Faculty of Science and Engineering
Department of Mathematics and Statistics

Optimisation Models for Medium-Long Term Logistics Planning in Mining

Jimmi Phangestu

This thesis is presented for the degree of
Doctor of Philosophy
of
Curtin University

February 2017

Declaration

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgment has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

17th February 2017

Jimmi Phangestu

Date

Abstract

Many applications in surface mining have been solved by optimisation techniques. These applications include finding the optimal ultimate pit limit, scheduling the production or extraction plan, allocating or selecting equipment or resources, routing and scheduling transportation systems, etc. Mathematical programming approaches, such as linear programming and integer programming, are commonly used to solve these problems. In this research, we focus on the use of mixed-integer programming to solve a logistics planning problem in iron ore mining.

In the iron ore mining industry, the logistics planning problem involves a mining value chain that comprises stockpiles at multiple mines, a rail network, stockpiles at multiple ports, and shipping. The use of mathematical programming is crucial to provide a tool that plans and schedules the movement of material from mine to port efficiently. The main objective of this research is to provide an iron ore mining company with a logistics planning model that can be applied to their medium to long term mining operation.

In this research, we extend the work of an existing model centred around the mining operation in the Pilbara region, Western Australia, and based on the existing literature. This model is discussed thoroughly in Chapter 3 of this thesis. The model aims to allocate trains to mines such that the total throughput of iron ore is maximised and various operational constraints are satisfied. Furthermore, a blending process, which involves mixing different types of materials, is considered for grade preservation purposes. The objective is to produce shipped products that are in compliance with the desirable grade quality in accordance with the demand.

The mathematical formulation of this model uses the challenging mixed-integer non-linear programming method. The non-linearity occurs due to the existence of bilinear terms in the blending requirement constraints. Both the integral and non-linear conditions complicate this problem, making it extremely difficult to find a global solution. As a solution approach, an iterative method was developed to estimate the values for the decision variables appearing in the non-linear terms. Whilst reasonable solutions are obtained with this approach, there is no rigorous basis for the methodology used. As part of this research, we implement the model on a range of test cases and identify the issues arising in the applications of this methodology.

Our main contributions are outlined in Chapter 4. In this chapter, we

address the non-linearity issue by developing a reformulation of the existing model that represents the same mining logistics operation, but is linear. In this model, we generate a convex relaxation of the problem by providing convex underestimating and concave overestimating functions for each non-linear term. Unlike the other solution approach, this approach is developed based on a sound optimisation theory. We implement the model on test cases and present the results obtained. We analyse our results by comparing the solutions obtained from the two models presented. Our computational results on a range of test cases establish the effectiveness of the new model.

In addition to the test cases, two case studies are considered in this thesis based on two real life industry problems that represent iron ore mining operations in the Pilbara region. The two data sets have different time intervals in the sense that the first data set involves 12 periods, each of which represents one month, whereas the second data set involves 52 periods in which each period is typically a week. The descriptions of the case studies are outlined in Chapter 5. We then implement the two different models to both case studies and present the computational results.

Although a number of iterations are required, the first model generates solutions in a faster CPU time. The second model, however, performs better in minimising the grade deviations, hence maximising the total profit. As our attempt to reduce the CPU time, we implement an aggregated model to reduce the problem size. This has been proven as a useful way to cut the solving time greatly while maintaining the quality of the solutions, thus making the model more practical in real life.

All implementations are done in AIMMS 4.21 software with CPLEX 12.6.3 MIP solver.

Acknowledgments

This research is conducted as a conjunct research project between Curtin University and an industry partner, Rio Tinto Iron Ore, through the IDTC (Industrial Doctoral Training Centre) program. The completion of this thesis would never have been possible without the help of many. For their academic support prior and during my long years of PhD, I would like to thank the following people:

- Prof. Louis Caccetta, my main supervisor from Curtin, for his guidance and great supervision for this research;
- Dr. Palitha Welgama and Dr. Mei-Mei Zhang, my associate supervisors from Rio Tinto Iron Ore, for the interesting problem that they proposed, as well as their input;
- Dr. Mark Grigoleit, for his help with AIMMS and other advice;
- Assoc. Prof. Ryan Loxton, for being responsible for my long struggling years (he was the one who encouraged me to do PhD in the first place!) and his listening ear for me to complain about anything;
- Dr. Greg Gamble, for introducing me to LaTeX during my third year undergraduate project which came in very handy during my thesis writing and his help with my bibliography.

I would also like to extend my gratitude to all family and friends who have been a great encouragement, especially these two most awesome people:

- my mum, for her love and dedication in raising three stubborn boys on her own;
- and Joe, for being the best friend and big brother in the world; not related by blood, but by grace.

Last but not least, I thank God for always being present in my life, and his grace abounding.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The mining industry is an important sector that contributes greatly to the world's economy (Walser, 2000). In Australia, mining is a significant industry as it generates considerable revenue and employment, provides raw materials for various products, and raises the level of living standard in the country (McKay et al., 2000). Mining has contributed to Australia's growing infrastructure and substantial increase of per capita income since the discovery of gold.

Optimisation or operation research has been used in mining operations in order to make better decisions, mainly in mine planning, scheduling, and distributing. In this thesis, we focus on the problem of logistics planning for iron ore mining which includes train scheduling and product blending for maintaining the ore's demanded quality.

In general, a logistics problem involves the movement of material from a source location to a destination. The aim is to move the right amount of the right material from the right source to the right destination effectively and efficiently. In our operation, the mining logistics considers the value chain from a mine to a shipping location. The optimal solution must satisfy the operational and non-operational constraints specified in the problem.

Throughout this chapter, we will introduce some necessary background and context for the iron ore mining and its operational research approaches. The organisation of this chapter is as follows:

- We initiate the chapter by stating some facts to describe the significance of iron ore mining in Section 1.1.
- We then describe the specific mining process at the Pilbara region in Section 1.2. This process involves operations at the mines, rail network, ports, and stockpiles.

- Section 1.3 gives a brief background of general optimisation problems. In this section, we will also describe some topics in optimisation which are the focus of our research.
- Finally, the last section of this chapter (Section 1.4) presents an overview of the whole thesis.

1.1 Iron ore

Iron is the fourth most abundant element in the Earth's crust, preceded by oxygen, silicon, and aluminium (Geoscience Australia, 2015). There is no metal more commonly used in the world than iron. Iron is primarily used as the key ingredient for steel production. About 98% of mined iron ore is processed to produce steel. Iron can also be refined in the form of cast iron or wrought iron. Iron in its various forms is widely used in everyday life. It is used in the construction of buildings, railroads, tunnels, transport materials, machine parts, pipelines, household appliances, and more.

Over the past 40 years, iron ore has also become one of Australia's most significant exports. The quantity of iron ore exported from Australia exceeded 390 million tonnes in 2009–2010, according to the latest Year Book Australia (Australian Bureau of Statistics, 2012), earning around 35 billion dollars and making iron ore the second largest Australian mineral export after black coal in that period. China, Japan, and South Korea were the major destinations of Australian iron ore export.

In iron ore mining, the product is commonly classified as lump or fines iron ore, depending on its size (Geoscience Australia, 2015). Lump iron ore takes more than 6.3 mm in size, whereas fines ore is like a powder less than 6.3 mm in size. Mined lump products can go directly to blast furnace for smelting process, while fines products must be sintered first.

1.2 Pilbara operations

The mines in the Pilbara region of Western Australia (location map shown in Figure 1.1) have produced most of the iron ore exported from Australia (Singh et al., 2014). In 2015, mining operations at the Pilbara produced more than 300 million tonnes of iron ore. Rio Tinto Iron Ore (RTIO) is one of the major miners of iron in this region. Its iron ore operations in the Pilbara region currently have an annual capacity of 240 million tonnes with possibility of further expansion. The mining value chain network of RTIO's Pilbara operation consists of 15 mines, a heavy freight rail network, and four shipping terminals across two different ports. In addition, stockpiles are used as storage locations at the mines and the ports.

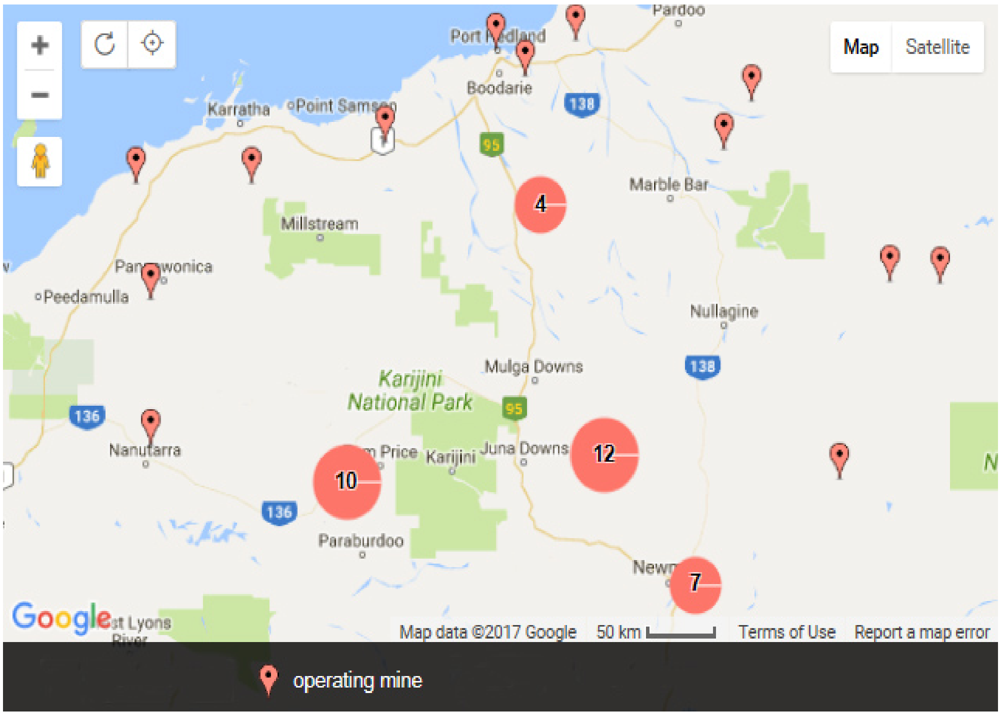

Figure 1.1: Location map of iron ore mines in the Pilbara region of Western Australia (Australian Mining, 2015)

RTIO's mining operations at the Pilbara involve four stages, namely the resource development stage, operations at the mines, the rail system, and operations at the ports (Rio Tinto, 2009). This mining process is described clearly in Figure 1.2. The resource development stage includes exploration, evaluation, metallurgical assessment, and mine planning and scheduling. The aim of this stage is to locate potential resources to be mined so that the planners can come up with a production scheduling plan.

The operation at the mine starts with drilling and blasting the drill holes using explosives to break the iron ore for digging. The mined ore is then loaded onto trucks and hauled to the crushing and screening plants. The processed product is stockpiled before it gets transported by a train to the ports. Once arriving at port, the ore is transferred to stockpiles through ore car dumpers and conveyors. Lastly, the final product is loaded onto ships before it is delivered to the destination countries.

In 2008, RTIO launched Mine of the Future™ which includes automated drilling and operation systems. This new innovation enables the transportation and material handling systems in the Pilbara to be operated from the operations centre in Perth, greatly improving efficiency and safety, as well as reducing the total cost and negative impacts on the environment. More information about this programme can be found in Rio Tinto (2014).

