OPTIMIZATION OF AN AUTONOMOUS VEHICLE DISPATCH SYSTEM IN AN UNDERGROUND MINE

by

Paul Saayman

Submitted in partial fulfillment of the requirements for the degree
Master of Engineering (Electronic Engineering)
in the
Faculty of Engineering, the Built Environment and Information Technology
UNIVERSITY OF PRETORIA

April 2005

Optimization of an autonomous vehicle dispatch system in an underground mine
Paul Saayman
Prof. I.K. Craig
Dr. F.R. Camisani-Calzolari
Department of Electrical, Electronic and Computer Engineering
Master of Engineering (Electronic Engineering)

ABSTRACT

The mining industry can greatly benefit from automation. A great deal of work has been done on this subject and is still ongoing. With automation comes the possibility for optimization, because more information is available, and actions can be repeated with more accuracy. Many factors in an underground environment make mining automation a challenging prospect. These factors include the difficulty and cost of installing the needed infrastructure. The work described in this dissertation focuses on a mining setup where vehicles such as LHDs and trucks are used to collect and transport ore underground. Considerable progress has been made in automating underground vehicles, and successful tests have been done underground. The next obvious step is to find ways of using the increased data to optimize the decisions that are made with regards to the dispatching of the vehicles.

Possible solutions to the problem of optimizing the autonomous vehicle dispatch system in an underground mine are investigated. Possible optimization strategies are evaluated using a simulated environment. In the simulated environment a block cave mine is modelled, and the simulation setup is discussed in detail. The operation of a block cave mine as it is operated currently is simulated to obtain a benchmark for the evaluation of further results. The simulation results for the developed strategies are evaluated against specific criteria, and indicate definite improvements on current methods used in mines. Some important things that must be kept in mind for the physical implementation of the dispatching strategies, as well as mining automation in general, are also discussed.

Keywords: automation, simulation, block cave mine, vehicles, optimization.

Optimization of an autonomous vehicle dispatch system in an underground mine
Paul Saayman
Prof. I.K. Craig
Dr. F.R. Camisani-Calzolari
Departement Elektriese, Elektroniese en Rekenaar Ingenieurswese
Magister in Ingenieurswese (Elektroniese Ingenieurswese)

OPSOMMING

Outomatisasie hou groot voordele in vir die mynbou industrie. Baie werk is alreeds gedoen, en word steeds gedoen, in verband met hierdie onderwerp. Outomatisasie skep die geleentheid vir optimering, want meer informasie is beskikbaar, en aksies kan baie akuuraat herhaal word. Baie faktore in ‘n ondergrondse omgewing maak outomatisasie ‘n groot uitdaging. Hierdie faktore sluit onder meer in die koste en die probleme geässosieer met die installering van die nodige infrastruktuur. Die werk wat in die dissertasie beskryf word fokus op ‘n myn wat LHD’s en trokke gebruik om erts ondergronds op te laai en vervoer. Groot vooruitgang is al gemaak in terme van outomatisasie van ondergrondse voertuie, en suksesvolle toetse is al ondergronds gedoen. Die volgende logiese stap is om die verhoogde hoeveelheid data wat beskikbaar is te gebruik om meer optimale besluite te maak in verband met die bewegings van die voertuie.

Moontlike oplossings vir die optimeringsprobleem word hier ondersoek. Moontlike optimeringsstrategieë word geëvalueer met behulp van ‘n gesimuleerde omgewing. Die simulasie omgewing modelleer ‘n ‘*block cave*’ myn, en die simulasie opstelling word in detail beskryf. Die werking van ‘n ‘*block cave*’ myn soos dit huidig daaruit sien, word gesimuleer om ‘n verwysingspunt te kry vir verdere resultate. Die simulasie resultate vir die ontwikkelde strategieë word geëvalueer ten opsigte van spesifieke criteria, en dit dui op definitiewe verbeterings teenoor huidige metodes in myne. Daar word ook aandag geskenk aan ‘n paar belangrike dinge wat in gedagte gehou moet word as die strategieë fisies geïmplementeer word, en ook vir outomatisasie in myne oor die algemeen

Sleutelterme: outomatisasie, simulasie, ‘*block cave*’ myn, voertuie, optimering.

ACKNOWLEDGEMENTS

I would like to acknowledge a few people and institutions, without whom this dissertation would not have been possible. Firstly Professor I.K. Craig, as promoter, who guided me technically but also in general through the process of research. Dr. F.R. Camisani-Calzolari, as co-promoter, for crucial and appreciated inputs. This research was initiated by De Beers and I would like to extend a special thank-you for the financial contribution made by them. I would also like to specially thank the following people at De Beers for much needed technical inputs: Gunter Metzner, Giel Marais (formerly at De Beers) as well as Eduard Winter.

There are still some people that have to be thanked who had an indirect, but vital, impact on the success of this project. Firstly my parents and sisters for their support and love throughout my years of study. Mariëtte for your love and understanding during the last few months. Above all I have to thank my Father in heaven and give Him all the credit.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

CAM	Computer-aided mining
TRM	Telerobotic mining
LHD	Load Haul Dump
AGV	Automated Guided Vehicle
LAN	Local Area Network
GPS	Global Positioning System

CHAPTER 1: INTRODUCTION

1.1 MOTIVATION

Because of the hazardous environment of underground mines, a great deal of attention is given to automation of mining activities. The goal is to reduce or even eliminate the need for human presence in underground mining areas. With autonomous mining equipment and vehicles an operator can be in a control room safely above ground, and efficiency can actually be increased because more information of the state of the mine can be made available to the operator.

Conventional mining methods rely strongly on the experience and intuition of shift supervisors. Although an experienced supervisor can achieve acceptable productivity, his/her decisions cannot be regarded as optimal. This is because the supervisor only relies on limited information, and can predominantly act only in a reactive manner. In an automated system there must be some central point of control and/or information storage. This means that all relevant information is easily accessible and therefore optimal solutions are possible. The importance of information in a mining environment, and the distribution thereof, for increased productivity are discussed in Penswick and Gilliland [1] and Pukkila and Särkkä [2].

The emphasis in the mining environment has been on automation of vehicles and not the optimal dispatching of the vehicles. Therefore the need exists for methods to optimize mining dispatch systems to increase efficiency.

1.2 BACKGROUND

1.2.1 Mining automation

According to Pukkila et al. [2] productivity of mines can be increased significantly by applying new mining methods, advanced mining technology, automation and good organizational quality. In this section the focus will be on automation, and more specifically, the automation of underground mining vehicles. The basic elements needed for automating a mine, according to Pukkila et al. [2] and Scoble and Daneshmend [3], can be summarized as follows:

- Communication network (bi-directional) connecting all the parts of the automated system for real-time monitoring and control.
- Information and data acquisition system (local sensors on vehicles, central database etc.).
- Computerized database with open access for mine planning, control and maintenance systems.
- Machinery that is automated and connected to the communication network. The machinery must have onboard monitoring, control and positioning systems.
- Centralized or decentralized computation of optimal signals and/or control commands.

Scoble et al. [3] give two possibilities for mining automation: computer-aided mining (CAM) and telerobotic mining (TRM). CAM is defined as a setup where human operators are still in control of the vehicles, either directly or remotely, and information is supplied to the operator by appropriate sensors. This enables the operator to be at a safe distance and still be in control. TRM is defined as a centralized control strategy where the system is totally controlled by computer-based controller with no (or very little) human intervention.

According to Penswick et al. [1] the past decade has seen widespread implementation of real-time computerized management systems in open pit mines. The focus of the research described in this report, however, is on underground mines, but the principles of vehicle dispatching stays the same for both environments. Underground mines generally lag behind in terms of automation, partly because of the difficulties associated with underground communication. The above mentioned reference also indicates that there exists a gap in the information hierarchy, so that optimizing an entire operation is not yet a reality.

A considerable amount of progress has been made in automating underground vehicles, and full scale, fully automated (and semi-automated with remote operator) vehicles have been successfully tested underground (Scheding, Dissanayake, Nebot and Durrant-Whyte [4] and Roberts, Duff and Corke [5]). Experiments using scale models of sensor-guided LHDs[1] have also been done, and are described in Steele, Ganesh and Kleve [6] and Hwang, Farmer and Hart [7]. The navigation of vehicles underground is obviously a crucial part of mining automation. The goal is to have reliable navigation without the addition of considerable extra infrastructure. Two possible solutions to this problem are proposed in Banta, Nutter and Xia [8] and Shaffer, Stentz, Whittaker and Fitzpatrick [9]. In Banta et al. [8] a strategy is proposed that use different modes of navigation, which include: wall following, collision avoidance and homing. The automated vehicle uses neural networks to acquire intelligence in terms of the environment, and therefore need very little infrastructure apart from what is included on the vehicle itself. In Shaffer et al. [9] a laser range sensor is used to detect features within the mine (line segments and corners) to place itself within a map of the mine layout.

1.2.2 Vehicle dispatch systems

Scheduling and dispatching of vehicles is a problem that has widespread applications. This problem has enjoyed considerable attention in the manufacturing and material handling environment, and the field of automated road traffic. Different approaches are used for scheduling of automated guided vehicles (AGVs). A few examples can be found in literature. Evers and Koppers [10] describe a control strategy for AGVs at a container terminal. With this approach the environment of the AGVs is divided into different areas and nodes. The movement of vehicles at a node is controlled using

---

[1] LHD: Load haul dump. A low profile vehicle with front end loader used to transport ore in mining applications

semaphores. A semaphore indicates if a node is occupied or not, and in this way a node is 'controlled' by a specific vehicle until it leaves the node. In this way the control is more localized and less communication is needed in the system. In Bing [11] a strategy is proposed which determines certain priorities in terms of the current status of the system, and then control is done based on these priorities and the available resources. Kim and Hwang [12] propose a dispatching strategy that is based on an evolutionary process. By using such an approach the control strategy can adapt to a changing environment. This strategy is applied to manufacturing environment and some results are given. A program for dispatching AGVs in a flexible manufacturing environment is developed in Smith and Sarin [13]. This program uses inputs like priorities, travel speeds etc. to make dispatching decisions.

The basic principles of vehicle scheduling and dispatching stays the same for most applications. The goals of a dispatching system are basically to control a certain number of vehicles to complete tasks while specific objectives are kept in mind. These objectives can be in terms of economic or time factors or both. In some cases these different objectives are in conflict, and then a compromise must be found. Therefore, an important step in designing a dispatch system is to determine the relative importance of the different parameters that have to be taken into account.

1.2.3 Vehicle dispatch systems in mining

As already mentioned in the previous section the general principles of vehicle dispatching can also be implemented in the mining environment. There are however certain unique factors that have to be taken into account. An underground mine consists of tunnels, which means that the possible routes for vehicles are fixed. This is the same as in manufacturing plants where the workstations are fixed and the AGVs follow fixed routes between them. The difference in an underground mine is, however, that new tunnels are developed while older tunnels are exhausted. This means that the active area in a mine is constantly shifting. The layout of the environment of the vehicles must therefore be updated regularly according to the development of the mine. Another important factor to take into consideration in a mining environment is road conditions. The vehicles that are used in mining applications are large and heavy. This means that the roads have to take considerable strain and regular maintenance is needed. The dispatch system must therefore take into account any delays that may result from undesirable road conditions and maintenance.

One of the most troubling factors in an underground mine is the establishment of a reliable and flexible communication network. A physical hardwire communication infrastructure is not an acceptable solution because of the changing nature of the mine layout, the use of explosives and large mobile equipment. The use of wireless LAN technology is a possible solution that uses a flexible infrastructure. In the underground environment technologies such as GPS (Global Positioning System) are not suitable, because communication with satellites is inhibited. The positioning of vehicles must therefore be done using local sensors (on the vehicles or in the tunnels). To find reliable and suitable sensors is another challenge for the design of an underground vehicle dispatch system.

In any underground mine there exists some bottlenecks. An example of a bottleneck is the crusher in a block cave mining setup. The crusher reduces the size of rocks extracted from the mine for further processing. The crusher has a limited capacity, and this means that trucks that are scheduled to unload into the crusher must wait until the crusher level is low enough. Even if the dispatching system operates very effectively and ore is moved very quickly to the crusher, the trucks will still have to wait. By developing a dispatch system in isolation and not considering factors that is outside of its control (e.g. crusher), very little, or no improvement in productivity may be the result. The dispatch system must therefore be designed with the big picture in mind, and some compromises must be made to ensure better overall efficiency.

1.3 PROBLEM STATEMENT

The problem statement consists of the following parts:

- Develop a simulation to accurately simulate the movement and other activities of autonomous vehicles in an underground mine.
- Implement a dispatch algorithm using the simulation, to represent current dispatching techniques used in underground mines. Use this as baseline to compare future results to.
- Determine evaluation criteria that can be used to measure the performance of dispatch strategies.
- Develop and implement dispatch strategies with the evaluation criteria in mind.
- Evaluate the improvement (if any) obtained by implementing the dispatch strategies, as compared to the baseline.

1.4 CONTRIBUTION

Algorithms are developed specifically for the mining environment to optimize the dispatching of autonomous underground vehicles. The simulation results will indicate the efficiency of the algorithms and possibly lead to the physical implementation in underground mines. The specific contribution is therefore the implementation of existing techniques in an environment where the use of these techniques is very limited or non-existent. An added benefit of this work is that it might help to motivate the powers that be to invest even more in research and implementation of dispatching systems.

1.5 DISSERTATION APPROACH

The steps followed during the dissertation are shown in the flowchart in figure 1-1. The different components of the flowchart are discussed below.

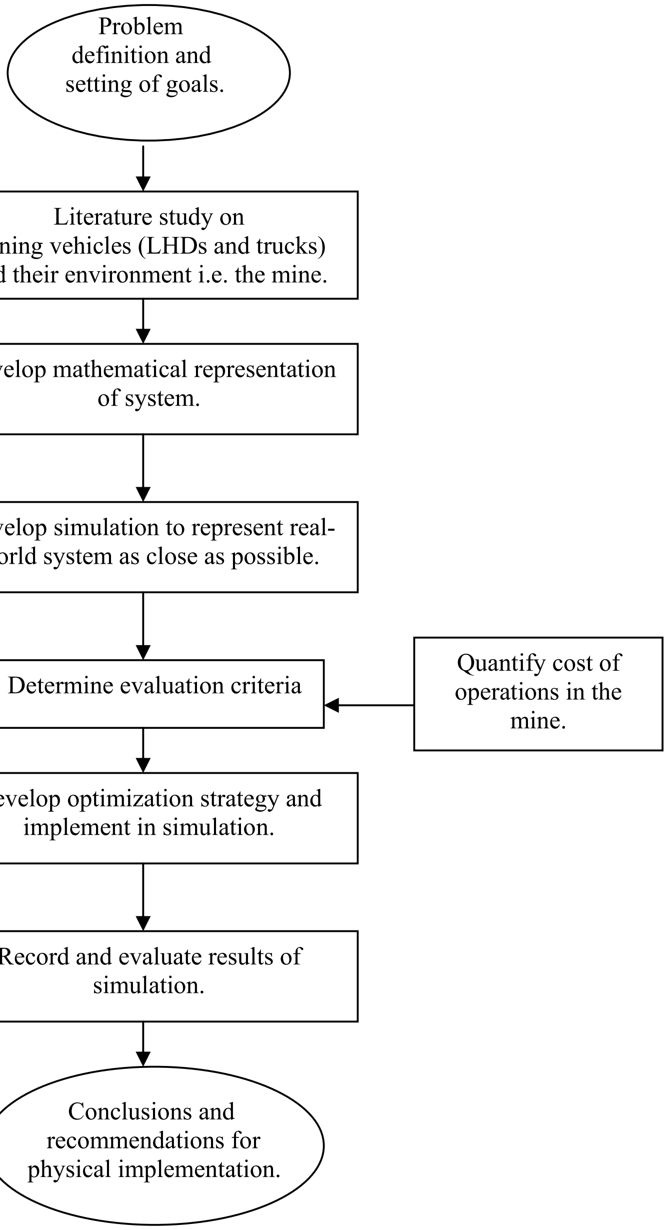


Figure 1-1.
The figure shows the steps that were followed during the dissertation.

The first stage of addressing any problem is to define the problem as clear as possible. While defining the problem the goals that must be reached are also implicitly defined. The problem statement given in section 1.3 is the result of this first step, and it can be seen that the problem statement contains the specific goals of the research as well.

A literature study is done to gain a better understanding of the underground mining environment, and more specifically mining vehicles. The simulation that will be discussed later in the dissertation simulates the environment found in a block cave mine, therefore the focus of the literature study is also on vehicles used in this specific environment. The literature study is restricted to the factors influencing the vehicles directly. These include, among others, the layout of the mine and road conditions.

The system is characterized in mathematical terms. The mathematical representation of the system is used as a basis for the design of control and optimization strategies.

The next step is to develop a simulated environment that models the real world underground system as close as possible. The simulation should model the layout in a block cave mine accurately, as well as the factors that influence the vehicles in the mine.

To determine the success of any strategy that is developed the results must be evaluated according to some relevant criteria. The cost of operations in the mine must therefore be quantified to obtain these criteria. This includes the output of the mine in terms of the amount of ore extracted. This is the main criterion used to evaluate the performance of dispatching algorithms in this research. The work described here is limited to the movement of the vehicles in the mine, although many other factors, for example the further processing of the ore, also contribute to the productivity of the mine. The goal of any dispatching algorithm should then be to minimize some appropriate cost function that contains the factors that it can have an effect on.

Dispatching algorithms are developed with the evaluation criteria, mentioned above, in mind. The specific goal of each strategy is to make the movements of the vehicles in the simulated mine more effective, so that productivity can be improved if compared to current dispatching methods. The developed strategies are implemented and tested using the simulated environment. The results of each strategy are recorded and compared to current methods in mines as well as the results obtained with other strategies. From these results it can be decided if a strategy is worthwhile to implement, and recommendations can be made on how to physically implement it.

1.6 ORGANISATION

The dissertation is organized as follows:

- Chapter 2 gives an overview of the processes found in a block cave mine. The emphasis is on the movements of the vehicles in such a mine. The system of vehicles is represented in terms of the different states of the vehicles as well as the conditions for transitions between states. This is done to give the reader a better understanding of the system, and to make the operation of the simulation clearer. The development of the simulation is also discussed. This includes the layout of the simulated environment, some fundamental assumptions that were made, and important parameters that the simulation uses, and the results it produces.
- In chapter 3 the different dispatching strategies that were developed are discussed. The specific objectives of each strategy are given. The connection between the functioning of each strategy and its specific objective is also discussed. As an introduction to the chapter the different factors that must be taken into account for the simulation are discussed. These include the controllable and non-controllable parameters, as well as collision avoidance.
- Chapter 4 is a collection of the results obtained from the simulations of the different dispatching strategies.
- In chapter 5 the results given in chapter 4 are discussed. The results obtained with the different dispatching strategies are compared with each other, and possible reasons are given for the performance of each strategy. The factors that will have to be taken into account for the physical implementation of these dispatching strategies are also discussed.
- Chapter 6 contains the conclusion and recommendations for future work.
- The final part of the dissertation is made up by the references and appendices. The appendices contain the data structures used in the simulation, and a detailed layout of the simulation environment.

CHAPTER 2: PROCESS OVERVIEW

2.1 INTRODUCTION

In this chapter an overview of the mining process in a block cave mine is presented, with the focus on the movements of the vehicles in the mine. These vehicles are LHDs and trucks, and are discussed in the following section. The layout of the mine, which obviously determines the possible routes of the vehicles, is also discussed.

2.2 VEHICLES

2.2.1 Load Haul Dump (LHD)

A typical LHD can be seen in figure 2-1. LHDs are used to load and transport ore from the areas where it is fragmented by blasting (or caving in the case of a block cave mine) to transfer points, from where it is either processed or transported further. The tunnels in which LHDs are used are very confined and therefore the LHD has a relatively low profile. An LHD consists of two parts joined by an articulation joint. The wheels cannot steer and are fixed perpendicular to the axles. Turning of the vehicle is achieved by actuators that change the articulation angle of the middle joint as shown in figure 2-2. This method of steering is also ideal for the confined and narrow nature of the working environment of an LHD. An illustration of this type of steering can be seen in figure 2-3, where it is compared to the steering of a normal vehicle.

The problem of mathematically modelling the motion (including slip) of an LHD, is addressed in Yavin [14], and a comprehensive mathematical model is proposed. The same problem is also addressed in Dragt, Craig and Camisani-Calzolari [15]. In this paper two vehicle models are used, firstly a no-slip model based on kinematical geometry and secondly a model which takes into account slip by means of an

Extended Kalman Filter. Simulations of a reactive navigation scheme, using both of these models, are described in Dragt et al. [15].

A lot of work has already been done on the automation of LHDs, and successful test have been done with full-size vehicles underground. A problem that still remains a challenge is to optimize the automated scooping up of ore. This is because the ore consists of randomly sized rocks and these rocks are randomly arranged at collection points. Solutions for this problem are investigated by Banks, Bennamoun and Corke [16], and Hemami [17]. Figure 2-4 shows the range of movement of the scoop that is situated on the front of the vehicle and is used for loading of ore.

The availability of LHDs is obviously very important to obtain the best possible production in a mine. The reliability of an LHD as a whole depends on the reliability of its separate subsystems (transmission, hydraulics, etc.). There are many factors that influence the reliability and maintainability of the subsystems in an LHDs. These factors are discussed in Samanta, Sarkar and Mukherjee [18] and recommendations are made to improve reliability.

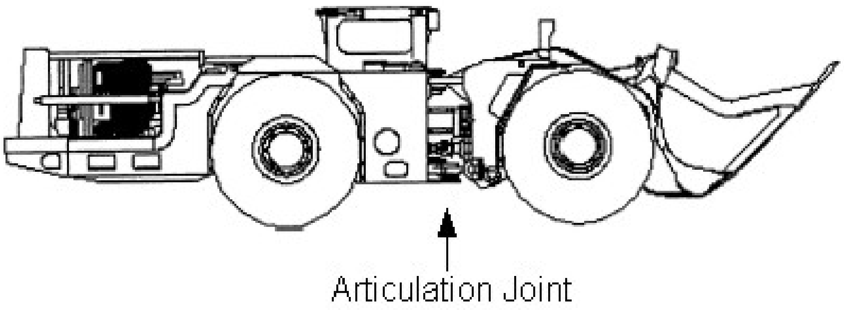


Figure 2-1.
The figure shows an illustration of a typical LHD. The articulation joint used for steering is clearly indicated. (Taken from Dragt et al. [15])

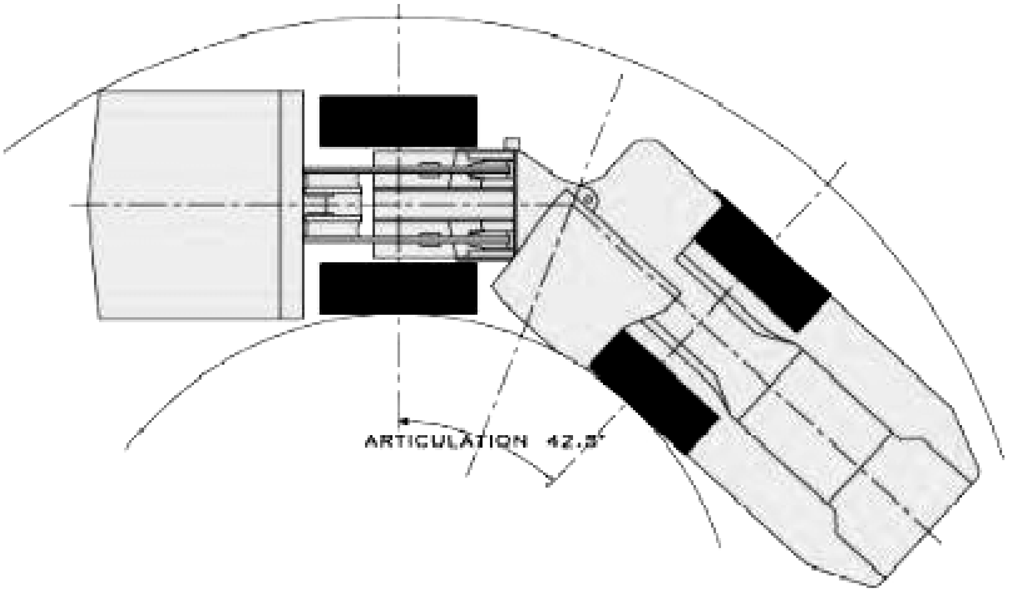
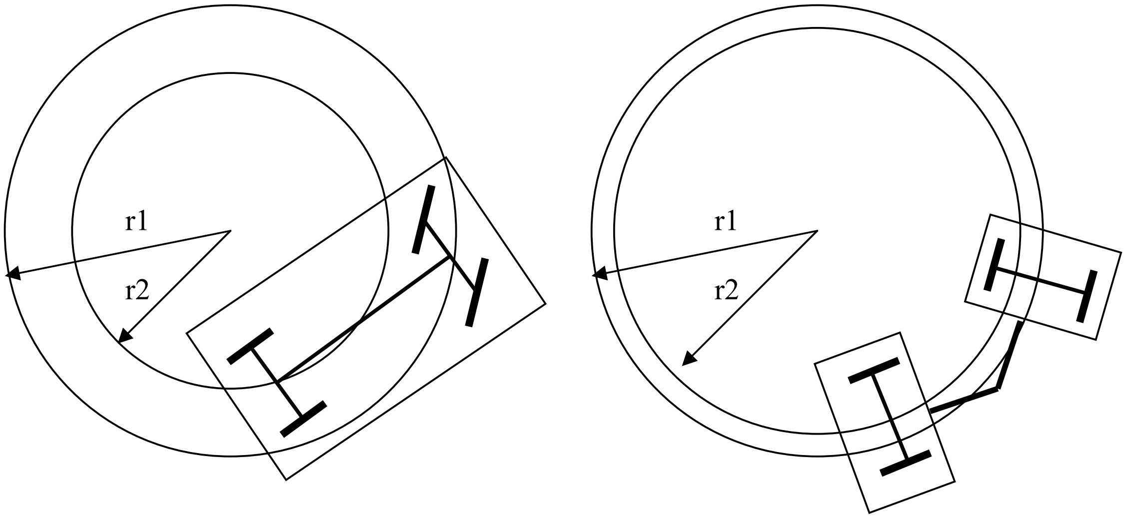


Figure 2-2.
The steering mechanism of an LHD. The articulation angle is changed by actuators to achieve the desired turning angle. (Taken from www.catelphinstone.com)



Car-like vehicle **Articulated vehicle**

Figure 2-3.
The figure shows the advantage of the method of steering of an articulated vehicle compared to that of a conventional vehicle. The difference in turning radii of the front and rear axle is much smaller for an articulated vehicle, which makes it ideal for maneuvering in confined environments.

