA(1.35,2.4,1)
        DPT = (ATRIB(7)*TIME1 + ATRIB(6)*TIME2)/60.
        IF (ATRIB(4).EQ.1.) THEN
           TIME(1,1,NNRUN) = TIME(1,1,NNRUN) + DPT
           TIME(2,1,NNRUN) = TIME(2,1,NNRUN) + DPT/3.
        ELSE IF (ATRIB(4).EQ.2.) THEN
           TIME(1,2,NNRUN) = TIME(1,2,NNRUN) + DPT
           TIME(2,2,NNRUN) = TIME(2,2,NNRUN) + DPT/3.
        ELSE
           CALL ERROR(1)
        END IF
        USERF = DPT
        RETURN
C*****CLEAN HOLES.
 4      TIME1 = 30. + 15.*BETA(1.38261,2.53043,1)
        TIME2 = 1.5 + 0.7*BETA(1.42857,3.57143,1)
        TIME3 = 3. + 4.54*BETA(1.55947,5.52053,1)
        IF (ATRIB(7).GT.0.) THEN
           IPT = 0.125*ATRIB(7)*TIME3/60.
           DPT = 0.125*(ATRIB(7)*TIME1 + ATRIB(6)*TIME2)/60.
        ELSE IF (ATRIB(7).EQ.0.) THEN
           IPT = 0.5*GETARY(PBRN,7)*TIME3/60.
           DPT = (0.5*GETARY(PBRN,7)*TIME1 +
     +             0.3*GETARY(PBRN,6)*TIME2)/60.
        ELSE
           CALL ERROR(1)
        END IF
        IF (ATRIB(4).EQ.1.) THEN
           TIME(2,1,NNRUN) = TIME(2,1,NNRUN)+4.*IPT/3.+DPT/3.
           TIME(1,1,NNRUN) = TIME(1,1,NNRUN) + DPT
        ELSE IF (ATRIB(4).EQ.2.) THEN
           TIME(2,2,NNRUN) = TIME(2,2,NNRUN)+4.*IPT/3.+DPT/3.
           TIME(1,2,NNRUN) = TIME(1,2,NNRUN) + DPT
        ELSE
           CALL ERROR(1)
        END IF
        USERF = IPT + DPT
        RETURN
C*****MEASURE HOLES.
 5      TIME1 = 3.5 + BETA(1.06667,1.15556,1)
        TIME2 = 2.4 + 0.8*BETA(1.43182,4.02273,1)
        IF (ATRIB(7).GT.0.) THEN
           CREW = (TIME1 + ATRIB(7)*TIME2)/60.
        ELSE IF (ATRIB(7).EQ.0.) THEN
           CREW = (TIME1 + GETARY(PBRN,7)*TIME2)/60.
        ELSE
```

```
          CALL ERROR(1)
       END IF
       BOT = BOT + 4.*CREW/3.
       USERF = CREW
       RETURN
C*****LOADING EXPLOSIVES.
 6     TIME1 = 63. + 174.*BETA(1.27586,4.72414,1)
       TIME2 = 2.8 + 3.2*BETA(1.18478,2.29348,1)
       TIME3 = 0.5 + 0.75*BETA(1.2,3.8,1)
       TIME4 = 0.08 + 0.07*BETA(2.28571,5.71429,1)
       IF (ATRIB(7).GT.0.) THEN
          CREW = (TIME1 + ATRIB(7)*TIME2 + ATRIB(8)*TIME3 +
     +          ATRIB(6)*TIME4)/60.
       ELSE IF (ATRIB(7).EQ.0.) THEN
          CREW = (TIME1 + GETARY(PBRN,7)*TIME2 +
     +       ATRIB(8)*TIME3 + 0.6*GETARY(PBRN,6)*TIME4)/60.
       ELSE
          CALL ERROR(1)
       END IF
       BOT = BOT + 4.*CREW/3.
       USERF = CREW
       RETURN
C*****GUARD AND BLAST FOR DRILLS.
 7     USERF = (60. + 72.2*BETA(1.44598,3.77402,1))/60.
       RETURN
C*****WAIT TIME FOR LHDS.
 8     IPT = 0.1
       IF (ATRIB(11).EQ.3.) THEN
          TIME(2,3,NNRUN) = TIME(2,3,NNRUN) + 4.*IPT/3.
       ELSE IF (ATRIB(11).EQ.4.) THEN
          TIME(2,4,NNRUN) = TIME(2,4,NNRUN) + 4.*IPT/3.
       ELSE
          CALL ERROR(1)
       END IF
       USERF = IPT
       RETURN
C*****CHECK FOR ROOM IN ORE DUMP LOCATION.
 9     ID = ATRIB(12)
       IF (XX(ID).LT.CAPACITY(ID)) THEN
          USERF = 1.
       ELSE
          USERF = 0.
       END IF
       RETURN
C*****LHD2 - 9 CUBIC YARD - REMOTE CONTROL MUCKING.
 10    IF (ATRIB(4).EQ.ATRIB(9)) THEN
          IF (ATRIB(2).EQ.1.) THEN
             PSC(ATRIB(1),4,NNRUN) = TNOW/24.
             IF (ATRIB(1).EQ.1.) THEN
                S_MUCK = TNOW
                TIME(1,6,NNRUN) = 0.
                TIME(2,6,NNRUN) = 0.
                TIME(3,6,NNRUN)  = 0.
                UT_TRAM = 0.
                BT_TRAM = 0.
                PD_TRAM = 0.
                PI_TRAM = 0.
             END IF
          END IF
```

```
      END IF
      BKT = TRIAG(12.5,14.,15.5,1)
      TLHD2 = TLHD2 + BKT
      CALL MUCKHAUL(BKT)
      TIME1 = 2. + 3.*BETA(1.15556,3.17778,1)
      SPEED = 345. + 130.*BETA(1.48303,4.1052,1)
      DPT = TIME1/60. + ATRIB(13)/(SPEED*60.)
      TIME(1,4,NNRUN) = TIME(1,4,NNRUN) + DPT
      TIME(2,4,NNRUN) = TIME(2,4,NNRUN) + DPT/3.
      USERF = DPT
      RETURN
C*****LHD2 - 9 CUBIC YARD - MANUAL (NON-REMOTE CONTROL) MUCKING.
 11   IF (ATRIB(4).EQ.ATRIB(9)) THEN
         IF (ATRIB(2).EQ.1.) THEN
            PSC(ATRIB(1),4,NNRUN) = TNOW/24.
            IF (ATRIB(1).EQ.1.) THEN
               S_MUCK = TNOW
               TIME(1,6,NNRUN) = 0.
               TIME(2,6,NNRUN) = 0.
               TIME(3,6,NNRUN)  = 0.
               UT_TRAM = 0.
               BT_TRAM = 0.
               PD_TRAM = 0.
               PI_TRAM = 0.
            END IF
         END IF
      END IF
      BKT = TRIAG(12.5,14.,15.5,1)
      TLHD2 = TLHD2 + BKT
      CALL MUCKHAUL(BKT)
      TIME1 = 1.2 + 1.3*BETA(1.05769,1.69231,1)
      SPEED = 345. + 130.*BETA(1.48303,4.1052,1)
      DPT = TIME1/60. + ATRIB(13)/(SPEED*60.)
      TIME(1,4,NNRUN) = TIME(1,4,NNRUN) + DPT
      TIME(2,4,NNRUN) = TIME(2,4,NNRUN) + DPT/3.
      USERF = DPT
      RETURN
C*****LHD1 - 6 CUBIC YARD - REMOTE CONTROL MUCKING.
 12   IF (ATRIB(4).EQ.ATRIB(9)) THEN
         IF (ATRIB(2).EQ.1.) THEN
            PSC(ATRIB(1),4,NNRUN) = TNOW/24.
            IF (ATRIB(1).EQ.1.) THEN
               S_MUCK = TNOW
               TIME(1,6,NNRUN) = 0.
               TIME(2,6,NNRUN) = 0.
               TIME(3,6,NNRUN)  = 0.
               UT_TRAM = 0.
               BT_TRAM = 0.
               PD_TRAM = 0.
               PI_TRAM = 0.
            END IF
         END IF
      END IF
      BKT = TRIAG(6.,7.,8.,1)
      TLHD1 = TLHD1 + BKT
      CALL MUCKHAUL(BKT)
      TIME1 = 1.5 + 2.5*BETA(1.34341,2.80293,1)
      SPEED = 345. + 95.*BETA(1.28289,3.40461,1)
      DPT = TIME1/60. + ATRIB(13)/(SPEED*60.)
```

```
      TIME(1,3,NNRUN) = TIME(1,3,NNRUN) + DPT
      TIME(2,3,NNRUN) = TIME(2,3,NNRUN) + DPT/3.
      USERF = DPT
      RETURN
C*****LHD1 - 6 CUBIC YARD - MANUAL (NON-REMOTE CONTROL) MUCKING.
 13   IF (ATRIB(4).EQ.ATRIB(9)) THEN
         IF (ATRIB(2).EQ.1.) THEN
            PSC(ATRIB(1),4,NNRUN) = TNOW/24.
            IF (ATRIB(1).EQ.1.) THEN
               S_MUCK = TNOW
               TIME(1,6,NNRUN) = 0.
               TIME(2,6,NNRUN) = 0.
               TIME(3,6,NNRUN)  = 0.
               UT_TRAM = 0.
               BT_TRAM = 0.
               PD_TRAM = 0.
               PI_TRAM = 0.
            END IF
         END IF
      END IF
      BKT = TRIAG(6.,7.,8.,1)
      TLHD1 = TLHD1 + BKT
      CALL MUCKHAUL(BKT)
      TIME1 = 0.75 + 1.25*BETA(1.25217,2.0087,1)
      SPEED = 345. + 95.*BETA(1.28289,3.40461,1)
      DPT = TIME1/60. + ATRIB(13)/(SPEED*60.)
      TIME(1,3,NNRUN) = TIME(1,3,NNRUN) + DPT
      TIME(2,3,NNRUN) = TIME(2,3,NNRUN) + DPT/3.
      USERF = DPT
      RETURN
C*****BACKFILLING.
 14   IF (ATRIB(9).EQ.ATRIB(4)) THEN
         PSC(ATRIB(1),6,NNRUN) = TNOW/24.
         IF (ATRIB(1).EQ.1.) THEN
           S_BACK = TNOW
        END IF
      END IF
      ATRIB(4) = ATRIB(4) - TRIAG(11.5,12.5,13.5,1)
      TIME1 = 9. + 5.8*BETA(3.62069,17.3793,1)
      SPEED = 350. + 112.5*BETA(2.4,6.6,1)
      DPT = TIME1/60. + ATRIB(14)/(SPEED*60.)
      TIME(1,5,NNRUN) = TIME(1,5,NNRUN) + DPT
      TIME(2,5,NNRUN) = TIME(2,5,NNRUN) + DPT/3.
      USERF = DPT
      RETURN
C*****IF 25% OF ORE MUCKED FROM BLAST SECTION THEN FREE BLAST ROOM.
 15   IF (ATRIB(9).LT.(0.75*ATRIB(15)).AND.ATRIB(15).NE.0.) THEN
         ATRIB(15) = 0.
         USERF = 1.
      ELSE
         USERF = 0.
      END IF
      RETURN
C*****REMOVE TRAINLOAD OR ORE, UPDATE AMOUNTS, AND TRANSFER.
 16   CALL LARGEST(ID)
      CARLOAD = TRIAG(13.8,14.8,15.8,1)
      XX(ID) = XX(ID)-NOCPT*CARLOAD
      XX(6) = XX(6)+NOCPT*CARLOAD
      XX(7) = XX(7)+NOCPT*CARLOAD
```

```
      ATRIB(12) = ID
      CALL TRANSFER
C*****DETERMINE ORE TRAINLOAD CYCLE TIME.
      TIME1 = 2.3 + 1.7*BETA(1.27451,3.05882,1)
      IF (ID.EQ.10) THEN
         DPT = (17. + 11.*BETA(1.76653,5.05165,1) +
     +         NOCPT*TIME1)/60.
      ELSE IF (ID.EQ.15) THEN
         DPT = (17.5 + 11.*BETA(1.78662,5.15783,1) +
     +         NOCPT*TIME1)/60.
      ELSE IF (ID.EQ.20) THEN
         DPT = (22. + 11.5*BETA(1.99209,5.07861,1) +
     +         NOCPT*TIME1)/60.
      ELSE IF (ID.EQ.25) THEN
         DPT = (23.5 + 10.3*BETA(1.61783,5.47308,1) +
     +         NOCPT*TIME1)/60.
      ELSE
         CALL ERROR(1)
      END IF
      TIME(1,6,NNRUN) = TIME(1,6,NNRUN) + DPT
      TIME(2,6,NNRUN) = TIME(2,6,NNRUN) + DPT/3.
      USERF = DPT
      RETURN
C*****CHECK IF TRAINLOAD OF ORE TO HAUL.
 17   CALL LARGEST(ID)
      IF (XX(ID).GT.(NOCPT*TRIAG(13.8,14.8,15.8,1))) THEN
         USERF = 1.
      ELSE
         USERF = 0.
      END IF
      RETURN
C*****REMOVE TRAINLOAD OF ROCK, UPDATE AMOUNTS AND TRANSFER.
 18   CARLOAD = TRIAG(4.,4.5,5.,1)
      XX(TRID) = XX(TRID)-NRCPT*CARLOAD
      XX(8) = XX(8)+NRCPT*CARLOAD
      XX(9) = XX(9)+NRCPT*CARLOAD
      ATRIB(12) = DRID
      CALL TRANSFER
C*****DETERMINE ROCK TRAINLOAD CYCLE TIME.
      TIME1 = 1.4 + 2.1*BETA(1.73469,5.40816,1)
      IF (TRID.EQ.10) THEN
         DPT = (17. + 11.*BETA(1.76653,5.05165,1) +
     +         NRCPT*TIME1)/60.
      ELSE IF (TRID.EQ.15) THEN
         DPT = (17.5 + 11.*BETA(1.78662,5.15783,1) +
     +         NRCPT*TIME1)/60.
      ELSE IF (TRID.EQ.20) THEN
         DPT = (22. + 11.5*BETA(1.99209,5.07861,1) +
     +         NRCPT*TIME1)/60.
      ELSE IF (TRID.EQ.25) THEN
         DPT = (23.5 + 10.3*BETA(1.61783,5.47308,1) +
     +         NRCPT*TIME1)/60.
      ELSE
         CALL ERROR(1)
      END IF
      TIME(1,6,NNRUN) = TIME(1,6,NNRUN) + DPT
      TIME(2,6,NNRUN) = TIME(2,6,NNRUN) + DPT/3.
      USERF = DPT
      RETURN
```

```
C*****CHECK IF TRAINLOAD OF ROCK TO HAUL.
 19   IF (XX(TRID).GT.NRCPT*TRIAG(4.,4.5,5.,1)) THEN
         USERF = 1.
      ELSE
         USERF = 0.
      END IF
      RETURN
C*****FINISHED MUCKING & HAULING FOR LAST BLAST OF BLOCK.
 20   IF (ATRIB(2).EQ.ATRIB(3)) THEN
         PSC(ATRIB(1),5,NNRUN) = TNOW/24.
      END IF
      USERF = 0.
      RETURN
C*****DRILLCR1 UPTIME UPDATE/CHECK.
 21   IF (UT_D1.EQ.0.) THEN
         UT_D1 = 2. + GAMA(18.606,0.7,1)
      END IF
      BT_D1 = BT_D1 + (TIME(1,1,NNRUN)+TIME(2,1,NNRUN))-
     +                       (PD_D1+PI_D1)
      PD_D1 = TIME(1,1,NNRUN)
      PI_D1 = TIME(2,1,NNRUN)
      IF (BT_D1.GT.UT_D1) THEN
         USERF = 1.
      ELSE
         USERF = 0.
      END IF
      RETURN
C*****SET UPTIME VALUE FOR DRILLCR1.
 22   BT_D1 = BT_D1-UT_D1
      USERF = UT_D1
      RETURN
C*****DRILLCR1 DOWNTIME.
 23   RT = 1. + GAMA(4.10714,1.4,1)
      TIME(3,1,NNRUN) = TIME(3,1,NNRUN) + RT
      UT_D1 = 0.
      XX(30) = 0.
      USERF = RT
      RETURN
C*****DRILLCR2 UPTIME UPDATE/CHECK.
 24   IF (UT_D2.EQ.0.) THEN
         UT_D2 = 2. + GAMA(18.606,0.7,1)
      END IF
      BT_D2 = BT_D2 + (TIME(1,2,NNRUN)+TIME(2,2,NNRUN))-
     +                       (PD_D2+PI_D2)
      PD_D2 = TIME(1,2,NNRUN)
      PI_D2 = TIME(2,2,NNRUN)
      IF (BT_D2.GT.UT_D2) THEN
         USERF = 1.
      ELSE
         USERF = 0.
      END IF
      RETURN
C*****SET UPTIME VALUE FOR DRILLCR2.
 25   BT_D2 = BT_D2-UT_D2
      USERF = UT_D2
      RETURN
C*****DRILLCR2 DOWNTIME.
 26   RT = 1. + GAMA(4.10714,1.4,1)
      TIME(3,2,NNRUN) = TIME(3,2,NNRUN) + RT
```

```
      UT_D2 = 0.
      XX(31) = 0.
      USERF = RT
      RETURN
C*****LHD1 UPTIME UPDATE/CHECK.
 27   IF (UT_LHD1.EQ.0.) THEN
         UT_LHD1 = 2. + GAMA(37.0703,0.7,1)
      END IF
      BT_LHD1 = BT_LHD1 + (TIME(1,3,NNRUN)+TIME(2,3,NNRUN))-
     +                     (PD_LHD1+PI_LHD1)
      PD_LHD1 = TIME(1,3,NNRUN)
      PI_LHD1 = TIME(2,3,NNRUN)
      IF (BT_LHD1.GT.UT_LHD1) THEN
         USERF = 1.
      ELSE
         USERF = 0.
      END IF
      RETURN
C*****SET UPTIME VALUE FOR LHD1.
 28   BT_LHD1 = BT_LHD1-UT_LHD1
      USERF = UT_LHD1
      RETURN
C*****LHD1 DOWNTIME.
 29   RT = 2. + GAMA(5.58571,1.4,1)
      TIME(3,3,NNRUN) = TIME(3,3,NNRUN) + RT
      UT_LHD1 = 0.
      XX(32) = 0.
      USERF = RT
      RETURN
C*****LHD2 UPTIME UPDATE/CHECK.
 30   IF (UT_LHD2.EQ.0.) THEN
         UT_LHD2 = 1. + GAMA(15.9714,0.7,1)
      END IF
      BT_LHD2 = BT_LHD2 + (TIME(1,4,NNRUN)+TIME(2,4,NNRUN))-
     +                     (PD_LHD2+PI_LHD2)
      PD_LHD2 = TIME(1,4,NNRUN)
      PI_LHD2 = TIME(2,4,NNRUN)
      IF (BT_LHD2.GT.UT_LHD2) THEN
         USERF = 1.
      ELSE
         USERF = 0.
      END IF
      RETURN
C*****SET UPTIME VALUE FOR LHD2.
 31   BT_LHD2 = BT_LHD2-UT_LHD2
      USERF = UT_LHD2
      RETURN
C*****LHD2 DOWNTIME.
 32   RT = 1. + GAMA(5.58571,1.4,1)
      TIME(3,4,NNRUN) = TIME(3,4,NNRUN) + RT
      UT_LHD2 = 0.
      XX(33) = 0.
      USERF = RT
      RETURN
C*****TRUCK UPTIME UPDATE/CHECK.
 33   IF (UT_TRUC.EQ.0.) THEN
         UT_TRUC = 1. + GAMA(47.6964,0.7,1)
      END IF
      BT_TRUC = BT_TRUC + (TIME(1,5,NNRUN)+TIME(2,5,NNRUN))-
```