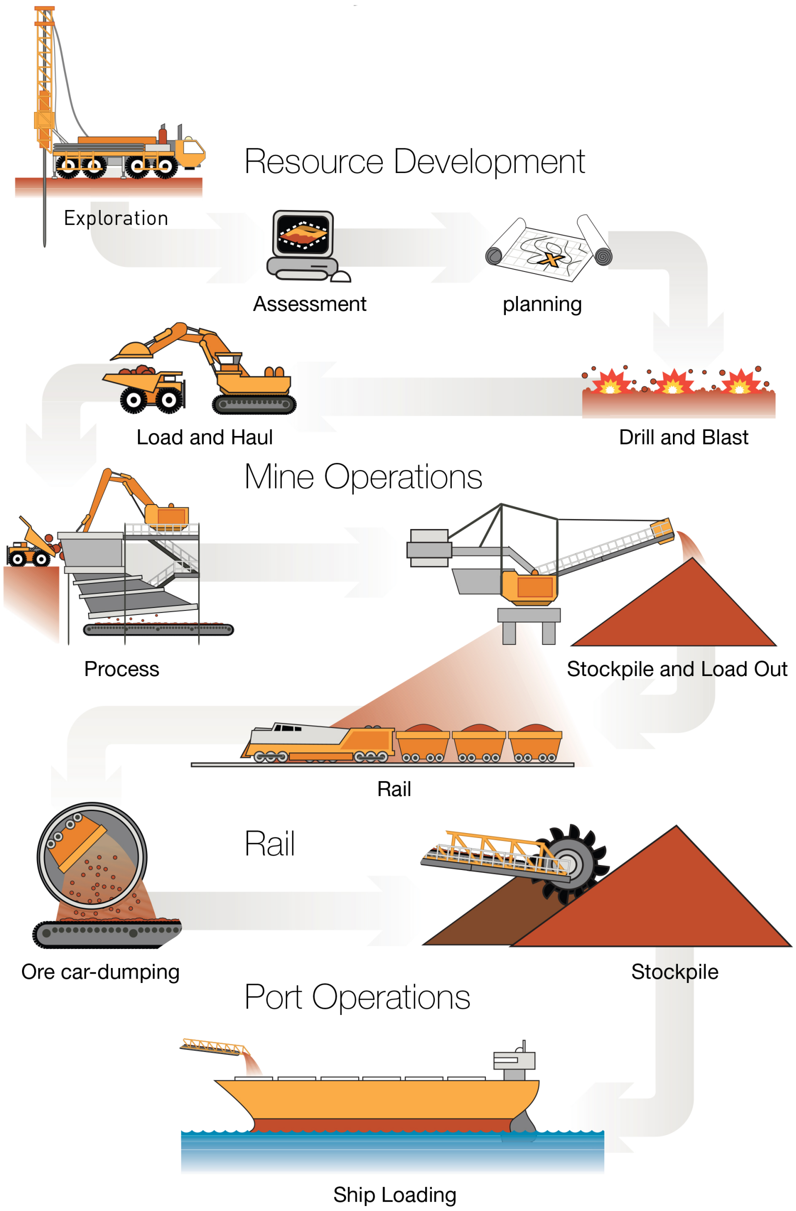

Figure 1.2: Rio Tinto's mining process in the Pilbara (Rio Tinto, 2009)

1.2.1 Mines

RTIO's current Pilbara operations involve 15 mines across several regions, namely East Pilbara, West Pilbara, and North West Pilbara. Most of these mines produce both lump and fines product types and normally separates the two different types on different piles. Figure 1.3 shows a picture of an iron ore mine in the Pilbara and Figure 1.4 shows the map of locations of RTIO's active mines in 2013.

Figure 1.3: An iron ore mine in the Pilbara region of Western Australia (Mining-Technology, 2016b)

Haul trucks and overland conveyors are the transportation modes used to move materials from the mine pits to the process plants. The processed product is then moved to the stockpile before being loaded onto trains for transport to the shipping ports. Most mines have two different types of stockpiles with one being the main production line, and the other one serving as storage.

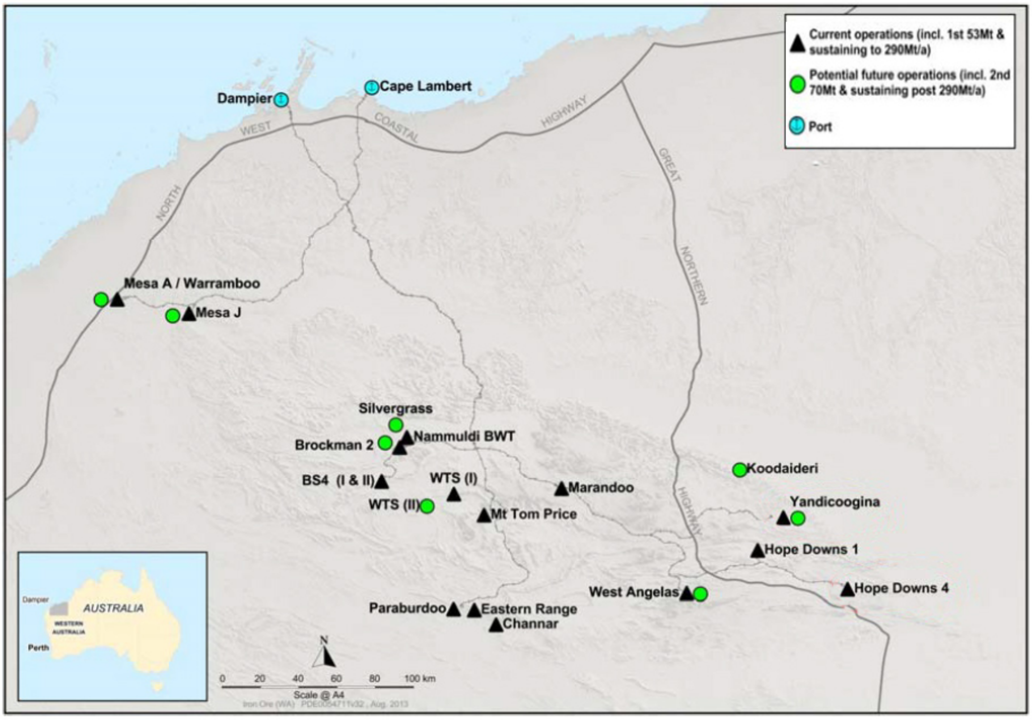

Figure 1.4: RTIO's mine locations in the Pilbara (Rio Tinto, 2013)

Figure 1.5: Rio Tinto's train operating in the Pilbara region of Western Australia (Rio Tinto, 2016)

1.2.2 Rail network

In the Pilbara region, RTIO owns the largest private freight rail network in Australia with a total operating route length about 1,700 kilometres.

The company's trains can be categorised into two different train fleets; each of which serves specific mines. The Robe Valley fleet only serves the mines in the north west region, whereas the pooled fleet serves all the other mines. Figure 1.5 shows one of the trains operating in the Pilbara region. The train goes from the mine to a specific port depending on the type of product it carries.

1.2.3 Ports

The mining value chain of RTIO in the Pilbara involves four shipping terminals. These terminals are located at two different ports, namely Dampier and Cape Lambert. The two terminals at Dampier, namely Parker Point and East Intercourse Island, are owned by Rio Tinto, whereas Cape Lambert A and B terminals are owned by the joint venture between Rio Tinto and other mining companies. Each terminal has its own facilities for train unloading, material stockpiling, product blending, lump product re-screening, and ship loading.

Figure 1.6: A car dumper operating at the port (Rio Tinto, 2013)

A car dumper [Figure 1.6] is used as the mechanism for unloading the train containing the iron ore product. The car dumper holds and rotates the ore cars to dump out the material which will be discharged onto conveyors. The material is then stacked onto the stockpile where the blending process takes place. Before loading onto the ship, the lump product is re-screened and a proportion of smaller ore is transferred to the fine product stockpile.

1.2.4 Stockpiles

In the mining logistics process, a stockpile is a pile of material which serves different purposes: as storage, buffering, or material blending (Singh et al., 2014). In the Pilbara region, most of the iron ore mines have two different types of stockpile, namely live and bulk stockpiles. The live stockpile normally serves as the main production line and blending, whereas the bulk stockpile serves as a buffer. These two types of stockpile can also be found at the ports as storage locations before shipping.

Figure 1.7: Stacking iron ore to stockpile at the Yandicoogina iron ore mine in Western Australia (Mining-Technology, 2016a)

A stacker and a reclaimer are normally used to pile and recover the material respectively. The stacking process at one of the Pilbara mines is pictured in Figure 1.7. There are at least three common methods to pile the material into the stockpile. They are the Cone Shell, Windrow, and Chevron methods.

If the Cone Shell method is used, the material will be piled from a fixed point to form a cone. When the required height of a cone is formed, the material will be piled from the next position so that a cone is formed against the first cone shell. This step continues in the longitudinal direction until all mined material is piled or the stockpile reaches its capacity. In the Chevron method, the stacker constantly moves backward and forward over the center of the stockpile while depositing the material. The Windrow method is done by depositing small separate piles across the stockyard. More piles are then formed to fill in the gaps between the small piles until the stockpile is full.

The stockpile reclaiming process depends on the type of reclaimer being used. There are portal scraper, bucket wheel, bridge scraper, and drum reclaimers. The necessity of the blending effects is considered when choosing the stacking and reclaiming methods. If a consistency of material grade quality is required, the combination of chevron method with bridge scraper or drum reclaimer is a better choice in the stockpiling process.

1.3 Optimisation

In mathematics, optimisation or operations research involves finding the best amongst many possible solutions in solving quantitative decision problems (Luenberger and Ye, 2008). Using a theoretical framework, optimisation translates a problem to a mathematical model or formulation which can be solved using mathematical algorithms. A simple optimisation problem involves minimising (or maximising, depending on the problem) a single objective function, whose purpose is to be a measurement of the quality of the decision, subject to a set of constraints which limit the selection of the decision.

A general mathematical optimisation problem, also called the mathematical programming problem, can be formulated as follows:

$$\begin{aligned} \text{Minimise} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & h_i(\mathbf{x}) = 0, \ i = 1, 2, ..., m \\ & g_j(\mathbf{x}) \leq 0, \ j = 1, 2, ..., r \\ & \mathbf{x} \in S. \end{aligned}$$

where $\mathbf{x}$ is an n-dimensional vector of unknown variables, that is, $\mathbf{x} = (x_1, x_2, ...x_n)$. A real function $f(\mathbf{x})$ is known as the objective function, while real functions $h_i(\mathbf{x})$ and $g_j(\mathbf{x})$ are equality and inequality constraints respectively. The set S is a subset of n-dimensional space.

If the objective function $f(\mathbf{x})$, and all the constraints, $h_i(\mathbf{x})$ and $g_j(\mathbf{x})$ are linear, we have a **linear programming** problem. Many simple linear programming problems can be demonstrated graphically, but most practical problems in industry require extensive computational effort to generate solutions. Examples of linear programming include simple transportation and network flow problems.

If the objective function and/or at least one of the constraints are non-linear, the problem becomes a **non-linear programming** problem.

If all variables are restricted to be integers, the problem becomes an **integer programming** problem. The integer programming methods determine the optimal solution among all discrete solutions in the continuous feasible solution space (Taha, 1975). An integer programming problem can be either linear or non-linear.

If the integrality condition is absent for at least one of the decision variables, we have a **mixed integer programming** problem. A mixed integer programming problem can be at times computationally difficult due to the integrality condition and formation of the constraint matrix. Similarly, a mixed integer programming problem can be either linear or non-linear.

Optimisation methods have solved many problems in industries. The use of optimisation methods is widespread in mining, manufacturing, utilities, energy, logistics, transportation, financial services, government, defence, and many other industries. This thesis focuses on the logistics part of the surface mining industry.