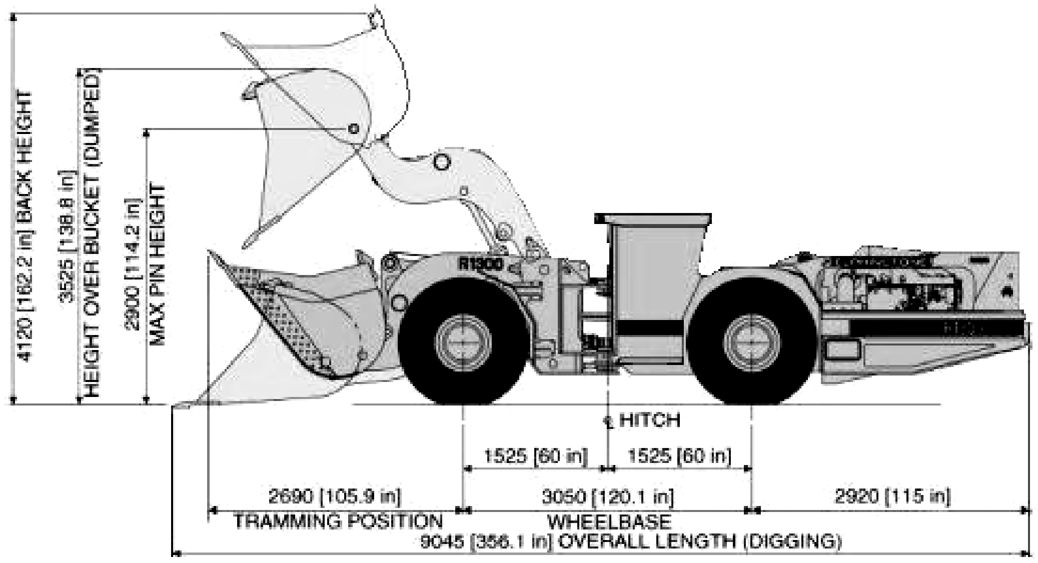


Figure 2-4.
The figure shows the range of movement possible with the front loading scoop of an LHD. (Taken from www.catelphinstone.com)

2.2.2 Truck

The trucks used in a block cave mine are typical mining trucks. Because the tunnels in which they move are not that confined they are a lot more bulky than LHDs. A full truck load can typically be in the region of 50 tons and consist of a few LHD loads. In figure 2-5 an example of a mining truck is shown.

Figure 2-5.
A typical mining truck. (Taken from http://artzia.com/Gallery/Photos/ Photo.net/)

2.3 PROCESS DESCRIPTION

The mining method implemented in a block cave mine will not be discussed in detail. The focus will instead be on the portion of the process where the vehicles play a role. In a block cave mine tunnels are developed underground. Above these tunnels, funnel shaped openings are created at regular intervals. The areas underneath these funnels are called the drawpoints. The rock above the drawpoints is blasted and the loose rocks are captured by the funnels and guided down to the drawpoints. The blasting forms a cave above the drawpoint level, and at a certain stage blasting is ceased and the rock caves in under its own weight. This caving is controlled by planning the initial blasting very carefully. A simple representation of a block cave mine can be seen in figure 2-6.

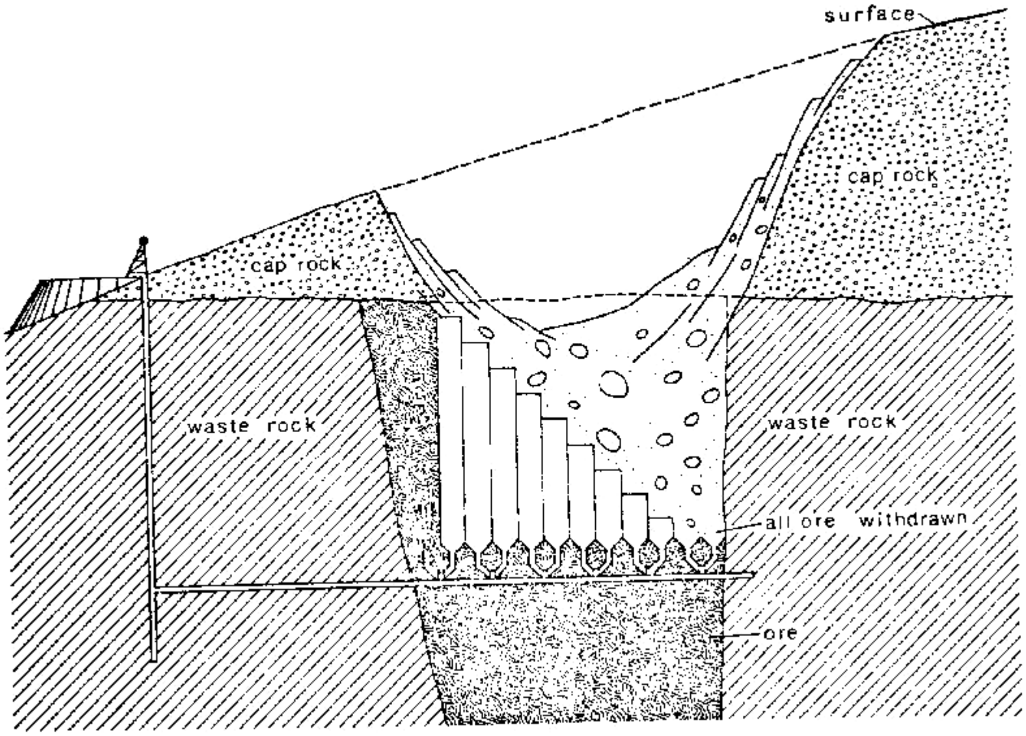


Figure 2-6.
The figure shows a cross cut through a typical block cave mine. The funnels can clearly be seen, and also the exhaustion of older drawpoints. (Taken from US Forest Service General Technical Report INT-35: "Anatomy of a mine from prospect to production")

The ore that collects at the drawpoints is loaded by LHDs, and transported to the transfer points. At the transfer points the ore is transferred to trucks. The trucks then transport the ore to the crusher where it is crushed and hauled to an extraction plant for further processing. The layout of the tunnels in a typical block cave mine can be seen in figure 2-7. The movements of the vehicles as well as the overall working of the mine will become clearer when the simulation is discussed.

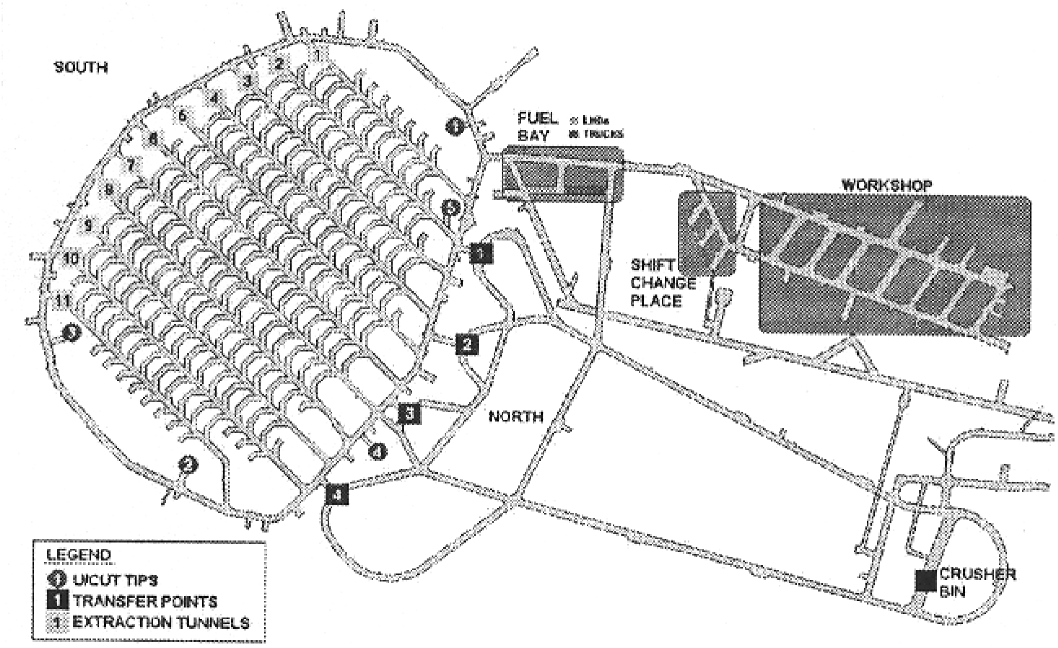


Figure 2-7.
The figure shows the layout of a typical block cave mine. The drawpoints are located above the network of tunnels on the left, and the LHDs move in these tunnels. The tunnel loop on the right connects the transfer points and the crusher, and the trucks move in this loop. (Figure courtesy of De Beers)

2.4 SYSTEM MODELING

The system of vehicles in an underground mine can be described as a hybrid system. A hybrid system is a dynamical system composed of discrete and continuous states. A more complete and generic definition of hybrid systems can be found in Tomlin, Lygeros and Sastry [19]. In the context of autonomous vehicles it can be said that the vehicles can be in one of a number of discrete states. In each of these modes the behaviour of the autonomous vehicle is governed by continuous dynamics.

The possible states of each vehicle are defined by the application for which it is used. For the case of vehicles in an underground mine the possible states will depend on the vehicles' locations, their destinations, whether they are busy loading or off-loading, whether servicing or repairs are needed *etc*. The movement of each vehicle is then determined by certain continuous dynamics in each mode.

In Bemporad, Ferrari-Trecate and Morari [20] a simple temperature control system is used as an example of a hybrid system. The graphical representation of the temperature control system is called a hybrid automaton. The hybrid automatons representing the dispatch system of the vehicles in an underground mine, can be seen in figure 2-8 and figure 2-9. In Morari and Bemporad [21] a unified model form for a wide range of discrete-time hybrid systems is developed. This model is called the Mixed Logical Dynamical (MLD) form. The system of vehicles in an underground mine can be represented in the MLD form. It is however not done here because it would not add any more value than the automatons, in terms of understanding the system. More detail regarding MLD systems as well as the control of these systems can be found in Colmenares, Cristea and Villegas [22] and Borrelli, Bemporad and Morari [23].

It is possible to develop a more thorough mathematical description for the system of mining vehicles. This could then be used as a basis for using existing optimization techniques to obtain dispatching strategies. The future goal of this work is however to develop dispatching strategies that can be implemented in a physical mine. In an underground mine the controllable parameters are very limited with regards to vehicle dispatching, e.g. the possible routes are limited, the possible destinations are limited and the actions needed at the different destinations are fixed. Therefore to follow a complicated mathematical approach would have academic value, but would not offer great advantages in terms of producing realistic strategies that can be implemented. It was therefore decided to follow a more heuristic approach to obtain strategies that can be implemented without too much difficulty. Past successes using a proven control systems approach are not denied, but it would have been excessive for this specific application.

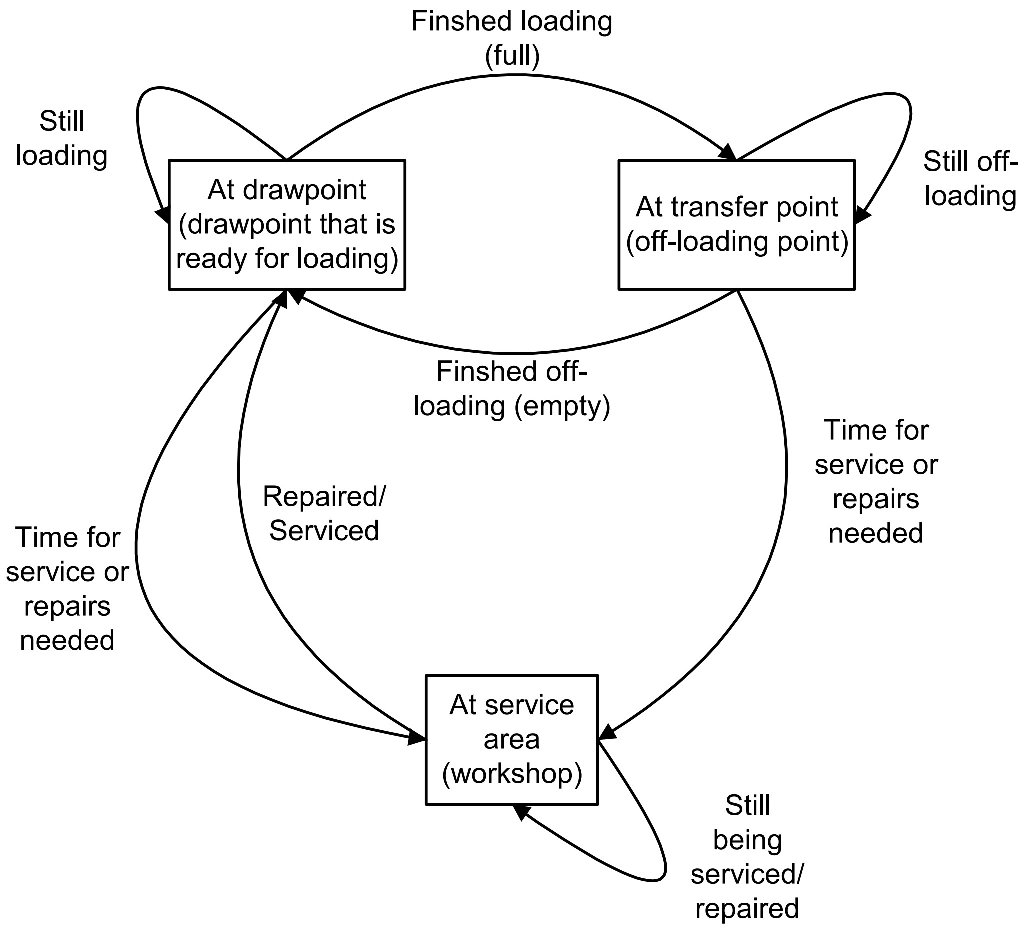


Figure 2-8.
The figure shows a simple representation of the different states of each LHD in the underground mine environment. The blocks indicate the states, and it can be seen that the states are defined in terms of the position of the LHD. The arrows indicate the transitions between states, and the requirements for each transition are given beside the arrow. The transitions between states as well as the activities within each state will be under control of the dispatching strategy. The dispatching algorithm will decide which drawpoint or transfer point will be the destination (this is not explicitly shown in the figure).

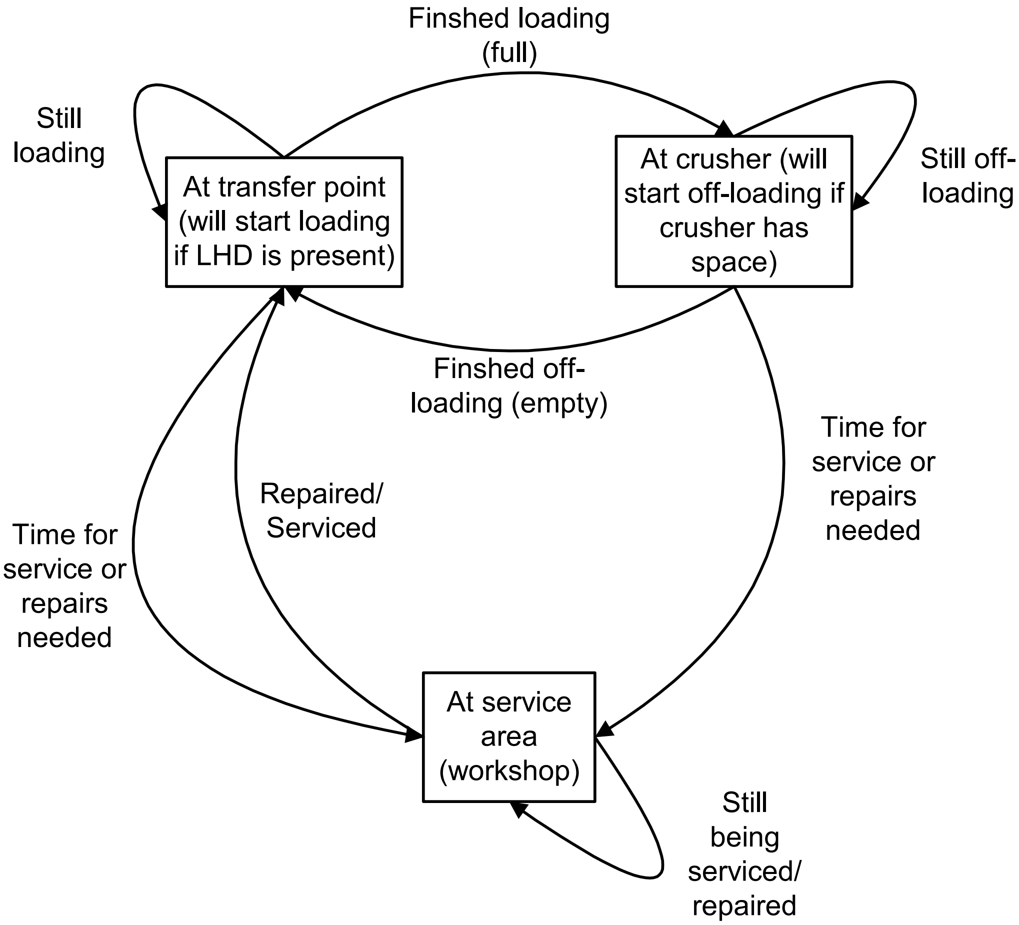


Figure 2-9.
The figure shows a simple representation of the different states of each truck in the underground mine environment. The blocks indicate the states, and it can be seen that the states are defined in terms of the position of the truck. The arrows indicate the transitions between states, and the requirements for each transition are given beside the arrow. The transitions between states as well as the activities within each state will be under control of the dispatching strategy. The dispatching algorithm will decide which transfer point will be the destination (this is not explicitly shown in the figure).

2.5 SIMULATION MODEL

The ultimate goal of the simulation is to model the real-life system of vehicles in an underground mine as close as possible. In a real-life system however, a lot of unpredictable factors play a role in the overall performance of the system. These factors are very hard, and sometimes impossible, to model accurately. Other factors can be incorporated without too much difficulty, but it may make the simulation too complicated for the specific application. It is therefore important to determine what the desired outcome of the simulation is, and to develop the simplest model that is adequate. The factors that were not taken into account in the simulation are discussed in more detail in a later chapter.

The mine layout used in the simulation is shown graphically in figure 2-10. In a real mine, tunnels get exhausted while new tunnels are developed. This means that the active area in the mine will shift with time. The simulation model therefore models this active area, although it is not explicitly included, a scenario where the number of tunnels exhausted is equal to the number of new tunnels developed is included implicitly. The active area that is simulated consists of 7 tunnels with 15 drawpoints on both sides of each tunnel, as shown in figure 2-10. The drawpoints between two adjacent tunnels can be accessed from both sides.

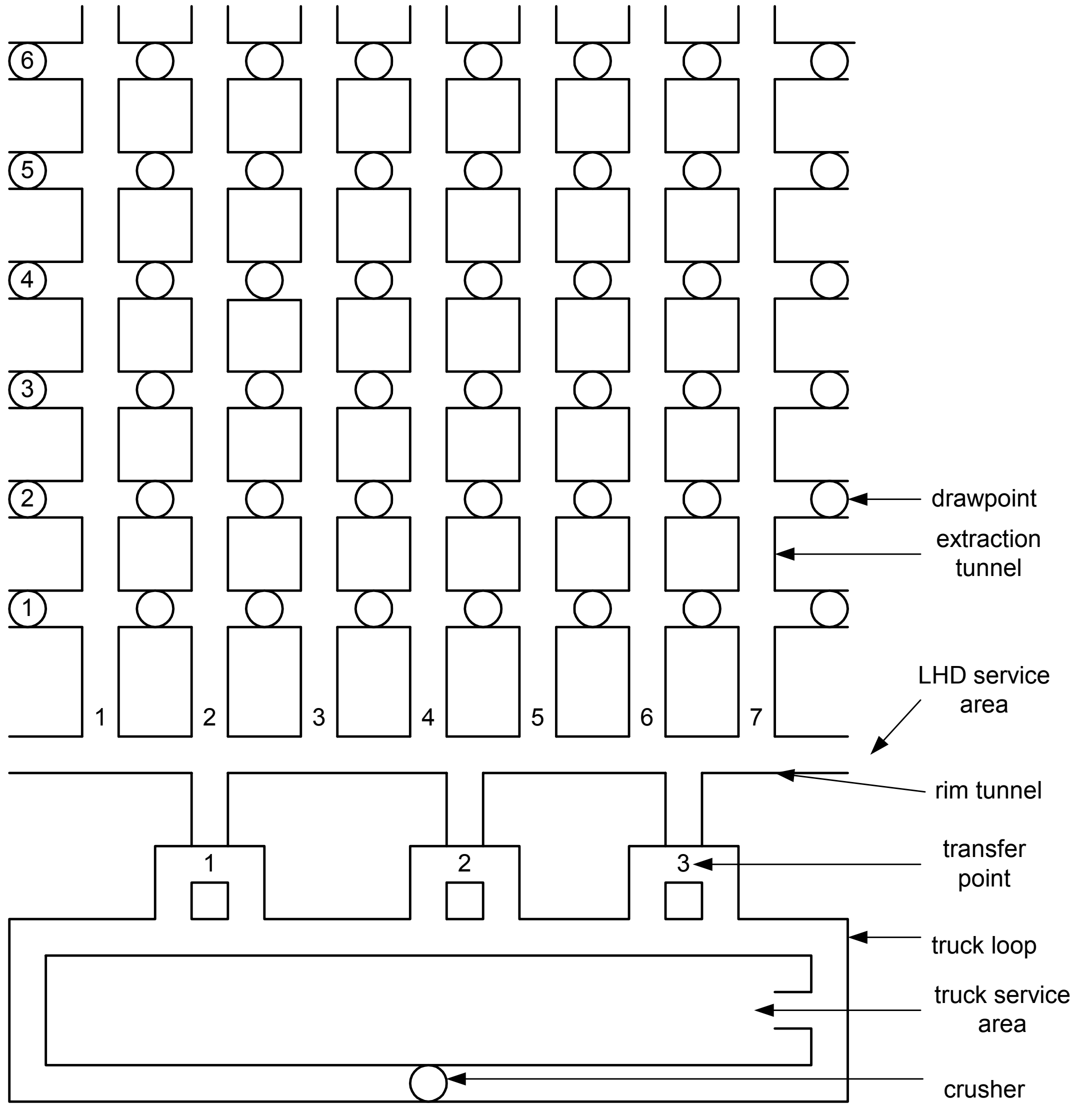


Figure 2-10.
The simulation environment used for the dissertation. The tunnel outlines can be seen and the circles represent drawpoints. The circle at the bottom represents the crusher, and the loop at the bottom is the tunnel connecting the transfer points with the crusher (this is not drawn according to scale). The position of the LHDs and trucks are not shown in the figure, but during simulation they are indicated with coloured dots, and updated each sampling period (e.g. 1 second). Only a portion of the total layout is shown. In the complete simulated mine layout more drawpoints are present.