```
     +                           (PD_TRUC+PI_TRUC)
      PD_TRUC = TIME(1,5,NNRUN)
      PI_TRUC = TIME(2,5,NNRUN)
      IF (BT_TRUC.GT.UT_TRUC) THEN
         USERF = 1.
      ELSE
         USERF = 0.
      END IF
      RETURN
C*****SET UPTIME VALUE FOR TRUCK.
 34   BT_TRUC = BT_TRUC-UT_TRUC
      USERF = UT_TRUC
      RETURN
C*****TRUCK DOWNTIME.
 35   RT = 1. + GAMA(3.96429,1.4,1)
      TIME(3,5,NNRUN) = TIME(3,5,NNRUN) + RT
      UT_TRUC = 0.
      XX(34) = 0.
      USERF = RT
      RETURN
C*****TRAIN UPTIME UPDATE/CHECK.
 36   IF (UT_TRAM.EQ.0.) THEN
         UT_TRAM = 4. + GAMA(170.714,0.7,1)
      END IF
      BT_TRAM = BT_TRAM + (TIME(1,6,NNRUN)+TIME(2,6,NNRUN))-
     +                           (PD_TRAM+PI_TRAM)
      PD_TRAM = TIME(1,6,NNRUN)
      PI_TRAM = TIME(2,6,NNRUN)
      IF (BT_TRAM.GT.UT_TRAM) THEN
         USERF = 1.
      ELSE
         USERF = 0.
      END IF
      RETURN
C*****SET UPTIME VALUE FOR TRAIN.
 37   BT_TRAM = BT_TRAM-UT_TRAM
      USERF = UT_TRAM
      RETURN
C*****TRAIN DOWNTIME.
 38   RT = 2. + GAMA(3.21429,1.4,1)
      TIME(3,6,NNRUN) = TIME(3,6,NNRUN) + RT
      UT_TRAM = 0.
      XX(35) = 0.
      USERF = RT
      RETURN
C*****FINISHED BACKFILLING.
 39   PSC(ATRIB(1),7,NNRUN) = TNOW/24.
      USERF = 0.
      RETURN
C*****MARK TIME AND PRINT FINISHED BLASTING FOR LAST BLAST OF BLOCK.
 40   ATRIB(8) = TNOW
      IF (ATRIB(2).EQ.ATRIB(3)) THEN
         PSC(ATRIB(1),3,NNRUN) = TNOW/24.
      END IF
C*****IF LAST BLAST CALL STATISTICS SUBROUTINE.
      CBLAST = CBLAST + 1
      IF (CBLAST.EQ.NBLASTS) THEN
         CALL STATS
      END IF
```

```
      ATRIB(15) = ATRIB(9)
      ATRIB(4) = ATRIB(9)
      USERF = 0.
      RETURN
C*****CHECK FOR TRUCK BLAST DELAY.
 41   IF (EBD_TRUC.EQ.0.) THEN
         DELAY = (45. + 37.45*BETA(1.46595,4.02405,1))/60.
         EBD_TRUC = TNOW + DELAY
         USERF = 1.
      ELSE IF (TNOW.GT.EBD_TRUC) THEN
         EBD_TRUC = 0.
         USERF = 0.
      ELSE
         USERF = 1.
      END IF
      RETURN
C*****TIME LEFT FOR BLAST DELAY AND CHECK VENTS FOR TRUCK.
 42   IF (EBD_TRUC.EQ.0.) THEN
         IPT = (45. + 37.45*BETA(1.46595,4.02405,1))/60.
         TIME(2,5,NNRUN) = TIME(2,5,NNRUN) + 4.*IPT/3.
      ELSE
         IPT = EBD_TRUC - TNOW
         IF (IPT.LT.0.) THEN
            IPT = 0.
         END IF
         TIME(2,5,NNRUN) = TIME(2,5,NNRUN) + 4.*IPT/3.
      END IF
      USERF = IPT
      RETURN
C*****CHECK FOR LHD1 BLAST DELAY.
 43   IF (EBD_LHD1.EQ.0.) THEN
         DELAY = (45. + 37.45*BETA(1.46595,4.02405,1))/60.
         EBD_LHD1 = TNOW + DELAY
         USERF = 1.
      ELSE IF (TNOW.GT.EBD_LHD1) THEN
         EBD_LHD1 = 0.
         USERF = 0.
      ELSE
         USERF = 1.
      END IF
      RETURN
C*****TIME LEFT FOR BLAST DELAY AND CHECK VENTS FOR LHD1.
 44   IF (EBD_LHD1.EQ.0.) THEN
         IPT = (45. + 37.45*BETA(1.46595,4.02405,1))/60.
         TIME(2,3,NNRUN) = TIME(2,3,NNRUN) + 4.*IPT/3.
      ELSE
         IPT = EBD_LHD1 - TNOW
         IF (IPT.LT.0.) THEN
            IPT = 0.
         END IF
         TIME(2,3,NNRUN) = TIME(2,3,NNRUN) + 4.*IPT/3.
      END IF
      USERF = IPT
      RETURN
C*****LHD2 BLAST DELAY.
 45   IF (EBD_LHD2.EQ.0.) THEN
         DELAY = (45. + 37.45*BETA(1.46595,4.02405,1))/60.
         EBD_LHD2 = TNOW + DELAY
         USERF = 1.
```

```
      ELSE IF (TNOW.GT.EBD_LHD2) THEN
         EBD_LHD2 = 0.
         USERF = 0.
      ELSE
         USERF = 1.
      END IF
      RETURN
C*****TIME LEFT FOR BLAST DELAY AND CHECK VENTS FOR LHD2.
 46   IF (EBD_LHD2.EQ.0.) THEN
         IPT = (45. + 37.45*BETA(1.46595,4.02405,1))/60.
         TIME(2,4,NNRUN) = TIME(2,4,NNRUN) + 4.*IPT/3.
      ELSE
         IPT = EBD_LHD2 - TNOW
         IF (IPT.LT.0.) THEN
            IPT = 0.
         END IF
         TIME(2,4,NNRUN) = TIME(2,4,NNRUN) + 4.*IPT/3.
      END IF
      USERF = IPT
      RETURN
C*****3RD SHIFT.
 47   IF (S3_D1.EQ.8.) THEN
         IF (NNACT(5).EQ.1.) THEN
            XX(30) = 8.
         ELSE
            XX(30) = 0.
         END IF
      END IF
      USERF = S3_D1
      RETURN
C*****3RD SHIFT.
 48   IF (S3_D2.EQ.8.) THEN
         IF (NNACT(6).EQ.1.) THEN
            XX(31) = 8.
         ELSE
            XX(31) = 0.
         END IF
      END IF
      USERF = S3_D2
      RETURN
C*****3RD SHIFT.
 49   IF (S3_LHD1.EQ.8.) THEN
         IF (NNACT(7).EQ.1.) THEN
            XX(32) = 8.
         ELSE
            XX(32) = 0.
         END IF
      END IF
      USERF = S3_LHD1
      RETURN
C*****3RD SHIFT.
 50   IF (S3_LHD2.EQ.8.) THEN
         IF (NNACT(8).EQ.1.) THEN
            XX(33) = 8.
         ELSE
            XX(33) = 0.
         END IF
      END IF
      USERF = S3_LHD2
```

```
      RETURN
C*****3RD SHIFT.
 51   IF (S3_TRUC.EQ.8.) THEN
         IF (NNACT(9).EQ.1.) THEN
            XX(34) = 8.
         ELSE
            XX(34) = 0.
         END IF
      END IF
      USERF = S3_TRUC
      RETURN
C*****3RD SHIFT.
 52   IF (S3_TRAM.EQ.8.) THEN
         IF (NNACT(10).EQ.1.) THEN
            XX(35) = 8.
         ELSE
            XX(35) = 0.
         END IF
      END IF
      USERF = S3_TRAM
      RETURN
C*****UPDATE.
 53   IBLK = ATRIB(1)
      BLKTONS(IBLK) = ATRIB(4)
      FTDRILL(IBLK) = ATRIB(6)
      ATRIB(4) = .6*ATRIB(4)
      ATRIB(9) = ATRIB(4)
      ATRIB(8) = TNOW
      USERF = 0.
      RETURN
      END
```

Appendix A.6.4 File VARIOUS

```
C      ***************************************************************
C      * SUBROUTINE: EVENT                                            *
C      ***************************************************************
C      *                                                              *
C      * SUBROUTINE EVENT IS USED FOR SPECIFIC EVENTS AS OUTLINED     *
C      * BELOW FOR INDIVIDUAL EVENT NUMBERS.                          *
C      *                                                              *
C      * ARGUMENTS: IEN - INTEGER EVENT NUMBER                        *
C      *                                                              *
C      * SEE "COMVAR" FOR VARIABLES NOT DEFINED BELOW.                *
C      *                                                              *
C      ***************************************************************
C
C      ROCKPASS : TONS OF ROCK IN THE ROCKPASS
C
       SUBROUTINE EVENT (IEN)
       INTEGER ROW,COL
$INCLUDE:'PARAM.INC'
$INCLUDE:'SCOM1.COM'
$INCLUDE:'COMVAR.FOR'
       SELECT CASE (IEN)
       CASE(1)
C*****INITIALIZE ATTRIBUTE VALUES.
       ROW = XX(1)
       DO 100 COL = 1,15
          ATRIB(COL) = GETARY(ROW,COL)
100    CONTINUE
       CASE(2)
C*****DEPOSIT DEVELOPMENT ORE & ROCK EVERY 2 HRS FOR 1ST & 2ND SHIFTS.
       XX(DDOID) = XX(DDOID)+ADODD/8.
       ATRIB(12) = DDOID
       CALL TRANSFER
       XX(DRID) = XX(DRID)+ARDD/8.
       ATRIB(12) = DRID
       CALL TRANSFER
       CASE(3)
C*****PRINT AMOUNTS AT END OF DAY AND REINTIALIZE DAILY VALUES.
       ROCKPASS = XX(TRID)+XX(TRID+1)+XX(TRID+2)+XX(TRID+3)+XX(TRID+4)
       WRITE (38,101) TNOW/24.,XX(2),XX(3),XX(4),XX(5),XX(6),XX(7),
      +XX(8),XX(9),ROCKPASS
101    FORMAT (F6.0,F7.0,F8.0,F7.0,F8.0,F7.0,2F8.0,F7.0,F8.0,F9.0)
       XX(2) = 0.
       XX(4) = 0.
       XX(6) = 0.
       XX(8) = 0.
       END SELECT
       RETURN
       END
C      ***************************************************************
C      * SUBROUTINE: LARGEST                                          *
C      ***************************************************************
C      *                                                              *
C      * SUBROUTINE LARGEST IS USED TO IDENTIFY THE OREPASS WITH THE *
C      * GREATEST NUMBER OF TONS OF ORE IN IT.                        *
```

```
C      *                                                              *
C      * RETURNS: ID - I.D. NUMBER OF THE LOWEST SECTION OF THE       *
C      *                OREPASS WITH THE MOST ORE IN IT               *
C      *                                                              *
C      * SEE "COMVAR" FOR VARIABLES NOT DEFINED BELOW.                *
C      *                                                              *
C      ****************************************************************
C
C      ID   : OREPASS SECTION I.D. NUMBER
C      MOST : VARIABLE FOR THE GREATEST AMOUNT OF ORE IN THE OREPASS
C
       SUBROUTINE LARGEST(ID)
$INCLUDE:'PARAM.INC'
$INCLUDE:'SCOM1.COM'
$INCLUDE:'COMVAR.FOR'
       IF (TRID.EQ.10) THEN
           MOST = XX(15)
           ID = 15
           IF (XX(20).GT.MOST) THEN
              MOST = XX(20)
              ID = 20
           END IF
           IF (XX(25).GT.MOST) THEN
              MOST = XX(25)
              ID = 25
           END IF
       ELSE IF (TRID.EQ.15) THEN
           MOST = XX(10)
           ID = 10
           IF (XX(20).GT.MOST) THEN
              MOST = XX(20)
              ID = 20
           END IF
           IF (XX(25).GT.MOST) THEN
              MOST = XX(25)
              ID = 25
           END IF
       ELSE IF (TRID.EQ.20) THEN
           MOST = XX(10)
           ID = 10
           IF (XX(15).GT.MOST) THEN
              MOST = XX(15)
              ID = 15
           END IF
           IF (XX(25).GT.MOST) THEN
              MOST = XX(25)
              ID = 25
           END IF
       ELSE IF (TRID.EQ.25) THEN
           MOST = XX(10)
           ID = 10
           IF (XX(15).GT.MOST) THEN
              MOST = XX(15)
              ID = 15
           END IF
           IF (XX(20).GT.MOST) THEN
              MOST = XX(20)
              ID = 20
           END IF
```

```
      ELSE
          CALL ERROR(1)
      END IF
      RETURN
      END
C     ****************************************************************
C     * SUBROUTINE: MUCKHAUL                                          *
C     ****************************************************************
C     *                                                               *
C     * SUBROUTINE MUCKHAUL IS USED FOR MUCKING AND HAULING RELATED *
C     * ACTIVITIES AS OUTLINED BELOW.                                 *
C     *                                                               *
C     * ARGUMENTS: BKT - TONS OF ORE PER LHD BUCKET                   *
C     *                                                               *
C     * SEE "COMVAR" FOR VARIABLES NOT DEFINED BELOW.                 *
C     *                                                               *
C     ****************************************************************
C
      SUBROUTINE MUCKHAUL(BKT)
$INCLUDE:'PARAM.INC'
$INCLUDE:'SCOM1.COM'
$INCLUDE:'COMVAR.FOR'
C****REMOVE BUCKET THEN CALCULATE ORE & NI HAULAGE.
      ATRIB(9) = ATRIB(9)-BKT
      XX(2) = XX(2)+BKT
      XX(3) = XX(3)+BKT
      XX(4) = XX(4)+(ATRIB(10)/100.)*BKT*2000.
      XX(5) = XX(5)+(ATRIB(10)/100.)*BKT*2000.
C*****DUMP ORE INTO OREPASS SECTION THEN TRANSFER TO LOWEST SECTION.
      ID = ATRIB(12)
      XX(ID) = XX(ID)+BKT
      CALL TRANSFER
      RETURN
      END
C     ****************************************************************
C     * SUBROUTINE: TRANSFER                                          *
C     ****************************************************************
C     *                                                               *
C     * SUBROUTINE TRANSFER IS USED TO TRANSFER ORE/ROCK TO THE      *
C     * LOWEST OREPASS/ROCKPASS SECTION POSSIBLE.  IT IS USED IN      *
C     * MUCKING AND HAULING, AS WELL AS IN TRAIN TRAMMING.            *
C     *                                                               *
C     * SEE "COMVAR" FOR VARIABLES NOT DEFINED BELOW.                 *
C     *                                                               *
C     ****************************************************************
C
C     PTA : POTENTIAL TRANSFER AMOUNT
C     ROOM : ROOM IN LOWEST SECTION OF THE OREPASS/ROCKPASS
C     L1700 : LEVEL 1700 I.D.
C     L1800 : LEVEL 1800 I.D.
C     L1900 : LEVEL 1900 I.D.
C     L2000 : LEVEL 2000 I.D.
C     L2050 : LEVEL 2050 I.D.
C
      SUBROUTINE TRANSFER
      INTEGER L1700,L1800,L1900,L2000,L2050
$INCLUDE:'PARAM.INC'
$INCLUDE:'SCOM1.COM'
$INCLUDE:'COMVAR.FOR'
```

```
C*****DETERMINE OREPASS/ROCKPASS LEVEL I.D.s
      ID = ATRIB(12)
      IF (ID.GE.10.AND.ID.LE.14) THEN
         L1700 = 14
         L1800 = 13
         L1900 = 12
         L2000 = 11
         L2050 = 10
      ELSE IF(ID.GE.15.AND.ID.LE.19) THEN
         L1700 = 19
         L1800 = 18
         L1900 = 17
         L2000 = 16
         L2050 = 15
      ELSE IF(ID.GE.20.AND.ID.LE.24) THEN
         L1700 = 24
         L1800 = 23
         L1900 = 22
         L2000 = 21
         L2050 = 20
      ELSE IF(ID.GE.25.AND.ID.LE.29) THEN
         L1700 = 29
         L1800 = 28
         L1900 = 27
         L2000 = 26
         L2050 = 25
      ELSE
         CALL ERROR(1)
      END IF
C*****TRANSFER ORE/ROCK TO LOWEST SECTION POSSIBLE.
      PTA = XX(L2000)+XX(L1900)+XX(L1800)+XX(L1700)
      ROOM = CAPACITY(L2050)-XX(L2050)
      IF (PTA.LE.ROOM) THEN
         XX(L2050) = XX(L2050)+PTA
         XX(L2000) = 0.
         XX(L1900) = 0.
         XX(L1800) = 0.
         XX(L1700) = 0.
      ELSE
         XX(L2050) = XX(L2050)+ROOM
         IF (XX(L1700).GT.0.) THEN
            XX(L1700) = XX(L1700)-ROOM
            IF (XX(L1700).LT.0.) THEN
               XX(L1800) = XX(L1800)+XX(L1700)
               XX(L1700) = 0.
               IF (XX(L1800).LT.0.) THEN
                  XX(L1900) = XX(L1900)+XX(L1800)
                  XX(L1800) = 0.
                  IF (XX(L1900).LT.0.) THEN
                     XX(L2000) = XX(L2000)+XX(L1900)
                     XX(L1900) = 0.
                  END IF
               END IF
            END IF
         ELSE IF (XX(L1800).GT.0.) THEN
            XX(L1800) = XX(L1800)-ROOM
            IF (XX(L1800).LT.0.) THEN
               XX(L1900) = XX(L1900)+XX(L1800)
               XX(L1800) = 0.
```

```
         IF (XX(L1900).LT.0.) THEN
            XX(L2000) = XX(L2000)+XX(L1900)
            XX(L1900) = 0.
         END IF
      END IF
   ELSE IF (XX(L1900).GT.0.) THEN
      XX(L1900) = XX(L1900)-ROOM
      IF (XX(L1900).LT.0.) THEN
         XX(L2000) = XX(L2000)+XX(L1900)
         XX(L1900) = 0.
      END IF
   ELSE
      XX(L2000) = XX(L2000)-ROOM
   END IF
END IF
RETURN
END
```

Appendix A.6.5 File STATS

```
C     ***********************************************************
C     * SUBROUTINE: STATS                                       *
C     ***********************************************************
C     *                                                         *
C     * SUBROUTINE STATS IS USED TO CALCULATE VARIOUS STATISTICS AS *
C     * OUTLINED BELOW.                                         *
C     *                                                         *
C     * SEE "COMVAR" FOR VARIABLES NOT DEFINED BELOW.           *
C     *                                                         *
C     ***********************************************************
C
C     RATIO()    : AVAILABILITY RATIOS
C     DTOM()     : DAILY TONS OF ORE MUCKED
C     DPNM()     : DAILY POUNDS OF NICKEL MUCKED
C     DTOT()     : DAILY TONS OF ORE TRAMMED
C     DTRT()     : DAILY TONS OF ROCK TRAMMED
C     C_TIME()   : CURRENT VALUE OF TIME() FOR VARIOUS ACTIVITIES
C     S_RATIO()  : SUMS FOR AVAILABILITY RATIOS
C     S_TIME()   : SUMS FOR ACTIVITY TIMES
C     A_RATIO()  : AVERAGES FOR AVAILABILITY RATIOS
C     A_TIME()   : AVERAGES FOR ACTIVITY TIMES
C     Y_RATIO()  : TOP TERM OF VARIANCE EQUATION FOR AVAILABILTIY RATIOS
C     Y_TIME()   : TOP TERM OF VARIANCE EQUATION FOR ACTIVITY TIMES
C     SP_RATIO(): SPREAD FOR AVAILABILITY RATIOS
C     SP_TIME()  : SPREAD FOR ACTIVITY TIMES
C     T()        : T-STATISTIC TABLE VALUES FOR 90, 95 AND 98%
C
C     A_DTOM     : AVERAGE FOR DAILY TONS OF ORE MUCKED
C     A_DPNM     : AVERAGE FOR DAILY POUNDS OF NICKEL MUCKED
C     A_DTOT     : AVERAGE FOR DAILY TONS OF ORE TRAMMED
C     A_DTRT     : AVERAGE FOR DAILY TONS OF ROCK TRAMMED
C     ID1        : NUMBER OF SHIFTS FOR DRILL 1
C     ID2        : NUMBER OF SHIFTS FOR DRILL 2
C     ILHD1      : NUMBER OF SHIFTS FOR LHD 1
C     ILHD2      : NUMBER OF SHIFTS FOR LHD 2
C     ITRUC      : NUMBER OF SHIFTS FOR THE TRUCK
C     ITRAM      : NUMBER OF SHIFTS FOR TRAMMING
C     LVL        : T-STATISTIC ROW NUMBER FOR CONFIDENCE LEVEL
C     S_DTOM     : SUM FOR DAILY TONS OF ORE MUCKED
C     S_DPNM     : SUM FOR DAILY POUNDS OF NICKEL MUCKED
C     S_DTOT     : SUM FOR DAILY TONS OF ORE TRAMMED
C     S_DTRT     : SUM FOR DAILY TONS OF ROCK TRAMMED
C     SP_DTOM    : SPREAD FOR DAILY TONS OF ORE MUCKED
C     SP_DPNM    : SPREAD FOR DAILY POUNDS OF NICKEL MUCKED
C     SP_DTOT    : SPREAD FOR DAILY TONS OF ORE TRAMMED
C     SP_DTRT    : SPREAD FOR DAILY TONS OF ROCK TRAMMED
C     Y_DTOM     : TOP TERM OF VARIANCE EQUATION FOR DTOM
C     Y_DPNM     : TOP TERM OF VARIANCE EQUATION FOR DPNM
C     Y_DTOT     : TOP TERM OF VARIANCE EQUATION FOR DTOT
C     Y_DTRT     : TOP TERM OF VARIANCE EQUATION FOR DTRT
C
      SUBROUTINE STATS
      REAL RATIO(4,6,10),DTOM(10),DPNM(10),DTOT(10),DTRT(10),
     + S_RATIO(4,6),A_RATIO(4,6),Y_RATIO(4,6),SP_RATIO(4,6),
```