1.3.1 Optimisation in surface mining

Surface mining is a process of extracting a mineral when it is found close to the earth's surface (less than 500m in depth), in contrast to underground mining which is used when a mineral is excavated from well below the surface (Kennedy, 1990). Minerals such as iron, coal, copper, and gold are extracted in surface mines. In surface mining, there are five main methods that can be used to remove the mineral. These include stripping, open-pit, mountain-top removal, dredging, and highwall mining. The iron ore mines are usually open-pit, where the ore is removed from a large open hole in the ground.

Optimisation methods have been applied to many areas of open-pit mining, including the pit design problem, the block-sequencing problem, the equipment allocation problem, and many others (Caccetta and Giannini, 1986, 1988, 1990). In this thesis, we focus on the logistics and supply chain problem which combines the movement of material from the mines to the shipping facilities, train scheduling, inventory, and material blending.

1.3.2 Logistics and supply chain

A supply chain management involves a complex network of moving materials from the suppliers to the customers (Christopher, 2011). A typical supply chain features manufacturers, warehouses, distributors, and retailers in the network. A supply chain network may also feature a transportation system. In this case, the network allows the movement of materials through road, rail, air, or sea while considering the availability, capacity, and cost of transporting.

Logistics is the part of supply chain process that manages the flow and movement of materials or services efficiently and effectively in between two points so that the customers' demands and requirements are satisfied (Christopher, 2011). The objective of a typical logistics and supply chain problem is to move the right materials in the right quantities and in the right quality from the right source to the right destination at the right time.

Optimisation methods are often applied in solving logistics and supply chain problems. Through mathematical models, it is easier to identify the crucial features and parameters, evaluate the problem quantitatively, and make decisions and policies to achieve the optimal results. There are many optimisation problems arising in logistics and supply chain area, such as optimal fleet sizing problem, production or transportation scheduling problem, vehicle routing problem, inventory or warehousing problem, etc.

1.3.3 Transportation problem

A simple transportation problem is an example of linear programming problem that arises from the logistics and supply chain process. According to Williams (1967), the concern of this problem is to decide the most efficient routes over which the materials are distributed from points of origin to points of destination. The efficiency of the routes is usually determined by the distance, the price, or the time taken. In those cases, the most efficient route is the shortest, the cheapest, or the quickest route. An instance of a simple transportation problem involves sending goods from factories where the goods are being produced, to storage locations or customers.

Luenberger and Ye (2008) describe a general transportation problem in mathematical terms. Suppose m is the number of origins and n is the number of destinations. In this problem, we assume that the total supply at the origins is equal to the total demand at the destinations. The aim is to find the transportation schedule of sending the goods between origins and destinations such that all the requirements are satisfied and the total cost is minimised. The problem then can be expressed by the following mathematical formulation:

$$\text{Minimise} \qquad \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

$$\text{subject to} \qquad \sum_{j=1}^{n} x_{ij} = a_i \ , \ i = 1, 2, ..., m$$

$$\sum_{i=1}^{m} x_{ij} = b_j \ , \ j = 1, 2, ..., n$$

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j \ , \ \text{for all } i \text{ and } j$$

$$x_{ij} \geq 0 \ , \ \text{for all } i \text{ and } j.$$

where x_{ij} is the amount of goods sent from origin i to destination j, c_{ij} is the unit cost associated with sending the goods from origin i to destination j, a_i is the supply at origin i, and b_j is the demand at destination j.

The problem stated above is a simple transportation problem which can be solved by the simplex method. In this thesis, we model the more complex transportation scheduling in mining which also involves integer restrictions and non-linear constraints.

1.4 Thesis overview

In this thesis, we present two models that describe the optimisation of logistics planning in iron ore mining in the Pilbara region. The first model is the existing model which was previously developed by CSIRO and currently implemented by RTIO. The model is formulated as a mixed integer non-linear programming problem and solved by applying an iterative algorithm to estimate the non-linear terms.

As RTIO has been going through a major expansion for its mining operations in the Pilbara, the current iterative approach has become more inconsistent. Subsequently, there is a need to improve its mining logistics by developing a more effective and efficient logistics planning model.

The objectives of this research, therefore, are outlined as follows:

- To evaluate the existing formulation and solution approach to the optimisation tool representing the Pilbara iron ore mining value chain and identify the arising issues.

- To develop an improved model and solution approach to represent the medium to long term logistics planning in terms of a rail freight scheduling problem in iron ore mining that maximises the total profit.

- To implement both models using AIMMS modelling software, linked to the CPLEX optimisation solver; and make a thorough analysis and comparison of the results from the two models.

In this first chapter, we have introduced some preliminary background information which includes some facts about iron ore, the significance of Australian iron ore mining, the iron ore mining operation in the Pilbara, and some optimisation topics in surface mining. In regards to the Pilbara operations section, we described the iron ore mining value chain in detail. This value chain involves the mines, a rail network, the ports, and the stockpiles at both mines and ports.

Chapter 2 of this thesis reviews literature that includes substantive theories and findings in the surface mining area. We initiate the chapter by providing a review and consolidation of the literature in surface mining area in general. The aim is to look at the applications and see the need of optimisation in the field before addressing the more particular problems. We then pay particular attention to the focus of our research problem which involves the transportation scheduling problem, the blending problem, and the logistics planning problem. Some optimisation techniques associated with our problem are also looked at in this chapter. This includes recent uses, applications, and solution approaches of mixed integer linear and non-linear programming in the relevant area.

Chapter 3 describes the existing logistics model that was developed by Singh et al. (2014) in full detail. It covers the problem description and formulation, the solution approach, our own implementations of the model for model testing, and a discussion on the performance of the optimisation tool and how it can be improved.

The major contribution of this thesis to the mining industry is presented in Chapter 4. In this chapter, we develop a reformulated model which linearises the problem formulation in Chapter 3 by providing a convex relaxation of the non-linear constraints. We outline the procedure of the convex relaxation and an example, problem formulation, and implementation of test cases. The results of these test cases are discussed to conclude the chapter.

After the two models in Chapter 3 and Chapter 4 are tested on various test cases, we execute our implementations thereof using real life case studies in Chapter 5. The aim of the case studies is to analyse and compare the performance of the two different tools presented in Chapter 3 and Chapter 4. We consider two case studies; both of which are real data sets provided by RTIO. Moreover, an aggregation approach is also implemented as an attempt to reduce the solving time.

Finally, Chapter 6 concludes this thesis with a summary of our findings. We highlight the main end results that include a comparative analysis of the computational results and the effectiveness of the approaches; and state the advantages and disadvantages of both models. In addition, we provide the opportunities for future research in the relevant area.

In summary, the objectives of this thesis have been achieved in the following manner:

- The solving method of the existing model is tested and evaluated in Chapter 3, and further in the discussion of our case studies in Chapter 5. Rather than dealing with the non-linearity in the formulation, the approach in this model avoids it by completely ignoring it in the first iteration and using the solutions gained to replace the non-linear terms in the next iterations. We conclude that this approach is not reliable in utilising the grade quality restrictions and does not have any theoretical basis.

- An improved model is developed in Chapter 4. This model describes the medium to long term logistics planning and has a strong theoretical basis in dealing with the non-linearity. In this model, we utilise a convex relaxation approach to provide a reformulation of the existing model. That is, we remove the non-linear constraints in the problem and add the convex and concave estimating functions of those constraints. A global solution of the reformulated problem can be obtained by running a MILP solver.

- Both models are implemented using AIMMS modelling software, linked to the CPLEX solver, to solve test cases in Chapters 3 and 4, and full-sized case studies in Chapter 5. Our implementation results show that the iterations for the existing model do not lead to convergence of solutions, but rather inconsistencies. It also does not reduce the grade deviation cost as expected. The new model that we developed generates solutions with an improvement of 24–30% of the grade deviation costs. As an exact method, of course, the improvement comes with a significant increase in computational time. An aggregation technique is developed to decrease the problem size greatly. The result of this approach is that we are able to generate good quality solutions in a suitable time frame.

Chapter 2

Literature review

There is an almost limitless number of literature and sources on the applications and optimisation approaches of mining. In this chapter, we briefly review some key literature on the mining problems in general before we look at some specific problems that are relevant to our research problem. We will focus on the use of mixed integer linear and mixed integer non-linear programming to model and solve such problems.

2.1 Introduction

Based on the distance of the extracted material from the surface, the mining process can be classified into two different methods; that is, surface and underground mining. A combination of both methods is also possible when the extraction depth increases.

Optimisation, or operations research, has been widely applied in mine planning for at least the past 50 years. There are many areas of both surface and underground mining in which optimisation has been applied. Early work on potential areas of applications in mining are discussed in Topuz and Duan (1989). Newman et al. (2010) provide a more recent and comprehensive literature review on the applications of operations research in mine planning problems. They categorise their problems in terms of the decision levels (strategic, tactical, or operational) in both surface and underground mining.

Kozan and Liu (2011) did similar work on operations research methodologies in both surface and underground mining. The problems discussed are classified into four categories, namely mine design, mine production, mine transportation, and mine evaluation.

In this chapter, we review the literature on the operational research methods in surface or open-pit mining. We will briefly discuss some important mining problems in general and then focus on the transportation scheduling, material blending, and logistics planning problems. These topics are consolidated in Section 2.2. Subsequently in Section 2.3, we discuss some optimisation techniques which focus on the applications of mixed integer linear and non-linear programming in solving some problems in mining industries. Finally, in Section 2.4, we summarise and discuss our findings from the literature review.

2.2 Optimisation in surface mining

Surface mining refers to the mining methods used to excavate minerals (usually hard rock or metal ore) that are found close to the surface of the Earth (Kennedy, 1990). In surface mining, the overlying rock and waste, often called overburden, are first removed before depositing the minerals. Iron ore, coal, and gold are some examples of minerals that often utilise a surface mining method when the minerals are found not too deep below the surface.

Optimisation methods have been applied to solve various problems that arise in the surface mining area. Some of these applications include the ultimate pit limit problem, the production scheduling problem, the equipment allocation problem, the blending problem, and many others (see Caccetta and Giannini, 1986, 1988, 1990). More of these applications in the mining industry are discussed in Weiss (1979)

2.2.1 Production scheduling problem

The optimum ultimate pit limit problem, also called the optimum open-pit mine design problem (see Lerchs and Grossmann, 1965) is fundamental in the mine planning process. This problem aims to determine the optimal contour which is the result of extracting the volume of material that generates a maximum total profit whilst considering the operational constraints. The ultimate pit limit problem is crucial in the production scheduling problem in mining.

The production scheduling, or the block sequencing problem, aims to determine the sequence of blocks to be removed from the mine such that the total profit is maximised and various constraints are satisfied. This problem was initiated by Johnson (1968) who proposed an LP model to optimise the timing of mineral extraction.

Fytas et al. (1993) present a computer package that generates alternative production scheduling strategies. Optimisation models and algorithms to solve the long-term open-pit production scheduling problem are reviewed by Osanloo et al. (2008). Gershon (1983b) addresses the issues and computational difficulties arising in the open-pit production scheduling problem. In addressing this problem, Caccetta and Hill (2003) propose an application of the branch and cut method.

Ramazan and Dimitrakopoulos (2007) and Boland et al. (2008) extend the open-pit production scheduling problem to a stochastic case with uncertainty in the geological properties of the material. Kumral (2011) also discusses a mine production scheduling problem which takes into account the fluctuations in geo-metallurgical variability.