To obtain relevant results the simulation must be a close representation of the real mine. Therefore factors like drawpoint hang-ups, vehicle breakdowns, and vehicle servicing are included in the simulation. A drawpoint hang-up occurs when large rocks get stuck in the funnels (see figure 2-6) above the drawpoint. This prevents any other rocks to collect at the drawpoint for loading. If a drawpoint is hung up and not the adjacent drawpoints, the difference in ore level above it and adjacent drawpoints can become large. In a block cave mine it is very important to keep the ore level above the funnels as even as possible. Therefore it is important to clear hang-ups in time, so as not to cause too much damage to the infrastructure of the mine. The hang-up is cleared by implementing secondary drilling and blasting. This is, however, not always successful and might have to be repeated a number of times. Secondary blasting, in a specific tunnel, is done only when 55% or more of the drawpoints in that tunnel are inactive due to hang-ups, or when a drawpoint has been inactive for more than 3 days. During secondary blasting the tunnel is closed for production until all the drawpoints are active again. There is also a waiting period, of about 60 minutes, after blasting before production vehicles can re-enter the specific tunnel. The simulation incorporates historical data obtained from the industrial partner (De Beers) to generate realistic random hang-ups, based on the amount of ore loaded from each drawpoint. Secondary blasting is modelled only as a time delay. This delay is also determined based on historical data, including the probability of the number of blasting iterations needed to successfully clear each hang-up. Different types of hang-ups can occur. The type of hang-up depends on the height at which the rock or rocks get stuck. The different types of hang-ups cause delays of different lengths of time.

According to the industrial partner the service intervals for both an LHD and a truck should be after 125 hours of production time. The vehicles also have a mean time between failures, which was incorporated into the simulation. If a vehicle breaks down, or it reaches its service time, it moves to the service area at the farthest right end of the rim tunnel (figure 2-7) where it is fixed or serviced. This is also just modelled as a time delay based on recorded data from a physical mine. When the vehicle is repaired or serviced both the service and breakdown counters are reset, and the vehicle returns to production.

Nominal values are used for the amount of ore that an LHD can load at one time, as well as the load that a truck can transport. These values are based on manufacturer's figures as well as practical experience of the industrial partner. The specific values used in the simulation are 9 tons for an LHD load, and 45 tons (5 LHD loads) for a truckload.

In real-life the crusher has a limited capacity and the simulation takes this into account. The level of ore in the crusher is reduced according to a certain crushing rate. If a loaded truck arrives at the crusher the level of the crusher is first checked, and if the crusher has enough capacity the truck dumps its load into the crusher. Otherwise the truck waits for the crusher level to recede sufficiently. The crusher is switched off if it stays empty for more than 12 minutes. When a truck arrives with a load at the crusher, and it is switched off, it must be switched on again. Because starting the crusher is time consuming, and energy inefficient, the number of crusher start-ups per hour must be kept as low as possible.

According to the theory of block cave mining (according to De Beers) and experience gained in physical mines, there is a limit on the amount of ore that can be drawn daily. A general estimate for this limit is 200mm of solid rock per day. By taking the area above each drawpoint, the limit per drawpoint per day can be obtained in tons. These limits are also taken into account by the simulation. The value of this limit is nominal and may vary somewhat for varying rock densities.

The simulation assumes that all the vehicles are fully automated, and a central control system exists which can send commands to all the vehicles. Alternatively it is assumed that the driver of each vehicle executes the central commands perfectly. It is also assumed that all the information regarding the mine status (drawpoint status, crusher status, layout etc.) and vehicle status (speed, acceleration, load etc.) are always known. In a simulation environment this does not really make a difference because the sensing and communication infrastructure is inherently present. In a physical mine, however, an extensive infrastructure has to be installed to make the assumptions mentioned above true. The principles of dispatching are the same for a manually operated and an automated mine. In both cases some central controller must make decisions in terms of the destinations of the vehicles. In the case of a manually operated mine this controller will typically be a person telling the drivers of the vehicles where to go. In an automated mine this controller will be a computer that sends commands to the remote computers located in each vehicle. These remote computers will then generate the corresponding control signals for the physical systems in the vehicle. Both of these scenarios could benefit from a system that uses the current state of the mine and vehicles to generate optimal dispatching commands. The method of relaying these commands to the vehicles and the physical low-level control of the vehicles will obviously be different for a manual and automated system. A requirement for both these systems is that an extensive sensing and communication infrastructure must be in place to make all the relevant information readily available. Finally the conclusion of the discussion in this last paragraph is that the results of this research are not only applicable to automated mines, but manually operated mines can benefit from it as well.

2.6 IMPLEMENTATION

All simulation work discussed here is done in Matlab[2]. Some of the essential parts of the Matlab source code are included here. A flow diagram (figure 2-11) is used to explain the overall operation of the source code. The flow diagram is followed by the different pieces of source code that implements the functions in the flow diagram. The source code used to implement the different dispatching strategies will be discussed in the next chapter. The constant parameter values, and data structures used to describe the state of each entity (vehicle, tunnel, drawpoint, and crusher) are included in an appendix to make the source code more understandable.

---

[2] Matlab: A technical computing language. The MathWorks, Inc.

2.6.1 Flow diagram

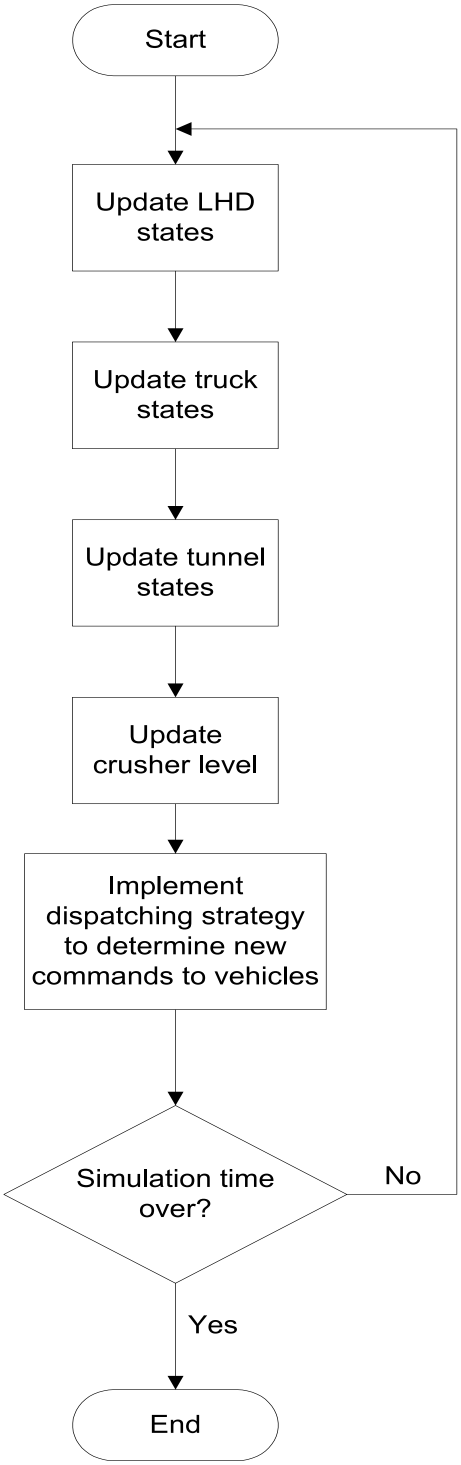


Figure 2-11.
The flow diagram describing the processes executed in each cycle of the main loop of the simulation program.

2.6.2 Source code

For debugging purposes a graphical representation of the mine layout and vehicle positions was created. This slows the simulation down considerably and therefore it was removed as soon as debugging was finished, to speed up the collection of results. The functions related to the graphical representation are not included in the flow diagram and the source code implementing these functions are also not shown. The source code implementing the main loop of the simulation looks as follows (text preceded by '%' is ignored by Matlab and acts only as comments):

```
%start simulation

while (counter<sim_time)

%Determine the new position of each LHD

LHD_state = lhd_move(LHD_state);

%Determine the new position of each truck

truck_state = truck_move(LHD_state,truck_state);

   x_prev=x_pos;   %save lhd previous position
   y_prev=y_pos;

   xt_prev=xt_pos;  %save truck previous position
   yt_prev=yt_pos;

%Determine if tunnel should be closed for secondary breaking

for i=1:num_tunnels
   if tunnel_state(i,1)==1 %then closed for secondary breaking
      tunnel_state = handle_hangup(tunnel_state,i);
   elseif tunnel_state(i,1)==0
      inactive=0;
      for j=((i-1)*num_load+1):((i-1)*num_load+2*num_load)
         if (drawpoint_state(j,5)==1)&(drawpoint_state(j,1)==0)
            inactive=inactive+1;
         end
      end
      if (inactive/tunnel_state(i,2))>=0.55   %more than 55% drawpoints inactive
         wait_lhd=0;
         for j=1:num_lhd
            if (LHD_state(j,1)==i)&(LHD_state(j,2)>0)
               wait_lhd=wait_lhd+1;
            end
         end
         if wait_lhd==0
            tunnel_state(i,1)=1;    %then tunnel is closed for secondary breaking
            for j=((i-1)*num_load+1):((i-1)*num_load+2*num_load)
                   if (drawpoint_state(j,5)==1)
                   tunnel_state(i,3)=tunnel_state(i,3)+drawpoint_state(j,10);
```

```
               end
            end
         end
      end
   end
end


%Update crusher level

if crusher_power==1
   crusher_level=crusher_level-crush_rate/3600;

   if crusher_level<=30
      crusher_level=30;
      crusher_run_empty=crusher_run_empty-1;
      if crusher_run_empty==0
         %crusher_level=30;   %minimum crusher level
         %crusher_power=0;    %switch off crusher if minimum level is reached
         crusher_limit_counter=crusher_limit_counter+1;
         crusher_run_empty=1; %12*60/time_int;
      end
   else
      crusher_run_empty=1; %12*60/time_int;
   end
end


%==================
%DISPATCH ALGORITHM
%==================

%LHD

for i=1:num_lhd
   if LHD_state(i,29)==1       %waiting for new command
      LHD_state(i,20)=0;  %not in idle any more
      LHD_state(i,29)=0;  %will get new command now
      LHD_state(i,12)=0;  %finished loading/off-loading
      LHD_state(i,:)=lhd_dispatch_3(LHD_state,drawpoint_state,tunnel_state,i);
   end
end

%truck

for i=1:num_trucks
   if truck_state(i,20)==1     %waiting for new command
      truck_state(i,9)=0; %not in idle any more
      truck_state(i,20)=0;    %will get new command now
      truck_state(i,8)=0; %finished loading or off-loading
      truck_state(i,:)=truck_dispatch_2(truck_state,LHD_state,i);
   end
end

%determine shift and day
shift_counter=round(counter/(8*3600));
day_counter=round(shift_counter/3);
week_counter=round(day_counter/5);

if mod(counter,(24*3600))==0
```

```
    for i=1:(num_load*(num_tunnels+1))
        if drawpoint_state(i,3)>drawpoint_state(i,4)
            drawpoint_state(i,1)=1;
        end
    end
    drawpoint_state(:,3)=0;
end

counter=counter+1;

end    %while counter<sim_time
```

The source code used to implement the different updating functions (LHD and truck positions, crusher level and tunnel state) is not included, because it is intuitive and rather long.

CHAPTER 3: DISPATCHING STRATEGIES

3.1 INTRODUCTION

The system of vehicles in a mine can be described as a control system with certain inputs and some desired outputs. On a high level the input to the system is the request to achieve maximum productivity, and the output is then the productivity. The factors used to determine the productivity includes the amount of ore loaded etc. but these factors will depend on the priorities of the specific mine. On a lower level the inputs to the system are the current status of each drawpoint, the status of each vehicle, and the status of the crusher. The status of a drawpoint contains flags indicating if the drawpoint is active or not, if a hang-up occurred, and if the allowed quota of ore for the day have been loaded from it. Other parameter values included in the drawpoint status are the level of the drawpoint (in terms of the amount of ore loaded from it), and the type of hang-up that has occurred, if any. The status of a vehicle includes the following parameter values: current position, speed, acceleration, destination, and load. It also includes flags indicating whether the vehicle is full/empty, whether it is busy loading/off-loading/in transit, and if it needs a service or repairs. The crusher status indicates the crusher level and a flag indicating if it has been running empty for a certain period of time, as well as a flag indicating if it is on/off. A complete breakdown of each status structure is given in Appendix B.

3.1.1 Control parameters

A control system has controllable parameters, and also fixed parameters. The controllable parameters are those things that can be manipulated in order to get more desirable outputs. The fixed parameters are things that are inherent to the system and cannot be changed, and include the minimum and maximum limits of the controllable parameters for example. These parameters usually limit the maximum efficiency of control that can be achieved. The controllable parameters in the case of a vehicle dispatching system are vehicle speed and direction of movement as well as the number of vehicles in the system. The fixed parameters are the maximum speed, maximum acceleration, service intervals, fuel consumption, maximum load size of vehicles, capacity of the crusher, and the maximum loading quotas per day. The occurrence of hang-ups and vehicle breakdowns are random, of which the probability distribution can be found, but it is also not controllable. An underground mine consists of tunnels. This means that the possible routes that the vehicles can follow are fixed. Specific loading and off-loading points exist in a mine, which effectively limits the possible destinations of the vehicles. Although old tunnels get exhausted and new tunnels are developed, the layout is constant for considerable periods at a time. The layout of the mine can therefore also be classified as a fixed parameter in the system, because the simulated production time is short enough (one week).

3.1.2 Collision avoidance

The assumption was made that the tunnels in the simulation are too narrow for vehicles to pass each other. Vehicles moving in the same direction with not too much of a difference in velocity can be in the same tunnel at the same time, but in other scenarios some form of collision avoidance is needed. It is also important to note that only one LHD is allowed in a specific extraction tunnel (figure 2-10) at one time. This is because the back-end of an LHD protrudes into the extraction tunnel while it is loading, and therefore another LHD cannot pass. This means that the majority of the collision avoidance is done in the rim tunnel (figure 2-10) and where an extraction tunnel intersects the rim tunnel. The scenarios shown in figure 3-1 illustrate the collision avoidance principles implemented in the simulation. The collision avoidance is implemented identically for all the dispatching strategies discussed in section 3.3. It will not be explicitly mentioned again, but nevertheless it is a crucial part of most of the vehicle movements. An example of a collision avoidance strategy for an AGV system can be found in Ho [24]. This strategy divides the environment into non-overlapping zones, and only one AGV may be inside a zone at a time. Collisions between vehicles are therefore avoided. A further feature of the strategy proposed by Ho [24] is that the zones are dynamically adjusted according to the needs of the system.

Closely related to collision avoidance is obstacle avoidance. In the simulation the assumption was also made that the tunnels are clear of any stationary obstacles. In a real mine it might happen that stationary obstacles exist (temporarily or permanent). A good example is a vehicle that breaks down. However it was assumed that when a vehicle breaks down it could move to the service area immediately in some or other way. Therefore obstacle avoidance is not taken into account, but only collision avoidance between moving vehicles. In Xu, Brussel, Nuttin and Moreas [25] the problem of obstacle avoidance in underground navigation is addressed, and a possible solution is introduced. This solution includes defining possible ways out and obstacle definition using a 'Polar Object Chart', which is essentially a simplified representation of the surrounding environment. It must, however, be said that all the vehicles must avoid the tunnel walls, which can be regarded as obstacles.

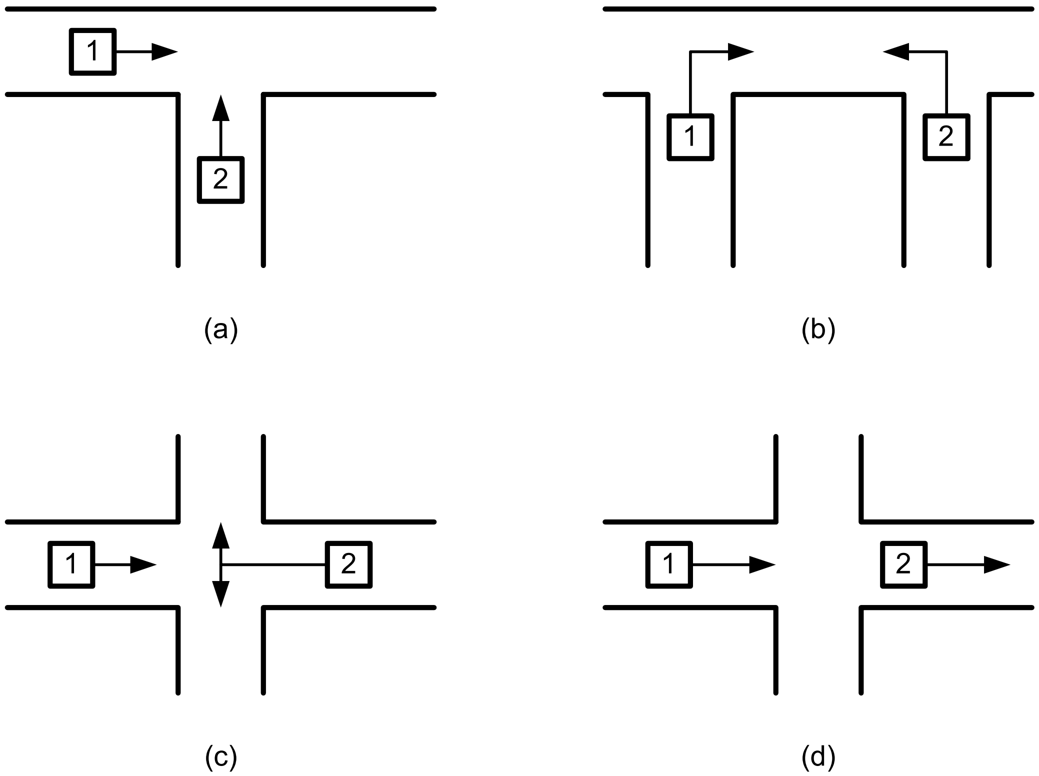


Figure 3-1.
The different possible scenarios where collision avoidance has to be implemented. The figure indicates the position of the vehicles with regards to an intersection, as well as the direction of movement of each vehicle. This figure is intended to illustrate the concept of collision avoidance and not to accurately reflect the layout of the mine.

The scenarios shown in figure 3-1 will now be discussed shortly:

a) Vehicle 1 is moving in the rim tunnel and vehicle 2 wants to enter the rim tunnel from an extraction tunnel. To avoid a collision vehicle 2 must wait until vehicle 1 has passed the intersection before it can enter the rim tunnel.

b) Vehicle 1 and 2 want to enter the rim tunnel at the same time, and their intended directions will cause a collision. To avoid a collision one of the vehicles, in this case vehicle 2 (the one with the higher number), must wait until vehicle 1 has passed the intersection where vehicle 2 is situated.

c) Vehicle 1 and 2 are in the rim tunnel and moving towards each other. The positions of the vehicles ensure that a collision is inevitable if neither of them takes evasive action. To avoid a collision the following possibilities exist: If both of them can still stop before they pass the destination intersection (intersection where the vehicle must turn) of the other vehicle, then vehicle 2 stops and waits until vehicle 1 has turned out of the rim tunnel. If one of them has already passed the destination intersection of the other vehicle, then it will just carry on while the other vehicle stops before the intersection until the rim tunnel is clear. If both are past the other vehicle’s respective destination intersection, then one of the vehicles must reverse so that the other can reach the intersection, and wait until the rim tunnel is clear.

d) Both vehicles are moving in the rim tunnel in the same direction. To avoid a collision some safety distance must be defined that has to be present between the vehicles. This is achieved by just controlling the speed of the following vehicle (vehicle 1 in figure 3-1).

Although it is not explicitly stated the collision avoidance strategies used for the LHDs, are similarly applied to the trucks. Possibilities of collisions exist at the entrances and exits to the transfer areas, and also at the crusher. The trucks always move in the same direction in the tunnel connecting the crusher and transfer areas. Therefore the collision avoidance will mainly focus on controlling the speed of a following truck to keep a safe distance from the truck in front of it.

3.2 PERFORMANCE EVALUATION CRITERIA

The major objective of most businesses is to maximize profit. A mine is no exception. Therefore one of the most important evaluation criteria for the dispatch system simulation is the production achieved, as there is usually a strong link between maximum production and profit. This is measured in terms of the total tons of ore drawn, or equivalently the total tons dumped into the crusher. The main objective of the different dispatching strategies is therefore to maximize the total tons produced.

Another important objective specific to a block cave mine, is to keep the ore level above the drawpoints as even as possible. To achieve this, the visits to active drawpoints must be scheduled so that the ore level is evenly drawn, and hang-ups must be cleared as soon as possible. The effectiveness of the different simulated strategies is therefore also evaluated according to the variation in the ore level over the active mining area. A lower variation increases the possibility for maximum future production.

The last objective is to have as few crusher shutdowns as possible. As already mentioned a crusher shutdown occurs when the crusher runs empty for a predetermined time (section 2.4). To achieve this, the arrival of the trucks at the crusher must be spaced evenly with long enough intervals so that the trucks don’t have to wait, but also short enough so that the crusher does not run empty. Therefore the last objective of the simulated dispatching strategies is to minimize the number of crusher shutdowns.

Other factors exist that are more difficult to quantify, but are also influenced, directly or indirectly, by the dispatching strategy. An example is the cost of developing and maintaining tunnels and drawpoints (clearing hang-ups etc.), that are influenced by how even the ore level is drawn. For these factors to be incorporated a much longer simulation period, and a more complex model, are required. The possible advantage of including these factors does not justify the extra development time and effort, in terms of the goals of this research. Therefore this will not be used as an evaluation criterion. Operating costs of the vehicles can be quantified easily, and it includes the cost of fuel, services, and repairs. These costs can however not be controlled by a dispatching strategy, and will also not be used as an evaluation criterion.

3.3 IMPLEMENTATION

The different strategies that were developed and simulated are discussed below. The Matlab source code used to implement each strategy is also given. In reality the dispatching system in a block cave mine consists of two systems interacting with each other. The one system consists of a number of LHDs and the other of a number of trucks. Therefore some of the strategies focus only on the movements of the LHDs and others on the movements of the trucks. Combinations of these strategies are also implemented to determine if a more optimal solution can be found for the overall system. It is possible to find very effective strategies for the LHD and truck systems separately, but because there is interaction between the systems, the overall best solution might not simply be a combination of two effective strategies. To find the most effective dispatching strategy some trade-offs will have to be made.

If one looks at the objectives discussed in the previous section, it is clear that good performance with regard to a specific objective might cause degraded performance with regard to another objective. Trade-offs will therefore also be needed to obtain the solution that will meet all the objectives sufficiently well.

To make the discussions that follow clearer the reader is referred to figure 2-10 in the previous chapter. Section 2.6 might also be helpful to clarify where the source code, given in this chapter, fits into the simulation as a whole.