```
     + S_TIME(5,6),A_TIME(5,6),Y_TIME(5,6),SP_TIME(5,6),
     + C_TIME(5,6,10),T(10,3)
$INCLUDE:'PARAM.INC'
$INCLUDE:'SCOM1.COM'
$INCLUDE:'COMVAR.FOR'
      DATA T/6.31,2.92,2.35,2.13,2.02,1.94,1.9,1.86,1.83,1.81,
     +       12.71,4.3,3.18,2.78,2.57,2.45,2.36,2.31,2.26,2.23,
     +       31.82,6.97,4.54,3.75,3.37,3.14,3.0,2.9,2.82,2.76/
C*****DETERMINE PST FOR RUN AND NUMBER OF SHIFTS FOR ALL RUNS.
      IF (S3_D1.EQ.2.) THEN
         TIME(5,1,NNRUN) = TNOW - S_DRIL
         ID1 = 3
      ELSE
         TIME(5,1,NNRUN) = (TNOW-S_DRIL) - (TNOW-S_DRIL)/3.
         ID1 = 2
      END IF
C
      IF (S3_D2.EQ.2.) THEN
         TIME(5,2,NNRUN) = TNOW - S_DRIL
         ID2 = 3
      ELSE
         TIME(5,2,NNRUN) = (TNOW-S_DRIL) - (TNOW-S_DRIL)/3.
         ID2 = 2
      END IF
C
      IF (S3_LHD1.EQ.2.) THEN
         TIME(5,3,NNRUN) = TNOW - S_MUCK
         ILHD1 = 3
      ELSE
         TIME(5,3,NNRUN) = (TNOW-S_MUCK) - (TNOW-S_MUCK)/3.
         ILHD1 = 2
      END IF
C
      IF (S3_LHD2.EQ.2.) THEN
         TIME(5,4,NNRUN) = TNOW - S_MUCK
         ILHD2 = 3
      ELSE
         TIME(5,4,NNRUN) = (TNOW-S_MUCK) - (TNOW-S_MUCK)/3.
         ILHD2 = 2
      END IF
C
      IF (S3_TRUC.EQ.2.) THEN
         TIME(5,5,NNRUN) = TNOW - S_BACK
         ITRUC = 3
      ELSE
         TIME(5,5,NNRUN) = (TNOW-S_BACK) - (TNOW-S_BACK)/3.
         ITRUC = 2
      END IF
C
      IF (S3_TRAM.EQ.2.) THEN
         TIME(5,6,NNRUN) = TNOW - S_MUCK
         ITRAM = 3
      ELSE
         TIME(5,6,NNRUN) = (TNOW-S_MUCK) - (TNOW-S_MUCK)/3.
         ITRAM = 2
      END IF
C*****CALCULATE STANDBY TIME FOR RUN.
      DO 700 I=1,6
         TIME(4,I,NNRUN) = TIME(5,I,NNRUN)-TIME(1,I,NNRUN)
```

```
     +                          -TIME(2,I,NNRUN)-TIME(3,I,NNRUN)
700   CONTINUE
C*****CALCULATE MECHANICAL AVAILABILITY FOR RUN.
      DO 702 I=1,6
         RATIO(1,I,NNRUN) =(TIME(1,I,NNRUN)+TIME(2,I,NNRUN))/
     +        (TIME(1,I,NNRUN)+TIME(2,I,NNRUN)+TIME(3,I,NNRUN)+.1)
702   CONTINUE
C*****CALCULATE PHYSICAL AVAILABILITY FOR RUN.
      DO 704 I=1,6
         RATIO(2,I,NNRUN) =(TIME(1,I,NNRUN)+TIME(2,I,NNRUN)
     +                 +TIME(4,I,NNRUN))/(TIME(5,I,NNRUN)+.1)
704   CONTINUE
C*****CALCULATE USE OF AVAILABILITY FOR RUN.
      DO 706 I=1,6
         RATIO(3,I,NNRUN) =(TIME(1,I,NNRUN)+TIME(2,I,NNRUN))/
     +        (TIME(1,I,NNRUN)+TIME(2,I,NNRUN)+TIME(4,I,NNRUN)+.1)
706   CONTINUE
C*****CALCULATE EFFECTIVE UTILIZATION FOR RUN.
      DO 708 I=1,6
         RATIO(4,I,NNRUN) =(TIME(1,I,NNRUN)+TIME(2,I,NNRUN))/
     +                     (TIME(5,I,NNRUN)+.1)
708   CONTINUE
C
      DTOM(NNRUN) = XX(3)/((TNOW-S_MUCK)/24.)
      DPNM(NNRUN) = XX(5)/((TNOW-S_MUCK)/24.)
      DTOT(NNRUN) = XX(7)/((TNOW-S_MUCK)/24.)
      DTRT(NNRUN) = XX(9)/(TNOW/24.)
C*****SUM VARIOUS TERMS OVER MULTIPLE RUNS.
      S_DTOM = S_DTOM + DTOM(NNRUN)
      S_DPNM = S_DPNM + DPNM(NNRUN)
      S_DTOT = S_DTOT + DTOT(NNRUN)
      S_DTRT = S_DTRT + DTRT(NNRUN)
C*****PRINT VALUES FOR RUN.
      WRITE(40,500)
      WRITE(40,501)NNRUN
      WRITE(40,504)
      WRITE(40,506)
      WRITE(40,508)
      WRITE(40,510)
      WRITE(40,512)(RATIO(I,1,NNRUN),I=1,4),
     +             (TIME(I,1,NNRUN),I=1,5),ID1
      WRITE(40,514)(RATIO(I,2,NNRUN),I=1,4),
     +             (TIME(I,2,NNRUN),I=1,5),ID2
      WRITE(40,516)(RATIO(I,3,NNRUN),I=1,4),
     +             (TIME(I,3,NNRUN),I=1,5),ILHD1
      WRITE(40,518)(RATIO(I,4,NNRUN),I=1,4),
     +             (TIME(I,4,NNRUN),I=1,5),ILHD2
      WRITE(40,520)(RATIO(I,5,NNRUN),I=1,4),
     +             (TIME(I,5,NNRUN),I=1,5),ITRUC
      WRITE(40,522)(RATIO(I,6,NNRUN),I=1,4),
     +             (TIME(I,6,NNRUN),I=1,5),ITRAM
C
      WRITE(40,524)DTOM(NNRUN)
      WRITE(40,526)DPNM(NNRUN)
      WRITE(40,528)DTOT(NNRUN)
      WRITE(40,530)DTRT(NNRUN)
C*****SET C_TIME(I,J,K) EQUAL TO CURRENT TIME(I,J,K) FOR RUN.
          DO 710 I=1,5
            DO 711 J=1,6
```

```
               C_TIME(I,J,NNRUN) = TIME(I,J,NNRUN)
 711         CONTINUE
 710       CONTINUE
C*****IF LAST RUN OF SIMULATION.
      IF (NNRUN.EQ.NRUNS) THEN
C********IF ONLY ONE RUN SELECTED PRINT NO CONFIDENCE LEVEL.
         IF (NRUNS.EQ.1.) THEN
          WRITE(41,502)
          WRITE(41,503)NNRUN
          WRITE(41,502)
          WRITE(41,505)
          WRITE(41,506)
          WRITE(41,508)
          WRITE(41,510)
          WRITE(41,512)(RATIO(I,1,NNRUN),I=1,4),
     +                 (TIME(I,1,NNRUN),I=1,5),ID1
          WRITE(41,514)(RATIO(I,2,NNRUN),I=1,4),
     +                 (TIME(I,2,NNRUN),I=1,5),ID2
          WRITE(41,516)(RATIO(I,3,NNRUN),I=1,4),
     +                 (TIME(I,3,NNRUN),I=1,5),ILHD1
          WRITE(41,518)(RATIO(I,4,NNRUN),I=1,4),
     +                 (TIME(I,4,NNRUN),I=1,5),ILHD2
          WRITE(41,520)(RATIO(I,5,NNRUN),I=1,4),
     +                 (TIME(I,5,NNRUN),I=1,5),ITRUC
          WRITE(41,522)(RATIO(I,6,NNRUN),I=1,4),
     +                 (TIME(I,6,NNRUN),I=1,5),ITRAM
          WRITE(41,*) '**NO CONFIDENCE LEVEL FOR 1 RUN**'
          WRITE(41,524)DTOM(NNRUN)
          WRITE(41,526)DPNM(NNRUN)
          WRITE(41,528)DTOT(NNRUN)
          WRITE(41,530)DTRT(NNRUN)
          WRITE(41,*) '**NO CONFIDENCE LEVEL FOR 1 RUN**'
         ELSE
C*********CALCULATE CONFIDENCE INTERVAL.
          IF (LEVEL.EQ.90) THEN
            LVL = 1
          ELSE IF (LEVEL.EQ.95) THEN
            LVL = 2
          ELSE IF (LEVEL.EQ.98) THEN
            LVL = 3
          ELSE
            CALL ERROR(1)
          END IF
C*********CALCULATE SUMS FOR RATIOS.
          DO 712 I=1,4
            DO 713 J=1,6
              DO 714 K=1,NNRUN
                S_RATIO(I,J)=S_RATIO(I,J)+RATIO(I,J,K)
714           CONTINUE
713         CONTINUE
712       CONTINUE
C*********CALCULATE AVERAGES FOR RATIOS.
          DO 716 I=1,4
            DO 717 J=1,6
              A_RATIO(I,J) = S_RATIO(I,J)/NNRUN
717         CONTINUE
716       CONTINUE
C*********CALCULATE TOP TERMS FOR VARIANCE EQUATIONS FOR RATIOS.
          DO 718 I=1,4
```

```
            DO 719 J=1,6
              DO 720 K=1,NNRUN
                Y_RATIO(I,J) = Y_RATIO(I,J)+(RATIO(I,J,K)-
     +            A_RATIO(I,J))*(RATIO(I,J,K)-A_RATIO(I,J))
720           CONTINUE
719         CONTINUE
718       CONTINUE
C*********CALCULATE SPREAD FOR RATIOS.
          DO 722 I=1,4
            DO 723 J=1,6
              SP_RATIO(I,J) = T(NNRUN-1,LVL)*
     +              SQRT((Y_RATIO(I,J)/(NNRUN-1))/NNRUN)
723         CONTINUE
722       CONTINUE
C*********CALCULATE SUMS FOR TIMES.
          DO 724 I=1,5
            DO 725 J=1,6
              DO 726 K=1,NNRUN
                S_TIME(I,J) = S_TIME(I,J)+C_TIME(I,J,K)
726           CONTINUE
725         CONTINUE
724       CONTINUE
C*********CALCULATE AVERAGES FOR TIMES.
          DO 728 I=1,5
            DO 729 J=1,6
              A_TIME(I,J) = S_TIME(I,J)/NNRUN
729         CONTINUE
728       CONTINUE
C*********CALCULATE TOP TERMS FOR VARIANCE EQUATIONS FOR TIMES.
          DO 730 I=1,5
            DO 731 J=1,6
              DO 732 K=1,NNRUN
                Y_TIME(I,J) = Y_TIME(I,J)+(C_TIME(I,J,K)-
     +           A_TIME(I,J))*(C_TIME(I,J,K)-A_TIME(I,J))
732           CONTINUE
731         CONTINUE
730       CONTINUE
C*********CALCULATE SPREAD FOR TIMES.
          DO 734 I=1,5
            DO 735 J=1,6
              SP_TIME(I,J) = T(NNRUN-1,LVL)*
     +              SQRT((Y_TIME(I,J)/(NNRUN-1))/NNRUN)
735         CONTINUE
734       CONTINUE
C*********CALCULATE AVERAGES FOR DAILY VALUES.
          A_DTOM = S_DTOM/NNRUN
          A_DPNM = S_DPNM/NNRUN
          A_DTOT = S_DTOT/NNRUN
          A_DTRT = S_DTRT/NNRUN
C*********CALCULATE TOP TERMS FOR VARIANCE EQUATIONS FOR DAILY VALUES.
          DO 736 J=1,NNRUN
            Y_DTOM=Y_DTOM+(DTOM(J)-A_DTOM)*(DTOM(J)-A_DTOM)
            Y_DPNM=Y_DPNM+(DPNM(J)-A_DPNM)*(DPNM(J)-A_DPNM)
            Y_DTOT=Y_DTOT+(DTOT(J)-A_DTOT)*(DTOT(J)-A_DTOT)
            Y_DTRT=Y_DTRT+(DTRT(J)-A_DTRT)*(DTRT(J)-A_DTRT)
736       CONTINUE
C*********CALCULATE SPREAD FOR DAILY VALUES.
          SP_DTOM=T(NNRUN-1,LVL)*SQRT((Y_DTOM/(NNRUN-1))/NNRUN)
          SP_DPNM=T(NNRUN-1,LVL)*SQRT((Y_DPNM/(NNRUN-1))/NNRUN)
```

```
          SP_DTOT=T(NNRUN-1,LVL)*SQRT((Y_DTOT/(NNRUN-1))/NNRUN)
          SP_DTRT=T(NNRUN-1,LVL)*SQRT((Y_DTRT/(NNRUN-1))/NNRUN)
C*********PRINT AVERAGES FOR RATIOS AND TIMES.
          WRITE(41,502)
          WRITE(41,503)NNRUN
          WRITE(41,502)
          WRITE(41,505)
          WRITE(41,506)
          WRITE(41,508)
          WRITE(41,510)
          WRITE(41,512)(A_RATIO(I,1),I=1,4),
     +                 (A_TIME(I,1),I=1,5),ID1
          WRITE(41,514)(A_RATIO(I,2),I=1,4),
     +                 (A_TIME(I,2),I=1,5),ID2
          WRITE(41,516)(A_RATIO(I,3),I=1,4),
     +                 (A_TIME(I,3),I=1,5),ILHD1
          WRITE(41,518)(A_RATIO(I,4),I=1,4),
     +                 (A_TIME(I,4),I=1,5),ILHD2
          WRITE(41,520)(A_RATIO(I,5),I=1,4),
     +                 (A_TIME(I,5),I=1,5),ITRUC
          WRITE(41,522)(A_RATIO(I,6),I=1,4),
     +                 (A_TIME(I,6),I=1,5),ITRAM
C*********PRINT SPREAD FOR RATIOS AND TIMES.
          WRITE(41,532)LEVEL
          WRITE(41,512)(SP_RATIO(I,1),I=1,4),
     +                 (SP_TIME(I,1),I=1,5),0
          WRITE(41,514)(SP_RATIO(I,2),I=1,4),
     +                 (SP_TIME(I,2),I=1,5),0
          WRITE(41,516)(SP_RATIO(I,3),I=1,4),
     +                 (SP_TIME(I,3),I=1,5),0
          WRITE(41,518)(SP_RATIO(I,4),I=1,4),
     +                 (SP_TIME(I,4),I=1,5),0
          WRITE(41,520)(SP_RATIO(I,5),I=1,4),
     +                 (SP_TIME(I,5),I=1,5),0
          WRITE(41,522)(SP_RATIO(I,6),I=1,4),
     +                 (SP_TIME(I,6),I=1,5),0
C*********PRINT AVERAGES AND SPREAD FOR DAILY VALUES.
          WRITE(41,550)LEVEL
          WRITE(41,552)A_DTOM,SP_DTOM
          WRITE(41,554)A_DPNM,SP_DPNM
          WRITE(41,556)A_DTOT,SP_DTOT
          WRITE(41,558)A_DTRT,SP_DTRT
        END IF
      END IF
C
500   FORMAT('*******************************************',
     +       '**************************************')
501   FORMAT('RESULTS FOR RUN:',I3)
502   FORMAT(19X,'******************************************')
503   FORMAT(19X,'**',2X,'SUMMARY REPORT BASED ON',I3,
     +       ' RUN(S)',2X,'**')
504   FORMAT(/13X,'AVAILABILITY RATIOS',15X,'HOURS')
505   FORMAT(/13X,'AVAILABILITY RATIOS',12X,'AVERAGE HOURS')
506   FORMAT(12X,'---------------------',2X,
     +       '----------------------------------')
508   FORMAT('RESOURCE',4X,'MA',4X,'PA',4X,'UA',4X,'EU',3X,'DPT',
     +       4X,'IPT',5X,'RT',4X,'SBT',4X,'PST',4X,'SHIFTS')
510   FORMAT('--------',3X,'---',3X,'---',3X,'---',3X,'---',2X,
     + '-----',2X,'-----',2X,'-----',2X,'-----',2X,'-----',3X,
```

```
     + '------')
512   FORMAT('  DRILL1',4F6.2,5F7.0,4X,I3)
514   FORMAT('  DRILL2',4F6.2,5F7.0,4X,I3)
516   FORMAT('    LHD1',4F6.2,5F7.0,4X,I3)
518   FORMAT('    LHD2',4F6.2,5F7.0,4X,I3)
520   FORMAT('   TRUCK',4F6.2,5F7.0,4X,I3)
522   FORMAT('   TRAIN',4F6.2,5F7.0,4X,I3)
C
524   FORMAT(/'AVERAGE DAILY TONS OF ORE MUCKED:',8X,F7.0)
526   FORMAT( 'AVERAGE DAILY POUNDS OF NICKEL MUCKED:',3X,F7.0)
528   FORMAT( 'AVERAGE DAILY TONS OF ORE TRAMMED:',7X,F7.0)
530   FORMAT( 'AVERAGE DAILY TONS OF ROCK TRAMMED:',6X,F7.0)
C
532   FORMAT(/I2,'% CONFIDENCE LEVEL FOR ABOVE (+/-):')
C
550   FORMAT(////I2,'% CONFIDENCE LEVEL FOR:')
552   FORMAT('AVERAGE DAILY TONS OF ORE MUCKED:',8X,F7.0,2X,
     +           '+/-',F7.0)
554   FORMAT('AVERAGE DAILY POUNDS OF NICKEL MUCKED:',3X,F7.0,2X,
     +           '+/-',F7.0)
556   FORMAT('AVERAGE DAILY TONS OF ORE TRAMMED:',7X,F7.0,2X,
     +           '+/-',F7.0)
558   FORMAT('AVERAGE DAILY TONS OF ROCK TRAMMED:',6X,F7.0,2X,
     +           '+/-',F7.0/)
      RETURN
      END
```

Appendix A.6.6 File OTPUT

```
C      **************************************************************
C      * SUBROUTINE: OTPUT                                          *
C      **************************************************************
C      *                                                            *
C      * SUBROUTINE OTPUT IS CALLED BY THE SLAM II MAIN PROGRAM AT  *
C      * THE END OF EACH SIMULATION RUN. OTPUT IS USED TO PERFORM   *
C      * VARIOUS FUNCTIONS AS OUTLINED BELOW.                       *
C      *                                                            *
C      * SEE "COMVAR" FOR VARIABLES NOT DEFINED BELOW.              *
C      *                                                            *
C      **************************************************************
C
C      A_ACT()   : AVERAGES FOR ACTIVITIES
C      A_PSC()   : AVERAGES FOR PRODUCTION SCHEDULE CHART
C      A_TPD()   : AVERAGES FOR MUCKING TONS PER DAY
C      S_ACT()   : SUMS FOR ACTIVITIES
C      S_PSC()   : SUMS FOR PRODUCTION SCHEDULE CHART
C      S_TPD()   : SUMS FOR MUCKING TONS PER DAY
C      SP_ACT()  : SPREAD FOR ACTIVITIES
C      SP_PSC()  : SPREAD FOR PRODUCTION SCHEDULE CHART
C      SP_TPD()  : SPREAD FOR MUCKING TONS PER DAY
C      T()       : T-STATISTIC TABLE VALUES FOR 90, 95 AND 98%
C      TPD()     : MUCKING TONS PER DAY
C      Y_ACT()   : TOP TERM IN VARIANCE EQUATION FOR ACTIVITIES
C      Y_PSC()   : TOP TERM IN VARIANCE EQUATION FOR PROD. SCHEDULE CHART
C      Y_TPD()   : TOP TERM IN VARIANCE EQUATION FOR MUCKING TONS PER DAY
C
C      LVL       : T-STATISTIC ROW NUMBER FOR CONFIDENCE LEVEL
C      ROCKPASS  : TONS OF ROCK IN THE ROCKPASS
C
       SUBROUTINE OTPUT
       REAL S_PSC(10,7),A_PSC(10,7),Y_PSC(10,7),SP_PSC(10,7),
     +      S_ACT(10,3),A_ACT(10,3),Y_ACT(10,3),SP_ACT(10,3),
     +      TPD(10,10),S_TPD(10),A_TPD(10),Y_TPD(10),SP_TPD(10),
     +      T(10,3)
$INCLUDE:'PARAM.INC'
$INCLUDE:'SCOM1.COM'
$INCLUDE:'COMVAR.FOR'
       DATA T/6.31,2.92,2.35,2.13,2.02,1.94,1.9,1.86,1.83,1.81,
     +        12.71,4.3,3.18,2.78,2.57,2.45,2.36,2.31,2.26,2.23,
     +        31.82,6.97,4.54,3.75,3.37,3.14,3.0,2.9,2.82,2.76/
C*****PRINT LAST LINE OF AMOUNTS.
       ROCKPASS = XX(TRID)+XX(TRID+1)+XX(TRID+2)+XX(TRID+3)+XX(TRID+4)
       WRITE (38,400) TNOW/24.,XX(2),XX(3),XX(4),XX(5),XX(6),XX(7),
     +XX(8),XX(9),ROCKPASS
C*****FOR EACH RUN CALCULATE MUCKING TONS PER DAY FOR EACH BLOCK.
       DO 600 I=1,10
         TPD(I,NNRUN)=BLKTONS(I)/(PSC(I,5,NNRUN)-PSC(I,4,NNRUN)+.001)
600    CONTINUE
C*****PRINT MUCKING TONS PER DAY FOR RUN.
       WRITE(40,440)
       WRITE(40,442)
       WRITE(40,444)
       WRITE(40,446)
```