In regards to real applications, Rehman and Asad (2010) develop a model to describe a production scheduling tool for a cement quarry. The model is developed for short-term, rather than long-term. An application of production scheduling problem in phosphate mining is considered by Busnach et al. (1985).

2.2.2 Transportation problem

There are at least four different modes that are commonly used to distribute goods in the logistics and supply chain. They are road (e.g. cars, trucks), rail (e.g. trains), sea (e.g. ships), and air (e.g. planes). Intermodal freight transport is an operations research area in which at least two modes of transportation are used to move the goods without a handling operation in between.

Macharis and Bontekoning (2004) did a review of intermodal freight transport research which focuses on inland transportation. Caris et al. (2008) provide an overview of a planning decision problem in intermodal freight transport which involves drayage operators, terminal operators, network operators, and intermodal operators. Li and Tayur (2005) integrate both pricing and operations planning aspects for their optimisation model in the context of intermodal transportation.

In this thesis, we pay more particular attention to the rail transportation mode. The most common problems in rail transportation are train routing and scheduling problems. A survey on optimisation models for both of these problems was done thoroughly by Cordeau et al. (1998). Although it is possible to integrate both problems into a single optimisation problem (Morlok et al., 1970), the train routing and scheduling problems are commonly treated separately.

The train routing problem produces an operation plan in terms of transportation routes, the number of trains, and their frequency (Eidenbenz et al., 2003) . The train scheduling problem generates a plan that specifies the timetable of the planned trains. Higgins et al. (1996) present an optimisation model to schedule trains on single line railroads. Extended works on train scheduling may consider various constraints such as the elasticity of the demand (Kuo et al., 2010). Newman and Yano (2000) note some comparisons between centralised and decentralised train scheduling methods for an intermodal network.

In terms of ocean transportation, a survey on cargo ship routing and scheduling problems are reported (Ronen, 1983; Christiansen et al., 2004). Mehrez et al. (1995) provide a good example of an ocean cargo shipping problem that includes shipping, unloading, and warehousing.

2.2.3 Blending problem

The quality of some mining products varies depending on various properties in the products. Iron ore, for instance, contains different components such as iron (Fe), silicon dioxide (SiO_2), aluminium oxide (Al_2O_3), phosphorus (P), etc. Iron ore buyers usually have specific demands on the target grade quality to be achieved for each component in the product. This target grade quality is typically 60% Fe, 4% SiO_2, 2% Al_2O_3, and 0.1% P (Everett, 2007). In this case, the blending process plays an important role in the whole operation to maintain the desired grade quality.

The blending problem involves mixing products of different compositions and grades in order to attain a reliable or required grades of products. The pooling problem is a type of blending problem, normally in the petroleum industry, where the products are sent to and blended in the intermediate pools to satisfy given quality requirements. Many blending problems are formulated as MILP. Nonetheless, according to Foulds et al. (1992), non-linearity seems to be inevitable in the formulation of a classic pooling problem due to the pooling restrictions. This is also shown in the earlier studies done by Haverly (1978, 1979, 1980).

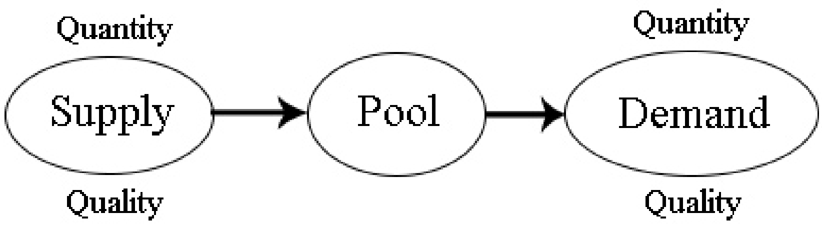

Figure 2.1: Simple network flow diagram for the pooling problem

The pooling problem is analyzed at length by Greenberg (1995). It can be described as a network flow problem in which both the quantities and the attribute qualities of products are considered in each flow. A simple network flow diagram for the pooling problem involves supplies at sources, intermediate pools, and demands at destinations, described in Figure 2.1.

Suppose we have three different sets of nodes, namely the sources, the pools, and the destinations. The network flow goes from a source to a pool and ends at a destination. We may assume that each intermediate pool must be connected with at least two sources and two destinations (Audet et al., 2004). If this condition is not satisfied, the intermediate pool becomes redundant as it is possible to merge the pool into either the source or destination.

We let S_i be the supply for source i and D_k to be the demand for destination k. Let x_{ij} be the flow from source i to pool j and y_{jk} be the flow from pool j to destination k. Figure 2.2 below is an instance of a network flow diagram for the pooling problem with 3 sources, 2 pools, and 2 destinations:

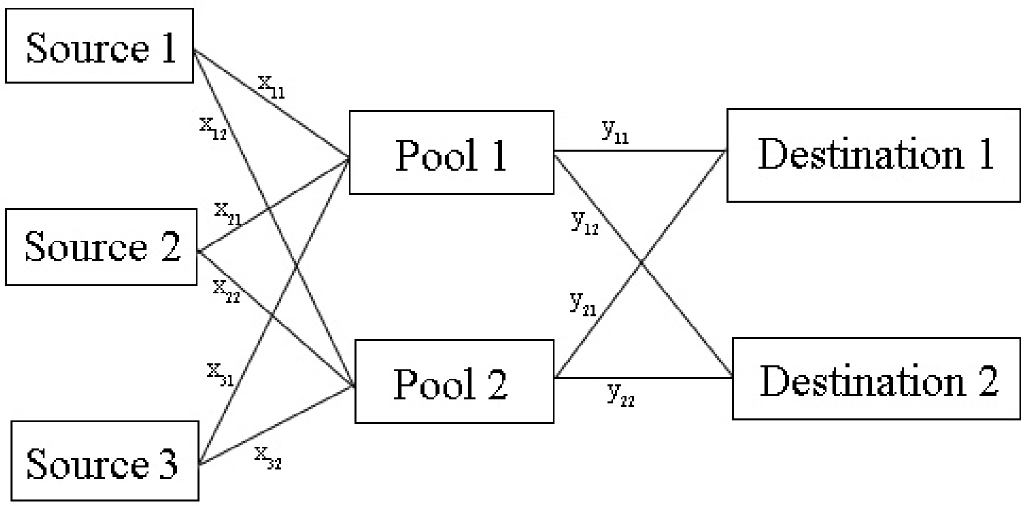

Figure 2.2: Network flow diagram for the pooling problem with 3 sources, 2 pools, and 2 destinations

To meet the supply and demand requirements, we have the following constraints:

$$\sum_j x_{ij} \leq S_i \text{ , for all } i$$

$$\sum_j y_{jk} = D_k \text{ , for all } k$$

As each pool assumes a flow balance, we also have:

$$\sum_i x_{ij} = \sum_k y_{jk} \text{ , for all } j.$$

The constraints above are the general constraints in a simple network flow problem. In the pooling problem, however, we also consider the attribute quality of the products.

Let a_{iq} be the quality of attribute q in the product at source i, b_{jq} be the quality of attribute q in the product at pool j, and c_{kq} be the quality of attribute q in the product at source k. Then we define the attribute quality b_{jq} as follows:

$$b_{jq} = \frac{\sum_i x_{ij} a_{iq}}{\sum_i x_{ij}} \text{ , for all } j, q$$

when the flow into the pool exists, that is, $\sum_i x_{ij} \geq 0$. Otherwise, b_{jq} is undefined. Similarly, we define the attribute quality c_{kq} by:

$$c_{kq} = \frac{\sum_j y_{jk} b_{jq}}{\sum_j y_{jk}} \text{ , for all } k, q$$

when the flow from the pool exists, that is, $\sum_j y_{jk} \geq 0$. Otherwise, c_{kq} is undefined.

It is evident that the equations for the quality constraints are bilinear. If the flow variables are restricted to be integers, we have a mixed-integer non-linear programming problem. In addition to the above constraints, the problem also requires the attribute quality of the products at the destinations to meet the target demand quality. Suppose c^L_{kq} and c^U_{kq} are the lower and upper bounds for the target demand quality. Thus, we have the bounds for c_{kq}:

$$c^L_{kq} \leq c_{kq} \leq c^U_{kq} \text{ , for all } k, q$$

.

The typical objective of the pooling problem is to minimise total costs. We define V_{ij} as the cost of moving goods from source i to pool j and W_{jk} as the cost of moving goods from pool j to destination k. Therefore the objective function is to minimise a linear function of x_{ij} and y_{jk}. In summary, the problem is formulated as follows:

$$\begin{aligned}
\text{Minimise} \quad & \sum_i \sum_j V_{ij} x_{ij} + \sum_j \sum_k W_{jk} y_{jk} \\
\text{subject to} \quad & \sum_j x_{ij} \leq S_i, && \text{for all } i \\
& \sum_j y_{jk} = D_k, && \text{for all } k \\
& \sum_i x_{ij} = \sum_k y_{jk}, && \text{for all } j \\
& b_{jq} = \frac{\sum_i x_{ij} a_{iq}}{\sum_i x_{ij}}, && \text{for all } j, q \\
& c_{kq} = \frac{\sum_j y_{jk} b_{jq}}{\sum_j y_{jk}}, && \text{for all } k, q \\
& c_{kq}^L \leq c_{kq} \leq c_{kq}^U, && \text{for all } k, q.
\end{aligned}$$

Audet et al. (2004) model the pooling problem through bilinear and nonconvex quadratic programming. They investigate the possibility of using a branch and cut algorithm for solving the nonconvex quadratically constrained optimisation problem (Audet et al., 2000). In a more recent work, Gupte et al. (2013) present new relaxations to the pooling problem and discretization methods to solve the general bilinear programming problem. Ulstein et al. (2007) formulate a model to describe the tactical planning of petroleum production for a Norwegian oil and gas company. This problem also involves a quality control of gas through blending and processing.

As we mentioned before, the problem of product blending is not limited to the petroleum industry. Coal, gold, and iron ore are some instances of ores that need blending of their multiple components as part of processing. The optimisation problem that includes blending coal as part of the optimal plan is presented by Liu and Sherali (2000). As this tool was developed for an electric utility company, it must consider the supply and quality of coal and the demand of electricity to find the optimal shipping and blending decisions of coal. Shih (1997) did a similar planning model of fuel coal imports with constrained power plants and harbours. Sandeman et al. (2010) provide a case study in gold mining with blending requirements in which they integrate optimisation within simulation models.

With regards to blending iron ore, Everett (2001) presents various analytical algorithms and simulation tools to improve the shipped material quality in iron ore production scheduling. This work considers different stages in the mining value chain to support the production scheduling decisions. Everett

(2007) provides a computer based tool to aid the product quality management in iron ore mining.

Other than ore mining and oil and gas, the blending process is also crucial to other industries such as agriculture (Fishman and McInnes, 2005), food processing (Kilic et al., 2013), etc.

2.2.4 Logistics planning problem

A logistics planning problem involves the movement of goods, typically from the suppliers to the customers. Logistics planning may integrate production scheduling, transportation, or blending problems into one model.

Mendez et al. (2006) present an optimisation model to describe the logistics planning of oil refinery operations that includes scheduling and blending. A model for planning and scheduling crude oil operations is also presented in Karuppiah et al. (2008). Wenkai et al. (2002) model the short-term scheduling problem for crude oil production that incorporates unloading, storing, and processing the crude oil.

Fröhling et al. (2010) develop supply chain planning systems for the integrated transportion and blending for multiple recycling plants. To address the blending requirements, their model derives linear input-output functions by multiple linear regression analysis.