3.3.1 Strategy 1 (Base case)

The first strategy is based on current dispatching strategies. This will therefore be used as the base case with which the other strategies are compared. The results obtained with this strategy will give a good estimate of what can currently be achieved in a block cave mine, in terms of productivity. To make the results even more relevant, realistic initial conditions are needed. To obtain these initial conditions strategy 1 is run for the equivalent of one week of production from a zero state. One week of production consists of 5 days, with 3 shifts of 8 hours each per day. Zero state indicates an even ore level, with no ore loaded from any drawpoint, and an empty crusher. These initial conditions are then used as the starting point for all the following simulations (including strategy 1). The counters used to indicate when vehicle breakdowns, vehicle services, and drawpoint hang-ups should occur, are reinitialised before a simulation is executed.

A simple representation of the LHD dispatching according to strategy 1 is given in figure 3-2. Figure 3-3 gives a simple representation of the truck movements according to strategy 1. The strategy works as follows:

Each LHD is assigned to a specific tunnel. Each tunnel has only one LHD assigned to it. The LHD sequentially visits each active drawpoint in its assigned tunnel. This is illustrated, for a reduced number of drawpoints, in figure 3-2. The first drawpoint that is visited as well as the last drawpoint is indicated. After all the drawpoints have been visited the sequence starts again from the first drawpoint. After each visit to a drawpoint the LHD visits a transfer point to unload before going to the next drawpoint. An LHD unloads only at a specific transfer point. This transfer point is the one closest to the tunnel that the LHD is assigned to. If another LHD is busy unloading, all other LHDs that want to go to the same transfer point must wait (in the extraction tunnel just outside the rim tunnel) until the transfer area is clear.

If an LHD has to be serviced, repaired or refuelled it moves to the service area. If it is loaded it first unloads before moving to the service area. If a tunnel becomes inactive because of hang-ups, the LHD assigned to that tunnel also moves to the service area (after unloading) until the tunnel is active again. This procedure stays the same for all the different strategies.

The trucks move in one direction around the loop connecting the transfer points and the crusher. In the simulation this direction is taken as clockwise. If a truck is empty it moves to the nearest open transfer point. An open transfer point is one where no other truck is present. At the transfer point the truck waits to receive LHD loads. The truck stays at the same transfer point until it has received enough LHD loads to fill it up (a truckload consists of 5 LHD loads), and then moves to the crusher. If all the transfer points are occupied the remaining trucks must wait just before the entry to the first transfer point until a transfer point becomes available. If a truck is due for a service or needs repairs, it moves to the service area after unloading any ore it might be carrying.

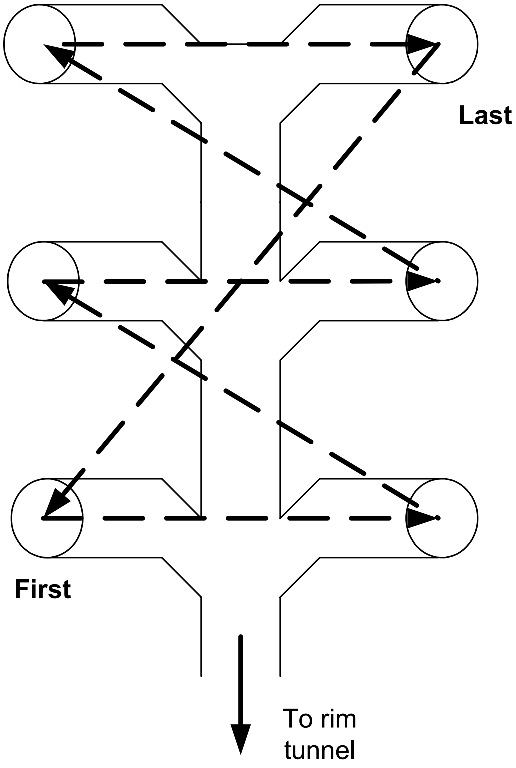


Figure 3-2.
The figure gives a simple representation of the sequence in which the drawpoints are visited by an LHD. The figure only shows 3 drawpoints on each side of the tunnel. The number of drawpoints used in the simulation is 15 on each side of a tunnel. The arrows only indicate the sequence, and not the physical movement of the LHD. Between successive drawpoints the LHD visits a transfer point to unload.

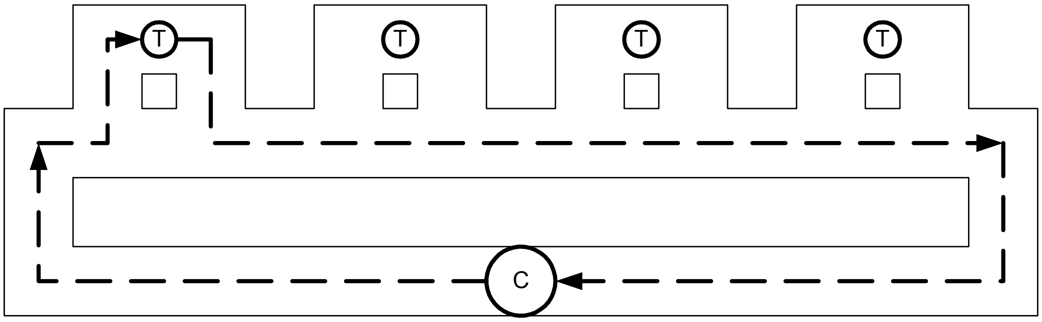


Figure 3-3.
The figure shows the direction of movement of the trucks around the loop connecting the transfer points and the crusher. The path actually indicates the movement of a truck that receives it load from transfer point 1 and then unloads in the crusher. The crusher is indicated with a 'C' and each transfer point is indicated with a 'T'..

3.3.1.1 Source code: LHD dispatching

```
function new_lhd_state = lhd_dispatch_1(LHD_state,drawpoint_state,tunnel_state,i)

global num_lhd num_tunnels previous_drawpoint num_load;

%for i=1:num_lhd

  if ((LHD_state(i,2)==-1)&(LHD_state(i,10)==0))|...    %then at transfer point
                                                                  (finished off-loading)
      (LHD_state(i,5)==(num_tunnels+1))
    LHD_state(i,7)=-1;
    if LHD_state(i,2)==-1
      LHD_state(i,17)=0;
    else
      LHD_state(i,17)=1;
    End

  %decide which drawpoint to go to

    if tunnel_state(i,1)==0 %then tunnel is active
      LHD_state(i,5)=i;
      if LHD_state(i,15)==1   %was on left side
        if (drawpoint_state(i*num_load+previous_drawpoint(i,1),1)==1)   %then next
drawpoint is active
          LHD_state(i,6)=previous_drawpoint(i,1);
          LHD_state(i,16)=0;      %go to right side
        else
          LHD_state(i,29)=1;
          LHD_state(i,15)=0;
          %LHD_state(i,6)=previous_drawpoint(i,1)+1;
          %LHD_state(i,16)=1;      %go to left side
        end
      else    %was on right side
        if (previous_drawpoint(i,1)+1)<=num_load
```

```
                if drawpoint_state((i-1)*num_load…
                            +previous_drawpoint(i,1)+1,1) == 1
%then next drawpoint is active
                  LHD_state(i,6)=previous_drawpoint(i,1)+1;
                  LHD_state(i,16)=1;
                else
                  LHD_state(i,29)=1;
                  LHD_state(i,15)=1;
                  previous_drawpoint(i,1)=previous_drawpoint(i,1)+1;
                end
              elseif drawpoint_state((i-1)*num_load+1,1)==1
                  LHD_state(i,6)=1;
                  LHD_state(i,16)=1;      %go to left side
              else
                LHD_state(i,29)=1;
                LHD_state(i,15)=1;
                previous_drawpoint(i,1)=1;
              end
            end
        else    %then tunnel not active
            LHD_state(i,5)=num_tunnels+1;   %wait for secondary breaking
            LHD_state(i,6)=1;
        end

    elseif (LHD_state(i,2)>0)&(LHD_state(i,10)==1)   %then at drawpoint
                                                        (finished loading)
        %previous_drawpoint(i,1)=LHD_state(i,2);
        LHD_state(i,6)=-1;
        LHD_state(i,17)=0;

    %decide which  transfer point to go to

        if i<=3 %tunnel 1 to 3
            LHD_state(i,5)=2;
            LHD_state(i,7)=1;
        elseif i<=5 %tunnel 4 or 5
            LHD_state(i,5)=5;
            LHD_state(i,7)=2;
        elseif i<=7 %tunnel 6 or 7
            LHD_state(i,5)=7;
            LHD_state(i,7)=3;
        elseif i<=10 %tunnel 8 to 10
            LHD_state(i,5)=10;
            LHD_state(i,7)=4;
        end
    end

new_lhd_state=LHD_state(i,:);
```

3.3.1.2 Source code: Truck dispatching

```
function new_truck_state = truck_dispatch_1(truck_state,LHD_state,i)

global crusher_level num_lhd num_trucks waiting;

%for i=1:num_trucks
   if (truck_state(i,4)==0)    %then at crusher (finished off-loading) or at
                                        transfer point but not full yet
   %decide which transfer point to go to
    done=0;
   one=0;
   two=0;
   three=0;
   four=0;
      if (truck_state(i,5)==0)  %then at crusher
         for j=1:num_trucks
            if (i~=j)
               %check if truck is at transfer point or on its way there
               if ((truck_state(j,2)==1)&(truck_state(j,3)==1))|…
                            ((truck_state(j,3)==1)&(truck_state(j,2)~=1)&…
                            (truck_state(j,1)>=truck_state(i,1)))|...
                            ((truck_state(j,3)==1)&(truck_state(j,2)~=1)&…
                            (truck_state(j,1)==truck_state(i,1))&(j<i))
                  one=one+1;
               elseif ((truck_state(j,2)==2)&(truck_state(j,3)==2))|…
                            ((truck_state(j,3)==2)&(truck_state(j,2)~=2)&…
                            (truck_state(j,1)>=truck_state(i,1)))|...
                            ((truck_state(j,3)==1)&(truck_state(j,2)~=1)&…
                            (truck_state(j,1)==truck_state(i,1))&(j<i))
                  two=two+1;
               elseif ((truck_state(j,2)==3)&(truck_state(j,3)==3))|…
                            ((truck_state(j,3)==3)&(truck_state(j,2)~=3)&…
                            truck_state(j,1)>=truck_state(i,1)))|...
                            ((truck_state(j,3)==1)&(truck_state(j,2)~=1)&…
                            (truck_state(j,1)==truck_state(i,1))&(j<i))
                  three=three+1;

               end
            end
         end
         if one==0
            truck_state(i,3)=1;
            waiting(i,1)=0;
            truck_state(i,13)=0;
         elseif two==0
            truck_state(i,3)=2;
            waiting(i,1)=0;
            truck_state(i,13)=0;
         elseif three==0
            truck_state(i,3)=3;
            waiting(i,1)=0;
            truck_state(i,13)=0;

         else
              %then go to spot before transfer points and wait.
            truck_state(i,3)=1;
            waiting(i,1)=1;     %will stop before transf point one and wait for
                                               transfer point to become available.
```

```
      end
    elseif (truck_state(i,5)>0)&(truck_state(i,5)<5)    %then at transfer point
                                                                              but not full yet
      truck_state(i,8)=1; %just wait at current transfer point for next LHD
    end

  elseif(truck_state(i,4)==1) %then at transfer point and full (5 bucket loads)
  %go to crusher
    truck_state(i,3)=-1;
  end

new_truck_state=truck_state(i,:);
```

3.3.2 Strategy 2

This strategy is basically the same as strategy 1. Each LHD is still assigned to a specific tunnel. However, the sequence in which the drawpoints are visited is reversed for every other tunnel. This is illustrated in figure 3-4. This strategy is aimed at keeping the ore level above the drawpoints more even, while still obtaining maximum production. By reversing the drawpoint visiting sequence of every other LHD, the loading is spread out more evenly throughout the active mining area. The expectation is that this will result in the ore level being drawn down more evenly. The design of this strategy is based on a scenario where all possible drawpoints are active. The performance of this strategy will therefore obviously be adversely affected by hang-ups. However, this is true for all the strategies.

The movement of the trucks correspond exactly to that found in strategy 1.

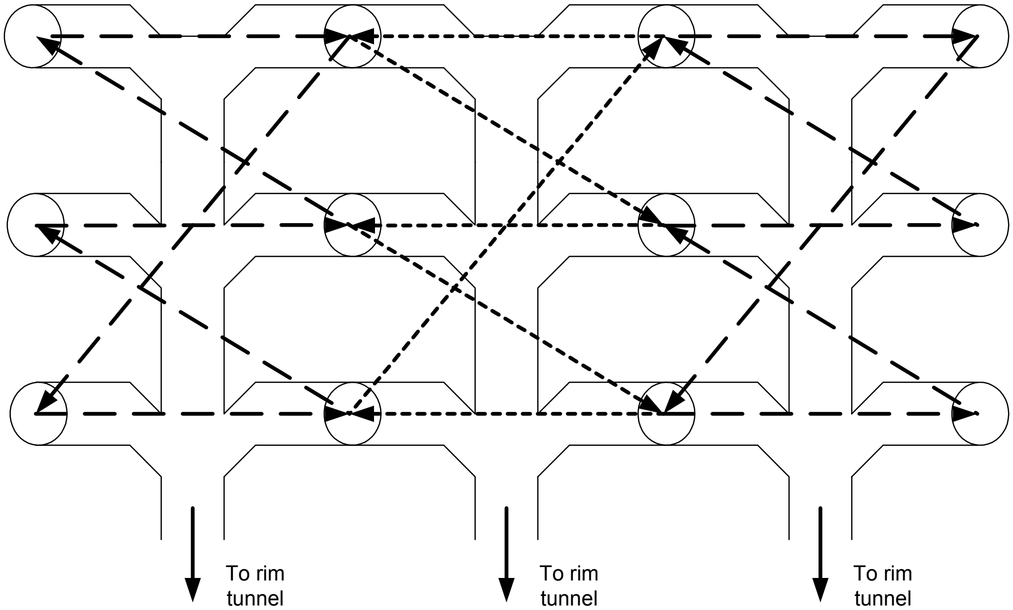


Figure 3-4.
The figure shows the drawpoint visiting sequence of three LHDs assigned to three adjacent tunnels. This is according to strategy 2. The figure again only shows a reduced number of drawpoints.

3.3.2.1 Source code: LHD dispatching

```
function new_lhd_state = lhd_dispatch_2(LHD_state,drawpoint_state,tunnel_state,i)

global num_lhd num_tunnels previous_drawpoint num_load;

%for i=1:num_lhd

if ((LHD_state(i,2)==-1)&(LHD_state(i,10)==0))|...    %then at transfer point
                                                        (finished off-loading)
     (LHD_state(i,5)==(num_tunnels+1))
  LHD_state(i,7)=-1;
     if LHD_state(i,2)==-1
        LHD_state(i,17)=0;
     else
        LHD_state(i,17)=1;
      end

  if mod(i,2)==1

   %decide which drawpoint to go to
     if tunnel_state(i,1)==0 %then tunnel is active
        LHD_state(i,5)=i;
        if LHD_state(i,15)==1   %was on left side
           if (drawpoint_state(i*num_load+previous_drawpoint(i,1),1)==1)   %then next
drawpoint is active
              LHD_state(i,6)=previous_drawpoint(i,1);
```

```
               LHD_state(i,16)=0;      %go to right side
            else
               LHD_state(i,29)=1;
               LHD_state(i,15)=0;
               %LHD_state(i,6)=previous_drawpoint(i,1)+1;
               %LHD_state(i,16)=1;      %go to left side
            end
         else    %was on right side
            if (previous_drawpoint(i,1)+1)<=num_load
               if drawpoint_state((i-1)*num_load+previous_drawpoint(i,1)+1,1)==1  %then next
drawpoint is
                                                                             active
                  LHD_state(i,6)=previous_drawpoint(i,1)+1;
                  LHD_state(i,16)=1;
               else
                  LHD_state(i,29)=1;
                  LHD_state(i,15)=1;
                  previous_drawpoint(i,1)=previous_drawpoint(i,1)+1;
               end
            elseif drawpoint_state((i-1)*num_load+1,1)==1
                  LHD_state(i,6)=1;
                     LHD_state(i,16)=1;      %go to left side
            else
               LHD_state(i,29)=1;
               LHD_state(i,15)=1;
               previous_drawpoint(i,1)=1;
            end
         end
      else    %then tunnel not active
         LHD_state(i,5)=num_tunnels+1;   %wait for secondary breaking
         LHD_state(i,6)=1;
      end
  elseif mod(i,2)==0
     %decide which drawpoint to go to
      if tunnel_state(i,1)==0 %then tunnel is active
         LHD_state(i,5)=i;
         if LHD_state(i,15)==1   %was on left side
            if (previous_drawpoint(i,1)-1)>=1
               if (drawpoint_state(i*num_load+previous_drawpoint(i,1)-1,1)==1)   %then next
drawpoint is active
                     LHD_state(i,6)=previous_drawpoint(i,1)-1;
                  LHD_state(i,16)=0;      %go to right side
               else
                  LHD_state(i,29)=1;
                  LHD_state(i,15)=0;
                  previous_drawpoint(i,1)=previous_drawpoint(i,1)-1;
                  %LHD_state(i,6)=previous_drawpoint(i,1)-1;
                  %LHD_state(i,16)=1;      %go to left side
               end
            elseif drawpoint_state((i)*num_load,1)==1
                  LHD_state(i,6)=num_load;
                  LHD_state(i,16)=0;      %go to right side
            else
               LHD_state(i,29)=1;
               LHD_state(i,15)=0;
               previous_drawpoint(i,1)=num_load;
            end
         else    %was on right side
               if drawpoint_state((i-1)*num_load+previous_drawpoint(i,1),1)==1  %then next
drawpoint is active
```

```
                LHD_state(i,6)=previous_drawpoint(i,1);
                LHD_state(i,16)=1;
              else
                LHD_state(i,29)=1;
                LHD_state(i,15)=1;
              end
          end
        else    %then tunnel not active
          LHD_state(i,5)=num_tunnels+1;   %wait for secondary breaking
          LHD_state(i,6)=1;
         end
  end

elseif (LHD_state(i,2)>0)&(LHD_state(i,10)==1) %then at drawpoint (finished loading)
        LHD_state(i,6)=-1;
        LHD_state(i,17)=0;
    %decide which  transfer point to go to
        if i<=3 %tunnel 1 to 3
          LHD_state(i,5)=2;
          LHD_state(i,7)=1;
        elseif i<=5 %tunnel 4 or 5
          LHD_state(i,5)=5;
          LHD_state(i,7)=2;
        elseif i<=7 %tunnel 6 or 7
          LHD_state(i,5)=7;
          LHD_state(i,7)=3;
        elseif i<=10 %tunnel 8 to 10
          LHD_state(i,5)=10;
          LHD_state(i,7)=4;
        end
end
new_lhd_state=LHD_state(i,:);
```

3.3.3 Strategy 3

The focus of this strategy is to keep the difference in the ore levels above adjacent drawpoints as low as possible. In other words it has the same objective as strategy 2. The LHDs are still assigned to specific tunnels, but the sequence in which the drawpoints are visited is not fixed. A cost function is used to determine the next drawpoint that an LHD visits after it has unloaded. When an LHD has finished off-loading it must go to a drawpoint to get another load. To determine the specific drawpoint that will be chosen as the new destination the cost function is computed for each active drawpoint in the specific tunnel assigned to the LHD. The drawpoint that gives the smallest value for the cost function is assigned as the LHD's new destination. To keep the ore level even the amount of ore loaded from each drawpoint must be equal. Therefore the cost function must take the amount of ore loaded from the specific drawpoint into account. Another important factor is the difference in ore level above adjacent drawpoints. This difference must be minimised and therefore the cost function must take this into account as well. According to the industrial partner the maximum allowable angular difference between the ore level above adjacent drawpoints is 7°. The maximum allowable difference in terms of mass or height will depend on the horizontal distances between adjacent drawpoints. This is illustrated graphically in figure 3-5.

It was also considered to include the distance between the LHD and drawpoint as a factor in the cost function. By going to the closest drawpoint the travelling time of the LHD is reduced and the production can be increased. This was, however, not done because it would have meant that the drawpoints closer to the rim tunnel would have been favoured until the ore level difference became too large. Then the drawpoints further away would be visited more frequently, and whatever increase in productivity that was gained, would have been undone. It would have been a case of getting a short term gain but in the process creating a future problem. By not including the distance factor the same productivity can be obtained while keeping the ore level more even throughout production.

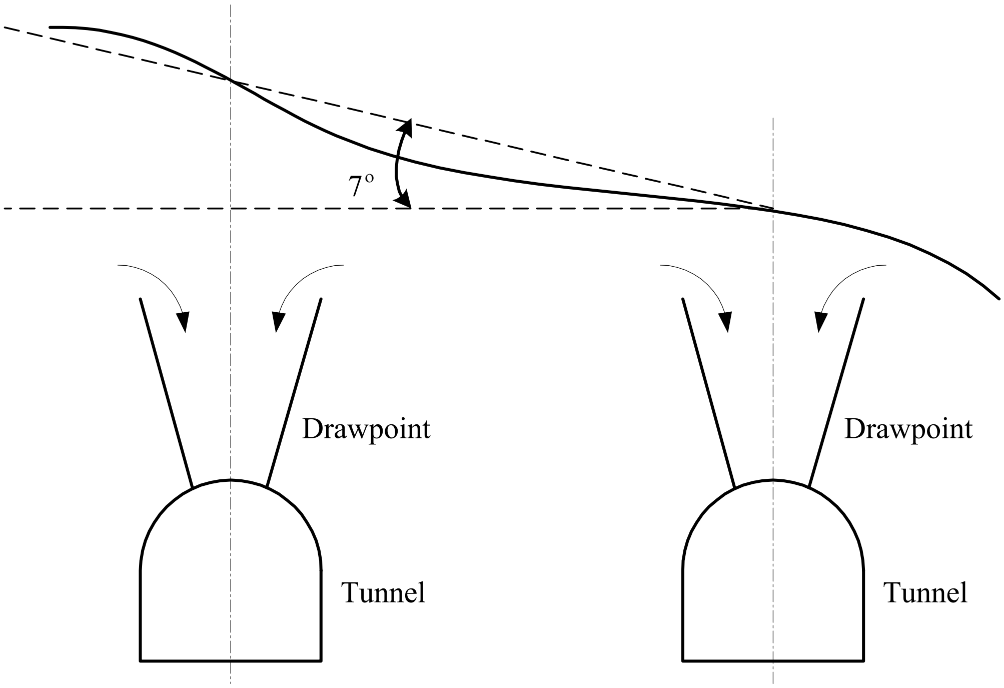


Figure 3-5.
A side view of a cross-cut through the centre of two drawpoints. The maximum allowable difference in ore level above adjacent drawpoints is shown graphically.

The cost function contains the following two terms:

$$T_1 = \frac{tonnes\_drawn\_today}{quota\_per\_day} \tag{1}$$

$$T_2 = \frac{adjacent\_level\_difference}{\max\_level\_difference} \tag{2}$$

In the equations above it can be seen that the two terms are normalized. This is done so that the terms represent fractions that can be used in the same equation. The normalization factor for the first term, T_1, is the maximum amount of ore that may be loaded from a single drawpoint in a day. As mentioned in chapter 2 this limit was obtained from an industrial partner as 200 *mm* of solid rock. For the mine layout used in the simulation, described in this document, this limit translates into 250 *t* of ore per drawpoint per day. For the second term, T_2, the normalization factor is the maximum allowable level difference between adjacent drawpoints. As already mentioned the maximum level difference in angular terms is 7°. For the mine layout used here this translates into a level difference of 1.84 *m* between adjacent drawpoints located between the same pair of extraction tunnels. For adjacent drawpoints between different pairs of extraction tunnels the maximum level difference translates into 3.68 *m*. A detailed description of the mine layout can be found in the appendix, with all the important measurements indicated. Using those measurements and the maximum angular difference, the level differences can easily be verified.

The cost function then looks as follows:

$$J_k = T_1 + T_{2,1} + T_{2,2} + T_{2,3} + T_{2,4} \tag{3}$$

with

$$k = 1,2,...,\# drawpts\_in\_tunnel$$

Each drawpoint has four adjacent drawpoints and therefore term 2 is included four times in the cost function.