```
      WRITE(40,448) (TPD(I,NNRUN),I=1,10)
C*****PRINT PSCHART FOR RUN.
      WRITE(40,402)
      WRITE(40,404)
      WRITE(40,417)
      WRITE(40,418)
      WRITE(40,420) (PSC(I,1,NNRUN),I=1,10)
      WRITE(40,422) (PSC(I,2,NNRUN),I=1,10)
      WRITE(40,424) (PSC(I,3,NNRUN),I=1,10)
      WRITE(40,426) (PSC(I,4,NNRUN),I=1,10)
      WRITE(40,428) (PSC(I,5,NNRUN),I=1,10)
      WRITE(40,430) (PSC(I,6,NNRUN),I=1,10)
      WRITE(40,432) (PSC(I,7,NNRUN),I=1,10)
C*****PRINT TIME FOR DRILLING & BLASTING, MUCKING & BACKFILL FOR RUN.
      WRITE(40,401)
      WRITE(40,404)
      WRITE(40,406)
      WRITE(40,408)
      WRITE(40,410) (PSC(I,3,NNRUN)-PSC(I,2,NNRUN),I=1,10)
      WRITE(40,412) (PSC(I,5,NNRUN)-PSC(I,4,NNRUN),I=1,10)
      WRITE(40,413) (PSC(I,7,NNRUN)-PSC(I,6,NNRUN),I=1,10)
C*****IF LAST RUN OF SIMULATION.
      IF (NNRUN.EQ.NRUNS) THEN
        IF (NRUNS.EQ.1.) THEN
C*********PRINT MUCKING TONS PER DAY FOR RUN.
          WRITE(41,440)
          WRITE(41,442)
          WRITE(41,444)
          WRITE(41,446)
          WRITE(41,448) (TPD(I,NNRUN),I=1,10)
          WRITE(41,*) '**NO CONFIDENCE LEVEL FOR 1 RUN**'
C*********PRINT PSCHART FOR RUN.
          WRITE(41,402)
          WRITE(41,404)
          WRITE(41,417)
          WRITE(41,418)
          WRITE(41,420) (PSC(I,1,NNRUN),I=1,10)
          WRITE(41,422) (PSC(I,2,NNRUN),I=1,10)
          WRITE(41,424) (PSC(I,3,NNRUN),I=1,10)
          WRITE(41,426) (PSC(I,4,NNRUN),I=1,10)
          WRITE(41,428) (PSC(I,5,NNRUN),I=1,10)
          WRITE(41,430) (PSC(I,6,NNRUN),I=1,10)
          WRITE(41,432) (PSC(I,7,NNRUN),I=1,10)
          WRITE(41,*) '**NO CONFIDENCE LEVEL FOR 1 RUN**'
C*********PRINT TIME FOR DRILLING & BLASTING, MUCKING & BACKFILL FOR
RUN.
          WRITE(41,401)
          WRITE(41,404)
          WRITE(41,406)
          WRITE(41,408)
          WRITE(41,410) (PSC(I,3,NNRUN)-PSC(I,2,NNRUN),I=1,10)
          WRITE(41,412) (PSC(I,5,NNRUN)-PSC(I,4,NNRUN),I=1,10)
          WRITE(41,414) (PSC(I,7,NNRUN)-PSC(I,6,NNRUN),I=1,10)
          WRITE(41,*) '**NO CONFIDENCE LEVEL FOR 1 RUN**'
        ELSE
C*********CALCULATE CONFIDENCE INTERVAL.
          IF (LEVEL.EQ.90) THEN
            LVL = 1
          ELSE IF (LEVEL.EQ.95) THEN
```

```
              LVL = 2
            ELSE IF (LEVEL.EQ.98) THEN
              LVL = 3
            ELSE
              CALL ERROR(1)
            END IF
C*********CALCULATE SUMS FOR MUCKING TONS PER DAY.
            DO 602 I=1,10
              DO 603 J=1,NNRUN
                S_TPD(I) = S_TPD(I) + TPD(I,J)
603           CONTINUE
602         CONTINUE
C*********CALCULATE AVERAGES FOR MUCKING TONS PER DAY.
            DO 604 I=1,10
              A_TPD(I) = S_TPD(I)/NNRUN
604         CONTINUE
C*********CALCULATE TOP TERMS FOR VARIANCE EQUATIONS FOR TONS PER DAY.
            DO 606 I=1,10
              DO 607 J=1,NNRUN
                Y_TPD(I) = Y_TPD(I)+(TPD(I,J)-A_TPD(I))*
     +                              (TPD(I,J)-A_TPD(I))
607           CONTINUE
606         CONTINUE
C*********CALCULATE SPREAD FOR MUCKING TONS PER DAY.
            DO 608 I=1,10
              SP_TPD(I) = T(NNRUN-1,LVL)*
     +                    SQRT((Y_TPD(I)/(NNRUN-1))/NNRUN)
608         CONTINUE
C*********PRINT AVERAGE MUCKING TONS PER DAY.
            WRITE(41,441)
            WRITE(41,442)
            WRITE(41,444)
            WRITE(41,446)
            WRITE(41,448) (A_TPD(I),I=1,10)
C*********PRINT AVERAGE MUCKING TONS PER DAY SPREAD.
            WRITE(41,419) LEVEL
            WRITE(41,448) (SP_TPD(I),I=1,10)
C*********CALCULATE SUMS FOR PSCHART.
            DO 610 I=1,10
              DO 611 J=1,7
                DO 612 K=1,NNRUN
                  S_PSC(I,J)=S_PSC(I,J)+PSC(I,J,K)
612             CONTINUE
611           CONTINUE
610         CONTINUE
C*********CALCULATE AVERAGES FOR PSCHART.
            DO 614 I=1,10
              DO 615 J=1,7
                A_PSC(I,J) = S_PSC(I,J)/NNRUN
615           CONTINUE
614         CONTINUE
C*********CALCULATE TOP TERMS FOR VARIANCE EQUATIONS FOR PSCHART.
            DO 616 I=1,10
              DO 617 J=1,7
                DO 618 K=1,NNRUN
                  Y_PSC(I,J) = Y_PSC(I,J)+(PSC(I,J,K)-A_PSC(I,J))*
     +                                    (PSC(I,J,K)-A_PSC(I,J))
618             CONTINUE
617           CONTINUE
```

```
616       CONTINUE
C*********CALCULATE SPREAD FOR PSCHART.
          DO 620 I=1,10
            DO 621 J=1,7
              SP_PSC(I,J) = T(NNRUN-1,LVL)*
     +                     SQRT((Y_PSC(I,J)/(NNRUN-1))/NNRUN)
621         CONTINUE
620       CONTINUE
C*********PSCHART AVERAGES.
          WRITE(41,403)
          WRITE(41,404)
          WRITE(41,417)
          WRITE(41,418)
          WRITE(41,420) (A_PSC(I,1),I=1,10)
          WRITE(41,422) (A_PSC(I,2),I=1,10)
          WRITE(41,424) (A_PSC(I,3),I=1,10)
          WRITE(41,426) (A_PSC(I,4),I=1,10)
          WRITE(41,428) (A_PSC(I,5),I=1,10)
          WRITE(41,430) (A_PSC(I,6),I=1,10)
          WRITE(41,432) (A_PSC(I,7),I=1,10)
C*********PRINT PSCHART SPREAD.
          WRITE(41,419) LEVEL
          WRITE(41,420) (SP_PSC(I,1),I=1,10)
          WRITE(41,422) (SP_PSC(I,2),I=1,10)
          WRITE(41,424) (SP_PSC(I,3),I=1,10)
          WRITE(41,426) (SP_PSC(I,4),I=1,10)
          WRITE(41,428) (SP_PSC(I,5),I=1,10)
          WRITE(41,430) (SP_PSC(I,6),I=1,10)
          WRITE(41,432) (SP_PSC(I,7),I=1,10)
C*********CALCULATE SUMS FOR ACTIVITIES.
          DO 622 I=1,10
            DO 623 J=1,3
              DO 624 K=1,NNRUN
              S_ACT(I,J)=S_ACT(I,J)+(PSC(I,1+2*J,K)-PSC(I,2*J,K))
624           CONTINUE
623         CONTINUE
622       CONTINUE
C*********CALCULATE AVERAGES FOR ACTIVITIES.
          DO 626 I=1,10
            DO 627 J=1,3
              A_ACT(I,J)=S_ACT(I,J)/NNRUN
627         CONTINUE
626       CONTINUE
C*********CALCULATE TOP TERMS FOR VARIANCE EQUATIONS FOR ACTIVITES.
          DO 628 I=1,10
            DO 629 J=1,3
              DO 630 K=1,NNRUN
                Y_ACT(I,J) = Y_ACT(I,J)+((PSC(I,1+2*J,K)-
     +             PSC(I,2*J,K))-A_ACT(I,J))*((PSC(I,1+2*J,K)-
     +             PSC(I,2*J,K))-A_ACT(I,J))
630           CONTINUE
629         CONTINUE
628       CONTINUE
C*********CALCULATE SPREAD FOR ACTIVITIES.
          DO 632 I=1,10
            DO 633 J=1,3
              SP_ACT(I,J) = T(NNRUN-1,LVL)*
     +                 SQRT((Y_ACT(I,J)/(NNRUN-1))/NNRUN)
633         CONTINUE
```

```
632       CONTINUE
C*********PRINT ACTIVITY AVERAGES.
          WRITE(41,434)
          WRITE(41,404)
          WRITE(41,406)
          WRITE(41,408)
          WRITE(41,410) (A_ACT(I,1),I=1,10)
          WRITE(41,412) (A_ACT(I,2),I=1,10)
          WRITE(41,414) (A_ACT(I,3),I=1,10)
C*********PRINT ACTIVITY SPREADS.
          WRITE(41,419) LEVEL
          WRITE(41,410) (SP_ACT(I,1),I=1,10)
          WRITE(41,412) (SP_ACT(I,2),I=1,10)
          WRITE(41,414) (SP_ACT(I,3),I=1,10)
        END IF
C*******CALCULATE COSTS.
        WRITE(42,460)
        WRITE(42,462)
        WRITE(42,464)
        WRITE(42,466)
C
        WRITE(42,468)
        WRITE(42,469) (FTDRILL(I)*3.65,I=1,10)
        WRITE(42,470) (FTDRILL(I)*.094,I=1,10)
        WRITE(42,471) (FTDRILL(I)*.573,I=1,10)
        WRITE(42,472) (FTDRILL(I)*.882,I=1,10)
        WRITE(42,473) (TIME(1,1,NNRUN)+TIME(2,1,NNRUN)+
     +                 TIME(1,2,NNRUN)+TIME(2,2,NNRUN))*30.
        WRITE(42,*)  ' '
C
        WRITE(42,480)
        WRITE(42,481) (BLKTONS(I)*.46,I=1,10)
        WRITE(42,482) (BLKTONS(I)*.25,I=1,10)
        WRITE(42,483) BOT*30.
        WRITE(42,*)  ' '
C
        WRITE(42,485)
        WRITE(42,486) TLHD1*.88
        WRITE(42,487) (TIME(1,3,NNRUN)+TIME(2,3,NNRUN))*30.
        WRITE(42,*)  ' '
C
        WRITE(42,490)
        WRITE(42,491) TLHD2*.43
        WRITE(42,492) (TIME(1,4,NNRUN)+TIME(2,4,NNRUN))*30.
        WRITE(42,*)  ' '
        WRITE(42,*) '**ABOVE COSTS ARE BASED ON ONE RUN ONLY**'
C
        CALL SUMRY
        CALL GETTIM (IHR,IMIN,ISEC,I100TH)
        WRITE(*,436)IHR,IMIN,ISEC,I100TH
        STOP
      END IF
C
400   FORMAT (F6.0,F7.0,F8.0,F7.0,F8.0,F7.0,2F8.0,F7.0,F8.0,F9.0)
C
401   FORMAT(/40X,'ACTIVITY TIMES (DAYS)')
402   FORMAT(/35X,'PRODUCTION SCHEDULE TABLE (DAYS)')
403   FORMAT(////////////////35X,
     +       'PRODUCTION SCHEDULE TABLE (DAYS)')
```

```
404   FORMAT(21X,'------------------------------------',
     +       '-----------------------')
406   FORMAT(6X,'ACTIVITY',7X,'BLK 1',1X,'BLK 2',1X,'BLK 3',1X,'BLK 4'
     + ,1X,'BLK 5',1X,'BLK 6',1X,'BLK 7',1X,'BLK 8',1X,'BLK 9',1X,
     + 'BLK10')
408   FORMAT('--------------------',1X,'-----',1X,'-----',1X,'-----',
     +  1X,'-----',1X,'-----',1X,'-----',1X,'-----',1X,'-----',1X,
     +'-----',1X,'-----')
410   FORMAT(' DRILLING & BLASTING',10F6.1)
412   FORMAT(' MUCKING AND HAULING',10F6.1)
413   FORMAT('         BACKFILLING',10F6.1/)
414   FORMAT('         BACKFILLING',10F6.1)
C
417   FORMAT(8X,'EVENT',8X,'BLK 1',1X,'BLK 2',1X,'BLK 3',1X,'BLK 4',1X,
     + 'BLK 5',1X,'BLK 6',1X,'BLK 7',1X,'BLK 8',1X,'BLK 9',1X,'BLK10')
418   FORMAT('--------------------',1X,'-----',1X,'-----',1X,'-----',1X,
     +'-----',1X,'-----',1X,'-----',1X,'-----',1X,'-----',1X,'-----',1X,
     +'-----')
419   FORMAT(/1X,I2,'% CONFIDENCE LEVEL FOR ABOVE (+/-):')
420   FORMAT('FINISHED PREPARATION',10F6.1)
422   FORMAT('      START DRILLING',10F6.1)
424   FORMAT('   FINISHED BLASTING',10F6.1)
426   FORMAT('       START MUCKING',10F6.1)
428   FORMAT('    FINISHED MUCKING',10F6.1)
430   FORMAT('      START BACKFILL',10F6.1)
432   FORMAT('   FINISHED BACKFILL',10F6.1)
434   FORMAT(////36X,'AVERAGE ACTIVITY TIMES (DAYS)')
C
436   FORMAT( /////// 1X,'End time of simulation:   ',I2.2,':',
     +                            I2.2,':',I2.2,':',I2.2)
C
440   FORMAT(/20X,'MUCKING TONS PER DAY')
441   FORMAT(///17X,'AVERAGE MUCKING TONS PER DAY')
442   FORMAT(1X,'------------------------------------'
     +       '-----------------------')
444   FORMAT(1X,'BLK 1',1X,'BLK 2',1X,'BLK 3',1X,'BLK 4'
     + ,1X,'BLK 5',1X,'BLK 6',1X,'BLK 7',1X,'BLK 8',1X,'BLK 9',1X,
     + 'BLK10')
446   FORMAT(1X,'-----',1X,'-----',1X,'-----',
     +  1X,'-----',1X,'-----',1X,'-----',1X,'-----',1X,'-----',1X,
     +'-----',1X,'-----')
448   FORMAT(10F6.0)
C
460   FORMAT(/45X,'BALLPARK COSTS ($)')
462   FORMAT(21X,'------------------------------------',
     +       '-----------------------')
464   FORMAT(3X,'DESCRIPTION',7X,'BLK 1',1X,'BLK 2',1X,'BLK 3',1X,
     + 'BLK 4',1X,'BLK 5',1X,'BLK 6',1X,'BLK 7',1X,'BLK 8',1X,
     + 'BLK 9',1X,'BLK10')
466   FORMAT('--------------------',1X,'-----',1X,'-----',1X,'-----',
     +  1X,'-----',1X,'-----',1X,'-----',1X,'-----',1X,'-----',1X,
     +'-----',1X,'-----')
C
468   FORMAT('           DRILLING:')
469   FORMAT('        DRILL REPAIR',10F6.0)
470   FORMAT('              HAMMER',10F6.0)
471   FORMAT('          COMPRESSOR',10F6.0)
472   FORMAT('         BITS & RODS',10F6.0)
473   FORMAT('    DRILLING LABOUR',F10.0,' (FOR ALL BLOCKS )')
```

```
C
480   FORMAT('          BLASTING:')
481   FORMAT('             POWDER',10F6.0)
482   FORMAT('        ACCESSORIES',10F6.0)
483   FORMAT('     BLASTING LABOUR',F10.0,' (FOR ALL BLOCKS )')
C
485   FORMAT('              LHD1:')
486   FORMAT('        LHD1 REPAIR',F10.0,' (FOR ALL BLOCKS )')
487   FORMAT(' LHD1 MUCKING LABOUR',F10.0,' (FOR ALL BLOCKS )')
C
490   FORMAT('              LHD2:')
491   FORMAT('        LHD2 REPAIR',F10.0,' (FOR ALL BLOCKS )')
492   FORMAT(' LHD2 MUCKING LABOUR',F10.0,' (FOR ALL BLOCKS )')
C
      RETURN
      END
```

APPENDIX B: SIMULATION OUTPUT FILES

Appendix B contains a sample of the output files generated by the simulation program. A complete copy of files SUMMARY, RUNS and COSTS have been included; however, due to the length of file AMOUNTS only one run has been included. The files can be found as indicated below.

Appendix B.1: Output File SUMMARY

FILE: SUMMARY