Bilgen and Ozkarahan (2007) address a logistics and supply chain problem for bulk grain that involves shipping and blending. A logistics planning model for iron ore which also involves rail transportation and blending of products is developed in Garcıa-Flores et al. (2011), discussed more thoroughly in Singh et al. (2014). Unlike the latter, the former formulate their blending requirements linearly. We will present the model in Singh et al. (2014) in more detail along with our implementations in the forthcoming chapter.

2.3 Optimisation techniques

Many mining problems are formulated as mathematical programs. In this section, we review the use of mixed-integer linear and non-linear programming in mining applications. Mixed-integer programs refer to mathematical programming problems that involve both discrete (integer) and continuous variables in the formulations. If the objective function and all constraints are linear, the problem is a mixed-integer linear programming (MILP) problem. If the problem includes a non-linearity, it is called a mixed-integer non-linear programming (MINLP) problem.

2.3.1 Mixed-integer linear program

The existence of the integer property in some of the variables makes MILP problems more difficult to solve than the normal LP problems. Regardless, the MILP type of mathematical modelling is considered significant in formulating many optimisation problems in mining. The branch and bound method is a general method in solving the IP or MIP problem (Land and Doig, 2010; Ivanchev et al., 1976). This method can only be used to solve the linear case of MIP (Taha, 1975), although the modified method can be applied to solve some non-linear cases.

MILP is often used for solving long-term production scheduling in open pit mining (Caccetta, 2007; Gershon, 1983a). Many solution strategies were developed to solve such problem. A lagrangian relaxation was mentioned in Caccetta et al. (1998). Gershon (1987) uses two heuristic approaches to approximate the results of the optimisation problem. A heuristic method, called a sliding time window heuristic, was introduced by Cullenbine et al. (2011) to solve a standard block sequencing problem. Caccetta and Hill (2003) propose a branch and cut method which generates a good set of solutions to the problem.

While the number of binary variables becomes the major problem when solving the problem, Ramazan and Dimitrakopoulos (2004) propose a method to reduce the binary variables in the formulation. The reduction, however, is usually not enough for very large open-pit mines. Ramazan (2007) then develops a new algorithm called a fundamental tree algorithm to further reduce the number of integers and constraints in the MILP formulation.

2.3.2 Mixed-integer non-linear program

If the MIP formulation involves a non-linearity in the objective function and/or the constraints, the problem is classified as MINLP. The use of MINLP in mining applications is also abundant. Several surveys of literature on the applications and algorithms for MINLP problems are reported (Bussieck and Pruessner, 2003; Grossmann, 2002; Leyffer et al., 2009).

Depending on the complexity of the non-linear terms, a MINLP can be addressed as a MILP problem by replacing the non-linear terms with individual variables. Lee et al. (1996) solve the crude oil scheduling and inventory management problem with integers and bilinear equations involved in the model. However, they maintain the problem as MILP by replacing the bilinear terms with individual component flows.

Mendez et al. (2006) approximate their complex MINLP problem by a sequence of MILP formulations. This method is often called the successive linear programming (SLP), or sequential linear programming, method. Wenkai et al. (2002) model their short-term scheduling problem for crude oil production in MINLP formulation. Rather than solving the MINLP directly, they also propose a solution algorithm that solves two MILPs and a NLP model iteratively.

2.3.3 Non-convex MINLP

Solving MINLP problems requires a lot of work due to the combination of the integrality condition and the non-linearity. Despite its challenges, there are still methods available for solving convex MINLP problems, that is, MINLP problems in which the functions involved in the objective and constraints are all convex. The outer-approximation algorithm, for example, was developed to solve such problems (Duran and Grossmann, 1986; Fletcher and Leyffer, 1994).

In the event where non-convexity is involved, finding a global solution is much more difficult. Standard MINLP methods may only lead to sub-optimal solutions or to no solution at all. Furthermore, some methods have been developed mainly for academic purposes but are not computable in real implementations.

The surveys on MINLP problems that have been reported (Bussieck and Pruessner, 2003; Grossmann, 2002; Leyffer et al., 2009) cover both convex and non-convex cases of MINLP. Adams and Sherali (1993) address more specific MINLP problems classified as mixed-integer bilinear programming problems. The integer variables in this paper are restricted to be binary valued. Burer and Letchford (2012) have done a good survey that focuses on non-convex MINLP problems.

The pooling problem is well known to be an application of the non-convex quadratic problem (Audet et al., 2004; Misener and Floudas, 2009). Although a standard pooling problem is generally only a bilinear program, some complex pooling problems are expressed as mixed-integer bilinear programs. Another application of non-convex MINLP is the transportation network design problem (Fügenschuh et al., 2010). The aim of this problem is to design transportation routes such that the number of cars or trains and the travel distances are minimised.

McCormick (1976) presents a general method for obtaining a global solution to a factorable non-convex programming problem. In this paper, McCormick outlines the details on how to generate underestimating convex and overestimating concave functions for factorable functions, including bilinear terms. These underestimators and overestimators are provided to relax the non-convex MINLP formulations. We will show the procedure in Section 4.2.

Karuppiah et al. (2008) present an outer approximation algorithm to their non-convex MINLP model. They use the concept of convex relaxation based on the work done by McCormick to obtain a rigorous lower bound on the global optimal solution. The spatial decomposition method is used to relax the MINLP problem in order to obtain the upper bound. The final solution is attained once the gap between lower and upper bound falls within a reasonable tolerance.

2.3.4 Heuristic methods

Heuristic methods are often developed to reduce the complexity of a mathematical programming problem, especially when the size of the problem is too large and/or the formulation is too complicated. Heuristics methods are commonly implemented as an attempt to shorten the solving time. In this section, we provide examples of heuristics methods used in Singh et al. (2014). These heuristic methods are useful when the size of the planning periods are large. The methods use sequences of iterations over a number of sub-intervals and are based on the sliding time window heuristic introduced by Cullenbine et al. (2011).

Heuristic 1

Each iteration in the first heuristic method solves a full problem with a complete time horizon. In the first iteration, the integer restrictions are only considered in the first I periods. In the second iteration, the integer restrictions are also considered in periods $[I, 2I]$ and we use the solutions from the first iteration for integer variables in periods $[0, I]$ with some tolerance to ensure feasibility. In general, in the ith iteration, we bring back the integer constraints for periods $[(i-1)I, iI]$ and use the previous iteration's solution with tolerance. This step continues until the last period is reached.

Heuristic 2

Unlike the first heuristic, in every iteration in the second heuristic, we solve the model for a limited time horizon with overlapping time horizons between consecutive iterations. In the first iteration, we solve the problem only over the first J periods. In the second iteration, we solve the model over the first $2J - J'$ periods where the solutions from the first iteration within tolerance are used for the integer constraints in the first $J - J'$ periods. In general, in the jth iteration, we solve the model over intervals $[0, J + (j-1)(J - J')]$ where the solutions from the previous iteration within tolerance are used for the integer variables for the first $(j-1)(J-J')$ periods. The additional tolerance is again used to ensure feasibility. This step continues until the last period is reached.

2.4 Discussion

It has been shown that there is a wide variety of problems that use the applications of optimisation in surface mining. Finding the optimum ultimate pit limit plays a major role in mine production scheduling. Optimisation is also used in routing and scheduling transportation system in order to distribute goods or products effectively and efficiently. Blending requirements must be considered to maintain the desired grade quality of products.

In this chapter, we have reported several medium-term planning problems in mining. Although a lot of research has focused on the problems of mine planning and blending separately, only a few have tried to integrate the two problems. Bilgen and Ozkarahan (2007) report a model to determine shipping schedules for the export of grain blends with blending requirements as one of their constraints. Singh et al. (2014) develop a medium-term rail scheduling for RTIO in iron ore mining which will be discussed at length in Chapter 3 of this thesis.

In this chapter, we have also indicated the use of MILP and MINLP in optimisation techniques in mining. We have seen that solving MINLP is more challenging than MILP. The constraints representing the blending requirements in Bilgen and Ozkarahan (2007) involve only two components and are linear. Their whole model is then formulated as MILP. On the contrary, the blending constraints in Singh et al. (2014) is non-linear, thus making the problem MINLP. They apply the successive linear programming approach to tackle the non-linearity (Mendez et al., 2006). To reduce the problem size, they also introduce two heuristic methods based on the sliding time window heuristic (Cullenbine et al., 2011).

In summary, we indicate the issues that arise from the relevant topic, namely:

- the importance of the blending requirement in logistics planning for minerals like iron ore to maintain the required grade quality;
- too little research has been focused on a logistics planning model that integrates transportation scheduling and blending requirements in one optimisation tool;
- evaluation of the solutions and the reliability of the effectiveness of the SLP method to solve the MINLP problem; and

- the need to implement the available methods to solve the challenging MINLP problem to obtain better results.

Chapter 3

Successive Linear Programming for Logistics Planning

A good logistics planning model is crucial in order to make good decisions in mining operations. Without one, the efficiency and optimality of the mining logistics and supply chain would be much harder to achieve. Before any optimisation tool was developed, RTIO performed a computer-based calculation using Excel spreadsheets for their mine planning process. The procedure required a lot of time and efforts and the results obtained from it were neither reliable nor flexible, thus impeding the mine planning process. A few years ago, CSIRO came up with a model which describes a logistics planning in iron ore mining for the company (Singh et al., 2014). Although there have been slight modifications since it was first developed due to requirement changes, the model is still primarily used in RTIO's current operation. In this chapter, we present the tool that RTIO currently implements for optimising their mining logistics. We model and implement the tool using AIMMS and present the computational results on a range of test cases.

3.1 Introduction

A rail scheduling model for medium to long term planning has been developed by CSIRO (Garcıa-Flores et al., 2011; Singh et al., 2014) for RTIO. This optimisation tool aims to maximise the total throughput of iron ore and manage the logistics and supply chain of Pilbara mining operations for a time horizon of up to two years. The tool allocates trains to mines and manages the mine-to-port value chain while considering various operational constraints inbetween. This model ensures that both quantity and grade quality targets of the shipped products are satisfied. Incentives are incorporated into the objective function to encourage a higher number of trains, hence undertaking scenario analysis.

The problem is categorised as a mixed integer non-linear programming problem due to the existence of both continuous and integer variables, as well as non-linear constraints. One of its output variables, namely the number of trains sent to each mine, is restricted to be integer, while other decision variables remain continuous. Some binary variables are also introduced in order to linearise maximisation and minimisation functions in the constraints. The non-linearity appears in the grade quality requirement constraints. The formulation for these constraints involve bilinear terms for the grade calculations.

As part of the optimisation tool, CSIRO developed a solution approach which involves a multi-stage algorithm and is derived from the successive linear programming (SLP) method. In the first stage, the integer restrictions are relaxed and the grade constraints are omitted, making the problem a simple linear programming problem. In the following stage, the grade constraints are brought back, but the solutions from the previous iteration are used the estimate the value of the decision variables in the non-linear terms, hence linearising the constraints. This step is repeated for a number of iterations until a reasonably good scheduling plan is attained. The tool has been implemented and solely relied on by RTIO for their current operation. It has produced better logistics plans than the traditional Excel spreadsheet based approach.

In Section 3.2, we show the procedure to linearise maximisation and minimisation functions before we describe the problem. This linearisation procedure will be needed when formulating the iron ore grade calculations. We then outline the full description of the problem in Section 3.3. The full formulation of the model is presented in Section 3.4. This formulation is a complete and extended version of the model developed for the current operation. We firstly list the assumptions (Section 3.4.1), the set notations (Section 3.4.2), then outline the objective function (Section 3.4.3), followed by the operational constraints (Section 3.4.4) and grade constraints (Section 3.4.5). The summary of notations is outlined in Section 3.4.6 and the summary of formulation in Section 3.4.7.