3.3.3.1 Source code: LHD dispatching

```
function new_lhd_state = lhd_dispatch_3(LHD_state,drawpoint_state,tunnel_state,i)

global num_lhd num_tunnels previous_drawpoint num_load load_distance;

%for i=1:num_lhd

  if ((LHD_state(i,2)==-1)&(LHD_state(i,10)==0))|...       %then at transfer point
       (LHD_state(i,5)==(num_tunnels+1))                   %(finished off-loading)
    LHD_state(i,7)=-1;
    if LHD_state(i,2)==-1
      LHD_state(i,17)=0;
    else
      LHD_state(i,17)=1;
    end
  %decide which drawpoint to go to
    if tunnel_state(i,1)==0 %then tunnel is active
      LHD_state(i,5)=i;
      min=10000000;
      for j=((i-1)*num_load+1):(i*num_load+num_load)
        %determine cost function for each drawpoint
        if drawpoint_state(j,1)==1
```

```
                cost=0;
                cost=cost+drawpoint_state(j,3)/drawpoint_state(j,4);      %term 1
                if (mod(j,num_load)~=0)
                     %term 2
                     cost=cost+(drawpoint_state(j,2)-drawpoint_state(j+1,2))/1449;
                end
                if (mod(j,num_load)~=1)
                     %term 2
                  cost=cost+(drawpoint_state(j,2)-drawpoint_state(j-1,2))/1449;
                end
                if (drawpoint_state(j,8)~=1)
                     %term 2
                  cost=cost+(drawpoint_state(j,2)-…
                                         drawpoint_state(j-num_load,2))/2898;
                end
                if (drawpoint_state(j,8)~=(num_tunnels+1))
                     %term 2
                  cost=cost+(drawpoint_state(j,2)-…
                                         drawpoint_state(j+num_load,2))/2898;
                end

                if cost<min
                   min=cost;
                   if (j/num_load)>i
                      LHD_state(i,16)=0;
                      LHD_state(i,6)=j-i*num_load;
                   else
                      LHD_state(i,16)=1;
                      LHD_state(i,6)=j-(i-1)*num_load;
                   end
                end
             end
          end

       else    %then tunnel not active
          LHD_state(i,5)=num_tunnels+1;   %wait for secondary breaking
          LHD_state(i,6)=1;
       end

    elseif (LHD_state(i,2)>0)&(LHD_state(i,10)==1) %then at drawpoint
                                                              (finished loading)
       %previous_drawpoint(i,1)=LHD_state(i,2);
       LHD_state(i,6)=-1;
       LHD_state(i,17)=0;
    %decide which  transfer point to go to
       if i<=3 %tunnel 1 to 3
          LHD_state(i,5)=2;
          LHD_state(i,7)=1;
       elseif i<=5 %tunnel 4 or 5
          LHD_state(i,5)=5;
          LHD_state(i,7)=2;
       elseif i<=7 %tunnel 6 or 7
          LHD_state(i,5)=7;
          LHD_state(i,7)=3;
       elseif i<=10 %tunnel 8 to 10
          LHD_state(i,5)=10;
          LHD_state(i,7)=4;
       end
    end
new_lhd_state=LHD_state(i,:);
```

3.3.4 Strategy 4

This strategy focuses on the dispatching of the trucks. The LHD movements correspond to that found in strategy 1. The aim of this strategy is to space the arrivals of the trucks at the crusher more evenly. This should result in less crusher shutdowns, and the time that trucks have to wait when the crusher is full should also be reduced. The trucks still move in one direction around the crusher loop (loop connecting the crusher and transfer points), but they don't have to receive their full load at one transfer point, as in strategies 1 to 3. A truck can move from one transfer point to another before going to the crusher. Before a truck goes to the crusher it has to be carrying a full load, i.e. 5 LHD loads. A truck can only move to another transfer point further on in the crusher loop, i.e. one with a higher number according to figure 2-10. The transfer point to which the truck moves must obviously not be occupied. The decision to move to another transfer point, or not, is made when a truck has just received a load from an LHD. If a transfer point further on in the crusher loop is not occupied, and no other truck is already on its way there, then the truck will move to that transfer point. If more than one transfer point satisfies this conditions then the truck will move to the closest one. If a truck is at the last transfer point in the crusher loop, and not yet full, then it must stay at that transfer point until it is full.

Another approach that was considered is to allow the trucks to go to the crusher even if it is not fully loaded. This was not pursued because of the long travelling distances between the crusher and transfer points. This approach will work well with a large number of trucks, but it will also increase the operating costs, in terms of fuel consumption and vehicle servicing, considerably.

3.3.4.1 Source code: Truck dispatching

```
function new_truck_state = truck_dispatch_1(truck_state,LHD_state,i)

global crusher_level num_lhd num_trucks waiting entry_crusher…
        exit_crusher_empty exit;

%for i=1:num_trucks
  if (truck_state(i,4)==0)    %then at crusher (finished off-loading) or at
                                                    transfer point but not full yet
   %decide which transfer point to go to
  done=0;
```

```
one=0;
two=0;
three=0;
four=0;
   if (truck_state(i,5)==0)  %then at crusher
      for j=1:num_trucks
         if (i~=j)
            %check if truck is at transf point or on its way there
            if ((truck_state(j,2)==1)&(truck_state(j,3)==1))|…
                          ((truck_state(j,3)==1)&(truck_state(j,2)~=1)&…
                          (truck_state(j,1)>=truck_state(i,1)))|...
                          ((truck_state(j,3)==1)&(truck_state(j,2)~=1)&…
                          (truck_state(j,1)==truck_state(i,1))&(j<i))
               one=one+1;
            elseif ((truck_state(j,2)==2)&(truck_state(j,3)==2))|…
                          ((truck_state(j,3)==2)&(truck_state(j,2)~=2)&…
                          (truck_state(j,1)>=truck_state(i,1)))|...
                          ((truck_state(j,3)==1)&(truck_state(j,2)~=1)&…
                          (truck_state(j,1)==truck_state(i,1))&(j<i))
               two=two+1;
            elseif ((truck_state(j,2)==3)&(truck_state(j,3)==3))|…
                          ((truck_state(j,3)==3)&(truck_state(j,2)~=3)&...
                          (truck_state(j,1)>=truck_state(i,1)))|...
                          ((truck_state(j,3)==1)&(truck_state(j,2)~=1)&…
                          (truck_state(j,1)==truck_state(i,1))&(j<i))
               three=three+1;

            end
         end
      end
      if one==0
         truck_state(i,3)=1;
         waiting(i,1)=0;
         truck_state(i,13)=0;
      elseif two==0
         truck_state(i,3)=2;
         waiting(i,1)=0;
         truck_state(i,13)=0;
      elseif three==0
         truck_state(i,3)=3;
         waiting(i,1)=0;
           truck_state(i,13)=0;

      else
         %then go to spot before transfer points and wait.
         truck_state(i,3)=1;
         waiting(i,1)=1;     %will stop before transf point one and wait for
                                             transfer point to become available.
      end
   elseif (truck_state(i,5)>0)&(truck_state(i,5)<5)    %then at transfer point
                                                                          but not full yet
         if(truck_state(i,2)==1)
            occupied_2=0;
            occupied_3=0;
            for j=1:num_trucks
               if ((truck_state(j,2)==2)&(truck_state(j,3)==2))|...
                     ((truck_state(j,3)==2)&(truck_state(j,2)~=2)&...
                     ((entry_crusher(2)-truck_state(j,1))<=…
                          (exit(1)+(entry_crusher(2)-exit_crusher_empty(1)))))
                  occupied_2=1;
```

```
                  elseif (truck_state(j,2)==3)&(truck_state(j,3)==3)|...
                        ((truck_state(j,3)==3)&(truck_state(j,2)~=3)&...
                        ((entry_crusher(3)-truck_state(j,1))<=…
                              (exit(1)+(entry_crusher(3)-exit_crusher_empty(1)))))
                     occupied_3=1;
                  end
               end
                  for j=1:num_lhd
                  if (LHD_state(j,3)==1)&(LHD_state(j,2)==-1)
                     occupied_2=1;
                     occupied_3=1;
                  end
               end
               if occupied_2==0
                  truck_state(i,3)=2;
               elseif occupied_3==0
                  truck_state(i,3)=3;
               end
            elseif truck_state(i,2)==2
               occupied_3=0;
               for j=1:num_trucks
                  if ((truck_state(j,2)==3)&(truck_state(j,3)==3))|...
                        ((truck_state(j,3)==3)&(truck_state(j,2)~=3)&...
                        ((entry_crusher(3)-truck_state(j,1))<=…
                              (exit(1)+(entry_crusher(3)-exit_crusher_empty(1)))))
                     occupied_3=1;
                  end
               end
               for j=1:num_lhd
                  if (LHD_state(j,3)==2)&(LHD_state(j,2)==-1)
                     occupied_3=1;
                  end
               end
               if occupied_3==0
                  truck_state(i,3)=3;
               end
            else
               truck_state(i,8)=1; %just wait at current transfer point for next
                                                LHD
            end
      end

   elseif(truck_state(i,4)==1) %then at transfer point and full (5 bucket loads)
   %go to crusher
      truck_state(i,3)=-1;
   end
   %end

new_truck_state=truck_state(i,:);
```

3.3.5 Strategy 5

This strategy again focuses on the movements of the LHDs. The truck movements correspond to that found in strategy 1. Strategy 5 is basically the same as strategy 3 but in this case each LHD is not assigned to a specific tunnel. The cost function is computed for all the active drawpoints in all the available tunnels. The drawpoint resulting in the lowest value for the cost function is chosen as the new destination of the LHD. An available tunnel is one in which no LHD is present, and no LHD is already on its way to. It is also not closed for secondary breaking. The number of LHDs is equal to the number of active tunnels. For each tunnel that is closed for secondary breaking, one LHD must go to the service area and wait for that tunnel to become active again. This strategy will obviously place a lot of strain on the collision avoidance algorithms. For this reason it was decided to keep the transfer points assigned to specific tunnels as in all the other strategies, so as not to place even more strain on the collision avoidance algorithms.

3.3.5.1 Source code: LHD dispatching

```
function new_lhd_state = lhd_dispatch_4(LHD_state,drawpoint_state,tunnel_state,i)

global num_lhd num_tunnels previous_drawpoint num_load load_distance;

production_tunnels=7;
temp=0;

%for i=1:num_lhd

   if ((LHD_state(i,2)==-1)&(LHD_state(i,10)==0))|...    %then at transfer point
                                                               (finished off-loading)
         (LHD_state(i,5)==(num_tunnels+1))
      LHD_state(i,7)=-1;
      if LHD_state(i,2)==-1
         LHD_state(i,17)=0;
      else
         LHD_state(i,17)=1;
      end
      %decide which drawpoint to go to
      min=10000000;
      for p=1:production_tunnels
         temp=0;
         if tunnel_state(p,1)==0 %then tunnel is active
            for k=1:num_lhd     %determine if another lhd is in tunnel or on its
                                            way there
               if k~=i
                  if ((LHD_state(k,5)==p)&(LHD_state(k,6)>=1))|…
                              ((LHD_state(k,2)>=1)&(LHD_state(k,1)==p))
```

```
          temp=temp+1;
        end
      end
    end
    if temp==0
      % LHD_state(i,5)=i;

      for j=((p-1)*num_load+1):(p*num_load+num_load)
        %determine cost function for each drawpoint
        if drawpoint_state(j,1)==1
          cost=0;
          cost=cost+drawpoint_state(j,3)/drawpoint_state(j,4);%term 1

          if (mod(j,num_load)~=0)

         %term 2
         cost=cost+(drawpoint_state(j,2)-drawpoint_state(j+1,2))/1449;
          end
          if (mod(j,num_load)~=1)

         %term 2
         cost=cost+(drawpoint_state(j,2)-drawpoint_state(j-1,2))/1449;
          end
          if (drawpoint_state(j,8)~=1)

         %term 2
   cost=cost+(drawpoint_state(j,2)-drawpoint_state(j-num_load,2))/2898;
          end
          if (drawpoint_state(j,8)~=(num_tunnels+1))

         %term 2
   cost=cost+(drawpoint_state(j,2)-drawpoint_state(j+num_load,2))/2898;
          end

          if cost<min
            min=cost;
            if (j/num_load)>p
              LHD_state(i,16)=0;
              LHD_state(i,6)=j-p*num_load;
              LHD_state(i,5)=p;
            else
              LHD_state(i,16)=1;
              LHD_state(i,6)=j-(p-1)*num_load;
              LHD_state(i,5)=p;
               end
          end
        end
      end
    end
    %else   %then tunnel not active
    % LHD_state(i,5)=num_tunnels+1;   %wait for secondary breaking
    % LHD_state(i,6)=1;
    %end
  end
end
  if LHD_state(i,6)==-1   %then no suitable tunnel could be found
    LHD_state(i,5)=num_tunnels+1;
    LHD_state(i,6)=1;
  end
```

```
  elseif (LHD_state(i,2)>0)&(LHD_state(i,10)==1) %then at drawpoint
                                                   (finished loading)
    %previous_drawpoint(i,1)=LHD_state(i,2);
    LHD_state(i,6)=-1;
    LHD_state(i,17)=0;
  %decide which  transfer point to go to
    if LHD_state(i,1)<=3 %tunnel 1 to 3
      LHD_state(i,5)=2;
      LHD_state(i,7)=1;
    elseif LHD_state(i,1)<=5 %tunnel 4 or 5
      LHD_state(i,5)=5;
      LHD_state(i,7)=2;
    elseif LHD_state(i,1)<=7 %tunnel 6 or 7
      LHD_state(i,5)=7;
      LHD_state(i,7)=3;
     elseif LHD_state(i,1)<=10 %tunnel 8 to 10
      LHD_state(i,5)=10;
      LHD_state(i,7)=4;
    end
  end
  %end

new_lhd_state=LHD_state(i,:);
```

3.3.6 Combinations

Some of the strategies focus on the LHD dispatching (strategy 1 to 3, and 5), and one focuses on the truck dispatching (strategy 4). It makes sense to combine these strategies to determine if a more optimal solution can be found for the overall system. Strategy 4 is effectively a combination of strategy 1 and 4, so this combination is inherently tested. The combinations that remain are: strategy 2 + 4, strategy 3 + 4, and strategy 5 + 4. The results obtained with these combinations are also discussed in the following chapters.

CHAPTER 4: SIMULATION RESULTS

The results obtained with the simulations are stated in this chapter and will be discussed in more detail in chapter 5.

4.1 INITIAL CONDITIONS

As discussed in the previous chapter the simulation was run for an equivalent of one week of production with strategy 1 implemented, from a zero state. The level of each drawpoint, in terms of the total tons of ore loaded, was recorded and used as the starting condition for subsequent simulations. A graphical representation of this initial condition can be seen in figure 4-1. The levels for tunnels 9 to 11 are zero because these tunnels were not active during the simulation.

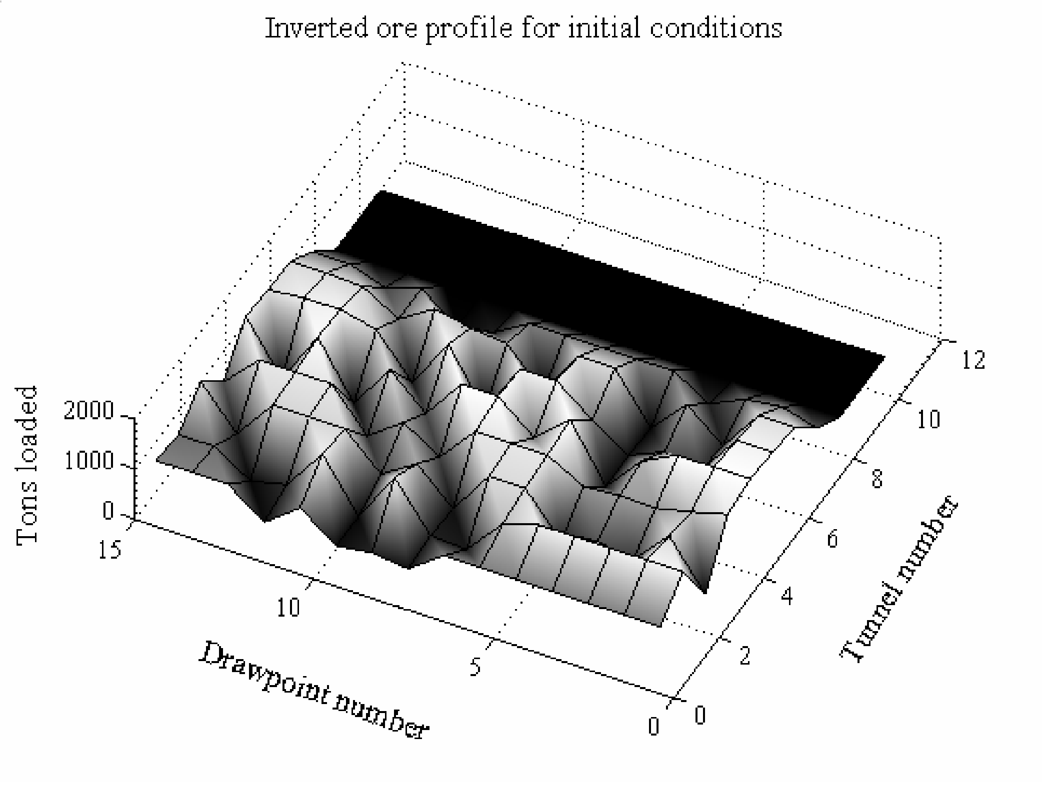


Figure 4-1.
The figure gives a graphical representation of the ore level above the extraction level. This representation is the inverse of the actual level, because it was drawn using the amount of ore loaded from each drawpoint. The higher points indicate higher amounts of ore loaded, and it would therefore be lower points in the physical ore level.

4.2 CRUSHER RESULTS

The results related to the crusher are shown in this section. These include the total amount of ore dumped into the crusher and the number of times the crusher had to shut down because it ran empty. These results will be shown for each strategy and combination of strategies. All the results in this section are for an equivalent of one week of production. As already mentioned this week of production, for each strategy, was started from the initial conditions illustrated in figure 4-1. In figure 4-2 the total amount of ore dumped into the crusher with each strategy is shown. The number of crusher shutdowns for each strategy can be seen in figure 4-3. These results are for a scenario with 7 LHDs and 6 trucks, and the mine layout as described in chapter 3.

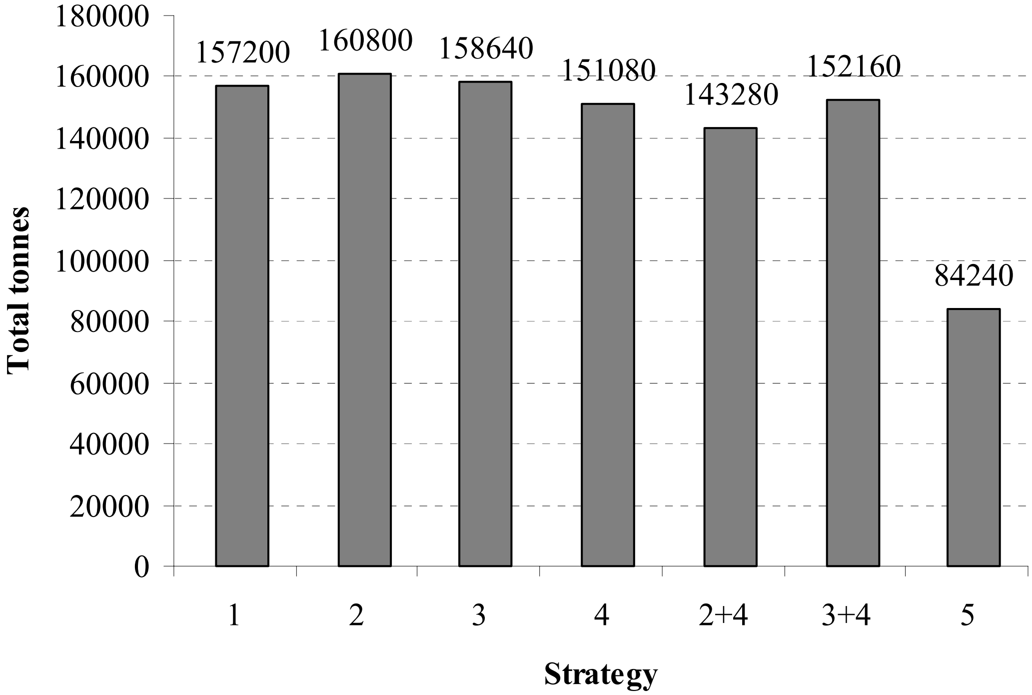


Figure 4-2.
The graph shows the total amount of ore dumped into the crusher, for each strategy, for one week of simulated production.

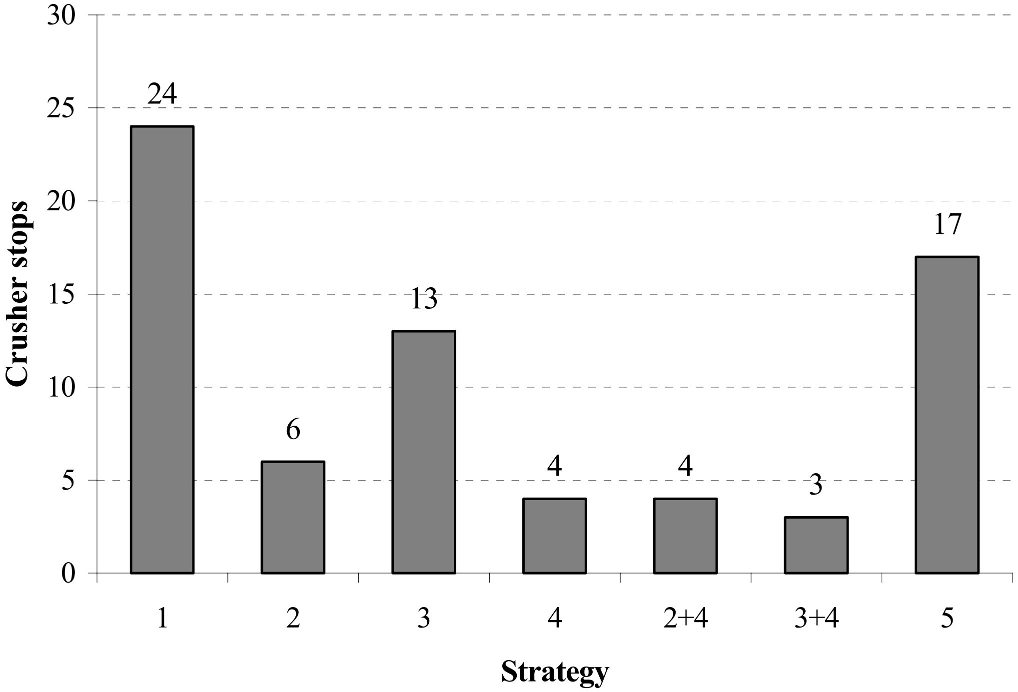


Figure 4-3.
The graph shows the number of crusher shutdowns, for each strategy, for one week of simulated production.

The number of vehicles is a parameter that can easily be controlled in a dispatch system. In a mining environment the purchasing cost and running cost of a vehicle can be very high. It is therefore very important to have the correct number of vehicles to achieve optimal productivity, but also to limit the costs associated with the vehicles. The constraint that only one LHD is allowed in a tunnel at a time means that the number of LHDs is fixed by the layout of the mine.

The number of trucks, however, is not a fixed parameter and therefore strategy 4 and the combinations containing strategy 4 were simulated with different numbers of trucks. Only these specific strategies were used for this particular experiment because strategy 4 is specifically aimed at the movements of the trucks. These simulations were run for the equivalent of one day of production. The influence of the number of trucks on the crusher results can be seen in figures 4-4 to 4-9. In the simulation only three transfer points were active. The maximum number of trucks used was 6. This ensures that even if all three transfer points become available at once, there would be a truck waiting to occupy it. Any more trucks would therefore not make a difference in production. The number of trucks used in the above mentioned simulation runs is two, three, four, five, and six.

A crusher shutdown is recorded by the simulation when the crusher level stays at its minimum value for a period of 12 minutes (in terms of real time and not simulation time). This criterion was changed to 1 second to illustrate the effect of the number of trucks on the number of crusher shutdowns more clearly. This means that the results indicate the time in seconds that the crusher level was at its minimum level (under 30 tons). In the following figures this is called crusher down-time.