```
******************************************
**  SUMMARY REPORT BASED ON 10 RUN(S)  **
******************************************
```

RESOURCE	AVAILABILITY RATIOS				AVERAGE HOURS					SHIFTS
	MA	PA	UA	EU	DPT	IPT	RT	SBT	PST	
DRILL1	.68	.79	.56	.44	469.	164.	301.	502.	1437.	3
DRILL2	.71	.84	.45	.38	405.	142.	231.	659.	1437.	3
LHD1	.73	.92	.23	.21	140.	50.	72.	646.	908.	2
LHD2	.61	.69	.71	.49	197.	245.	287.	180.	908.	2
TRUCK	.86	.86	.94	.82	341.	123.	79.	28.	571.	2
TRAIN	.97	.98	.69	.68	690.	230.	32.	410.	1363.	3
95% CONFIDENCE LEVEL FOR ABOVE (+/-):										
DRILL1	.04	.04	.03	.02	13.	5.	60.	61.	58.	0
DRILL2	.03	.02	.02	.02	11.	4.	35.	52.	58.	0
LHD1	.04	.02	.03	.03	23.	9.	20.	37.	38.	0
LHD2	.05	.04	.03	.05	13.	17.	44.	17.	38.	0
TRUCK	.04	.03	.02	.04	18.	7.	23.	9.	39.	0
TRAIN	.02	.01	.02	.02	9.	3.	17.	37.	57.	0

95% CONFIDENCE LEVEL FOR:

AVERAGE DAILY TONS OF ORE MUCKED:	1105.	+/-	46.
AVERAGE DAILY POUNDS OF NICKEL MUCKED:	34550.	+/-	1438.
AVERAGE DAILY TONS OF ORE TRAMMED:	1266.	+/-	47.
AVERAGE DAILY TONS OF ROCK TRAMMED:	149.	+/-	0.

AVERAGE MUCKING TONS PER DAY

BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
625.	639.	1002.	442.	0.	0.	0.	0.	0.	0.
95% CONFIDENCE LEVEL FOR ABOVE (+/-):									
56.	45.	93.	23.	0.	0.	0.	0.	0.	0.

PRODUCTION SCHEDULE TABLE (DAYS)

EVENT	BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
FINISHED PREPARATION	2.1	2.1	4.2	4.2	.0	.0	.0	.0	.0	.0
START DRILLING	2.1	2.1	21.3	28.9	.0	.0	.0	.0	.0	.0
FINISHED BLASTING	21.3	28.9	54.8	60.5	.0	.0	.0	.0	.0	.0
START MUCKING	5.2	6.6	34.2	35.3	.0	.0	.0	.0	.0	.0
FINISHED MUCKING	26.2	35.0	59.0	69.6	.0	.0	.0	.0	.0	.0
START BACKFILL	26.2	40.2	61.8	88.7	.0	.0	.0	.0	.0	.0
FINISHED BACKFILL	40.2	61.0	88.7	105.8	.0	.0	.0	.0	.0	.0
95% CONFIDENCE LEVEL FOR ABOVE (+/-):										
FINISHED PREPARATION	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
START DRILLING	.0	.0	1.6	1.8	.0	.0	.0	.0	.0	.0
FINISHED BLASTING	1.6	1.8	3.3	4.7	.0	.0	.0	.0	.0	.0
START MUCKING	.6	.5	3.0	2.1	.0	.0	.0	.0	.0	.0
FINISHED MUCKING	1.9	2.2	3.3	2.7	.0	.0	.0	.0	.0	.0
START BACKFILL	1.9	1.6	3.2	4.4	.0	.0	.0	.0	.0	.0
FINISHED BACKFILL	1.6	2.5	4.4	4.5	.0	.0	.0	.0	.0	.0

AVERAGE ACTIVITY TIMES (DAYS)

ACTIVITY	BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
DRILLING & BLASTING	19.2	26.8	33.5	31.7	.0	.0	.0	.0	.0	.0
MUCKING AND HAULING	21.1	28.4	24.8	34.3	.0	.0	.0	.0	.0	.0
BACKFILLING	14.0	20.8	27.0	17.1	.0	.0	.0	.0	.0	.0
95% CONFIDENCE LEVEL FOR ABOVE (+/-):										
DRILLING & BLASTING	1.6	1.8	2.5	3.3	.0	.0	.0	.0	.0	.0
MUCKING AND HAULING	1.9	1.9	2.3	1.7	.0	.0	.0	.0	.0	.0
BACKFILLING	.7	1.5	2.2	1.0	.0	.0	.0	.0	.0	.0

Appendix B.2: Output File RUNS

FILE: RUNS

****

RESULTS FOR RUN: 1

RESOURCE	AVAILABILITY RATIOS MA	PA	UA	EU	HOURS DPT	IPT	RT	SBT	PST	SHIFTS
DRILL1	.76	.85	.57	.48	493.	172.	207.	502.	1374.	3
DRILL2	.73	.85	.47	.40	409.	144.	201.	620.	1374.	3
LHD1	.76	.93	.25	.23	149.	53.	62.	611.	875.	2
LHD2	.61	.69	.71	.49	195.	236.	270.	173.	875.	2
TRUCK	.91	.91	.97	.89	377.	136.	50.	15.	578.	2
TRAIN	.98	.99	.71	.70	691.	230.	16.	374.	1312.	3

AVERAGE DAILY TONS OF ORE MUCKED: 1157.
AVERAGE DAILY POUNDS OF NICKEL MUCKED: 36117.
AVERAGE DAILY TONS OF ORE TRAMMED: 1318.
AVERAGE DAILY TONS OF ROCK TRAMMED: 149.

MUCKING TONS PER DAY

BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
699.	663.	874.	420.	0.	0.	0.	0.	0.	0.

PRODUCTION SCHEDULE TABLE (DAYS)

EVENT	BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
FINISHED PREPARATION	2.1	2.1	4.2	4.2	.0	.0	.0	.0	.0	.0
START DRILLING	2.1	2.1	19.0	28.4	.0	.0	.0	.0	.0	.0
FINISHED BLASTING	19.0	28.4	51.3	59.3	.0	.0	.0	.0	.0	.0
START MUCKING	4.7	6.3	26.1	34.3	.0	.0	.0	.0	.0	.0
FINISHED MUCKING	23.2	33.5	54.1	70.2	.0	.0	.0	.0	.0	.0
START BACKFILL	23.2	37.3	56.2	78.5	.0	.0	.0	.0	.0	.0
FINISHED BACKFILL	37.3	56.2	78.5	95.3	.0	.0	.0	.0	.0	.0

ACTIVITY TIMES (DAYS)

ACTIVITY	BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
DRILLING & BLASTING	16.9	26.3	32.3	30.9	.0	.0	.0	.0	.0	.0
MUCKING AND HAULING	18.6	27.2	28.0	35.9	.0	.0	.0	.0	.0	.0
BACKFILLING	14.1	18.9	22.3	16.8	.0	.0	.0	.0	.0	.0

****

RESULTS FOR RUN: 2

RESOURCE	AVAILABILITY RATIOS MA	PA	UA	EU	HOURS DPT	IPT	RT	SBT	PST	SHIFTS
DRILL1	.63	.78	.49	.38	447.	157.	354.	639.	1598.	3
DRILL2	.67	.83	.42	.35	410.	144.	272.	773.	1598.	3
LHD1	.73	.90	.29	.26	193.	72.	99.	661.	1024.	2
LHD2	.49	.60	.63	.38	171.	221.	407.	225.	1024.	2
TRUCK	.80	.81	.92	.75	359.	133.	124.	43.	660.	2
TRAIN	.94	.96	.65	.62	718.	239.	56.	522.	1536.	3

AVERAGE DAILY TONS OF ORE MUCKED: 987.
AVERAGE DAILY POUNDS OF NICKEL MUCKED: 30836.
AVERAGE DAILY TONS OF ORE TRAMMED: 1146.
AVERAGE DAILY TONS OF ROCK TRAMMED: 149.

MUCKING TONS PER DAY

BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
570.	589.	858.	403.	0.	0.	0.	0.	0.	0.

PRODUCTION SCHEDULE TABLE (DAYS)

EVENT	BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
FINISHED PREPARATION	2.1	2.1	4.2	4.2	.0	.0	.0	.0	.0	.0
START DRILLING	2.1	2.1	20.8	31.4	.0	.0	.0	.0	.0	.0
FINISHED BLASTING	20.8	31.4	59.3	68.7	.0	.0	.0	.0	.0	.0
START MUCKING	4.7	7.1	37.6	37.7	.0	.0	.0	.0	.0	.0
FINISHED MUCKING	27.4	37.6	66.2	75.1	.0	.0	.0	.0	.0	.0
START BACKFILL	27.4	42.3	67.2	93.3	.0	.0	.0	.0	.0	.0
FINISHED BACKFILL	42.3	64.5	93.3	112.1	.0	.0	.0	.0	.0	.0

ACTIVITY TIMES (DAYS)

ACTIVITY	BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
DRILLING & BLASTING	18.7	29.3	38.6	37.2	.0	.0	.0	.0	.0	.0
MUCKING AND HAULING	22.8	30.5	28.6	37.4	.0	.0	.0	.0	.0	.0
BACKFILLING	14.9	22.2	26.0	18.9	.0	.0	.0	.0	.0	.0

****

RESULTS FOR RUN: 3

RESOURCE	AVAILABILITY RATIOS MA	PA	UA	EU	HOURS DPT	IPT	RT	SBT	PST	SHIFTS
DRILL1	.64	.73	.65	.47	463.	162.	352.	341.	1318.	3
DRILL2	.71	.84	.48	.40	388.	136.	216.	577.	1318.	3
LHD1	.74	.93	.23	.21	132.	48.	63.	600.	843.	2
LHD2	.66	.73	.71	.52	201.	238.	225.	179.	843.	2
TRUCK	.93	.93	.97	.91	297.	107.	31.	10.	444.	2
TRAIN	1.00	1.00	.71	.71	675.	225.	4.	360.	1264.	3

AVERAGE DAILY TONS OF ORE MUCKED: 1183.
AVERAGE DAILY POUNDS OF NICKEL MUCKED: 37212.
AVERAGE DAILY TONS OF ORE TRAMMED: 1343.
AVERAGE DAILY TONS OF ROCK TRAMMED: 150.

MUCKING TONS PER DAY

BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
522.	710.	917.	507.	0.	0.	0.	0.	0.	0.

PRODUCTION SCHEDULE TABLE (DAYS)

EVENT	BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
FINISHED PREPARATION	2.1	2.1	4.2	4.2	.0	.0	.0	.0	.0	.0
START DRILLING	2.1	2.1	24.3	24.0	.0	.0	.0	.0	.0	.0
FINISHED BLASTING	24.3	24.0	57.0	43.8	.0	.0	.0	.0	.0	.0
START MUCKING	4.3	7.1	32.5	32.6	.0	.0	.0	.0	.0	.0
FINISHED MUCKING	29.2	32.4	59.2	62.3	.0	.0	.0	.0	.0	.0
START BACKFILL	29.2	42.5	60.4	82.6	.0	.0	.0	.0	.0	.0
FINISHED BACKFILL	42.5	60.4	82.6	98.1	.0	.0	.0	.0	.0	.0

ACTIVITY TIMES (DAYS)

ACTIVITY	BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
DRILLING & BLASTING	22.2	21.9	32.7	19.8	.0	.0	.0	.0	.0	.0
MUCKING AND HAULING	24.9	25.4	26.7	29.7	.0	.0	.0	.0	.0	.0
BACKFILLING	13.3	18.0	22.2	15.5	.0	.0	.0	.0	.0	.0

****

RESULTS FOR RUN: 4

RESOURCE	AVAILABILITY RATIOS MA	PA	UA	EU	HOURS DPT	IPT	RT	SBT	PST	SHIFTS
DRILL1	.68	.80	.54	.44	461.	162.	287.	520.	1430.	3
DRILL2	.71	.84	.48	.40	423.	148.	233.	626.	1430.	3
LHD1	.70	.91	.22	.20	132.	45.	75.	620.	873.	2
LHD2	.64	.70	.74	.52	201.	256.	260.	156.	873.	2
TRUCK	.83	.84	.95	.80	322.	115.	89.	21.	547.	2
TRAIN	.95	.97	.72	.69	680.	227.	43.	360.	1310.	3

AVERAGE DAILY TONS OF ORE MUCKED: 1143.
AVERAGE DAILY POUNDS OF NICKEL MUCKED: 35712.
AVERAGE DAILY TONS OF ORE TRAMMED: 1312.
AVERAGE DAILY TONS OF ROCK TRAMMED: 148.

MUCKING TONS PER DAY

BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
638.	587.	1195.	437.	0.	0.	0.	0.	0.	0.

PRODUCTION SCHEDULE TABLE (DAYS)

EVENT	BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
FINISHED PREPARATION	2.1	2.1	4.2	4.2	.0	.0	.0	.0	.0	.0
START DRILLING	2.1	2.1	22.1	30.4	.0	.0	.0	.0	.0	.0
FINISHED BLASTING	22.1	30.4	53.3	61.7	.0	.0	.0	.0	.0	.0
START MUCKING	7.1	6.4	37.1	37.1	.0	.0	.0	.0	.0	.0
FINISHED MUCKING	27.4	37.1	57.6	71.6	.0	.0	.0	.0	.0	.0
START BACKFILL	27.4	41.6	63.5	92.7	.0	.0	.0	.0	.0	.0
FINISHED BACKFILL	41.6	63.5	92.7	110.6	.0	.0	.0	.0	.0	.0

ACTIVITY TIMES (DAYS)

ACTIVITY	BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
DRILLING & BLASTING	20.1	28.3	31.2	31.2	.0	.0	.0	.0	.0	.0
MUCKING AND HAULING	20.4	30.7	20.5	34.5	.0	.0	.0	.0	.0	.0
BACKFILLING	14.1	21.9	29.1	17.9	.0	.0	.0	.0	.0	.0

****

RESULTS FOR RUN: 5

RESOURCE	AVAILABILITY RATIOS MA	PA	UA	EU	HOURS DPT	IPT	RT	SBT	PST	SHIFTS
DRILL1	.69	.80	.56	.45	472.	165.	280.	505.	1422.	3
DRILL2	.73	.86	.43	.37	392.	138.	198.	693.	1422.	3
LHD1	.79	.94	.26	.24	160.	57.	56.	627.	900.	2
LHD2	.56	.64	.71	.45	187.	222.	326.	165.	900.	2
TRUCK	.82	.83	.94	.78	341.	124.	99.	31.	594.	2
TRAIN	.98	.99	.69	.68	691.	230.	14.	414.	1350.	3

AVERAGE DAILY TONS OF ORE MUCKED: 1115.
AVERAGE DAILY POUNDS OF NICKEL MUCKED: 34829.
AVERAGE DAILY TONS OF ORE TRAMMED: 1276.
AVERAGE DAILY TONS OF ROCK TRAMMED: 149.

MUCKING TONS PER DAY

BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
679.	715.	911.	418.	0.	0.	0.	0.	0.	0.

PRODUCTION SCHEDULE TABLE (DAYS)

EVENT	BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10

EVENT	BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
FINISHED PREPARATION	2.1	2.1	4.2	4.2	.0	.0	.0	.0	.0	.0
START DRILLING	2.1	2.1	19.0	26.4	.0	.0	.0	.0	.0	.0
FINISHED BLASTING	19.0	26.4	53.1	61.3	.0	.0	.0	.0	.0	.0
START MUCKING	5.1	5.1	30.3	30.5	.0	.0	.0	.0	.0	.0
FINISHED MUCKING	24.2	30.3	57.2	66.6	.0	.0	.0	.0	.0	.0
START BACKFILL	24.2	39.2	61.3	90.3	.0	.0	.0	.0	.0	.0
FINISHED BACKFILL	39.2	61.3	90.3	108.3	.0	.0	.0	.0	.0	.0

ACTIVITY TIMES (DAYS)

ACTIVITY	BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
DRILLING & BLASTING	16.9	24.3	34.1	34.9	.0	.0	.0	.0	.0	.0
MUCKING AND HAULING	19.1	25.2	26.9	36.0	.0	.0	.0	.0	.0	.0
BACKFILLING	15.0	22.1	29.0	18.0	.0	.0	.0	.0	.0	.0

****

RESULTS FOR RUN: 6

RESOURCE	AVAILABILITY RATIOS MA	PA	UA	EU	HOURS DPT	IPT	RT	SBT	PST	SHIFTS
DRILL1	.79	.87	.56	.49	496.	174.	179.	519.	1369.	3
DRILL2	.75	.86	.49	.42	426.	148.	188.	607.	1369.	3
LHD1	.67	.95	.11	.11	70.	23.	45.	726.	865.	2
LHD2	.73	.77	.79	.61	236.	292.	197.	140.	865.	2
TRUCK	.91	.91	.96	.88	367.	132.	52.	19.	570.	2
TRAIN	.97	.98	.71	.70	681.	227.	25.	363.	1297.	3

AVERAGE DAILY TONS OF ORE MUCKED: 1157.
AVERAGE DAILY POUNDS OF NICKEL MUCKED: 36144.
AVERAGE DAILY TONS OF ORE TRAMMED: 1321.
AVERAGE DAILY TONS OF ROCK TRAMMED: 149.

MUCKING TONS PER DAY

BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
706.	737.	1054.	438.	0.	0.	0.	0.	0.	0.

PRODUCTION SCHEDULE TABLE (DAYS)

EVENT	BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
FINISHED PREPARATION	2.1	2.1	4.2	4.2	.0	.0	.0	.0	.0	.0
START DRILLING	2.1	2.1	19.3	27.3	.0	.0	.0	.0	.0	.0
FINISHED BLASTING	19.3	27.3	50.0	59.1	.0	.0	.0	.0	.0	.0
START MUCKING	5.1	6.7	31.1	32.1	.0	.0	.0	.0	.0	.0
FINISHED MUCKING	23.5	31.1	54.3	66.5	.0	.0	.0	.0	.0	.0
START BACKFILL	23.5	37.3	56.2	86.7	.0	.0	.0	.0	.0	.0
FINISHED BACKFILL	37.3	56.2	86.7	103.3	.0	.0	.0	.0	.0	.0

ACTIVITY TIMES (DAYS)

ACTIVITY	BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
DRILLING & BLASTING	17.2	25.2	30.7	31.8	.0	.0	.0	.0	.0	.0
MUCKING AND HAULING	18.4	24.4	23.2	34.4	.0	.0	.0	.0	.0	.0
BACKFILLING	13.8	18.9	30.5	16.6	.0	.0	.0	.0	.0	.0

****

RESULTS FOR RUN: 7

RESOURCE	AVAILABILITY RATIOS MA	PA	UA	EU	HOURS DPT	IPT	RT	SBT	PST	SHIFTS
DRILL1	.68	.80	.54	.43	473.	166.	305.	545.	1488.	3
DRILL2	.62	.77	.48	.37	408.	144.	339.	598.	1488.	3
LHD1	.64	.86	.28	.24	167.	57.	128.	583.	935.	2
LHD2	.54	.64	.67	.43	180.	218.	340.	197.	935.	2

RESOURCE	MA	PA	UA	EU	DPT	IPT	RT	SBT	PST	SHIFTS
TRUCK	.87	.87	.95	.83	345.	121.	70.	23.	560.	2
TRAIN	.97	.98	.68	.66	699.	233.	29.	442.	1403.	3

AVERAGE DAILY TONS OF ORE MUCKED: 1068.
AVERAGE DAILY POUNDS OF NICKEL MUCKED: 33355.
AVERAGE DAILY TONS OF ORE TRAMMED: 1230.
AVERAGE DAILY TONS OF ROCK TRAMMED: 149.

MUCKING TONS PER DAY

BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
553.	596.	1068.	410.	0.	0.	0.	0.	0.	0.

PRODUCTION SCHEDULE TABLE (DAYS)

EVENT	BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
FINISHED PREPARATION	2.1	2.1	4.2	4.2	.0	.0	.0	.0	.0	.0
START DRILLING	2.1	2.1	23.7	30.7	.0	.0	.0	.0	.0	.0
FINISHED BLASTING	23.7	30.7	55.1	64.1	.0	.0	.0	.0	.0	.0
START MUCKING	5.7	7.1	37.3	37.4	.0	.0	.0	.0	.0	.0
FINISHED MUCKING	29.1	37.3	60.2	74.2	.0	.0	.0	.0	.0	.0
START BACKFILL	29.1	42.3	63.5	91.4	.0	.0	.0	.0	.0	.0
FINISHED BACKFILL	42.3	63.5	91.4	106.5	.0	.0	.0	.0	.0	.0

ACTIVITY TIMES (DAYS)

ACTIVITY	BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
DRILLING & BLASTING	21.6	28.6	31.5	33.4	.0	.0	.0	.0	.0	.0
MUCKING AND HAULING	23.5	30.2	22.9	36.8	.0	.0	.0	.0	.0	.0
BACKFILLING	13.2	21.1	28.0	15.1	.0	.0	.0	.0	.0	.0

****

RESULTS FOR RUN: 8

RESOURCE	AVAILABILITY RATIOS				HOURS					
	MA	PA	UA	EU	DPT	IPT	RT	SBT	PST	SHIFTS
DRILL1	.72	.83	.53	.44	455.	160.	244.	541.	1400.	3
DRILL2	.73	.86	.42	.37	378.	134.	192.	696.	1400.	3
LHD1	.77	.94	.21	.20	127.	49.	53.	663.	892.	2
LHD2	.66	.72	.73	.53	207.	266.	246.	173.	892.	2
TRUCK	.84	.85	.94	.80	358.	127.	92.	32.	610.	2
TRAIN	.96	.97	.71	.69	691.	230.	34.	382.	1338.	3

AVERAGE DAILY TONS OF ORE MUCKED: 1130.
AVERAGE DAILY POUNDS OF NICKEL MUCKED: 35294.
AVERAGE DAILY TONS OF ORE TRAMMED: 1292.
AVERAGE DAILY TONS OF ROCK TRAMMED: 149.

MUCKING TONS PER DAY

BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
735.	620.	1186.	457.	0.	0.	0.	0.	0.	0.

PRODUCTION SCHEDULE TABLE (DAYS)

EVENT	BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
FINISHED PREPARATION	2.1	2.1	4.2	4.2	.0	.0	.0	.0	.0	.0
START DRILLING	2.1	2.1	18.7	28.7	.0	.0	.0	.0	.0	.0
FINISHED BLASTING	18.7	28.7	50.3	60.4	.0	.0	.0	.0	.0	.0
START MUCKING	4.7	5.6	34.6	35.6	.0	.0	.0	.0	.0	.0
FINISHED MUCKING	22.3	34.6	55.3	68.6	.0	.0	.0	.0	.0	.0
START BACKFILL	22.3	37.4	56.5	81.3	.0	.0	.0	.0	.0	.0
FINISHED BACKFILL	37.4	56.5	81.3	100.4	.0	.0	.0	.0	.0	.0

ACTIVITY TIMES (DAYS)

ACTIVITY	BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
DRILLING & BLASTING	16.6	26.6	31.7	31.8	.0	.0	.0	.0	.0	.0
MUCKING AND HAULING	17.7	29.0	20.6	33.0	.0	.0	.0	.0	.0	.0
BACKFILLING	15.1	19.1	24.8	19.2	.0	.0	.0	.0	.0	.0

****

RESULTS FOR RUN: 9

	AVAILABILITY RATIOS				HOURS					
RESOURCE	MA	PA	UA	EU	DPT	IPT	RT	SBT	PST	SHIFTS
DRILL1	.58	.68	.64	.44	488.	171.	477.	377.	1512.	3
DRILL2	.73	.87	.41	.36	397.	139.	196.	780.	1512.	3
LHD1	.82	.96	.20	.19	134.	46.	41.	739.	960.	2
LHD2	.59	.68	.70	.47	193.	259.	310.	198.	960.	2
TRUCK	.88	.89	.91	.81	340.	121.	61.	48.	570.	2
TRAIN	.92	.94	.67	.63	685.	228.	82.	443.	1439.	3

AVERAGE DAILY TONS OF ORE MUCKED: 1014.
AVERAGE DAILY POUNDS OF NICKEL MUCKED: 31915.
AVERAGE DAILY TONS OF ORE TRAMMED: 1176.
AVERAGE DAILY TONS OF ROCK TRAMMED: 150.

MUCKING TONS PER DAY

BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
533.	555.	874.	472.	0.	0.	0.	0.	0.	0.

PRODUCTION SCHEDULE TABLE (DAYS)

EVENT	BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
FINISHED PREPARATION	2.1	2.1	4.2	4.2	.0	.0	.0	.0	.0	.0
START DRILLING	2.1	2.1	24.0	32.8	.0	.0	.0	.0	.0	.0
FINISHED BLASTING	24.0	32.8	65.1	64.3	.0	.0	.0	.0	.0	.0
START MUCKING	5.1	7.1	39.5	39.6	.0	.0	.0	.0	.0	.0
FINISHED MUCKING	29.5	39.5	67.6	71.5	.0	.0	.0	.0	.0	.0
START BACKFILL	29.5	41.6	67.6	97.6	.0	.0	.0	.0	.0	.0
FINISHED BACKFILL	41.6	62.6	97.6	113.6	.0	.0	.0	.0	.0	.0

ACTIVITY TIMES (DAYS)

ACTIVITY	BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
DRILLING & BLASTING	21.9	30.7	41.1	31.6	.0	.0	.0	.0	.0	.0
MUCKING AND HAULING	24.4	32.4	28.0	31.9	.0	.0	.0	.0	.0	.0
BACKFILLING	12.2	20.9	30.1	16.0	.0	.0	.0	.0	.0	.0

****

RESULTS FOR RUN: 10

	AVAILABILITY RATIOS				HOURS					
RESOURCE	MA	PA	UA	EU	DPT	IPT	RT	SBT	PST	SHIFTS
DRILL1	.65	.78	.53	.41	442.	156.	327.	531.	1456.	3
DRILL2	.67	.81	.47	.39	417.	146.	271.	622.	1456.	3
LHD1	.65	.89	.23	.21	139.	50.	100.	631.	919.	2
LHD2	.61	.69	.69	.48	199.	240.	286.	195.	919.	2
TRUCK	.77	.79	.91	.72	310.	109.	123.	39.	581.	2
TRAIN	.98	.99	.68	.67	690.	230.	20.	438.	1378.	3

AVERAGE DAILY TONS OF ORE MUCKED: 1091.
AVERAGE DAILY POUNDS OF NICKEL MUCKED: 34086.
AVERAGE DAILY TONS OF ORE TRAMMED: 1251.
AVERAGE DAILY TONS OF ROCK TRAMMED: 148.

MUCKING TONS PER DAY

BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
614.	620.	1085.	456.	0.	0.	0.	0.	0.	0.

PRODUCTION SCHEDULE TABLE (DAYS)

EVENT	BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
FINISHED PREPARATION	2.1	2.1	4.2	4.2	.0	.0	.0	.0	.0	.0
START DRILLING	2.1	2.1	22.0	28.6	.0	.0	.0	.0	.0	.0
FINISHED BLASTING	22.0	28.6	53.8	62.8	.0	.0	.0	.0	.0	.0
START MUCKING	5.3	7.1	36.1	36.1	.0	.0	.0	.0	.0	.0
FINISHED MUCKING	26.5	36.1	58.7	69.2	.0	.0	.0	.0	.0	.0
START BACKFILL	26.5	40.5	65.5	93.1	.0	.0	.0	.0	.0	.0
FINISHED BACKFILL	40.5	65.5	93.1	110.1	.0	.0	.0	.0	.0	.0

ACTIVITY TIMES (DAYS)

ACTIVITY	BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
DRILLING & BLASTING	19.9	26.6	31.8	34.1	.0	.0	.0	.0	.0	.0
MUCKING AND HAULING	21.1	29.0	22.6	33.1	.0	.0	.0	.0	.0	.0
BACKFILLING	14.1	25.0	27.6	17.0	.0	.0	.0	.0	.0	.0

Appendix B.3: Output File AMOUNTS

FILE: AMOUNTS

****
RESULTS FOR RUN: 1

	MUCKING AND HAULING				TRAMMING				
	ORE		NICKEL		ORE		ROCK		ROCKPASS
TIME (Days)	DAILY (Tons)	TOTAL (Tons)	DAILY (lbs.)	TOTAL (lbs.)	DAILY (Tons)	TOTAL (Tons)	DAILY (Tons)	TOTAL (Tons)	TOTAL (Tons)
1.	0.	0.	0.	0.	88.	88.	136.	136.	14.
2.	0.	0.	0.	0.	174.	262.	130.	265.	35.
3.	0.	0.	0.	0.	178.	440.	171.	437.	13.
4.	0.	0.	0.	0.	87.	527.	130.	567.	33.
5.	99.	99.	2806.	2806.	269.	796.	185.	752.	-2.
6.	55.	154.	1557.	4363.	182.	978.	129.	881.	19.
7.	181.	334.	7125.	11488.	366.	1344.	129.	1010.	40.
8.	69.	403.	1960.	13447.	263.	1606.	179.	1189.	11.
9.	1059.	1463.	30078.	43526.	1159.	2765.	133.	1322.	28.
10.	637.	2100.	18101.	61627.	804.	3569.	140.	1462.	38.
11.	1516.	3616.	43060.	104687.	1686.	5255.	183.	1645.	5.
12.	1420.	5037.	50270.	154958.	1504.	6759.	48.	1692.	108.
13.	0.	5037.	0.	154958.	184.	6942.	222.	1914.	36.
14.	791.	5828.	31177.	186134.	976.	7919.	178.	2092.	8.
15.	965.	6793.	38018.	224152.	1072.	8990.	137.	2229.	21.
16.	1001.	7794.	39440.	263592.	1137.	10128.	132.	2361.	39.
17.	1052.	8845.	34455.	298047.	1233.	11361.	182.	2544.	6.
18.	1527.	10372.	43360.	341408.	1689.	13050.	137.	2680.	20.
19.	386.	10758.	10964.	352372.	526.	13576.	138.	2818.	32.
20.	1506.	12264.	42772.	395144.	1618.	15194.	175.	2994.	6.
21.	1511.	13775.	42910.	438054.	1673.	16867.	136.	3129.	21.
22.	1538.	15313.	43668.	481723.	1687.	18554.	135.	3265.	35.
23.	1483.	16796.	42123.	523845.	1688.	20242.	177.	3441.	9.
24.	1685.	18482.	61077.	584922.	1774.	22016.	134.	3575.	25.
25.	854.	19336.	33654.	618576.	1069.	23084.	137.	3712.	38.
26.	1820.	21156.	71708.	690284.	1957.	25042.	172.	3884.	16.
27.	320.	21476.	10504.	700788.	440.	25482.	133.	4017.	33.
28.	1713.	23188.	67477.	768265.	1866.	27348.	184.	4201.	-1.
29.	1677.	24865.	66056.	834321.	1870.	29218.	134.	4335.	15.
30.	1615.	26480.	63622.	897943.	1680.	30899.	131.	4466.	34.
31.	1529.	28009.	60249.	958192.	1695.	32593.	186.	4653.	-3.
32.	1734.	29743.	68336.	1026528.	1946.	34539.	131.	4784.	16.
33.	263.	30007.	10369.	1036897.	355.	34894.	133.	4917.	33.
34.	1508.	31514.	55435.	1092332.	1699.	36593.	172.	5088.	12.
35.	1069.	32583.	29272.	1121603.	1164.	37757.	134.	5222.	28.
36.	1507.	34090.	40394.	1161997.	1590.	39347.	179.	5401.	-1.
37.	1533.	35623.	41087.	1203084.	1685.	41032.	135.	5536.	14.
38.	1511.	37134.	40484.	1243567.	1678.	42711.	132.	5668.	32.
39.	1528.	38662.	40957.	1284525.	1699.	44410.	184.	5852.	-2.
40.	881.	39544.	23620.	1308145.	1051.	45461.	139.	5991.	9.
41.	0.	39544.	0.	1308145.	88.	45548.	139.	6130.	20.
42.	1342.	40886.	35975.	1344121.	1522.	47071.	134.	6264.	36.
43.	1537.	42423.	41187.	1385307.	1701.	48772.	176.	6440.	10.
44.	1562.	43984.	41850.	1427157.	1693.	50465.	135.	6576.	24.
45.	605.	44589.	16220.	1443377.	796.	51261.	136.	6712.	38.
46.	883.	45473.	23669.	1467046.	968.	52229.	179.	6890.	10.
47.	207.	45680.	5560.	1472606.	441.	52670.	138.	7028.	22.
48.	1277.	46957.	34220.	1506826.	1427.	54097.	136.	7164.	36.
49.	1510.	48467.	40456.	1547282.	1614.	55711.	178.	7342.	8.
50.	1517.	49984.	40664.	1587946.	1680.	57391.	138.	7480.	20.
51.	1481.	51465.	39682.	1627628.	1585.	58977.	137.	7617.	33.
52.	1446.	52911.	38765.	1666393.	1619.	60596.	186.	7803.	-3.
53.	1529.	54440.	40971.	1707364.	1669.	62264.	134.	7937.	13.
54.	1064.	55504.	28524.	1735887.	1231.	63496.	135.	8072.	28.

55.	986.	56490.	29726.	1765614.	888.	64384.	139.	8210.	40.
56.	1324.	57814.	41029.	1806643.	1760.	66144.	176.	8386.	14.
57.	1504.	59318.	46612.	1853255.	1587.	67731.	135.	8521.	29.
58.	1572.	60889.	48721.	1901976.	1770.	69501.	173.	8694.	6.
59.	1641.	62530.	50875.	1952852.	1844.	71345.	136.	8830.	20.
60.	1553.	64083.	48147.	2000999.	1682.	73027.	135.	8965.	35.
61.	57.	64140.	1760.	2002759.	182.	73209.	183.	9148.	2.
62.	1613.	65753.	50011.	2052770.	1510.	74718.	0.	9148.	152.
63.	516.	66270.	16002.	2068772.	805.	75523.	274.	9422.	28.
64.	1556.	67826.	48241.	2117013.	1781.	77304.	137.	9560.	40.
65.	0.	67826.	0.	2117013.	181.	77484.	175.	9735.	15.
66.	840.	68665.	26030.	2143044.	1057.	78541.	135.	9869.	31.
67.	0.	68665.	0.	2143044.	177.	78718.	172.	10041.	9.
68.	0.	68665.	0.	2143044.	85.	78803.	136.	10177.	23.
69.	0.	68665.	0.	2143044.	171.	78974.	174.	10351.	-1.
70.	1623.	70289.	50315.	2193359.	1796.	80770.	135.	10486.	14.
71.	377.	70665.	11683.	2205041.	440.	81210.	140.	10626.	24.
72.	0.	70665.	0.	2205041.	175.	81385.	135.	10761.	39.
73.	0.	70665.	0.	2205041.	179.	81565.	183.	10944.	6.
74.	0.	70665.	0.	2205041.	0.	81565.	0.	10944.	156.
75.	0.	70665.	0.	2205041.	268.	81832.	274.	11218.	32.
76.	0.	70665.	0.	2205041.	183.	82015.	176.	11394.	6.