We describe the solution method which involves a multi-stage algorithm in Section 3.5. For validation purposes, we run test cases of problems with small size periods, namely 5 and 11 periods. The results will be outlined and discussed in Section 3.6. We implement all cases in AIMMS linked to CPLEX 12.6.3 solver. Real case studies which involve a complete logistics plan in iron ore mining will be considered in Chapter 5. Finally, we provide the conclusion of this chapter in Section 3.7.

3.2 Maximum and minimum functions

Before we present the problem description and formulation, it is necessary to look at the linearisation procedure of maximum and minimum functions. A maximum function of continuous variables returns the highest-valued variable among the given set of variables. Similarly, a minimum function of continuous variables returns the lowest-valued variable in the given set of variables. Both maximum and minimum functions are non-linear. It is possible, however, to convert almost any maximum or minimum function of single continuous variables into equivalent linear formulations by introducing some binary variables.

Let y be a maximum function of continuous variables $x_1, x_2, x_3, \ldots x_n$, that is,

$$y = \max\{x_1, x_2, x_3, \ldots, x_n\}$$

It is assumed that the lower and upper bounds for $x_1, x_2, x_3, \ldots x_n$ are known. Thus we have:

$$x_i^L \leq x_i \leq x_i^U, \quad \text{for } i = 1, 2, 3, \ldots, n$$

Firstly, we introduce binary variables $b_1, b_2, b_3, \ldots b_n$. The variable b_i will return the value 1 if x_i is the maximum value and 0 otherwise. Let $x^U_{\max}$ be the highest value among the upper bounds, that is,

$$x^U_{\max} = \max\{x_1^U, x_2^U, x_3^U, \ldots, x_n^U\}$$

The generalisation of the MIP formulation to the maximum function is given by:

$$\begin{aligned} x_i^L \leq x_i &\leq x_i^U, && \text{for } i = 1, 2, 3, \ldots, n \\ y &\geq x_i, && \text{for } i = 1, 2, 3, \ldots, n \\ y &\leq x_i + (x^U_{\max} - x_i^L)(1 - b_i), && \text{for } i = 1, 2, 3, \ldots, n \\ \sum_i b_i &= 1 \end{aligned}$$

Now we let z to be a minimum function of continuous variables $x_1, x_2, x_3, \ldots x_n$, that is,

$$z = \min\{x_1, x_2, x_3, \ldots, x_n\}$$

and we have:

$$x^L_{\min} = \min\{x_1^L, x_2^L, x_3^L, \ldots, x_n^L\}$$

We introduce binary variables c_i such that c_i is 1 if x_i is the minimum value and 0 otherwise for all $i = 1, 2, 3, \ldots, n$. The generalisation of the MIP formulation to the minimum function is then given by:

$$
\begin{aligned}
x_i^L \leq x_i \leq x_i^U, & \qquad \text{for } i = 1, 2, 3, \ldots, n \\
z \leq x_i, & \qquad \text{for } i = 1, 2, 3, \ldots, n \\
z \geq x_i - (x_i^U - x_{\min}^L)(1 - c_i), & \qquad \text{for } i = 1, 2, 3, \ldots, n \\
\sum_i c_i = 1 &
\end{aligned}
$$

This MIP generalisation procedure is needed in our model as we will deal with some maximum and minimum functions in the iron ore grade calculations.

3.3 Problem description

This model aims to produce an optimal medium to long term plan for allocating trains to mines and maximising total throughput of iron ore while satisfying all the capacity, contractual obligation, and grade quality constraints. In this section, we describe the main features of the problem.

RTIO currently services 15 mines and 4 shipping facilities across 2 ports in their Pilbara mining operations. It owns the largest private rail network in Australia with a total of 191 locomotives and 11,500 wagons. Figure 3.1 shows a single-network version of RTIO's Pilbara operation process from mine to port.

Stockpiling plays an important role in the production line of iron ore mining operations. After the extraction process, the iron ore is stored in the stockpiles before it gets loaded onto trains. Likewise, the stockpiles at ports serve as storage before the product is loaded onto the ships. In addition, a blending process takes place in the stockpiles to meet grade requirements.

Most of the mines and ports have two different types of stockpiles, namely live and bulk stockpiles. Live stockpiles are used as the main production line, whereas bulk stockpiles serve as buffering and storage. The live stockpile levels have to meet the minimum and maximum capacity which are formulated by both soft and hard constraints in the model. The soft constraints can be violated, but penalties apply. The hard constraints of the stockpile capacity are described as yard limits and cannot be violated. Only maximum capacity hard constraints apply at the bulk stockpiles. The inloaders and outloaders are used to transfer the products from/to bulk stockpiles. However, transferring material between live and bulk stockpiles is not preferable as it will incur bulk handling costs. There are also maximum capacity limits for inloaders and outloaders. We have book keeping constraints to keep track of the inventory levels in the stockpiles. To sum up, the constraints and penalties involved within the stockpiling process are listed below:

- desirable capacities of minimum and maximum stockpile levels with penalties if violated,
- maximum yard limit capacities,
- maximum inloader and outloader capacities,
- bulk handling penalties, and
- book keeping constraints.

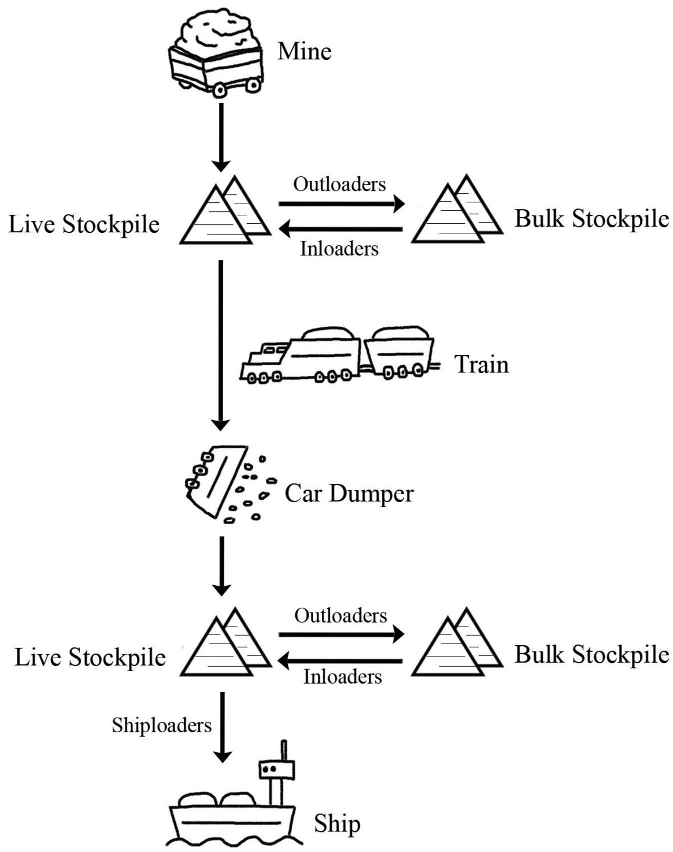

Figure 3.1: RTIO's Pilbara operations from mine to port

The products are transferred from mines to ports by trains. Maximum capacity and cycle time limits are expressed as hard and soft constraints respectively. The number of trains cannot exceed the available trains in each mine and region. Furthermore, there is also a maximum capacity for each fleet. The cycle time of a train trip is defined as the time it takes for a train to complete a port-mine-port journey. This cycle time also includes loading and unloading the iron ore. Violating the maximum hours of cycle time will result in penalties. For some of the mines, the number of trains must comply with the contractual obligations with the joint ventures which are expressed by annual quotas. Therefore, the constraints and penalties involved within the rail operation are as follows:

- maximum number of trains in each mine,
- maximum number of trains in each region,
- maximum number of trains that belong to each fleet,
- maximum hours of available cycle time with penalties if violated, and
- maximum number of trains for mines restricted by the joint ventures.

Car dumpers are used to dump the material from the trains onto conveyors and to port stockpiles. In the formulation, we include the cost of using the car dumping facilities. As some car dumpers are preferred to service specific types of products, the costs vary depending on the car dumper being used and the product type. There are capacity limits for the number of trains that can be serviced by the car dumpers. There is also an additional capacity limit for total number of trains serviced by car dumping facilities in the Cape Lambert-Western Creek region. The cost and constraints involving the car dumping facilities at ports are as follows:

- cost of dumping material,
- maximum capacity of car dumpers, and
- maximum capacity of car dumpers in Cape Lambert-Western Creek region.

The blending process takes place at the stockpiles. We include the calculations for the product grade for each component at each stockpile and train in our formulation to maintain the desired quality. The blending process at port stockpiles, in particular, is essential as it determines the final product quality. We penalise the slack and excess variables when the target composition is off target within tolerance. Hence the following constraints and penalty regarding the blending requirement are added to the formulation:

- grade calculations of product component at each stockpile and train, and
- grade deviation constraints with penalties if violated.

The final product is then transferred to the ships by the shiploaders. The total amount of materials loaded onto a ship cannot be greater than the shiploader capacity. Therefore this additional constraint in regards to shipping is added:

- maximum shiploader capacities.

Another important operation at this stage is the lump re-screening process, also called the return fines process. When the lump material is loaded onto the ships, we return a proportion that is undersize back to the stockpiles. This return fines product is mixed with the fines material and thus will affect the book keeping and the grade composition of the fines product.

This model was designed for time horizons varying from two weeks to two years. Each period represents a few days up to a week. In this thesis, we run the model for time horizons of up to one year.

3.4 Problem formulation

Part of the problem formulation was presented in Garcıa-Flores et al. (2011) and Singh et al. (2014). The literature, however, did not fully reflect RTIO's current operation as some details of the formulation were omitted. In this section, we outline the extended version of the problem formulation based on RTIO's current operation.

The objective of the problem is to maximise the total profit which is expressed by total revenue less total penalties. The revenue and all the cost functions appear linearly. The problem is subject to operational capacity constraints, contractual obligations, and grade quality requirements. The grade quality constraints involve some non-linear terms. In addition, both continuous and integer decision variables are involved, hence the problem is a mixed integer non-linear program.

3.4.1 Assumptions

Although the model in Singh et al. (2014) describes a real case of mining network, there are some assumptions and conditions that we need to apply to uncomplicate the problem. The assumptions are listed in the following:

Mine production amount is determined

The exploration process is scheduled separately and the amount of production from each mine is estimated.

No rail operational cost

To make the best use of the available trains, we apply no operational cost for running the trains. Nevertheless, we consider penalties for exceeding the total allowed number of trains.

Estimated lump screening proportion

The parameters describing the proportion of lump products to be removed and added to the fines stockpile at each port in each period are estimated.

Uniformity of grades

At each stockpile in the mines and ports, we assume that the grade composition of the product is uniform across the stockpile, that is, partial mixing that occurs after transferring new materials is ignored.

3.4.2 Sets

Let the set of all mines be M and the set of all ports be R. Some of the mines must comply with the joint ventures obligations. We use M^{JV} to describe the set of such mines where $M^{JV} \subseteq M$. Let the set of all train fleets be F and the set of all regions be G. Further, we let M_f and $M_g \subset M$ denote the set of all mines serviced by the fleet $f \in F$ and belonging to region $g \in G$ respectively. As we have two different regimes for loading the trains, we define M^{FF} to be the set of mines whose regime is FIFO (first in first out) and M^{LF} to be the set of mines whose regime is LIFO (last in first out). It is clear that M^{FF} and $M^{LF} \subset M$. The definition of these regimes will be given in the coming section.