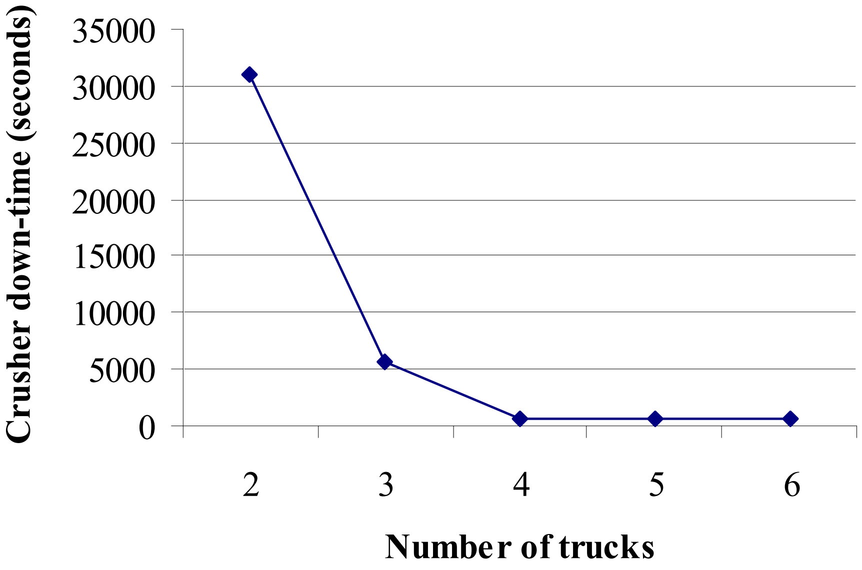


Figure 4-4.
The graph shows the number of crusher shutdowns for different numbers of trucks using strategy 4.

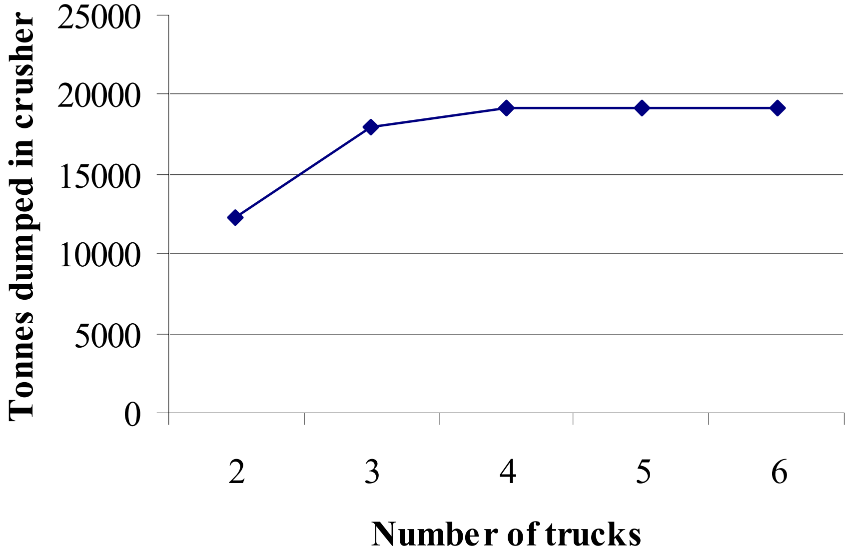


Figure 4-5.
The graph shows the total tons dumped into the crusher for different numbers of trucks using strategy 4.

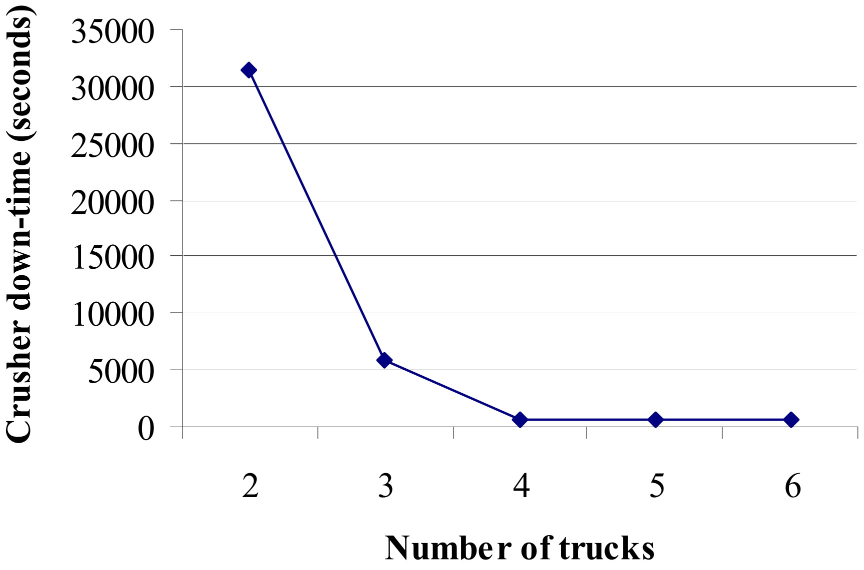


Figure 4-6.
The graph shows the number of crusher shutdowns for different numbers of trucks using the combination of strategy 2 and 4 strategy.

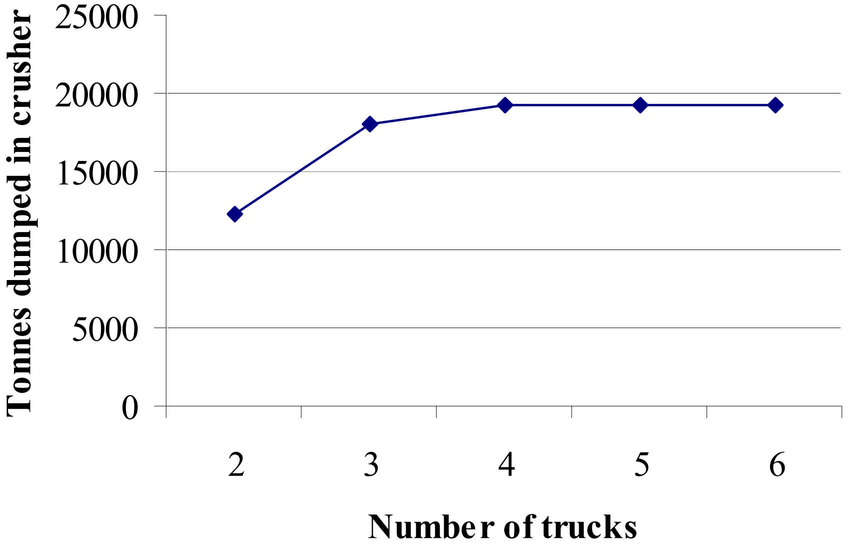


Figure 4-7.
The graph shows the total tons dumped into the crusher for different numbers of trucks using the combination of strategy 2 and 4.

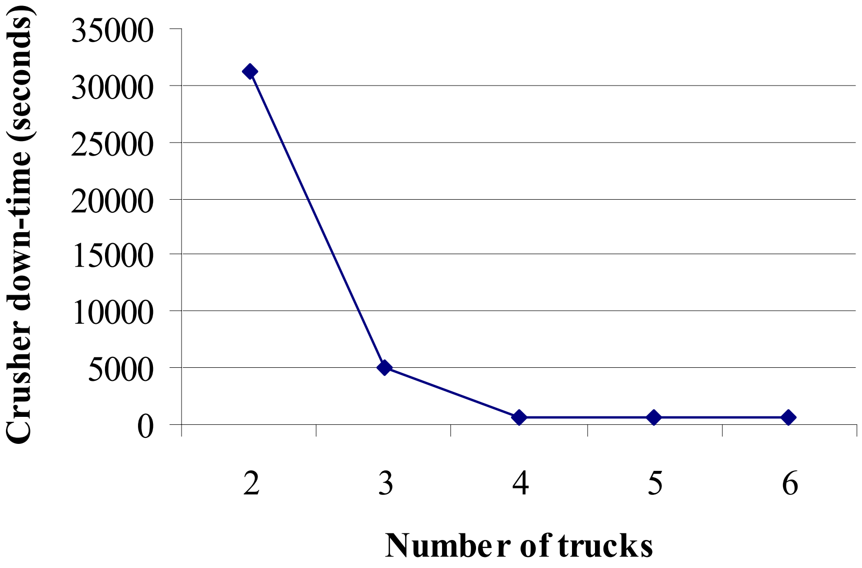


Figure 4-8.
The graph shows the number of crusher shutdowns for different numbers of trucks using the combination of strategy 3 and 4.

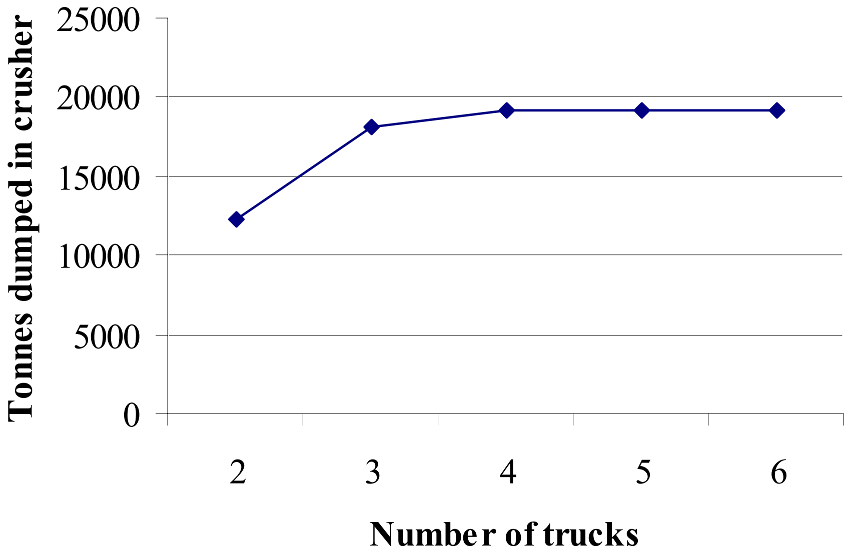


Figure 4-9.
The graph shows the total tons dumped into the crusher for different numbers of trucks using the combination of strategy 3 and 4.

4.3 DRAWPOINT RESULTS

The results related to the drawpoints are given in this section. These results include the average tons drawn per drawpoint, as well as the standard deviation from this average value. Graphical representations of the ore levels above the drawpoints (at the end of the simulation) are also given. These results will be shown for each strategy and combination of strategies. All the results in this section are for an equivalent of one week of production. As already mentioned this week of production, for each strategy, was started from the initial conditions illustrated in figure 4-1. Figure 4-10 shows the average tons drawn per drawpoint for each strategy. The standard deviation from this average value for each strategy can be seen in figure 4-11.

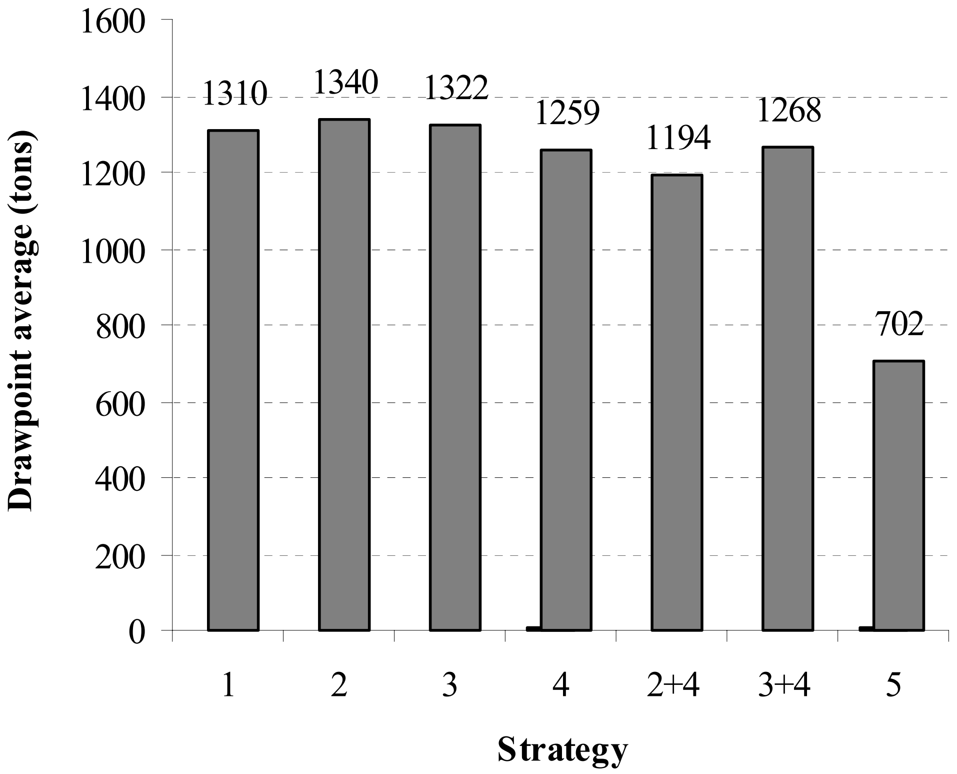


Figure 4-10.
The graph shows the average tons drawn per drawpoint, for each strategy, for one week of production.

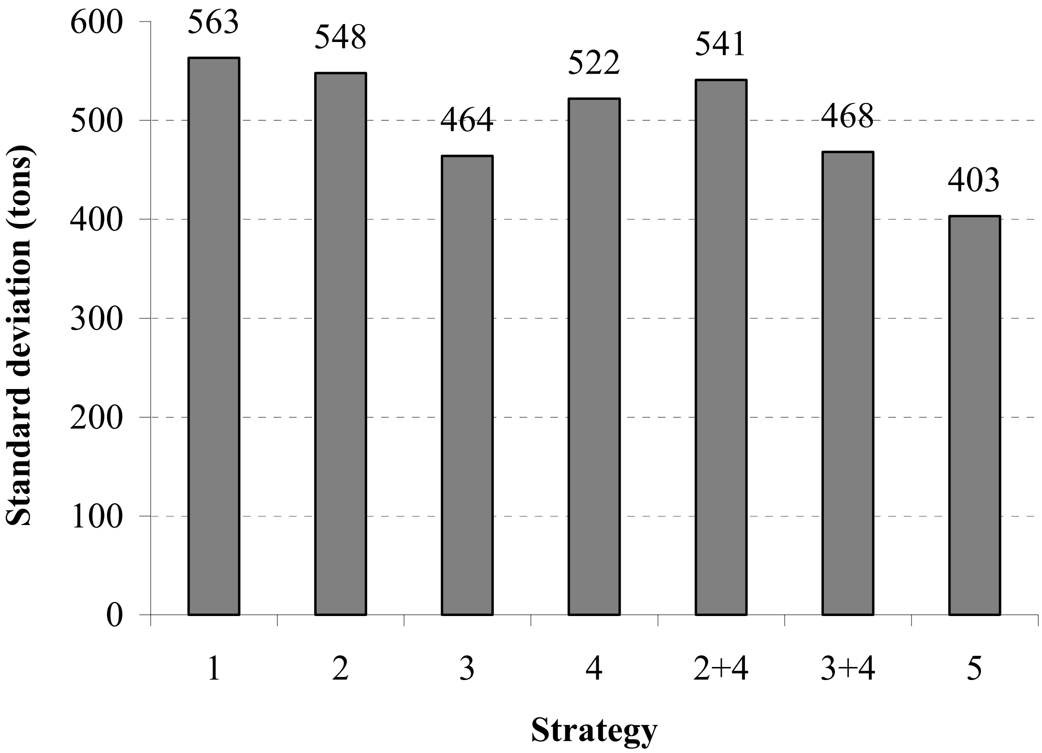


Figure 4-11.
The graph shows the standard deviation from the average tons drawn from each drawpoint, for each strategy, for one week of production.

Figures 4-12 to 4-18 gives graphical representations of the ore surface above the drawpoints. The surface profiles indicates the tons drawn, and therefore it is actually an inverted view of the ore surface. A higher quantity of ore drawn from a drawpoint will cause a lower ore level above that drawpoint, but the surface plots show the inverse of this. The reason for including these plots is to show the measure of success of each strategy to keep the ore level above the drawpoints even. These figures are important because one of the main goals stated for the dispatching strategies is to keep the ore level above the drawpoints as even as possible. Spaces have been put between each tunnel in the ore surface figures, so that it can be interpreted easier. The evenness of the surface can therefore be judged more easily for each tunnel individually.

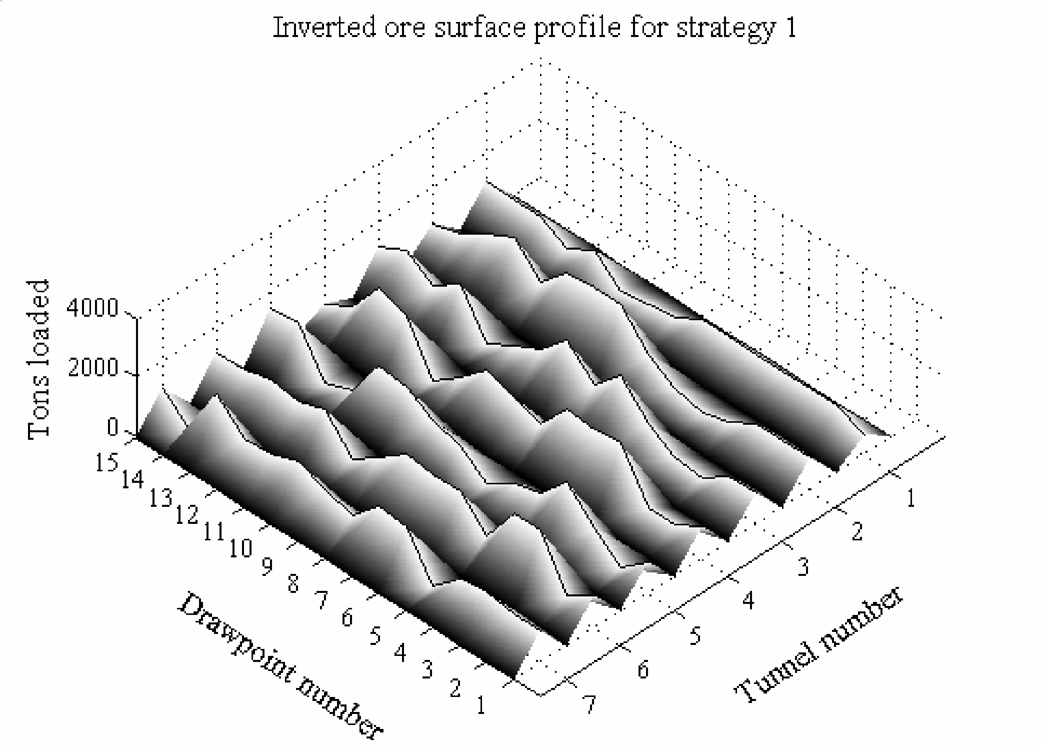


Figure 4-12.
The figure shows the inverted profile of the ore level above the drawpoints for one week of simulated production with strategy 1.

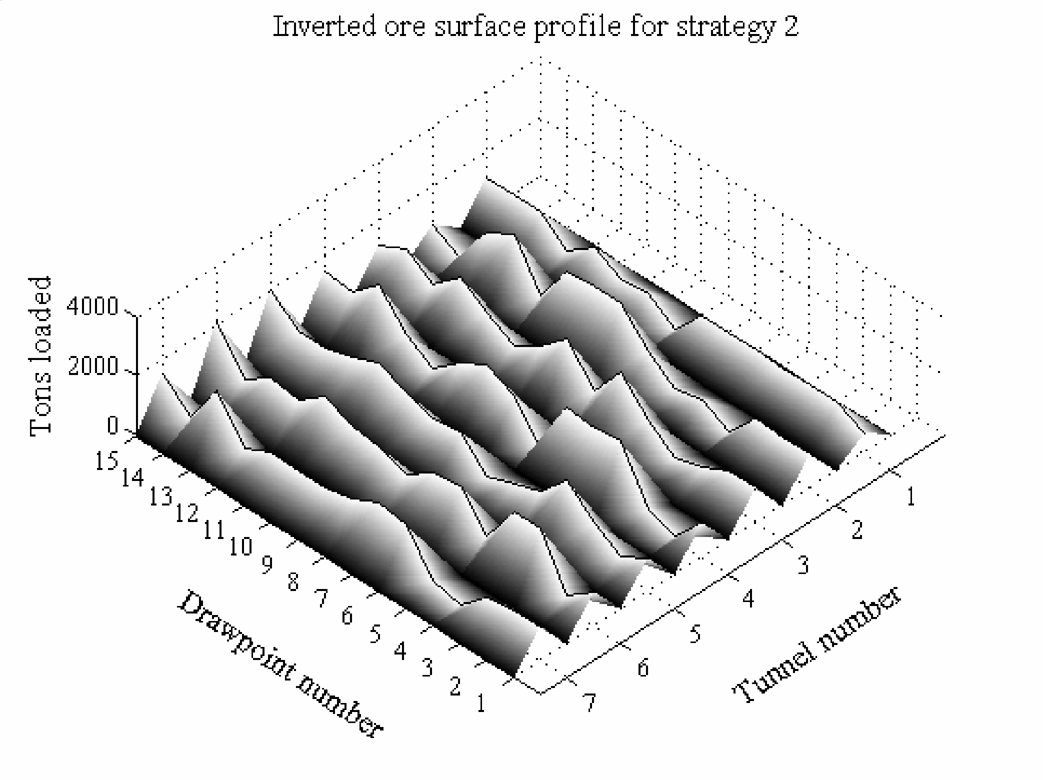


Figure 4-13.
The figure shows the inverted profile of the ore level above the drawpoints for one week of simulated production with strategy 2.

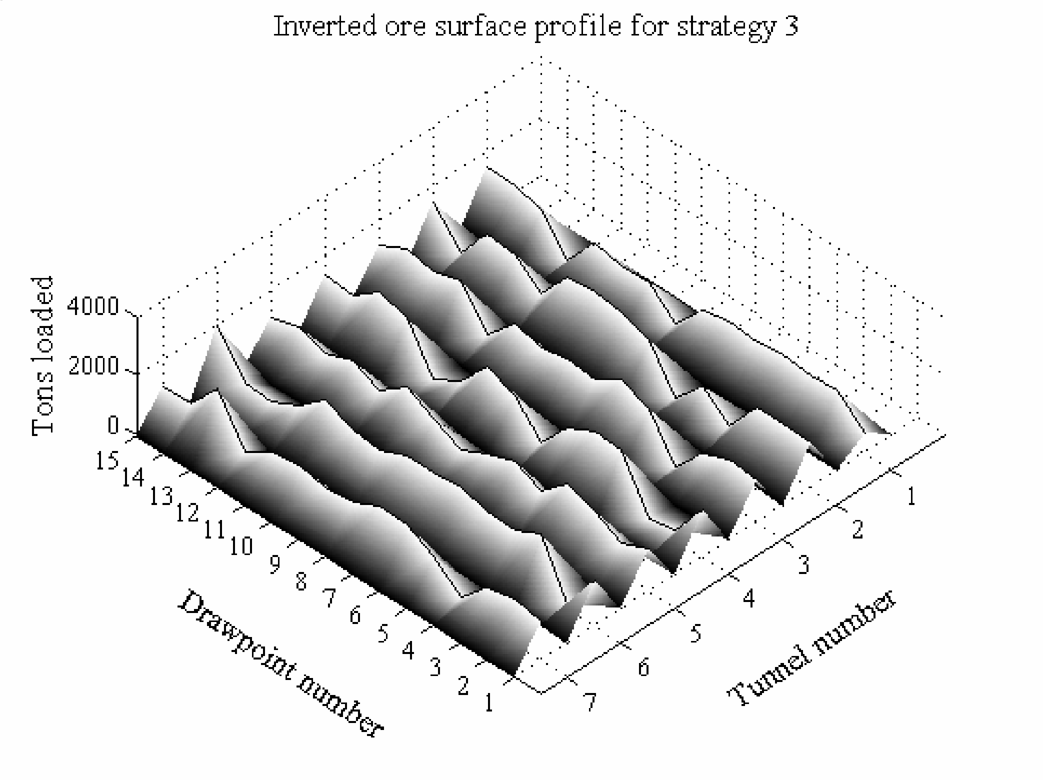


Figure 4-14.
The figure shows the inverted profile of the ore level above the drawpoints for one week of simulated production with strategy 3.