77.	0.	70665.	0.	2205041.	89.	82105.	132.	11526.	24.
78.	0.	70665.	0.	2205041.	172.	82276.	136.	11662.	38.
79.	0.	70665.	0.	2205041.	181.	82457.	179.	11841.	9.
80.	0.	70665.	0.	2205041.	173.	82631.	140.	11981.	19.
81.	0.	70665.	0.	2205041.	87.	82718.	131.	12112.	38.
82.	0.	70665.	0.	2205041.	181.	82899.	177.	12289.	11.
83.	0.	70665.	0.	2205041.	171.	83070.	131.	12420.	30.
84.	0.	70665.	0.	2205041.	91.	83161.	181.	12601.	-1.
85.	0.	70665.	0.	2205041.	181.	83342.	126.	12727.	23.
86.	0.	70665.	0.	2205041.	184.	83525.	137.	12864.	36.
87.	0.	70665.	0.	2205041.	86.	83612.	187.	13051.	-1.
88.	0.	70665.	0.	2205041.	177.	83789.	133.	13184.	16.
89.	0.	70665.	0.	2205041.	177.	83966.	133.	13316.	34.
90.	0.	70665.	0.	2205041.	88.	84054.	177.	13493.	7.
91.	0.	70665.	0.	2205041.	180.	84234.	135.	13628.	22.
92.	0.	70665.	0.	2205041.	185.	84418.	137.	13765.	35.
93.	0.	70665.	0.	2205041.	92.	84511.	182.	13948.	2.
94.	0.	70665.	0.	2205041.	181.	84691.	138.	14086.	14.
95.	0.	70665.	0.	2205041.	173.	84864.	137.	14223.	27.
95.	0.	70665.	0.	2205041.	0.	84864.	0.	14223.	83.

Appendix B.4: Output File COSTS

FILE: COSTS

BALLPARK COSTS ($)

DESCRIPTION	BLK 1	BLK 2	BLK 3	BLK 4	BLK 5	BLK 6	BLK 7	BLK 8	BLK 9	BLK10
DRILLING:										
DRILL REPAIR	10257.	13009.	19754.	12213.	0.	0.	0.	0.	0.	0.
HAMMER	264.	335.	509.	315.	0.	0.	0.	0.	0.	0.
COMPRESSOR	1610.	2042.	3101.	1917.	0.	0.	0.	0.	0.	0.
BITS & RODS	2478.	3143.	4773.	2951.	0.	0.	0.	0.	0.	0.
DRILLING LABOUR	34805. (FOR ALL BLOCKS)									
BLASTING:										
POWDER	5972.	8280.	11266.	6937.	0.	0.	0.	0.	0.	0.
ACCESSORIES	3246.	4500.	6123.	3770.	0.	0.	0.	0.	0.	0.
BLASTING LABOUR	9346. (FOR ALL BLOCKS)									
LHD1:										
LHD1 REPAIR	18204. (FOR ALL BLOCKS)									
LHD1 MUCKING LABOUR	6565. (FOR ALL BLOCKS)									
LHD2:										
LHD2 REPAIR	21493. (FOR ALL BLOCKS)									
LHD2 MUCKING LABOUR	15013. (FOR ALL BLOCKS)									

ABOVE COSTS ARE BASED ON ONE RUN ONLY

APPENDIX C: TIME AND AVAILABILITY RATIOS

The definitions for the time and the availability ratios used in the simulation model are based upon the work performed by Elbrond (1977) and Sense (1972); however, amendments to their definitions were performed as necessary in order to suit the specific needs of this project.

Basically time, or more specifically Calendar Time (CT), has been divided into Non-Production Scheduled Time (NPST) and Production Scheduled Time (PST). That is:

$$CT = NPST + PST \tag{C.1}$$

NPST consists of holidays, weekends, shift shutdowns, plant maintenance/shutdowns and closures of operations by force majeure (such as climatic conditions, accidents, incidents and strikes).

PST consists of the following:

1) Direct Production Time (DPT), which is the time the machine is directly used to carry out the assigned task.
2) Indirect Production Time (IPT), which consists of stops in DPT due to shift changes, lunch, coffee breaks, minor daily maintenance, moves required for accomplishing the task, and minor waiting.
3) Repair Time (RT), which is the time to repair equipment breakdowns, and to perform preventive maintenance that is not part of the daily maintenance (this includes stops in RT due to shift changes, lunch, coffee breaks and delays).

4) Stand By Time (SBT), which is the time the machine is prepared to accomplish its task but is not required.

Based upon the above, the following ratios were of interest:

Mechanical availability (i.e., available when needed), or MA,

$$= (DPT + IPT) / (DPT + IPT + RT) \quad (C.2)$$

Physical availability (i.e., actually available), or PA,

$$= (DPT + IPT + SBT) / PST \quad (C.3)$$

Use of availability (i.e., used when available), or UA,

$$= (DPT + IPT) / (DPT + IPT + SBT) \quad (C.4)$$

Effective utilization (i.e., used effectively), or EU,

$$= (DPT + IPT) / PST \quad (C.5)$$

Within the simulation model, the time for drill related activities have been classified as shown in Table C.1; the time for LHD related activities in Table C.2; the time for truck related activities in Table C.3; and the time for train related activities in Table C.4.

Table C.1 Drill related activities time classification.

Activity	Classification
Relocate drill (see Note 1)	IPT
Drilling holes (see Note 2)	DPT
Clean holes with drill (see Note 2)	DPT
Clean holes without drill (see Note 1)	IPT
Measure holes (see Note 4)	SBT
Load holes (see Note 4)	SBT
Guard and blast (see Note 4)	SBT
Drill downtime (see Note 3)	RT
All remaining time not accounted for above (see Note 4)	SBT

Note 1: Within the simulation model, for every 6 hours of IPT collected, an additional 2 hours is added to the cumulative IPT value since there are 2 hours of IPT (non-productive time) that are not collected for every 8 hour shift (see the shifts segment of the model). This is accomplished within the USERF FORTRAN code by adding an additional 1/3 of the incremental IPT value to the cumulative IPT value, that is,

cumulative IPT + incremental IPT + incremental IPT/3,

which can be written as

cumulative IPT + 4*incremental IPT/3.

For example, for every 6 hours of IPT collected, there is an increase of 8 hours in the cumulative IPT value.

Note 2: Within the simulation model, for every 6 hours of DPT collected, an additional 2 hours is added to the cumulative IPT value since there are 2 hours of IPT (non-productive time) that are not collected for every 8 hour shift (see the shifts segment of the model). This is accomplished within the USERF FORTRAN code by adding 1/3 of the incremental DPT value to the cumulative IPT value. For example, for every 6 hours of DPT collected, there is an increase of 2 hours to the cumulative IPT value.

Note 3: Within the simulation model, RT is not incremented in the same fashion as DPT and IPT since the 2 hours of non-productive time per repair shift is considered part of RT, therefore, it is collected, or accounted for, in the random generation of the incremental RT value.

Note 4: Within the simulation model, SBT is calculated when the FORTRAN STATS subroutine is called, and is equal to the time not accounted for in DPT, IPT and RT. That is, SBT = PST - DPT - IPT - RT.

Table C.2 LHD related activities time classification.

Activity	Classification
Wait due to orepass being full (see Note 1)	IPT
Mucking and hauling (see Note 2)	DPT
Blast delay and check vents (see Note 1)	IPT
LHD downtime (see Note 3)	RT
All remaining time not accounted for above (see Note 4)	SBT

For an explanations of notes see Table C.1.

Table C.3 Truck related activities time classification.

Activity	Classification
Hauling rockfill (see Note 2)	DPT
Blast delay and check vents (see Note 1)	IPT
Truck downtime (see Note 3)	RT
All remaining time not accounted for above (see Note 4)	SBT

For an explanations of notes see Table C.1.

Table C.4 Train related activities time classification.

Activity	Classification
Tramming rock or ore (see Note 2)	DPT
Train downtime (see Note 3)	RT
All remaining time not accounted for above (see Note 4)	SBT

For an explanations of notes see Table C.1.

APPENDIX D: MODEL ASSUMPTIONS, DATA AND EQUATIONS

Appendix D provides documentation of the assumptions made in building the model, as well as, a detailed listing of the model data and equations used. It has been divided as shown below:

Appendix D.1: Introduction

Appendices D.2 through D.5 each begin with a list of general assumptions that were applicable to the 83 orebody when the model was built. The activity times presented within these appendices in Tables D.1 through D.24, are based upon Inco's incentive standards hourly rates, or incentive rates. The incentive rates in turn are based upon time studies conducted by Inco personnel at Thompson, as well as Sudbury, and are used as a basis for incentive pay allowances; therefore, they are considered to be relatively accurate. Nevertheless, portions of the incentive rates did not accurately reflect some of the activities at the 83 orebody, consequently, they were adjusted as deemed appropriate. These adjustments were made following consultation with various individuals at Inco who were qualified to make such judgments (i.e., expert opinions), and basically encompassed deletions of activities not performed at the 83 orebody, as well as, modifications to various activity times. The adjusted incentive rates were then assumed to represent the average activity times, and used as a basis in estimating the minimum, maximum and most likely values for each activity. These values were then input into UniFit II, a probability distribution calculation software package, to obtain corresponding SLAM II equations.

Appendix D.6 provides the assumptions and data for the equipment uptime and downtime equations. Historical records, along with expert opinions, were used to estimate various values which were entered into UniFit II to obtain corresponding SLAM II equations.

Appendix D.7 lists various assumptions and estimates for chimney capacities based upon the existing chimney dimensions when the model was built.

Appendix D.2: Drilling and Blasting

Drilling and blasting (which includes relocating the drill, drilling holes, cleaning holes, measuring holes, loading explosives, and guarding and blasting) is based upon the following general assumptions:

- An In-The-Hole (ITH) drill is used,
- Drill rods are 3 inches in diameter by 5 feet in length,
- Holes drilled are 4.5 inches in diameter with a depth range from 0 to 100 feet,
- Slash blasting (vertical slices),
- Average penetration rate for drilling is 40 feet per hour,
- Average penetration rate for cleaning holes with the drill is 60 feet per hour, and
- Three shifts per day with 1 drill operator per shift (rates are also applicable for 2 drill operators per shift since there is no significant difference for the 4.5 inch holes).

The productive hours per shift for drilling and blasting have been reduced from 8 to 6 hours based upon the non-productive time activities listed in Table D.1.

Table D.1 Non-productive time per shift for drilling and blasting.

Activity	Unit	Minutes
Lunch	time/shift	30.00
Travel	time/shift	30.00
Supervision	time/shift	10.00
Personal	time/shift	10.00
General Preparation	time/shift	10.00
Service Equipment	time/shift	30.00
Total	time/shift	120.00

Relocating the Drill

Time to relocate the drill (TRD) has been divided into travel time plus preparation time for travel. That is:

TRD = Travel time + Preparation time (D.1)

Travel time is the time to move the drill from blast section of ore to blast section of ore, as well as, the time to move from mining block to mining block. Preparation time is the time to get ready for the move, which includes such activities as putting the mast to the horizontal position, washing down the machine and securing the panel for travel.

Assuming the drill travelling speed (DTS) varies with an average value of 110, minimum of 50, maximum of 200, and most likely value of 100 feet per minute, then the accompanying SLAM II equation obtained from UniFit II is:

DTS = 50 + 150*BETA(2,3,<stream>) (D.2)

Assuming preparation time (PT) varies with an average value of 20, minimum of 10, maximum of 30, and most likely value of 15 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

PT = 10 + 20*BETA(1.333,2,<stream>) (D.3)

Thus, using equations D.1 through D.3:

TRD (in hours) = (number of feet moved / DTS + PT) / 60 (D.4)

Drilling

Drilling includes set-up operation times as shown in Table D.2, collar and case hole operation times as shown in Table D.3, tear down operation times as shown in Table D.4, and drilling holes operation times as shown in Table D.5.

Table D.2 Set-up operation times for drilling.

Activity	Unit	Minutes
Move hole to hole	time/hole	2.88
Check print	time/hole	0.84
Line up mast over hole	time/hole	6.37
Set-up control panel & hoses	time/hole	0.30
Set-up rod platform	time/hole	0.38
Secure mast	time/hole	2.18
Prepare area for hole	time/hole	4.47
Set mast to required angle	time/hole	9.00
Total	time/hole	26.42

Table D.3 Collar and case hole operation times for drilling.

Activity	Unit	Minutes
Start compressor	time/hole	1.86
Collar and flush hole	time/hole	3.98
Prepare & install casing	time/hole	6.66
Prepare & install fabric seal	time/hole	6.93
Shovel cuttings around casing	time/hole	5.80
Insert plug in completed hole	time/hole	0.39
Total	time/hole	25.62

Table D.4 Tear down operation times for drilling.

Activity	Unit	Minutes
Shut off & bleed compressor	time/hole	1.48
Stingers down, or remove mast anchor	time/hole	0.69
Total	time/hole	2.17

Table D.5 Drilling holes operation times.

Activity	Unit	Minutes
Penetration rate	time/foot	1.50
Add rods	time/foot	0.40
Pull rods	time/foot	0.30

In general, the time for drilling for a blast section of ore (TD) is:

TD = Number of holes (setup time per hole + collar and case hole time per hole + tear down time per hole) + Number of feet drilled (penetration rate + time to add rods + time to pull rods) (D.5)

Substituting values from Tables D.2 through D.5 into equation D.5 and summing like terms:

TD (in minutes) = Number of holes (54.21) + Number of feet drilled (2.2) (D.6)

Assuming time 54.21 in equation D.6 (hereafter represented by the variable T1) varies with an average value of 54.21, minimum of 45, maximum of 80, and most likely value of 50 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

T1 = 45 + 35*BETA(1.56261,4.37564,<stream>) (D.7)

Assuming time 2.2 in equation D.6 (hereafter represented by the variable T2) varies with an average value of 2.2, minimum of 1.75, maximum of 3, and most likely value of 2 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

T2 = 1.75 + 1.25*BETA(1.35,2.4,<stream>) (D.8)

Thus, using equations D.6 through D.8:

TD (in hours) = (Number of holes * T1 + Number of feet drilled *T2) / 60 (D.9)

Cleaning Blast Holes

Cleaning blast holes operation times using the drill are listed in Table D.6, and cleaning blast holes operation times by hand are listed in Table D.7.

Table D.6 Cleaning blast holes operation times using the drill.

Activity	Unit	Minutes
Set-up operations	time/hole	26.42
Tear down operations	time/hole	8.88
Add rods	time/foot	0.40
Pull rods	time/foot	0.30
Lower rods & clean hole (60 feet/hour)	time/foot	1.00

Table D.7 Cleaning blast holes operation times by hand.

Activity	Unit	Minutes
General preparation	time/hole	0.07
Prepare gear	time/hole	0.14
Clean hole by hand	time/hole	3.58
Store gear	time/hole	0.21
Total	time/hole	4.00

In general, the time to clean blast holes for a blast section of ore (TCBH) is:

$$\text{TCBH} = \text{time to clean holes using drill} + \text{time to clean holes by hand} \quad \text{(D.10)}$$

Using equation D.10 and Tables D.6 and D.7, and assuming 12.5 % of the holes are cleaned using the drill and 12.5 % of the holes are cleaned by hand, then the time to clean blast holes when additional holes are drilled for a blast section of ore (TCBHA) is:

TCBHA (in minutes) = 0.125 (number of holes per blast*35.3 + number of feet drilled per blast*1.7) + 0.125 (number of holes per blast*4.0) (D.11)

Assuming time 35.3 in equation D.11 (hereafter represented by the variable T3) varies with an average value of 35.3, minimum of 30, maximum of 45, and most likely value of 33 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

T3 = 30 + 15*BETA(1.38261,2.53043,<stream>) (D.12)

Assuming time 1.7 in equation D.11 (hereafter represented by the variable T4) varies with an average value of 1.7, minimum of 1.5, maximum of 2.2, and most likely value of 1.6 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

T4 = 1.5 + 0.7*BETA(1.42857,3.57143,<stream>) (D.13)

Assuming time 4.0 in equation D.11 (hereafter represented by the variable T5) varies with an average value of 4.0, minimum of 3.0, maximum 7.54, and most likely value 3.5 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

T5 = 3 + 4.54*BETA(1.55947,5.52053,<stream>) (D.14)

Thus, using equations D.11 through D.14:

TCBHA (in hours) = [0.125 (number of holes per blast*T3 + number of feet drilled per blast*T4) + 0.125 (number of holes per blast*T5)] / 60 (D.15)

The time to clean blast holes when additional holes are not drilled for a blast section of ore (TCBHN) is based upon the holes from the previous blast section of ore. In these cases, assuming 50 % of the holes are cleaned using the drill, 50 % of the holes are cleaned by hand, and only 60 % of the total number of feet from the previous blast of ore is remaining, then the formula is:

TCBHN (in hours) =

[0.5 (number of holes per previous blast*T3) +

0.3 (number of feet drilled per previous blast*T4) +

0.5 (number of holes per previous blast*T5)] / 60 (D.16)

Measuring Blast Holes

Measuring blast hole operation times are listed in Table D.8.

Table D.8 Measuring blast hole operation times.

Activity	Unit	Minutes
General preparation	time/blast	2.17
Gather materials	time/blast	1.19
Store materials	time/blast	0.62
Prepare measuring line	time/hole	1.03
Set line and record reading	time/hole	0.58
Lower and pull out measuring line	time/hole	1.00

In general, the time to measure blast holes for a blast section of ore (TMBH) is:

TMBH = time for operations per blast + number of holes per

blast (time for operations per hole) (D.17)

Summing like terms in Table D.8 and substituting them into equation D.17:

TMBH (in minutes) = 3.98 +(number of holes per blast)*2.61 (D.18)

Assuming time 3.98 in equation D.18 (hereafter represented by the variable T6) varies with an average value of 3.98, minimum of 3.5, maximum of 4.5, and most likely value of 3.8 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

T6 = 3.5 + BETA(1.06667,1.15556,<stream>) (D.19)

Assuming time 2.61 in equation D.18 (hereafter represented by the variable T7) varies with an average value of 2.61, minimum of 2.4, maximum of 3.2, and most likely value of 2.5 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

T7 = 2.4 + 0.8*BETA(1.43182,4.02273,<stream>) (D.20)

Thus, using equations D.18 through D.20, the time to measure blast holes when additional holes are drilled for a blast section of ore (TMBHA) is:

TMBHA (in hours) = (T6 + number of holes per blast
*T7) / 60 (D.21)

When additional holes are not drilled for a blast section of ore (TMBHN), then the holes from the previous blast section of ore are used and the formula is:

TMBHN (in hours) = (T6 + number of holes per previous blast
* T7) / 60 (D.22)

Loading Explosives

Loading explosives operation times are listed in Table D.9. It is assumed that explosives are transported to the workplace by a worker not included in the simulation model and 15 feet of stemming is used for each hole loaded.

Table D.9 Loading explosives operation times.

Activity	Unit	Minutes
General preparation	time/blast	30.00
Scale workplace	time/blast	15.00
Gather & set-up miscellaneous material	time/blast	10.00
Nip hole plug	time/plug	1.20
Prepare plug for installation	time/plug	1.50
Lower plug into blast hole	time/foot of hole	0.04
Measure plug depth and set plug	time/plug	2.00
Prepare bags of ballast	time/plug	2.00
Cover plug with ballast	time/plug	2.00
Prepare explosive charges	time/primer	0.13

Table D.9 (Continued.)

Activity	Unit	Minutes
Prepare primer & lower into hole	time/primer	0.82
Unload stemming material	time/foot of hole loaded	0.10
Stem holes	time/foot of hole loaded	0.25
Wire up blast	time/hole	2.00
Clean up after loading	time/blast	30.00
Tie off detonator cord	time/hole	0.74
Inspect holes	time/hole	1.00
Cycle maintenance	time/blast	15.00
Measure depth of explosive charge	time/hole	0.15
Fill blast hole with explosive	time/foot of hole	0.06

Summing similar unit times in Table D.9:

Total for "time/blast" = 100.0

Total for "time/hole" = 3.89

Total for "time/foot of hole loaded" = 0.35

Total for "time/plug" = 8.7

Total for "time/primer" = 0.95

Total for "time/foot of hole" = 0.1

Based on the above rates, and assuming 41 feet loaded per plug and 8.2 feet loaded per primer, then the time to load explosives for a blast section of ore when additional holes are drilled (TLEA) is:

TLEA (in minutes) = 100.0 + number of holes per blast (3.89) + number of feet loaded per blast (0.35) + number of feet loaded per blast (1/41 feet loaded per plug) (8.7) + number of feet loaded per blast (1/8.2 feet loaded per primer)(0.95)+ number of feet per blast (0.1) (D.23)

Grouping like terms in equations D.23:

TLEA (in minutes) = 100 + number of holes per blast (3.89) +

number of feet loaded per blast (0.68) +

number of feet per blast (0.1) (D.24)

Assuming time 100 in equation D.24 (hereafter represented by the variable T8) varies with an average value of 100, minimum of 63, maximum of 237, and most likely value of 75 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

T8 = 63 + 174*BETA(1.27586,4.72414,<stream>) (D.25)

Assuming time 3.89 in equation D.24 (hereafter represented by the variable T9) varies with an average value of 3.89, minimum of 2.8, maximum of 6, and most likely value of 3.2 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

T9 = 2.8 + 3.2*BETA(1.18478,2.29348,<stream>) (D.26)

Assuming time 0.68 in equation D.24 (hereafter represented by the variable T10) varies with an average value of 0.68, minimum of 0.5, maximum of 1.25, and most likely value of 0.55 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

T10 = 0.5 + 0.75*BETA(1.2,3.8,<stream>) (D.27)

Assuming time 0.1 in equation D.24 (hereafter represented by the variable T11) varies with an average value of 0.1, minimum of 0.08, maximum of 0.15, and most likely value of 0.095 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