Each mine and port produces different types of iron ore. We let P be the set of all mined products and S be the set of all shipped products. We also let P_m be the set of all mined products produced in mine $m \in M$, S_r be the set of all shipped products produced in port $r \in R$, and S_{mp} be the set of all shipped products for mine $m \in M$ and product $p \in P$. Clearly, $P_m \subseteq P$ and $S_r, S_{mp} \subseteq S$. It is also required to define S^L and S^F to be the set of all lump and fines shipped products respectively, where $S^L, S^F \subset S$. As the grade composition of iron ore is crucial in the problem, we also consider the various components in the iron ore products. We let C to be the set of all different components in a product.

The logistics operation process involves the use of car dumpers at the ports. Suppose D be the set of all car dumpers. We let D_{mrp} be the set of all car dumpers at port $r \in R$ receiving mined product $p \in P$ from mine $m \in M$ where $D_{mrp} \subset D$. The car dumpers that belong to the Cape Lambert-Western Creek region have special maximum capacity. Therefore, it is necessary to let D^{WC} be the set of car dumpers that serve the Cape Lambert-Western Creek region where $D^{WC} \subseteq D$. Finally, T is a set of integers that represent the planning periods in the model. The length of the time interval for each period is typically one week time horizon.

3.4.3 Objective function

The objective is to maximise the total revenue while minimising various penalties from not complying with the operational constraints and the product quality specification (Singh et al., 2014). Thus we consider the objective function as maximising the total profit which is total revenue less total penalties plus an incentive. All penalties are modelled as soft constraints in the formulation. The motivation of adding an incentive is to encourage a higher number of trains in the solutions. Therefore the objective function is defined by:

Total profit = Total revenue - Cost of live stockpile violations - Cost of bulk stockpile violations - Cost of bulk handling - Cost of cycle time violations - Cost of dumping materials - Cost of grade non-compliance + Incentive.

Total revenue

The mining operation in Pilbara ships different types of products, each of which has different sale price. The total revenue is then given by:

$$\sum_{s \in S_r} SP_s \sum_{t \in T} \sum_{r \in R} z_{rst} (1+I)^{1-t} \tag{3.1}$$

where SP_s is the sale price dedicated to shipped product s and z_{rst} is the amount of product s shipped from port r in period t. We discount the revenue to the present using a discount factor I.

Cost of live stockpile violations

After the extraction process in a mine, the ore product is stored at a live stockpile. The same type of stockpiling process also applies in the ports after transferring iron ore from the mines. Each live stockpile at the mines and ports has minimum and maximum storage capacities which are described as both soft and hard constraints. For the soft capacity constraints, penalties will occur if the closing live stockpile level does not fall within the target limits. The cost of live stockpile limit violations is then formulated as:

$$\sum_{m\in M}\sum_{p\in P_m}\left[MS^{LE}_{mp}\sum_{t\in T}\alpha^{LE}_{mpt} + MS^{LS}_{mp}\sum_{t\in T}\alpha^{LS}_{mpt}\right] + \sum_{r\in R}\sum_{s\in S_r}\left[PS^{LE}_{rs}\sum_{t\in T}\beta^{LE}_{rst} + PS^{LS}_{rs}\sum_{t\in T}\beta^{LS}_{rst}\right] \quad (3.2)$$

where MS^{LE}_{mp} and MS^{LS}_{mp} represent the penalties when the respective maximum and minimum stockpile limits are violated at the live stockpiles in mine m for product p, PS^{LE}_{rs} and PS^{LS}_{rs} are similar penalties for violation at the live stockpiles at port r and of shipped product s, α^{LE}_{mpt} and α^{LS}_{mpt} are the excess and slack amounts of mined product p by which the live stockpile limits at mine m are violated in period t, and β^{LE}_{rst} and β^{LS}_{rst} are similar excess and slack variables for shipped product s at port r in period t.

Cost of bulk stockpile violations

Some of the mines and ports have bulk stockpiles whose purpose serve as buffers and additional storage. Similar stockpile level violations also apply at the bulk stockpiles. The bulk stockpiles, however, do not have any minimum stockpile capacity, thus only maximum level violations apply. The cost of bulk stockpile limit violations is as follows:

$$\sum_{m\in M}\sum_{p\in P_m} MS^{BE}_{mp}\sum_{t\in T}\alpha^{BE}_{mpt} + \sum_{r\in R}\sum_{s\in S_r} PS^{BE}_{rs}\sum_{t\in T}\beta^{BE}_{rst} \quad (3.3)$$

where MS^{BE}_{mp} represents the penalties for maximum limit violation at bulk stockpiles at mine m producing product p, PS^{BE}_{rs} is the maximum limit violation penalty for bulk stockpiles at port r and of shipped product s, α^{BE}_{mpt} the excess amount of mined product p by which the bulk stockpile limit at mine m are violated in period t and β^{BE}_{rst} is the similar excess variable for shipped product s at port r in period t.

Cost of bulk handling

The outloaders and inloaders are used to transfer materials between live and bulk stockpiles. The occurrence of this process should be kept minimum as it will incur some handling cost. The cost applies at stockpiles in both mines and ports. The cost of moving material to and from bulk stockpiles is

$$\sum_{m\in M}\sum_{p\in P_m}\left[MB^{out}_{mp}\sum_{t\in T}y^{out}_{mpt}+MB^{in}_{mp}\sum_{t\in T}y^{in}_{mpt}\right]+\sum_{r\in R}\sum_{s\in S_r}\left[PB^{out}_{rs}\sum_{t\in T}u^{out}_{rst}+PB^{in}_{rs}\sum_{t\in T}u^{in}_{rst}\right] \quad (3.4)$$

where $MB^{out}_{mp}, MB^{in}_{mp}, PB^{out}_{rs}$ and PB^{in}_{rs} are the handling costs of moving products to and from bulk stockpiles at mines and ports respectively, y^{out}_{mpt} and y^{in}_{mpt} are the transfers to and from bulk stockpiles at mine m for product p in period t, and u^{out}_{rst} and u^{in}_{rst} are similar variables for port r and product s in period t.

Cost of cycle time violations

Violating the cycle time limits of train trips will incur penalties. The cost of exceeding the cycle time of train trips is given by:

$$\sum_{f\in F}CP_f\sum_{t\in T}\mu_{ft} \quad (3.5)$$

where μ_{ft} is the additional cycle time needed at fleet f in period t and CP_f is the corresponding penalty.

Cost of dumping materials

Some of the car dumpers are preferred to service specific material. We express this by adding costs of dumping materials in the objective function. The cost of dumping a product in a specific car dumper is

$$\sum_{m\in M}\sum_{d\in D_m}DP_{md}\sum_{p\in P_m}\sum_{f\in F}\sum_{t\in T}TS_{mpft}x_{mpfdst} \quad (3.6)$$

where x_{mpfdst} is the number of trains that serve mine m, mined product p, fleet f, car dumper d, shipped product s and in period t, TS_{mpft} is the corresponding train capacity, and DP_{md} is the penalty for using the dumping facility d for product from mine m.

Cost of grade non-compliance

The final grade composition of the shipped products is required to fall within the target composition range. We penalise any deviation to this target range. The cost of target grade deviations of the shipped products is

$$\sum_{c\in C}\sum_{r\in R}\sum_{s\in S_r} GP_{rsc}\sum_{t\in T}(si_{rsct}+ei_{rsct}) \tag{3.7}$$

where GP_{rsc} is the penalty for violating the target grade of component c for shipped product s at port r, si_{rsct} and ei_{rsct} are the slack and excess variables respectively when component c of product s at port r is off target in period t.

Incentive

In order to enforce a higher number of trains sent to the mines, incentives are proposed to the model. These incentives are included in the objective function and expressed by the fraction of the total throughput of iron ore. Mathematically, the incentives can be written as:

$$\pi\sum_{s\in S} SP_s\sum_{m\in M}\sum_{p\in P_m}\sum_{f\in F}\sum_{d\in D}\sum_{t\in T} TS_{mpft}x_{mpfdst} \tag{3.8}$$

where π is the incentive fraction.

3.4.4 Operational constraints

In this section, we describe all operational constraints that are not related to iron ore grade quality. These constraints include inventory levels at stockpiles, transport capacities, transfer between live and bulk stockpiles, contractual obligation, and book keeping constraints.

Live stockpile levels

There are minimum and maximum desirable live stockpile capacity limits at both mines and ports. These capacity limits are allowed to be violated but penalties apply. These penalties have been described in (3.2). The capacity constraints for live stockpile levels are expressed by the following inequalities:

$$S^L_{mpt}-\alpha^{LS}_{mpt}\leq s_{mpt}\leq S^U_{mpt}+\alpha^{LE}_{mpt}, \qquad \forall m\in M, p\in P_m, t\in T \tag{3.9}$$

$$W^L_{rst}-\beta^{LS}_{rst}\leq w_{rst}\leq W^U_{rst}+\beta^{LE}_{rst}, \qquad \forall r\in R, s\in S_r, t\in T \tag{3.10}$$

where s_{mpt} and w_{rst} are the live stockpile levels of product p at mine m and of product s at port r respectively in period t, S^L_{mpt} and S^U_{mpt} are the desirable minimum and maximum live stockpile levels respectively of product p at mine m in period t, W^L_{rst} and W^U_{rst} are the desirable minimum and maximum live stockpile levels respectively of product s at port r in period t.

Bulk stockpile levels

The bulk stockpiles at both mines and ports only have maximum desirable stockpile levels. The associated penalty is given by (3.3). The capacity constraints are given by:

$$0 \leq b_{mpt} \leq B^U_{mpt} + \alpha^{BE}_{mpt}, \qquad \forall m \in M, p \in P_m, t \in T \tag{3.11}$$

$$0 \leq v_{rst} \leq V^U_{rst} + \beta^{BE}_{rst}, \qquad \forall r \in R, s \in S_r, t \in T \tag{3.12}$$

where b_{mpt} and v_{rst} are the bulk stockpile levels of product p at mine m and of product s at port r respectively in period t, B^U_{mpt} is the desirable maximum bulk stockpile level of product p at mine m in period t, and V^U_{rst} is the desirable maximum bulk stockpile of product s at port r in period t.

Yard limits

Besides the minimum and maximum desirable stockpile levels, each live stockpile is also constrained by a yard limit. The yard limit capacity constraints are described as hard constraints. The closing stockpile level cannot be greater than the site's yard limit.

$$S^U_{mpt} + \alpha^{LE}_{mpt} \leq YM_{mpt}, \qquad \forall m \in M, p \in P_m, t \in T \tag{3.13}$$

$$W^U_{rst} + \beta^{LE}_{rst} \leq YP_{rst}, \qquad \forall r \in R, s \in S_r, t \in T \tag{3.14}$$

where YM_{mpt} is the yard limit of live stockpile at mine m for product p in period t and YP_{rst} is the yard limit of live stockpile at port r for product s in period t.

Inloader and outloader capacities

Outloaders are used to tranport the iron ore from live to bulk stockpiles, whereas inloaders is vice versa. There are maximum capacities specified for inloaders and outloader at both mines and ports.

$$0 \leq y^{out}_{mpt} \leq y^{out,U}_{mp}, \qquad \forall m \in M, p \in P_m, t \in T \tag{3.15}$$
$$0 \leq y^{in}_{mpt} \leq y^{in,U}_{mp}, \qquad \forall m \in M, p \in P_m, t \in T \tag{3.16}$$
$$0 \leq u^{out}_{rst} \leq u^{out,U}_{rs}, \qquad \forall r \in R, s \in S_r, t \in T \tag{3.17}$$
$$0 \leq u^{in}_{rst} \leq u^{in,U}_{rs}, \qquad \forall r \in R, s \in S_r, t \in T \tag{3.18}$$

where $y^{out,U}_{mp}$ and $y^{in,U}_{mp}$ are the maximum tonnes of product p that can be outloaded and inloaded between stockpiles at mine m. Similarly, $u^{out,U}_{rs}$ and $u^{in,U}_{rs}$ are the maximum tonnes of product s that can be outloaded and inloaded between stockpiles at port r.