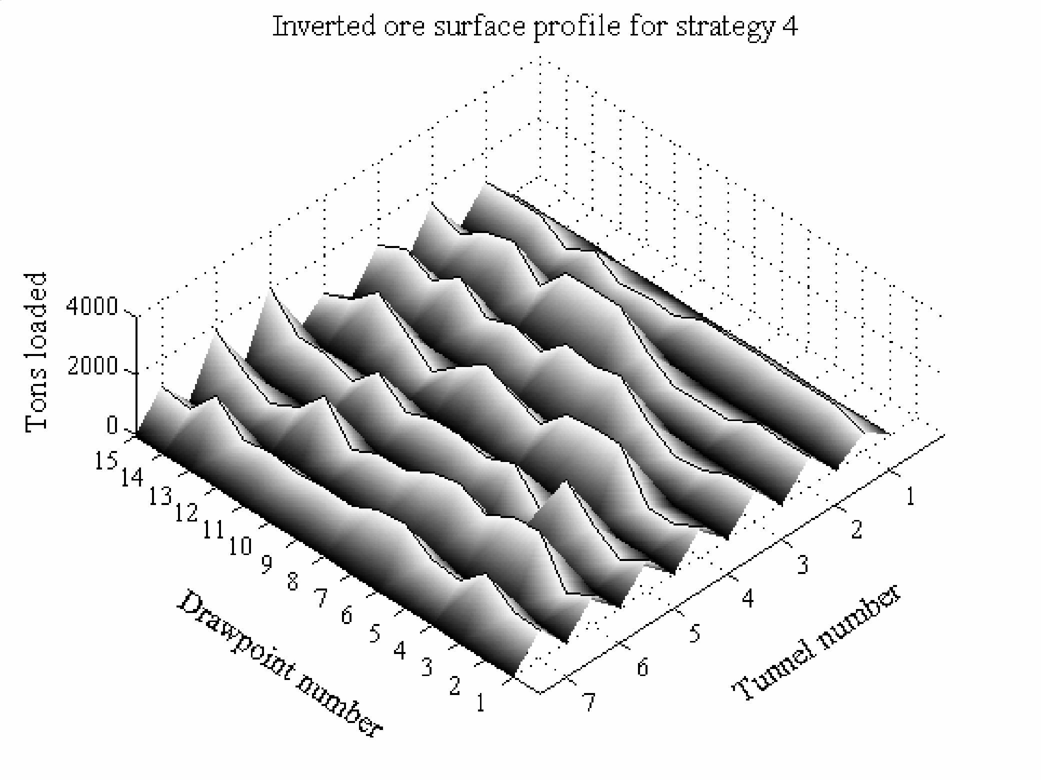


Figure 4-15.
The figure shows the inverted profile of the ore level above the drawpoints for one week of simulated production with strategy 4.

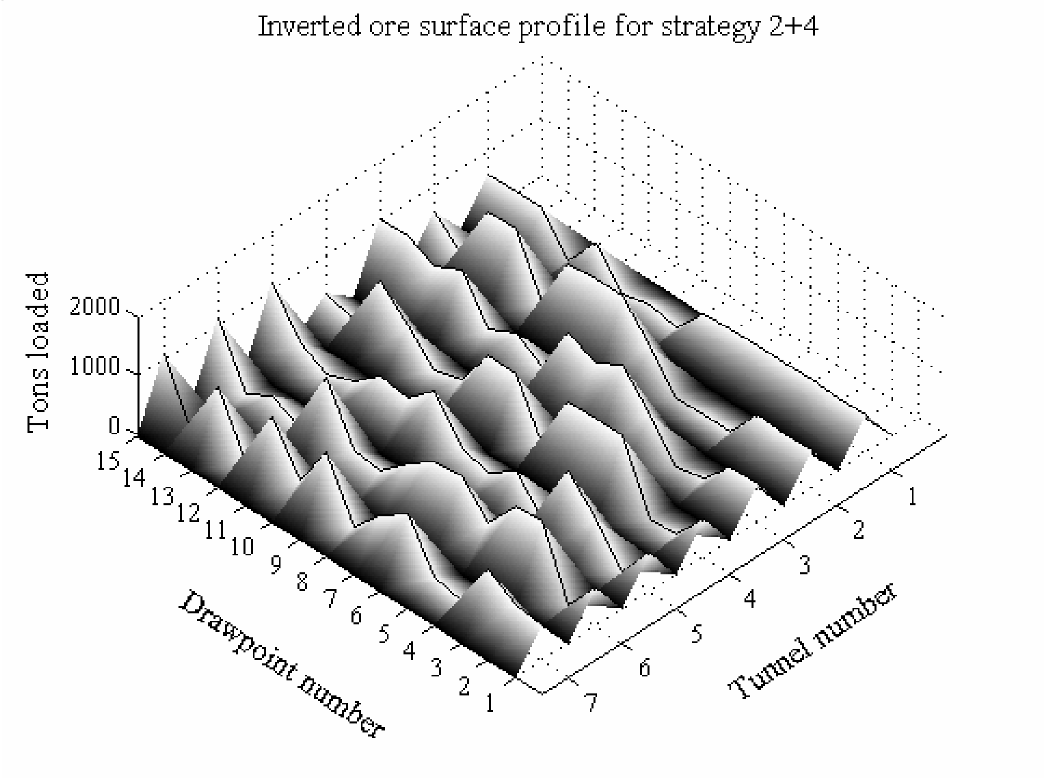


Figure 4-16.
The figure shows the inverted profile of the ore level above the drawpoints for one week of simulated production with the combination of strategy 2 and 4.

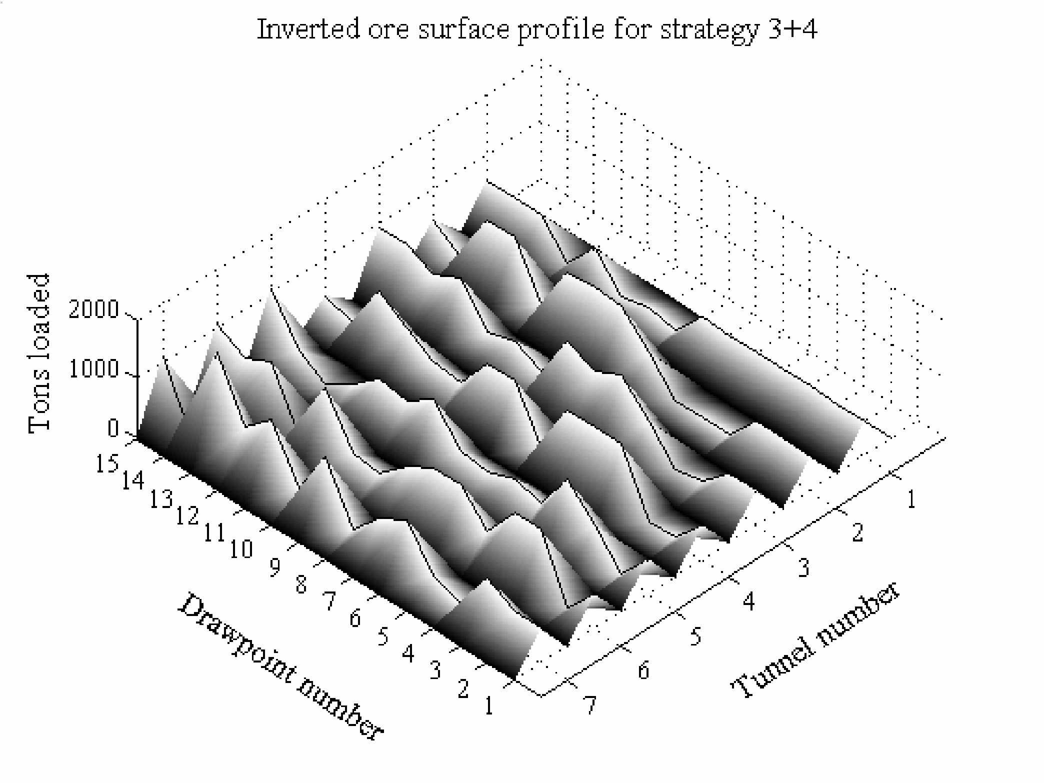


Figure 4-17.
The figure shows the inverted profile of the ore level above the drawpoints for one week of simulated production with the combination of strategy 3 and 4.

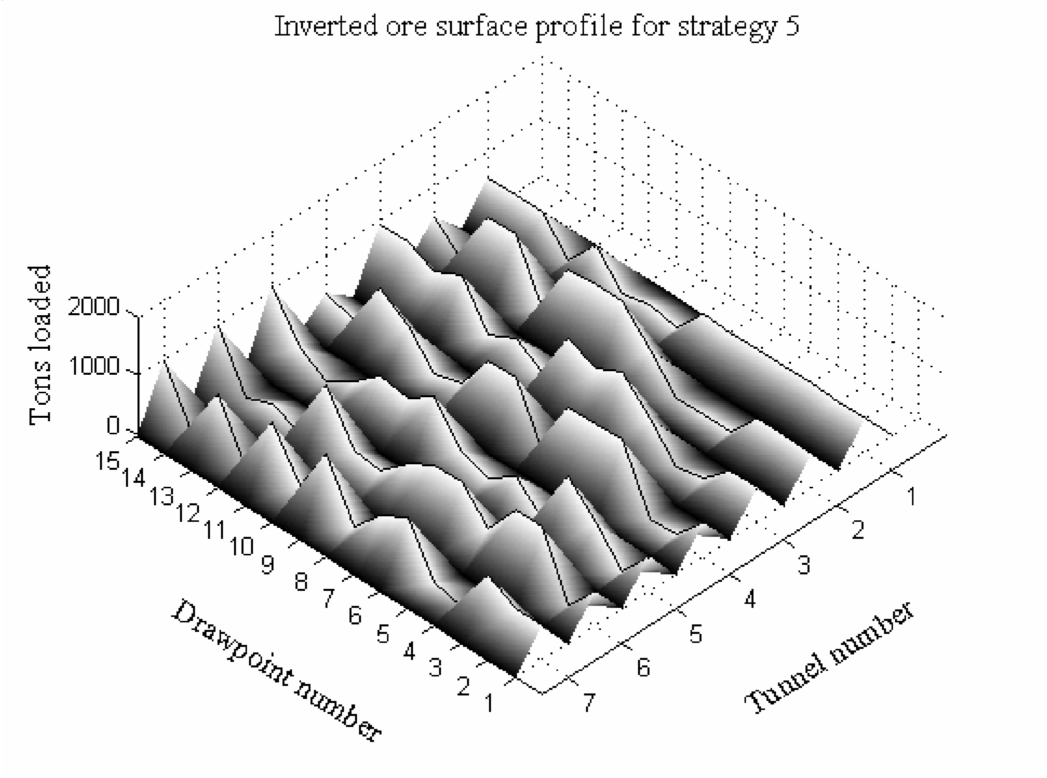


Figure 4-18.
The figure shows the inverted profile of the ore level above the drawpoints for one week of simulated production with strategy 5.

CHAPTER 5: DISCUSSION

In this chapter the results given in the previous chapter are discussed, as well as possible causes of certain features seen in these results. The important factors regarding the physical implementation of a dispatch system are also discussed later in this chapter.

To make the interpretation of the results easier, the evaluation criteria listed in chapter 3 are revisited:

The main objective of a mine is to produce as much as possible each day to maximize profit, but also to ensure that maximal future production is possible. In the context of a block cave mine this means that each active drawpoint must be visited as frequently as possible while still keeping the ore level relatively even. The production must also be as even as possible throughout the time the mine is active. In practical terms this means that the crusher level must stay within its operation limits. This implies that the arrival of the trucks at the crusher must be evenly spaced, not too far apart but also not too close together. If the crusher is empty for a specified time it switches off, when a truck then arrives with a load it must be switched on again. An important performance measure is therefore the number of times the crusher switches off. The results of chapter 4 were compiled with these criteria in mind, and it must also be kept in mind when reading the discussions in this chapter.

It must also be remembered that all the results are for a period of one week of production. The differences between the results of the different strategies might therefore not be that prominent, but will become more pronounced for longer periods of production.

5.1 CRUSHER RESULTS

From the results it is clear that strategy 2 performed the best in terms of total tons produced, and strategy 5 the worst. Strategy 2 also produced one of the lowest number of crusher shutdowns. In terms of the crusher results strategy 2 is by far the most effective. Strategies 1 and 3 produce more crusher stoppages. This is to be expected because strategy 3 is aimed at maximizing the tons drawn from the drawpoints and to keep the ore level as even as possible. It does not attempt to evenly space the arrivals of LHDs at transfer points and therefore also not to evenly space the arrival of trucks at the crusher. In strategy 1 the LHDs progressively move to further drawpoints and this means that the travelling distances for the individual LHDs are almost the same at any time. The LHDs therefore tend to arrive at the transfer points at the same time, and in between very little off-loading takes place. This causes the trucks to receive many loads in a short time, and at other times to wait long for loads. The arrival of the trucks at the crusher is therefore not evenly spaced, and the crusher runs empty more often and has to switch off more. This problem is addressed by strategy 2 because the travelling distance is different for adjacent LHDs most of the time. This is clear from the results in figure 4-3.

Strategy 4 and the combinations containing strategy 4 produced low values for the total tons dumped in the crusher, but on the other hand they produced the least number of crusher stoppages. This is because strategy 4 is aimed at spreading the arrivals of the trucks at the crusher evenly by reducing the travelling distances to the crusher if the truck is full, as well as to the nearest transfer point when the truck is empty. The lower total tons produced could be due to the fact that the different strategies combined to obtain these results were developed in isolation with different goals in mind. It could therefore be expected that the strategies will not necessarily compliment each other.

The expectation was that strategy 5 would not perform very well. This is because of the increased collision avoidance, travelling times, and travelling distances of the LHDs. The results obtained with strategy 5 confirm this prediction, and it shows that the performance of this strategy is very poor compared to all the other strategies.

The number of crusher shutdowns obtained with strategy 5 is not significantly higher than that obtained with the other strategies. It is important to keep in mind the time that the crusher is off before a new truckload arrives. This means that a lower number of crusher shutdowns do not necessarily imply better performance, and it must be evaluated in conjunction with the total produced tons. The low number of tons produced with strategy 5 (figure 4-2) clearly indicates that the crusher downtime was much higher if compared to that of the other strategies.

The effects of the number of trucks on the results related to the crusher can be seen in figures 4-4 to 4-9. These results reveal nothing surprising. As one would expect the total tons dumped into the crusher increase as the number of trucks increases. However, the total tons produced reach a limit, at the number of trucks equal to four, and it stays more or less constant for further increases in truck numbers. This limit occurs because the production of the trucks depends on the production of the LHDs, and the production of the LHDs is limited. The factors that limit the LHD production include the following:

- Only one LHD can off-load at a transfer point at a time.
- An LHD can only off-load at its assigned transfer point.
- Only one LHD is allowed in a tunnel at a time, and the number of LHDs equals the number of active tunnels.
- Each drawpoint in a tunnel must be serviced to keep the ore level even. This means that the travelling distances of the LHDs are not fixed. The travelling time is therefore also a limiting factor.

Other limiting factors for the trucks are:

- The capacity of a truck is limited.
- A truck must be full before it goes to the crusher. It must therefore wait for five LHD loads before it can off-load.
- Only one truck can off-load into the crusher at a time.
- Only one truck can occupy a transfer point at a time.

It can also be seen that the number of crusher shutdowns decrease as the number of trucks increases. This number again reaches a limit that stays more or less constant as the number of trucks increases further. The reasons for this crusher shutdown limit are basically the same as those given above for the crusher total limit. The limits in production and crusher shutdowns are first reached with four trucks. It can be concluded that the optimal number of trucks is four, because a further increase in the number of trucks yields no further improvement. The results in figures 4-4 to 4-9 look very similar for the three different strategies. The differences between these strategies are on the LHD side only. The dispatching of the LHDs has only an indirect effect on the crusher results, and therefore the similarity in figures 4-4 to 4-9 is expected.

5.2 DRAWPOINT RESULTS

Before the drawpoint results are discussed it must be kept in mind that some drawpoints became inactive due to hang-ups that occurred. The standard deviation values in figure 4-11 will therefore not reflect the true potential of the strategies because large level differences between inactive and active drawpoints are inevitable. The results are however sufficient to compare the performance of the different strategies with each other.

From the results in figure 4-10 it can be seen that the tons loaded from the drawpoints reflect the values of the crusher totals in figure 4-2. This makes sense because all the ore loaded from the drawpoints must go to the crusher.

The values of the standard deviations indicate that strategy 3 improves on strategy 1. Strategy 3 therefore succeeds in keeping the ore level more even, while still maintaining a high level of productivity. The standard deviation results for last four strategies cannot really be compared to that of the first three due to the lower production values. The lower deviations therefore do not carry any weight and it will not be discussed further. It is however worthy to note that the combination of strategy 3 and 4 produces the second lowest deviation and strategy 5 the lowest deviation (figure 4-10). These results could be expected because strategy 3 and strategy 5 are explicitly aimed at keeping the ore level even.

The ore surface profiles in figures 4-12 to 4-18 give a more complete picture of the effectiveness of each strategy in terms of keeping the ore level even. The cause of some of the deep dips in the profiles can be attributed to drawpoints that became inactive due to hang-ups.

5.3 PHYSICAL IMPLEMENTATION

As previously mentioned the simulations described in this document were done with the assumption that the vehicles are fully automated. It is easy to assume this with a simulation on a computer because all the needed information about the status of the mine and vehicles is available. In a physical mine an elaborate infrastructure (sensors, communication channels etc.) is needed to obtain the same information availability. Different factors will limit this infrastructure, and therefore some compromises will have to be made. The specific application will determine what the acceptable trade-offs are, and this must be looked at very carefully during the planning stages of mining automation.

The simulation results in the previous chapter are very useful to evaluate the performance of different dispatching strategies, but as with most simulations it has some shortcomings. The ideal situation is to have a simulation that models the real life system exactly. For the purpose of this research, however, some compromises are acceptable and some assumptions were made that do not correlate perfectly with the real life system. The main goal of the simulations is to compare different dispatching strategies. The assumptions are therefore the same for all the strategies and it influences the results for all the strategies in the same way. The assumptions are however, not unrealistic, and the results can be compared to real life results with reasonable accuracy.

Some of the factors are very difficult to implement in a simulation environment, and they might not even have a very prominent effect on the results. Therefore these factors (described in the next section) were not incorporated to minimize the development time and processing speed, and still obtain acceptable accuracy. These factors have to be taken into account during the automation of vehicles in a mine, and therefore also during the development and implementation of the dispatching system for that mine. The dispatching system will not be directly influenced by these factors because it will rely on the information supplied by the different sensors. It will however affect the performance, and decision making of the dispatching system indirectly. Some of the assumptions are discussed below.

5.3.1 Tunnels

The assumption was made that the tunnels in the mine are too narrow for two vehicles to pass each other. This might be untrue for some mines but it is a very realistic assumption for mines using automated vehicles. Automated vehicles must have some navigational system that uses physical indicators on the tunnel floor or walls, which are recognized, by a sensory system on the vehicles. In some cases the tunnel wall itself is used for navigation, as described in Mäkelä [26], and this reduces the necessary infrastructure. To make it possible for two vehicles to pass each other two lanes will have to be formed. This will mean an increase in infrastructure is needed in terms of indicators. The sensory systems on the vehicles will also have to be very accurate to prevent the vehicles from changing lanes accidentally. The collision avoidance systems will have to be very accurate as well. More accurate systems imply more expensive systems, and more difficulties implementing it.

The surface of the tunnel floors (roads) is assumed to be smooth, and that no wheel slip occurs. The level of the tunnel floors is assumed to be horizontal with no inclines. These two assumptions mean that the acceleration profile of the vehicles can be modelled easily and it can be kept fixed for a certain type of vehicle. The speed of a vehicle can be kept constant without adjusting the fuel consumption, because without inclines the vehicles do not have to work harder (or less in the case of a decline) to maintain its speed. It can therefore be assumed that the maximum acceleration for the vehicle can always be achieved. Different areas in a mine have different speed limits for the vehicles, which are within the capabilities of the vehicles. It is also assumed that the maximum allowed speed can be reached and maintained by any vehicle.

The vehicles in a mine are very heavy, and transport very heavy loads. This places a lot of strain on the road surfaces, and regular maintenance is needed. This is ignored in the simulation, and it is assumed that the road surface stays smooth for the duration of the simulation, and that no maintenance is done. This is not totally unrealistic because the simulation is only run for an equivalent of one week of production. If the simulation was run for longer periods of production then road maintenance would become an important issue.

5.3.2 Mine layout

The simulation environment models a fixed number of identical tunnels. This means that the tunnels have the same length and the same number of evenly spaced drawpoints. This is not very different from a real mine, although some tunnels may be of different lengths. This might just be because the mine is laid out that way, or because the development of the tunnel is not finished yet. This can easily be catered for in the simulation by making some drawpoints permanently inactive, which effectively shortens the relevant tunnel.

The only off-loading points available to the LHDs in the simulation are the transfer points. In a real mine, however, the LHDs can also off-load into undercut tips (figure 2-7) were it is taken to another level by gravity and then transported further. If the LHDs dump their loads into an undercut tip then the trucks are cut out of the process. To determine the efficiency of the despatching strategies the trucks must be utilized maximally, and therefore the undercut tips are not used as off-loading points for the LHDs.

The modelling of the service area is not very realistic in terms of its location. This will obviously depend on the specific layout of a mine, but in most cases it will be located further from the production area in a real mine than in the simulation. The travelling time to the service area will therefore be longer than in the simulation for most mines. In a mine one service area will be used for all the vehicles, i.e. trucks and LHDs. Separate service areas for the trucks and LHDs are simulated close to the production areas of the respective vehicles. This was done to simplify the modelling. The servicing and repairs of the vehicles are modelled as time delays of fixed lengths (different times for servicing and repairs). These times were obtained from the industrial partner and are based on the averages obtained from historical data. A certain degree of variance in these times will occur in a real life system, some room for improvement in accuracy therefore exists in the simulation.

5.3.3 Vehicles

The vehicle's load capacities, speed limits etc. are based on values obtained from the industrial partner. This is based on their specific application, and the specific vehicles they use. Different mines may use different LHDs or trucks with different characteristics. This can easily be catered for in the simulation by just changing the relevant parameters.

The vehicles are modelled as point bodies with no physical dimensions. This will have some implications if the collision avoidance strategies have to be implemented physically. If the point is seen as the front tip of the vehicle the collision avoidance strategies will still function even if the vehicle dimensions have to be taken into account. This is evident from figure 3-1. To make sure that the existing collision avoidance strategies will work a safety zone can be modelled around each vehicle (point designator) that is large enough to accommodate the vehicle's dimensions. From the layout used for the simulation (figure 2-10) it might seem that an LHD might pass another LHD that is busy loading at a drawpoint. This is because the point designator of the loading LHD is out of the extraction tunnel. However, if this point designator is seen as the front tip of the LHD, then its back end will protrude into the extraction tunnel which makes it impossible for another LHD to pass. This is why only one LHD is allowed per extraction tunnel at a specific time.

The loading of ore at a drawpoint by an LHD is modelled as a fixed time delay. In a real mine the arrangement of rocks at a drawpoint is random, as well as the size of the rocks. The loading will therefore not be identical each time, and varying loading times are inevitable. Another consequence of this is that the LHD loads will not all weigh exactly the same, as it is modelled in the simulation. The same obviously holds for the truck loads, because if the LHD loads vary in weight the truck loads will also vary. For the purpose of the simulation, however, it is sufficient to assume a fixed weight for all LHD loads.

For secondary breaking (clearing of hang-ups) other vehicles are also used. These vehicles include drilling and blasting vehicles. The movement of these vehicles are not modelled explicitly in the simulation, but are implicitly included in the time delay implemented for secondary breaking.

5.3.4 Crusher

One of the important motives for an optimized dispatching strategy is to minimize the number of crusher shutdowns. This implies that the process of shutting down and starting the crusher is not an insignificant one. In the simulation the processes of shutting down and starting the crusher are not modelled explicitly. A more realistic modelling would have been to implement a time delay to accommodate these processes.

5.3.5 Drawpoints

Some of the assumptions regarding the drawpoints have already been mentioned in section 5.3.1. When a hang-up occurs at a drawpoint, that drawpoint becomes inactive and it is not visited by an LHD again before the hang-up is cleared. In real life there will still be some ore available at the drawpoint although no new ore will accumulate because of the hang-up. This ore can therefore be loaded before the hang-up is cleared. This will, however, be very difficult to simulate because there is no way to accurately model the amount of ore still available at a drawpoint.