T11 = 0.08 + 0.07*BETA(2.28571,5.71429,<stream>) (D.28)

Thus, using equations D.24 through D.28:

TLEA (in hours) = (T8 +

number of holes per blast * T9 +

number of feet loaded per blast * T10 +

number of feet per blast * T11) / 60 (D.29)

When additional holes are not drilled for a blast of ore, then the holes from the previous blast section of ore are used. Assuming only 60 % of the total number of feet from the previous blast section of ore is remaining, then the time to load explosives when additional holes are not drilled (TLEN) is:

TLEN (in hours) = (T8 +

number of holes per previous blast * T9 +

number of feet loaded per blast * T10 +

0.6*number of feet per previous blast * T11) / 60 (D.30)

Guarding and Blasting

Guarding and blasting operation times are listed in Table D.10.

Table D.10 Guarding and blasting operation times.

Activity	Unit	Minutes
Call blast	time/blast	0.78
Light up blast	time/blast	1.42
Guard blast	time/blast	67.80
Clean up after blast	time/blast	10.00
Total	time/blast	80.00

Assuming time 80 in Table D.10 (hereafter represented by the variable T12) varies with an average value of 80, minimum of 60, maximum of 132.2, and most likely value of 70 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

T12 = 60 + 72.2*BETA(1.44598,3.77402,<stream>) (D.31)

Thus, using equation D.31, the time to guard and blast (TGB) is:

TGB (in hours) = T12/60 (D.32)

Appendix D.3: Mucking and Hauling

Mucking and hauling rates are based upon the following general assumptions:

- The amount of ore per bucket for the 6 cubic yard LHD follows a triangular distribution with a minimum of 6, mode of 7 and maximum value of 8 tons,
- The amount of ore per bucket for the 9 cubic yard LHD follows a triangular distribution with a minimum of 12.5, mode of 14 and maximum value of 15.5 tons,
- Broken ore is 14.5 cubic feet per ton,
- When 25 percent of a blast section of ore is mucked the blastroom resource is freed,
- Fifty percent is mucked remotely and 50 percent is mucked manually, and
- Two shifts per day are available for mucking and hauling.

The productive hours per shift for mucking and hauling have been reduced from 8 to 6 hours based upon the non-productive time activities listed in Table D.11.

Table D.11 Non-productive time per shift for mucking and hauling.

Activity	Unit	Minutes
Lunch	time/shift	30.00
Supervision	time/shift	10.00
Personal	time/shift	10.00
General inspection & preparation	time/shift	15.00
Service LHD	time/shift	30.00
Set up remote control unit	time/shift	3.97
Vent fan on	time/shift	1.45
Vent fan off	time/shift	1.34
Park and stop LHD	time/shift	1.62
Travel, clean & maintain haulage way, etc.	time/shift	16.62
Total	time/shift	120.00

Blast Delay and Check Vents

Blast delay and check vents times applicable to both the 6 and 9 cubic yard LHDs are given in Table D.12.

Table D.12 Blast delay and check vents times.

Activity	Unit	Minutes
Blast Delay	time/blast	32.55
Ventilation Check	time/blast	22.45
Total	time/blast	55.00

Assuming time 55 in Table D.12 (hereafter represented by the variable T13) varies with an average value of 55, minimum of 45, maximum of 82.45, and most likely value of 50 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

$$T13 = 45 + 37.45*BETA(1.46595,4.02405,<stream>) \quad (D.33)$$

Thus, using equation D.33, the time for each blast delay and check vents for the LHDs (TBDCV) is:

$$TBDCV \text{ (in hours)} = T13 / 60 \quad (D.34)$$

6 Cubic Yard LHD

Mucking and hauling operation times for manual loading/dumping for the 6 cubic yard LHD are given in Table D.13, and for the remote loading/dumping in Table D.14.

Table D.13 Manual loading/dumping operation times for 6 cubic yard.

Activity	Unit	Minutes
Initial load bucket	time/bucket	1.00
Dump bucket	time/bucket	0.23
Total	time/bucket	1.23

Table D.14 Remote loading/dumping operation times for 6 cubic yard.

Activity	Unit	Minutes
Initial load bucket	time/bucket	1.24
Dump bucket	time/bucket	0.23
Switch to remote control	time/bucket	0.28
Switch from remote control	time/bucket	0.32
LHD remote travel in drawpoint	time/bucket	0.24
Total	time/bucket	2.31

In general, the time to muck and haul a bucket (TMHB) is:

$$\text{TMHB} = \text{Load/Dump time} + \text{Travel time} \quad \text{(D.35)}$$

Assuming the travel speed (full and empty) for the 6 cubic yard (TS6) varies with an average value of 371, minimum of 345, maximum of 440, and most likely value of 355 feet per minute, then the accompanying SLAM II equation obtained from UniFit II is:

$$\text{TS6} = 345 + 95*\text{BETA}(1.28289,3.40461,<\text{stream}>) \quad \text{(D.36)}$$

Assuming time 1.23 in Table D.13 (hereafter represented by the variable T14) varies with an average value of 1.23, minimum of 0.75, maximum of 2, and most likely value of 1 minute, then the accompanying SLAM II equation obtained from UniFit II is:

$$\text{T14} = 0.75 + 1.25*\text{BETA}(1.25217,2.0087,<\text{stream}>) \quad \text{(D.37)}$$

Thus, using equations D.35 through D.37, the time per bucket for the 6 cubic yard for manual loading (TPB6M) is:

$$\text{TPB6M (in hours)} = [\text{T14} + \text{travelling distance in feet} / \text{TS6}] / 60 \quad \text{(D.38)}$$

Assuming time 2.31 in Table D.14 (hereafter represented by the variable T15) varies with an average value of 2.31, minimum of 1.5, maximum of 4, and most likely value of 1.9 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

$$\text{T15} = 1.5 + 2.5*\text{BETA}(1.34341,2.80293,\text{<stream>}) \quad \text{(D.39)}$$

Thus, using equations D.35, D.36 and D39, the time per bucket for the 6 cubic yard for remote control loading (TPB6R) is:

$$\text{TPB6R (in hours)} = [\text{T15} + \text{travelling distance in feet} / \text{TS6}] / 60 \quad \text{(D.40)}$$

9 Cubic Yard LHD

Mucking and hauling operation times for manual loading/dumping for the 9 cubic yard LHD are given in Table D.15, and for the remote loading/dumping in Table D.16.

Table D.15 Manual loading/dumping operation times for 9 cubic yard.

Activity	Unit	Minutes
Initial load bucket	time/bucket	1.49
Dump bucket	time/bucket	0.21
Total	time/bucket	1.70

Table D.16 Remote loading/dumping operation times for 9 cubic yard.

Activity	Unit	Minutes
Initial load bucket	time/bucket	1.75
Dump bucket	time/bucket	0.21
Switch to remote control	time/bucket	0.28
Switch from remote control	time/bucket	0.32
LHD remote travel in drawpoint	time/bucket	0.24
Total	time/bucket	2.80

Assuming the travel speed (full and empty) for the 9 cubic yard (TS9) varies with an average value of 379.5, minimum of 345, maximum of 475, and most likely value of 362.5 feet per minute, then the accompanying SLAM II equation obtained from UniFit II is:

TS9 = 345 + 130*BETA(1.48303,4.1052,<stream>) (D.41)

Assuming time 1.7 in Table D.15 (hereafter represented by the variable T16) varies with an average value of 1.7, minimum of 1.2, maximum of 2.5, and most likely value of 1.3 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

T16 = 1.2 + 1.3*BETA(1.05769,1.69231,<stream>) (D.42)

Thus, using equations D.35, D.41 and D.42 the time per bucket for the 9 cubic yard for manual loading (TPB9M) is:

TPB9M (in hours) = [T16 +
travelling distance in feet / TS9] / 60 (D.43)

Assuming time 2.8 in Table D.16 (hereafter represented by the variable T17) varies with an average value of 2.8, minimum of 2, maximum of 5, and most likely value of 2.2 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

T17 = 2 + 3*BETA(1.15556,3.17778,<stream>) (D.44)

Thus, using equations D.35, D.41 and D.44 the time per bucket for the 9 cubic yard for remote control loading (TPB9R) is:

TPB9R (in hours) = [T17 +

travelling distance in feet / TS9] / 60 (D.45)

Appendix D.4: Train Tramming

Tramming rates are based upon the following general assumptions:

- There are 4 chimneys (3 are used as orepasses and 1 is used as a rockpass),
- There are 6 cars in the ore train, and the amount of ore per car follows a triangular distribution with a minimum of 13.8, mode of 14.8 and maximum of 15.8 tons,
- There are 10 cars in the rock train, and the amount of rock per car follows a triangular distribution with a minimum of 4, mode of 4.5 and maximum of 5 tons, and
- There are 3 shifts per day for tramming, of which 2 are used for tramming ore and 1 is used for tramming rock; however, if there is no rock to tram on the third shift, then ore is trammed if available.

The productive hours per shift for tramming have been reduced from 8 to 6 hours based upon the non-productive time activities listed in Table D.17.

Table D.17 Non-productive time per shift for tramming.

Activity	Unit	Minutes
Lunch	time/shift	30.00
General preparation	time/shift	4.03
Inspect & service train	time/shift	10.15
Park train, end of shift	time/shift	2.70
Traffic interference	time/shift	6.06
Load/dump delays	time/shift	12.37
Electrical delays	time/shift	9.50
Supervision	time/shift	10.00
Personal	time/shift	10.00
Travel, etc.	time/shift	25.19
Total	time/shift	120.00

Ore Tramming Rates

Ore tramming operation times on a per trip basis are listed in Table D.18 and on a per car basis in Table D.19. The return tramming times (based on 330 feet per minute) for the chimneys are listed in Table D.20.

Table D.18 Per trip operation times.

Activity	Unit	Minutes
Remote onto motor	trip	0.36
Off motor, onto gangway	trip	0.66
Remote control on at orepass	trip	0.44
Pan down	trip	0.09
Pan up	trip	0.14
Remote off panel, off gangway	trip	0.26
Walk to motor	trip	0.52
Remote control to panel at dump, open gate	trip	0.74
Train into dump	trip	0.34
Reverse out of dump	trip	1.63
Remote off panel, close gate	trip	0.43
Walk to motor	trip	0.66
Total	trip	6.27

Table D.19 Per car operation times for ore tramming.

Activity	Unit	Minutes
Spot car under chute	car	0.22
Load car	car	0.65
Bar chute, large chunks	car	0.97
Bolts, etc. in muck	car	0.21
Trim load	car	0.11
Clean spill	car	0.31
Spot car at dump	car	0.17
Dump car	car	0.16
Total	car	2.80

Table D.20 Return tramming times.

Chimney	Unit	Minutes
#1	trip	13.58
#2	trip	14.06
#3	trip	18.97
#4	trip	19.58

In general, the tramming time per trip (TTPT) is:

TTPT = time for per trip operations + return tramming time + number of cars (time for per car operations) (D.46)

Assuming time 2.8 in Table D.19 (hereafter represented by the variable T18) varies with an average value of 2.8, minimum of 2.3, maximum of 4, and most likely value of 2.5 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

T18 = 2.3 + 1.7*BETA(1.27451,3.05882,<stream>) (D.47)

Summing appropriate values from Tables D.18 and D.20 results in a per trip time of 19.85 minutes for chimney #1. Assuming time 19.85 (hereafter represented by the variable T19) varies with an average value of 19.85, minimum of 17, maximum of 28, and most likely value of 18.75 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

T19 = 17 + 11*BETA(1.76653,5.05165,<stream>) (D.48)

Thus, using equations D.46 through 48, the tramming time for ore from chimney #1 (TOC1) is:

TOC1 (in hours) = (T19 + number of cars * T18) / 60 (D.49)

Summing appropriate values from Tables D.18 and D.20 results in a per trip time of 20.33 minutes for chimney #2. Assuming time 20.33 (hereafter represented by the variable T20) varies with an average value of 20.33, minimum of 17.5, maximum of 28, and most likely value of 19.25 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

T20 = 17.5 + 11*BETA(1.78662,5.15783,<stream>) (D.50)

Thus, using equations D.46, D47 and D.50, the tramming time for ore from chimney #2 (TOC2) is:

TOC2 (in hours) = (T20 + number of cars * T18) / 60 (D.51)

Summing appropriate values from Tables D.18 and D.20 results in a per trip time of 25.24 minutes for chimney #3. Assuming time 25.24 (hereafter represented by the variable T21) varies with an average value of 25.24, minimum of 22, maximum of 33.5, and most likely value of 24.25 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

T21 = 22 + 11.5*BETA(1.99209,5.07861,<stream>) (D.52)

Thus, using equations D.46, D47 and D.52, the tramming time for ore from chimney #3 (TOC3) is:

TOC3 (in hours) = (T21 + number of cars * T18) / 60 (D.53)

Summing appropriate values from Tables D.18 and D.20 results in a per trip time of 25.85 minutes for chimney #4. Assuming time 25.85 (hereafter represented by the variable T22) varies with an average value of 25.85, minimum of 23.5, maximum of 33.8, and most likely value of 24.75 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

T22 = 23.5 + 10.3*BETA(1.61783,5.47308,<stream>) (D.54)

Thus, using equations D.46, D47 and D.54, the tramming time for ore from chimney #4 (TOC4) is:

TOC4 (in hours) = (T22 + number of cars * T18) / 60 (D.55)

Rock Tramming Rates

Rock tramming operation times on a per trip basis are given in Table D.18 (the same as the ore tramming rates), and on a per car basis in Table D.21. The return tramming times for the chimneys are listed in Table D.20 (the same as the ore tramming rates).

Table D.21 Per car operation times for rock tramming.

Activity	Unit	Minutes
Spot car under chute	car	0.22
Load car	car	0.38
Bar chute, large chunks	car	0.12
Bolts, etc. in muck	car	0.21
Trim car	car	0.02
Clean spill	car	0.51
Spot car at dump	car	0.27
Dump car	car	0.18
Total	car	1.91

Assuming time 1.91 in Table D.21 (hereafter represented by the variable T23) varies with an average value of 1.91, minimum of 1.4, maximum of 3.5, and most likely value of 1.7 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

T23 = 1.4 +2.1*BETA(1.73469,5.40816,<stream>) (D.56)

Thus, using equations D.46, D.48 and D.56, the tramming time for rock from chimney #1 (TRC1) is:

TRC1 (in hours) = (T19 + number of cars * T23) / 60 (D.57)

Thus, using equations D.46, D.50 and D.56, the tramming time for rock from chimney #2 (TRC2) is:

TRC2 (in hours) = (T20 + number of cars * T23) / 60 (D.58)

Thus, using equations D.46, D.52 and D.56, the tramming time for rock from chimney #3 (TRC3) is:

$$TRC3 \text{ (in hours)} = (T21 + \text{number of cars} * T23) / 60 \quad (D.59)$$

Thus, using equations D.46, D.54 and D.56, the tramming time for rock from chimney #4 (TRC4) is:

$$TRC4 \text{ (in hours)} = (T22 + \text{number of cars} * T23) / 60 \quad (D.60)$$

Appendix D.4: Backfilling

Backfilling is based upon the following general assumptions:

- The tons of rockfill required to fill a cavity of a mined block is equal to 60 percent of the tons of ore removed,
- Rockfill is loaded onto the dump truck from a chute,
- The amount of rockfill per truckload follows a triangular distribution with a minimum of 11.5, mode of 12.5 and maximum of 13.5 tons,
- The mining block which is finished mucking and hauling first is backfilled first,
- Only 1 mining block is backfilled at a time, and
- There are 2 shifts per day for backfilling.

The productive hours per shift for backfilling have been reduced from 8 to 6 hours based upon the non-productive time activities listed in Table D.22.

Table D.22 Non-productive time per shift for backfilling.

Activity	Unit	Minutes
Lunch	time/shift	30.00
Travel	time/shift	30.00
Supervision	time/shift	5.00
Personal	time/shift	10.00
General Preparation	time/shift	10.00
Minor mechanical repairs	time/shift	10.00
Service truck	time/shift	25.00
Total	time/shift	120.00

Blast Delay and Check Vents

Blast delay and check vents times for backfilling are given in Table D.23.

Table D.23 Blast delay and check vents times.

Activity	Unit	Minutes
Blast Delay	time/blast	32.55
Ventilation Check	time/blast	22.45
Total	time/blast	55.00

Assuming time 55 in Table D.23 (hereafter represented by the variable T24) varies with an average value of 55, minimum of 45, maximum of 82.45, and most likely value of 50 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

$$T24 = 45 + 37.45*BETA(1.46595,4.02405,<stream>) \quad (D.61)$$

Thus, using equation D.61, the time for each blast delay and check vents for the truck (TBDCVT) is:

$$TBDCVT \text{ (in hours)} = T24/60 \quad (D.62)$$

Backfilling Rates

Loading and dumping truck operation times are listed in Table D.24.

Table D.24 Loading and dumping truck operation times.

Activity	Unit	Minutes
Spot truck under chute	time/load	0.80
Onto gangway & uncover	time/load	1.01
Gate pin out	time/load	0.09
Load 26 ton truck	time/load	1.23
Bar chute	time/load	2.08
Blast chute	time/load	0.54
Loading delays	time/load	0.48

Table D.24 (Continued.)

Activity	Unit	Minutes
Start & move truck ahead	time/load	1.18
Gate pin in	time/load	0.08
Cover gangway	time/load	0.51
Dump	time/load	0.40
Turn around & backup, etc.	time/load	1.60
Total	time/load	10.00

In general, the backfilling time per truckload (BTPT) is:

$$BTPT = \text{load and dump time} + \text{travel time} \quad (D.63)$$

Assuming time 10 in Table D.24 (hereafter represented by the variable T25) varies with an average value of 10, minimum of 9, maximum of 14.8, and most likely value of 9.8 minutes, then the accompanying SLAM II equation obtained from UniFit II is:

$$T25 = 9 + 5.8*BETA(3.62069,17.3793,<stream>) \quad (D.64)$$

Assuming the travel speed for the truck (TST) varies with an average value of 380, minimum of 350, maximum of 462.5, and most likely value of 372.5 feet per minute, then the accompanying SLAM II equation obtained from UniFit II is:

$$TST = 350 + 112.5*BETA(2.4,6.6,<stream>) \quad (D.65)$$

Using equations D.63 through D.65:

$$BTPT\text{ (in hours)} = (T25 + \text{travel distance in feet} / TST) / 60 \quad (D.66)$$

Appendix D.6: Equipment Uptime and Downtime

Appendix D.6 provides the assumptions and data for the equipment uptime and downtime equations. Historical records, along with expert opinions, were used to estimate various values which were entered into UniFit II to obtain corresponding SLAM II equations.

For the drills (CD 360), assuming a machine efficiency of 0.69, minimum possible downtime of 1 hour, average downtime of 6.75 hours, and minimum possible uptime (busy time) of 2 hours, the accompanying SLAM II equations obtained from UniFit II are:

Uptime = 2 + GAMA (18.606, 0.7, <stream>) (D.67)

Downtime = 1 + GAMA (4.10714, 1.4, <stream>) (D.68)

For the 6 cubic yard LHD (EJC 210), assuming a machine efficiency of 0.74, minimum possible downtime of 2 hours, average downtime of 9.82 hours, and minimum possible uptime (busy time) of 2 hours, the accompanying SLAM II equations obtained from UniFit II are:

Uptime = 2 + GAMA (37.0703, 0.7, <stream>) (D.69)

Downtime = 2 + GAMA (5.58571, 1.4, <stream>) (D.70)

For the 9 cubic yard LHD (JCI 900M), assuming a machine efficiency of 0.58, minimum possible downtime of 1 hour, average downtime of 8.82 hours, and minimum possible uptime (busy time) of 1 hour, the accompanying SLAM II equations obtained from UniFit II are:

Uptime = 1 + GAMA (15.9714, 0.7, <stream>) (D.71)

Downtime = 1 + GAMA (5.58571, 1.4, <stream>) (D.72)

For the truck (JCI 2604C), assuming a machine efficiency of 0.84, minimum possible downtime of 1 hour, average downtime of 6.55 hours, and minimum possible uptime (busy time) of 1 hour, the accompanying SLAM II equations obtained from UniFit II are:

$$\text{Uptime} = 1 + \text{GAMA}\ (47.6964,\ 0.7,\ \text{<stream>}) \tag{D.73}$$

$$\text{Downtime} = 1 + \text{GAMA}\ (3.96429,\ 1.4,\ \text{<stream>}) \tag{D.74}$$

For the train, assuming a machine efficiency of 0.95, minimum possible downtime of 2 hours, average downtime of 6.5 hours, and minimum possible uptime (busy time) of 4 hours, the accompanying SLAM II equations obtained from UniFit II are:

$$\text{Uptime} = 4 + \text{GAMA}\ (170.714,\ 0.7,\ \text{<stream>}) \tag{D.75}$$

$$\text{Downtime} = 2 + \text{GAMA}\ (3.21429,\ 1.4,\ \text{<stream>}) \tag{D.76}$$

Appendix D.7: Chimney Capacities

Appendix D.7 lists various assumptions and estimates for chimney capacities based upon the existing chimney dimensions when the model was built.

The chimneys and dump location I.D. numbers are given in Figure D.6.1. The capacities of chimney #1 (I.D. #10 and #11), chimney #2 (I.D. #15 and #16), chimney #3 (I.D. #20) and chimney #4 (I.D. #25 and #26) have been set to 213 tons (see file INTLC in Appendix A.6.2). This is based upon the estimated dimensions of 50 feet high, 8 feet by 8 feet across, and assuming 15 cubic feet per ton.

The capacities of chimney #2 (I.D. #17, #18 and #19) and chimney #4 (I.D. #27, #28 and #29) have been set to 426 tons. This is based upon the estimated dimensions of 100 feet high, 8 feet by 8 feet across, and assuming 15 cubic feet per ton.

The capacities of chimney #1 (I.D. #12, #13 and #14) and chimney #3 (I.D. #21, #22, #23, and #24) have been set to 0 since they did not exist at the time the model was created.

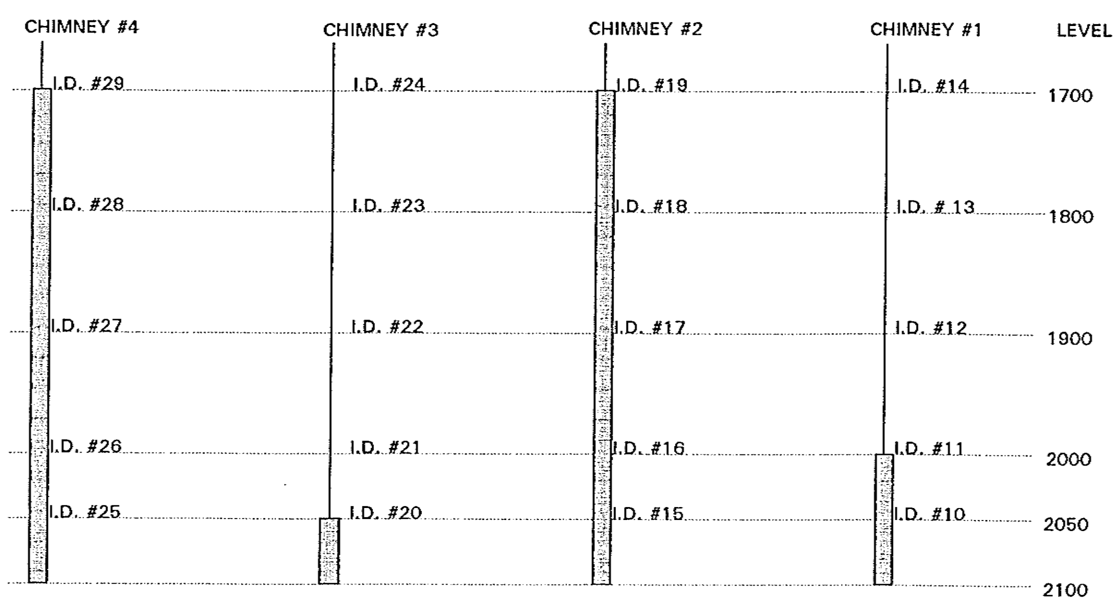


Figure D.6.1: Chimneys and dump location I.D. numbers

APPENDIX E: RECOMMENDED PROGRAM REFINEMENTS

During the final proofreading of this thesis, several program refinements were identified. These refinements have not been incorporated into the model; however, they are recorded here for future reference. The recommended program refinements are as follows:

1. The activity time calculated in USERF(7), or a portion thereof, should be included as part of the blasting operation time (i.e., BOT), which is used to calculate the blasting labour cost in the OTPUT subroutine. To include the entire activity time, USERF(7) should be changed as follows:

```
CREW = (60. + 72.2*BETA(1.44598,3.77402,1))/60.
BOT = BOT + 4.*CREW/3.
USERF = CREW
```

2. XX(7), the variable for the total tons of ore trammed, should be reinitialized in USERF(10), USERF(11), USERF(12) and USERF(13) immediately following S_MUCK = TNOW. The reason for this change is the development ore trammed during the first few days at the start of the simulation run are included in the calculation of the average daily tons of ore trammed for the run; however, the associated time is not included (see the STATS subroutine, i.e., DTOT(NNRUN) = XX(7)/((TNOW-S_MUCK)/24.).

3. The warm-up and cool-down stages for the drill/crew resources should be handled individually since one drill/crew may have a different starting and/or ending time than the other. For example, for a particular simulation run, Drill/Crew 1 may be used from

day 5 to day 50, whereas Drill/Crew 2 may be used from day 5 to day 60. Presently, the model would allocate the time from day 50 to day 60 for Drill/Crew 1 into stand by time, which in turn affects various Drill/Crew 1 availability ratios. In the real world, it is more likely that Drill/Crew 1 would be relocated to another mining block rather than on stand by until Drill/Crew 2 completed its mining blocks.

4. The current program includes allowances for the non-productive time of 2 hours per 8 hour repair shift within the random generation of the incremental RT value. However, note when the RT activity is preempted by the shifts segment, it returns with the remaining time for the RT activity. Although this does not affect the RT value used in the calculation of various availability ratios, it does, however, increase the overall simulated time and ultimately the PST and SBT values (which in turn affect the availability ratios). This oversight is illustrated in Example 1 (note the model is currently set-up in this manner), with the recommended approach given in Example 2.

   Example 1: Assume the random generation of the incremental RT value is 8 hours, which includes allowances for the non-productive time of 2 hours per 8 hour repair shift. Since the RT activity time within the model is equated to this value, the program performs similar to that shown in Figure F.1. Thus, the shift change preemption of the RT activity results in a total simulated time of 10 rather than 8 hours. This affects both PST and SBT as follows:

   PST = TNOW - START = 10 hrs. (It should be 8 hrs.)

   SBT = PST - DPT - IPT - RT = 10 - 0 - 0 - 8 = 2 hrs. (It should be 0 hrs.)

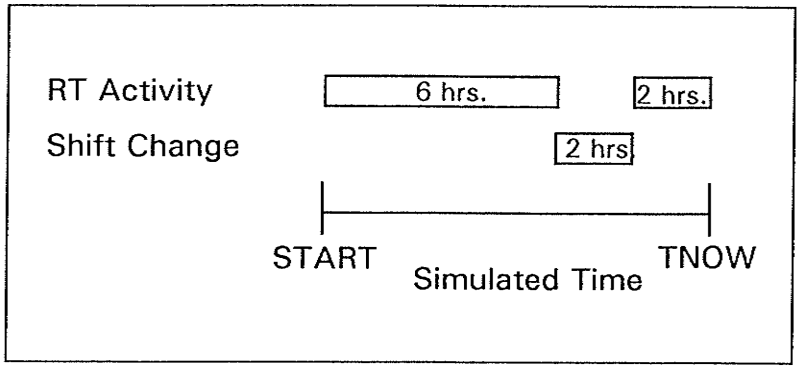


Figure F.1: Non-productive time included in the incremental RT value.

Example 2: Assume the random generation of the incremental RT value is 6 hours, which does not include the allowances for non-productive time of 2 hours per 8 hour repair shift. Since the RT activity time within the model is equated to this value, the program would perform similar to that shown in Figure F.2. Thus, the shift change preemption of the RT activity results in a total simulated time of 8 hours, with RT, PST and SBT calculated as shown below:

RT = cumulative RT + incremental RT + incremental RT/3.

= 0 + 6 + 6/3 = 8 hrs.

PST = TNOW - START = 8 hrs.

SBT = PST - DPT - IPT - RT = 8 - 0 - 0 - 8 = 0 hrs.

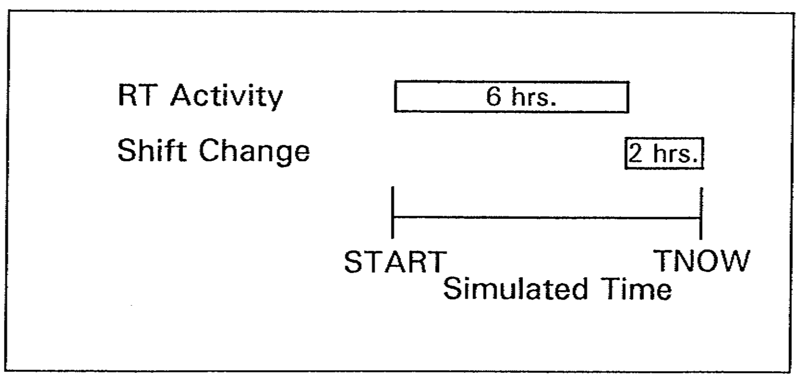


Figure F.2: Non-productive time not included in the incremental RT value.

Consequently, as demonstrated using Examples 1 and 2 above, the program should be modified so that RT is handled the same way as DPT and IPT. That is, the random generation of the incremental RT value should <u>not</u> include allowances for the non-productive time of 2 hours per 8 hour repair shift. In addition to changing the necessary equations, the RT value used to calculate the availability ratios should be determined as follows:

cumulative RT + incremental RT + incremental RT/3.,

which can be written as,

cumulative RT + 4.*incremental RT/3.