CHAPTER 6: CONCLUSION

6.1 OVERVIEW

With the advances in the field of mining automation a number of new problems are created. One of these problems is to have a central intelligent point of decision making, to control the processes in the mine. A vast amount of information can be made available in an automated mine because of the increased sensing infrastructure. A human operator will find it difficult to take into account all the relevant information when making a decision, even if he is experienced. Therefore the need exists for a computerized central system that can use all the relevant information to make more optimal decisions.

The work described in this report focuses on the decision-making with regards to the movement of underground mining vehicles. Specifically the vehicles found in a block cave mine. The scope of the research includes the development, implementation, and simulation of different dispatching strategies. Each strategy was developed to achieve a specific goal, and the success (or lack thereof) of each strategy was clearly indicated by the results. Another advantage of using a simulation is that different system configurations can be tested with little difficulty. In this specific case it means that the same scenario can be tested with different numbers of vehicles. This means that the simulation can be used to test dispatching strategies but also to determine the most optimal number of vehicles for a specific scenario. This is a very useful feature because the number of vehicles is a parameter that can easily be controlled in a mine.

In reality the mine consists of two separate systems of vehicles, as already mentioned in chapter 3. This is due to the physical layout of a block cave mine. These two systems are the system of LHDs and the system of trucks respectively. It can therefore be seen as two separate "dispatching systems" that must be optimized. Therefore some of the strategies focus on the dispatching of the LHDs (strategies 1 to 3) and others on the dispatching of the trucks (strategy 1 and 4). However, there is interaction between these systems, and therefore they are not totally isolated. Each of these “dispatching systems” is optimized to achieve certain criteria which are specific to that system. Some of the optimization criteria of one system might conflict with that of the other. For example the main objective of the LHD system, and in fact of the mine as a whole, is to produce as much ore as possible. This might conflict with the objective of minimizing the number of crusher shutdowns by evenly spacing the arrivals of trucks at the crusher.

It is clear that these strategies can very easily be developed in isolation and not specifically to compliment each other. To find the most optimal solution some trade-offs will have to be made. The best solution will also depend on the priorities of the specific application. In some cases one of the strategies described in this document, or a combination of two, will be sufficient. In other cases another approach will be needed.

6.2 MAJOR FINDINGS

The main objective of each strategy is listed below as a reminder:

- Strategy 1: To represent current dispatching in mines.
- Strategy 2: To keep the ore level even.
- Strategy 3: To keep the ore level even.
- Strategy 4: To minimize the number of crusher shutdowns.
- Strategy 5: To keep the ore level even.

The objective of obtaining maximum production is mutual to all the strategies. Different approaches were used to pursue the same objective in strategies 2, 3 and 5. These strategies each modify the standard LHD movement pattern, found in strategy 1, in a unique way. These modifications were discussed in detail in chapter 3.

All the strategies were successful to some degree in terms of their specific objectives. The only strategy that can be classified as a failure is strategy 5, but this was expected. Strategies 2 and 3 delivered high quantities of ore, and strategy 3 produced one of the lowest standard deviations in terms of the ore level above the drawpoints. Strategy 4 and the combinations containing strategy 4 produced the lowest number of crusher shutdowns. The combinations of strategies, however, did not produce quantities of ore as high as strategy 2 and 3. This can be attributed to the separated nature of the systems of vehicles in the mine. These strategies were also simulated with different numbers of trucks, and the optimal number of trucks was found to be four. The strategy that performed the best overall according to the results presented earlier is strategy 3. The best solution will however depend on the specific application. These strategies are good starting points from which more refined solutions can be developed for real life situations. It is difficult to compare these results with existing results in literature, mainly because no work could be found that specifically focus on the dispatching in a mining environment. Related work found in literature also did not focus on the specific mine layout that is used in a block cave mine.

The success of these dispatching strategies in a practical implementation will depend heavily on the available infrastructure, and the quantity and quality of the information that is available. If the objectives are too ambitious it might seem that the cost of implementing a dispatching system is not worth it. However, in most situations it will offer great advantages in terms of safety and productivity.

An important outflow of this work is the development of a relatively inexpensive simulation platform for the testing of dispatching strategies aimed at underground mining applications.

6.3 FUTURE WORK

In chapter 5 the assumptions used to develop the simulation are discussed. These assumptions mean that the simulation is not a perfect reflection of a true system. The assumptions are also justified in chapter 5, and it is concluded that the accuracy of the simulation is sufficient for the purpose of the research. However, a more accurate simulation can only offer advantages, and make the results more credible. Therefore possible future work exists in terms of refining the simulation to reflect a true system even more accurately.

Something that might be considered is to develop a model, which explicitly describes the processes in a mine in terms of rands and cents. The dispatching strategies can then be refined to minimize the inputs in terms of money, and maximize the outputs. A clearer financial basis will make the results more understandable to people in industry who have to make decisions based on these results. It will also make the marketing of proposed dispatching strategies easier.

Obvious future work will be to improve on the proposed strategies, and to investigate the physical implementation of these strategies more thoroughly. Finally something has to be implemented in a real life mine.

REFERENCES

[1] Penswick D. and Gilliland K. (2000), "The e-Mine", *Proceedings of MassMin 2000,* Brisbane, Queensland, 29 Oct – 2 Nov 2000, pp. 839 – 842.

[2] Pukkila J. and Särkkä P. (2000), "Intelligent mine technology program and its implementation", *Proceedings of MassMin 2000*, Brisbane, Queensland, 29 Oct – 2 Nov 2000, pp. 135 – 143.

[3] Scoble M. and Daneshmend L. (1999), "Mine of the year 2020: Technology and human resources", *The Australian Coal Review,* April 1999.

[4] Scheding S., Dissanayake G., Nebot E.M. and Durrant-Whyte H. (1999), "An experiment in autonomous navigation of an underground mining vehicle", *IEEE Transactions on Robotics and Automation,* Vol. 15, No. 1, Feb 1999, pp. 85 – 95.

[5] Roberts J.M., Duff E.S. and Corke P.I. (2001), "Reactive navigation and opportunistic localization for autonomous underground mining vehicles", *Information Sciences 145 (2002),* pp. 127 – 146.

[6] Steele J., Ganesh C. and Kleve A. (1993), "Control and scale model simulation of sensor-guided LHD mining machines", *IEEE Transactions on Industry Applications,* Vol. 29, No. 6, Nov/Dec 1993, pp. 1232 – 1238.

[7] Hwang Y.S., Farmer N. and Hart J. (1999), "Mining automation in the next millennium: Engineering a tele-operated load haul dump model", *Proceedings of the Second International Conference on Intelligent Processing and Manufacturing of Materials*, 10 – 15 July 1999, Vol. 2, pp. 879 - 885.

[8] Banta L.E., Nutter R.S. and Xia Y. (1992), "Mode-based navigation for autonomous mine vehicles", *IEEE Transactions on Industry Applications,* Vol. 28, No. 1, Jan/Feb 1992, pp. 181 – 185.

[9] Shaffer G.K., Stentz A., Whittaker W.L. and Fitzpatrick K.W. (1992), "Position estimator for underground mine equipment", *IEEE Transactions on Industry Applications,* Vol. 28, No. 5, Sep/Oct 1992, pp. 1131 – 1140.

[10] Evers J.J.M. and Koppers S.A.J. (1996), "Automated guided vehicle traffic control at a container terminal", *Transportation Research Part A: Policy and Practice,* Vol. 30, Issue 1, January 1996, pp. 21-34.

[11] Xiao Bing W.U. (1998), "The application of analytic process of resource in AGV scheduling", *Computers and Industrial Engineering*, Vol. 35, 1998, pp. 169 – 172.

[12] Kim S. H. and Hwang H. (1999), "An adaptive dispatching algorithm for automated guided vehicles based on an evolutionary process", *International Journal of Production Economics,* 60 – 61 (1999), pp. 465 – 472.

[13] Smith T. and Sarin S. (1992), "A program for dispatching and routing AVGs", *Computers and Industrial Engineering*, Vol. 23, 1992, pp. 187 – 190.

[14] Yavin Y. (2004), "Modelling the motion of an underground mining vehicle", accepted for publication in *Mathematical and Computer Modelling*.

[15] Dragt B., Craig I.K., Camisani-Calzolari F.R. (2004), "Modelling of an autonomous underground mine vehicle", *Proceedings of the 11th IFAC Symposium on Automation in Mining, Mineral and Metal Processing (MMM2004),* September 2004, Nancy, France.

[16] Banks J., Bennamoun M. and Corke P. (1999), "Fast and robust stereo matching algorithms for mining automation", *Digital Signal Processing,* Vol. 9, 1999, pp. 137 – 148.

[17] Hemami A. (1994), "Motion trajectory study in the scooping operation of an LHD-loader", *IEEE Transactions on Industry Applications,* Vol. 30, No. 5, September/October 1994, pp. 1333 – 1338.

[18] Samanta B., Sarkar B. and Mukherjee S.K. (2004), "Reliability modelling and performance analysis of an LHD system in mining", *The Journal of the South African Institute of Mining and Metallurgy,* January/February 2004, pp. 1 – 8.

[19] Tomlin, C.J., Lygeros, J., and Sastry, S.S. (2000), "A game theoretic approach to controller design for hybrid systems", *Proceedings of the IEEE,* Vol. 88, No. 7, pp. 949 – 969.

[20] Bemporad A., Ferrari-Trecate G., and Morari M. (2000), "Observability and controllability of piecewise affine and hybrid systems", *IEEE Transactions on Automatic Control,* Vol. 45, No. 10, pp. 1864 – 1876.

[21] Morari M. and Bemporad A. (1999), "Control of systems integrating logic, dynamics, and constraints.", *Automatica*, 35(3), pp. 407 – 428.

[22] Colmenares W., Cristea S., and Villegas T. (2001), "MLD systems: modeling and control. Experience with a pilot process", *Proceedings of the 2001 IEEE International conference on Control Apllications*, Mexico City , Mexico, 5 -7 September 2001.

[23] Borrelli F., Bemporad A., and Morari M. (2000), "Optimal controllers for hybrid systems: stability and piecewise linear explicit form", *Proceedings of the 39th IEEE Conference on Decision and Control,* Sydney, Australia, December 2000.

[24] Ho Y. (2000), "A dynamic-zone strategy for vehicle-collision prevention and load balancing in an AGV system with a single-loop guide path", *Computers in Industry,* Vol. 42, 2000, pp. 159 – 176.

[25] Xu F., Van Brussel H., Nuttin M., and Moreas R. (2003), "Concepts for dynamic obstacle avoidance and their extended application in underground navigation", *Robotics and Autonomous Systems,* Vol. 42, 2003, pp. 1 – 15.

[26] Mäkelä H. (2001), "Overview of LHD navigation without artificial beacons", *Robotics and Autonomous Systems,* Vol. 36, 2001, pp. 21 – 35.

[27] Camisani-Calzolari F., Dragt B., and Craig I.K. (2005), "An overview of the automation of load-haul-dump vehicles in an underground mining environment", *Accepted for 16th IFAC World Congress*, Prague, Czech Republic, July 2005.

[28] Craig I.K., Dragt B., and Camisani-Calzolari F. (2005), "Modelling the dynamics of a Load-Haul-Dump vehicle", *Accepted for 16th IFAC World Congress*, Prague, Czech Republic, July 2005.

APPENDIX A: DETAIL DRAWINGS OF MINE LAYOUT

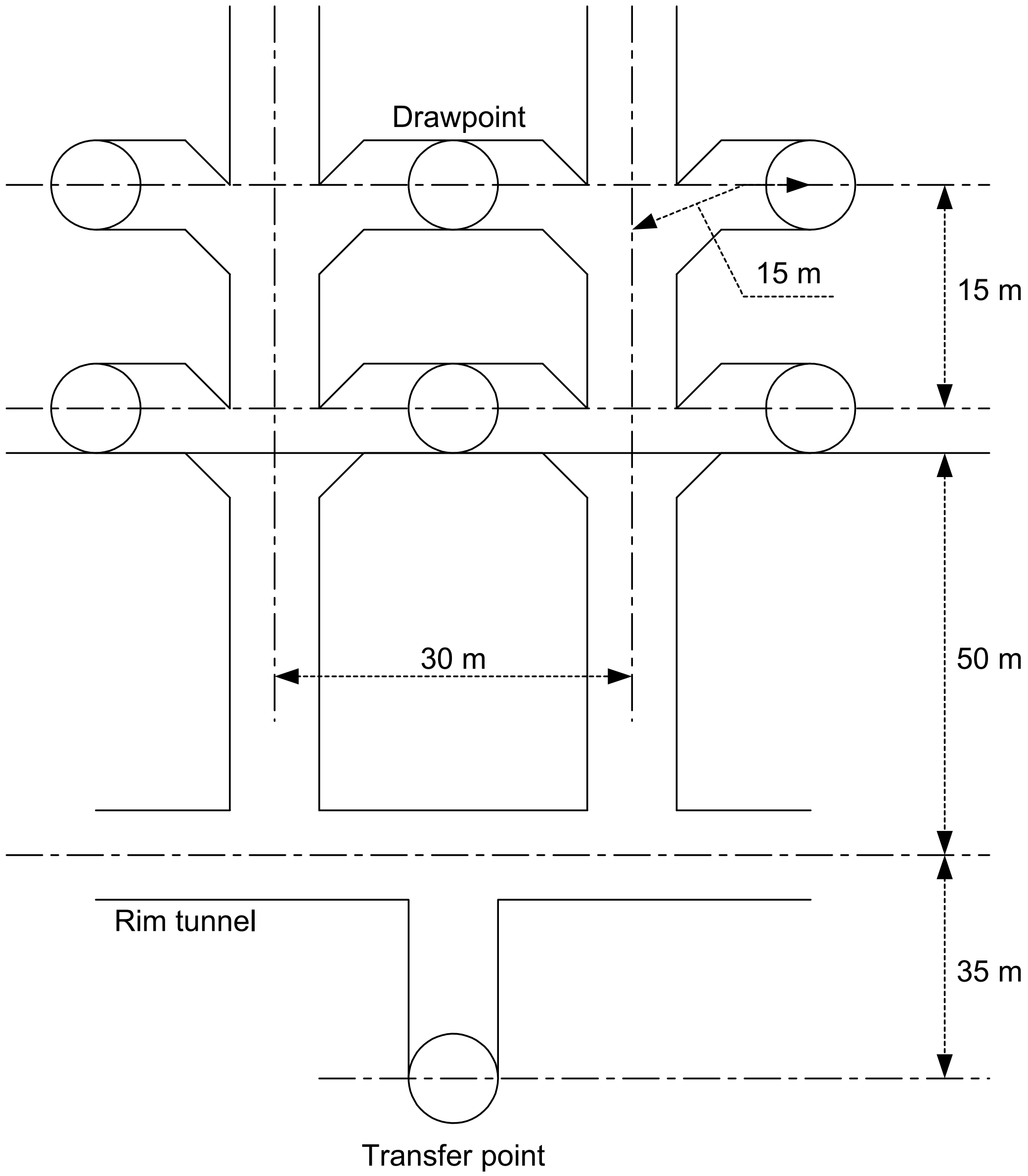


Figure A1.
The figure shows a detail representation of the simulated LHD environment, including dimensions.

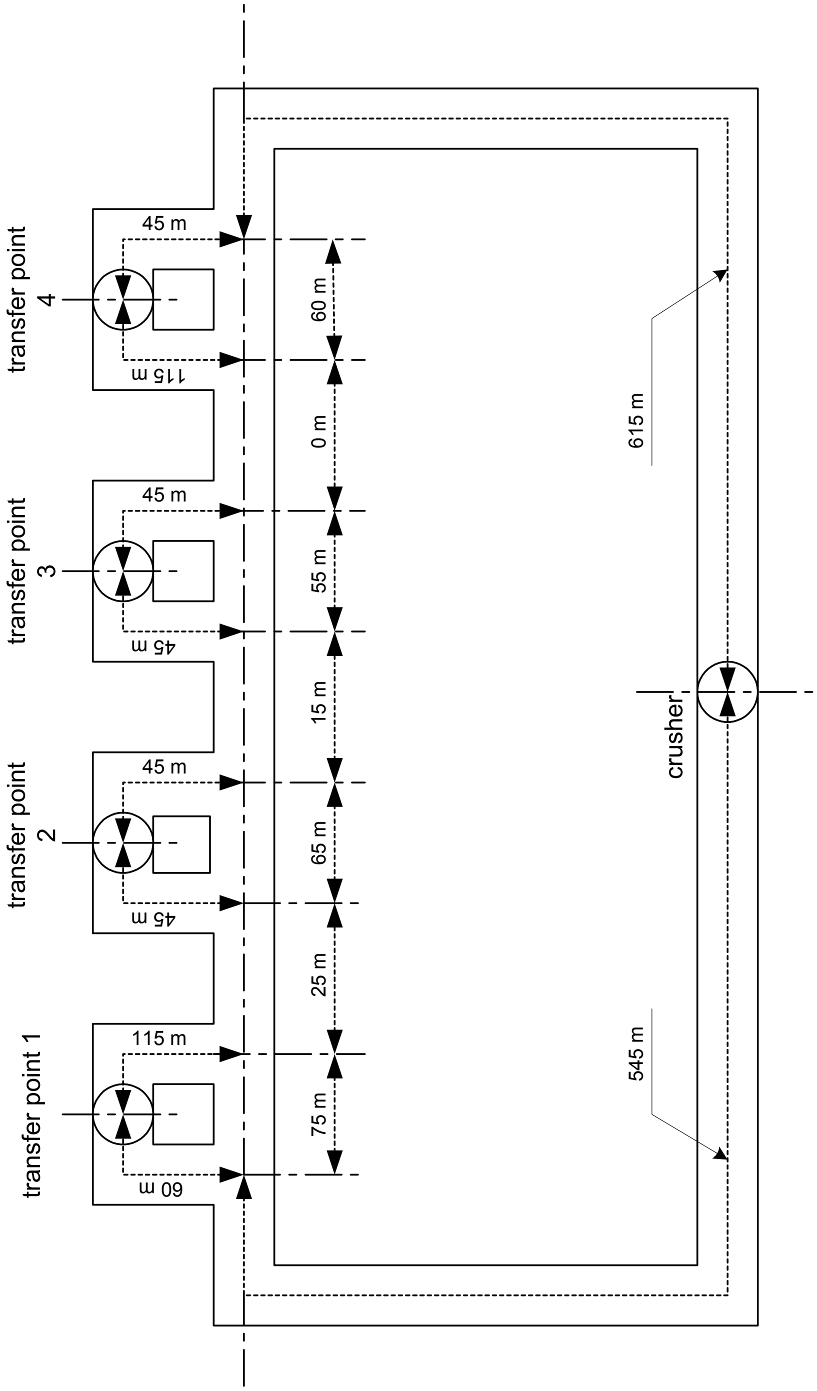


Figure A2.
The figure shows a detail representation of the simulated truck environment, including dimensions.

APPENDIX B: PARAMETER VALUES AND DATA STRUCTURES

B.1 CONSTANT PARAMETER VALUES

The constant parameter values used in the simulation are shown in table B1. The values that have min, average and maximum values are determined using a normal statistical distribution described by these three values.

Table B1. Constant parameter values.

Parameter	**Min**	**Typical**	**Max**
LHD parameters			
Bucket load		9 tons	
Load delay	0.5 minutes	0.6 minutes	0.75 minutes
Off-load time	0.25 minutes	0.33 minutes	0.45 minutes
Acceleration empty		0.4 m/s^2	
Acceleration full		0.35 m/s^2	
Deceleration		-1 m/s^2	
Max speed rim tunnel		15 km/h	
Max speed extraction tunnel		16 km/h	
Max speed drawpoint tunnel		4 km/h	
Turn speed		4 km/h	
Max speed limit in transfer area		16 km/h	
Fuel tank		450 liters	
Refuel time		13 minutes	
Service interval		125 hours	

Parameter	Min	Typical	Max
Service duration	240 minutes	360 minutes	600 minutes
MTBF	40 hours	45 hours	60 hours
MTTR	1.5 hours	2 hours	2.5 hours
Fuel consumption	40 liters/hour	45 liters/hour	50 liters/hour
Truck parameters			
Speed limit		25 km/h	
Intersection speed limit		5 km/h	
Acceleration empty		0.49 m/s^2	
Acceleration full		0.35 m/s^2	
Deceleration		-1 m/s^2	
Off-load time		38 seconds	
Safety zone		10 meters	
Bucket size		45 tons	
Fuel tank		700 liters	
Refuel time		13 minutes	
Fuel consumption	70 liters/hour	80 liters/hour	90 liters/hour
Service interval		125 hours	
Service duration	240 minutes	360 minutes	600 minutes
MTBF	40 hours	45 hours	60 hours
MTTR	1.5 hours	2 hours	2.5 hours
Crusher parameters			
Crush rate		800 tons/hour	
Capacity		140 tons	
Minimum level		30 tons	

B.2 DATA STRUCTURES

The LHD state matrix contains a row for each LHD and each column describes a specific parameter of the LHD. The same holds for each truck, tunnel, and drawpoint. The meaning of each column in the different matrices is defined in tables B2 to B5.

Table B2. Parameters contained in LHD state vector.

Column	**Parameter**	
1	Current Position	Tunnel number
2		Drawpoint number
3		Transfer point number
4		Distance from nearest drawpoint or tunnel entry (m)
5	Destination	Tunnel number
6		Drawpoint number (-1 if transfer point is destination)
7		Transfer point number (-1 if drawpoint is destination)
8	Speed (m/s)	
9	Acceleration (m/s^2)	
10	Load	Full/empty (1/0)
11		Weight (tonnes)
12	Busy loading (yes = 1/no = 0)	
13	Not used	
14	Not used	
15	Current drawpoint side (right = 0/left = 1)	
16	Destination drawpoint side (right =0/left = 1)	
17	Direction of movement (away from rim = 1/towards rim tunnel = 0)	
18	Busy avoiding collision (yes = 1/no = 0)	
19	Slowing down (yes = 1/no = 0)	
20	Idle (waiting = 1/ busy = 0)	
21	Number of LHD it is avoiding (if busy avoiding collision)	
22	MTBF (mean time between failure) (seconds)	
23	MTTR (mean time to repair) (seconds)	
24	Service interval (seconds)	

Column	Parameter
25	Service duration (seconds)
26	Fuel consumption (liters per second)
27	Loading time (seconds)
28	Off-loading time (seconds)
29	Waiting for new command (yes = 1/no = 0)

Table B3. Parameters contained in truck state vector.

Column	Parameter	
1	Current Position	Distance from crusher/transfer point (m)
2		Number of current or last transfer point (or crusher = -1, haulage tunnel = 0)
3	Destination (transfer point number/crusher = -1/service area = 0)	
4	Load (full = 1/empty = 0)	
5	Number of LHD loads received (0 to 5)	
6	Speed (m/s)	
7	Acceleration (m/s^2)	
8	Busy loading/off-loading (yes = 1/no = 0)	
9	Idle (yes = 1/no = 0)	
10	LHD number that is off-loading into truck	
11	Direction (away from transfer point/crusher = 1/towards = 0)	
12	Side relative to crusher or transfer point (right = 0/left = 1)	
13	Slowing down (yes = 1/no = 0)	
14	MTBF (seconds)	
15	MTTR (seconds)	
16	Service interval (seconds)	
17	Service duration (seconds)	
18	Fuel consumption (liters/second)	
19	Off-load delay (seconds)	
20	Waiting for new command (yes = 1/no = 0)	

Table B4. Parameters contained in drawpoint state vector.

Column	Parameter
1	Active? (yes = 1/no = 0)
2	Level (tonnes drawn from point)
3	Tones drawn in current day
4	Production target for day
5	Hang-up? (yes = 1/no = 0)
6	Type of hangup (1=high,2=low,3=high cluster,4=low cluster)
7	Footwall oversize? (1 = yes/0 = no)
8	Tunnel number where drawpoint is located
9	Time until hangup (i.t.o. tonnes drawn)
10	Time to handle hangup (seconds)
11	# of attempts before blasting is successful

Table B5. Parameters contained in tunnel state vector.

Column	Parameter
1	Active? (yes = 0/no = 1)
2	Number of drawpoints
3	Time for secondary breaking