Maximum number of trains

The maximum allowed number of trains to be sent from each mine cannot be exceeded. Therefore, mathematically, we have the following inequality:

$$\sum_{p \in P_m} \sum_{f \in F} \sum_{d \in D_{mp}} \sum_{s \in S} x_{mpfdst} \leq x^U_{mt}, \quad \forall m \in M, t \in T \tag{3.19}$$

where x^U_{mt} is the maximum number of allowed trains at mine m in period t.

Furthermore, the maximum allowed number of trains in a region also cannot be exceeded. This constraint was initially modelled as a soft constraint in the original formulation. We modified it as a hard constraint in our formulation.

$$\sum_{m \in M_g} \sum_{p \in P_m} \sum_{f \in F} \sum_{d \in D_{mrp}} \sum_{s \in S} x_{mpfdst} \leq MT_{gt}, \quad \forall g \in G, t \in T \tag{3.20}$$

where MT_{gt} is the maximum number of allowed trains at region g in period t.

Fleet capacity

Each fleet has a maximum capacity by which the number of trains serviced by that fleet in a given period cannot exceed.

$$\sum_{m \in M_f} \sum_{p \in P_m} \sum_{d \in D_{mrp}} \sum_{s \in S} x_{mpfdst} \leq MF_{ft}, \quad \forall f \in F, t \in T \tag{3.21}$$

where MF_{ft} is the available consist numbers for fleet f in period t.

Fleet hours capacity

Fleet hours capacity is defined by a soft constraint. The associated penalty is given by (3.5). Hence the capacity constraint of fleet hours of the trains is given by:

$$\sum_{m\in M_f}\sum_{p\in P_m} CT_{mpt}\sum_{d\in D_{mrp}}\sum_{s\in S} x_{mpfdst} \leq PF_{ft} + \mu_{ft}, \quad \forall f \in F, t \in T \tag{3.22}$$

where CT_{mpt} is the cycle time of the train sent to mine m carrying product p in period t, PF_{ft} is the available pooled hours of fleet f in period t and μ_{ft} is the additional cycle time needed.

Joint ventures obligation

There is a contractual obligation between the joint ventures which some of the mines must comply with. This obligation is expressed as maximum and minimum capacity limits for cumulative number of trains to be sent to the associated mines in each period. In the original model, this constraint is described as both soft and hard constraints. In our formulation, only the hard constraint is employed. At period $t = n$, the contractual obligation constraint is defined by:

$$\sum_{t=1}^{n} JV_{mt}^{L} \leq \sum_{p\in P_m}\sum_{f\in F}\sum_{d\in D_{mp}}\sum_{s\in S}\sum_{t=1}^{n} x_{mpfdst} \leq \sum_{t=1}^{n} JV_{mt}^{U}, \quad \forall m \in M^{JV}, t \in T \tag{3.23}$$

where JV_{mt}^{L} and JV_{mt}^{U} are cumulative train delivery targets at mine m constrained by the joint venture in period t.

Car dumpers capacity

Each car dumper has a maximum capacity for the number of trains that can be facilitated. This capacity constraints are expressed by the following inequality:

$$\sum_{m\in M}\sum_{p\in P_m}\sum_{f\in F}\sum_{s\in S} x_{mpfdst} \leq DC_{dt}, \quad \forall d \in D, t \in T \tag{3.24}$$

where DC_{dt} is the maximum capacity of car dumper d in period t.

In addition, car dumpers serving the Cape Lambert-Western Creek region are also constrained by per-period total dumping capacity.

$$\sum_{m\in M}\sum_{p\in P_m}\sum_{f\in F}\sum_{d\in D^{WC}}\sum_{s\in S} x_{mpfdst} \leq WC_t, \quad \forall t \in T \tag{3.25}$$

where WC_t is the total maximum capacity of car dumpers serving the Cape Lambert-Western Creek region in period t.

Shipping capacity

The total amount of shipped product cannot exceed the maximum shipping capacity.

$$0 \leq \sum_{s \in S_r} z_{rst} \leq Z^U_{rt}, \quad \forall r \in R, t \in T \tag{3.26}$$

where Z^U_{rt} is the maximum tonnes of product that can be shipped at port r in period t.

Book keeping

Book keeping constraints are designed to keep track of the closing stockpile levels at each period. In a mine, the stockpile level at the end of a period is equal to the opening stockpile level plus the quantity of products coming into stockpile less the quantity of products going out of stockpile. The process of stockpile book keeping at the mine is pictured in Figure 3.2.

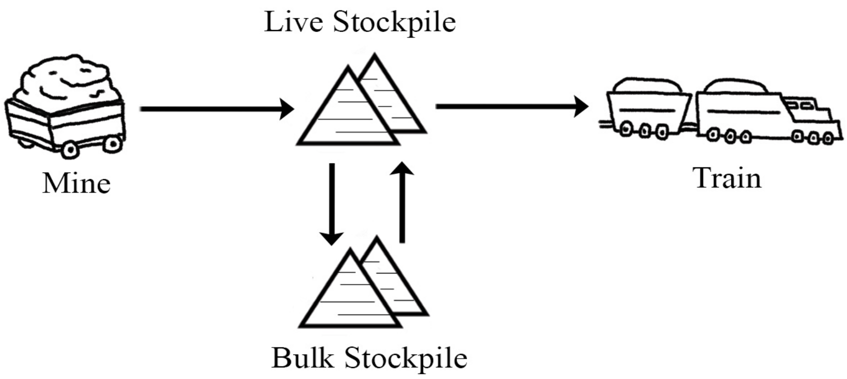

Figure 3.2: Book keeping process at a mine

Hence for the book keeping of iron ore at mines, we have the following equations:

$$s_{mpt} = s_{mp,t-1} + IOP_{mpt} + y^{in}_{mpt} - y^{out}_{mpt} - \sum_{f \in F} TS_{mpft} \sum_{d \in D_{mp}} \sum_{s \in S_{mp}} x_{mpfdst}, \quad \forall m \in M, p \in P_m, t \in T \tag{3.27}$$

$$b_{mpt} = b_{mp,t-1} + y^{out}_{mpt} - y^{in}_{mpt}, \qquad \forall m \in M, p \in P_m, t \in T \quad (3.28)$$

where IOP_{mpt} is a parameter that indicates the amount of product p produced at mine m in period t.

The book keeping constraint at a port live stockpile is more complex, taking into account the lump screening process. We describe the book keeping process at the port in Figure 3.3. Before loading onto a ship, a certain proportion of lump products which are too small are added into the associated fines ore stockpile (see Section 1.1 for the definition of lump and fines ore).

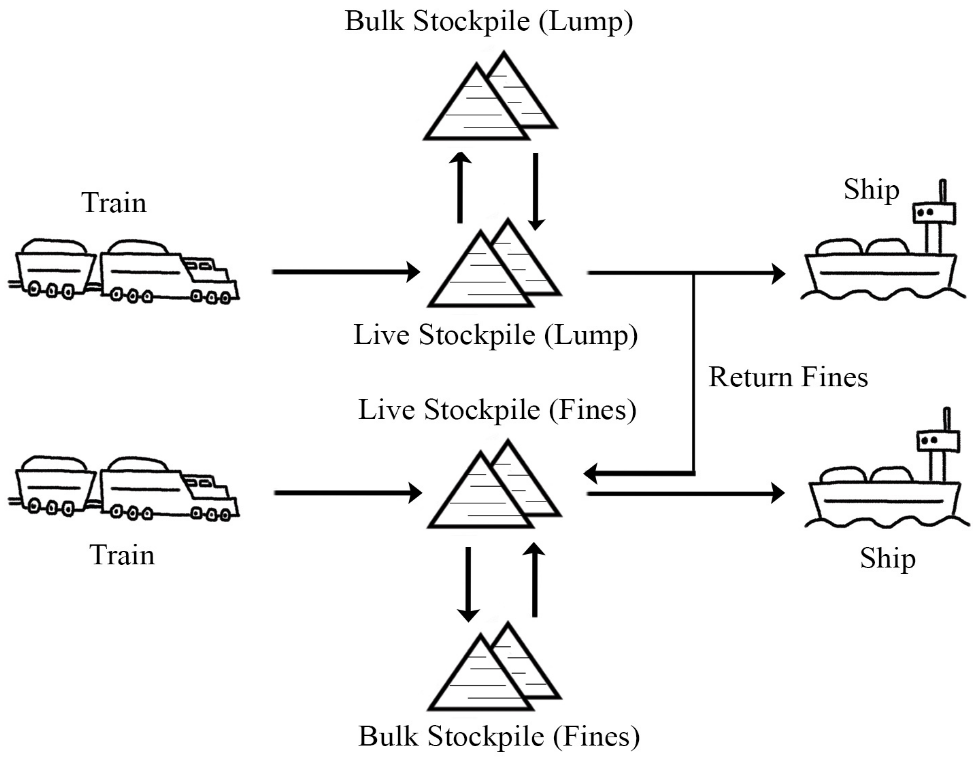

Figure 3.3: Book keeping process at a port

Therefore the book keeping constraint for lump products at a port live stockpile is as follows:

$$w_{rst} = w_{rs,t-1} + \sum_{m \in M} \sum_{p \in P_m} \sum_{f \in F} TS_{mpft} \sum_{d \in D_{mrp}} x_{mpfdst} + u^{in}_{rst} - u^{out}_{rst} - \frac{z_{rst}}{1 - RF_{rst}}, \quad \forall r \in R, s \in S^L, t \in T \quad (3.29)$$

where and RF_{rst} is the return fines proportion in percent for a lump product s at port r in period t.

For fines product, the stockpile level also depends on the return fines transferred from the associated lump stockpile.

$$w_{rst} = w_{rs,t-1} + \sum_{m \in M} \sum_{p \in P_m} \sum_{f \in F} TS_{mpft} \sum_{d \in D_{mrp}} x_{mpfdst} + u^{in}_{rst} - u^{out}_{rst} - z_{rst} + \frac{z^*_{rst} RF^*_{rst}}{1 - RF^*_{rst}}, \quad \forall r \in R, s \in S^F, t \in T \quad (3.30)$$

where z^*_{rst} is the shipping amount of associated lump product of fines product s at port r in period t and RF^*_{rst} is the return fines in percent of the associated lump product.

The book keeping constraint at a port bulk stockpile is similar to that at mines.

$$v_{rst} = v_{rs,t-1} + u^{out}_{rst} - u^{in}_{rst}, \qquad \forall r \in R, s \in S_r, t \in T \qquad (3.31)$$

3.4.5 Iron ore grades

The blending process plays an important role in the entire model as it will determine the final product grade quality. The aim is to achieve final grade composition within target range. Any grade non-compliance will result in penalty as shown in (3.7). This section outlines constraints associated with the blending process which are expressed as grade calculations at the stockpiles, trains, and ships. The constraints outlined in this section are in similar fashion with equations (3.27)–(3.31), namely the book keeping constraints, except that most terms are bilinear.

Live stockpile grades at mines

The iron ore grades at live stockpile in a mine are calculated by dividing the total component of products left at the mine live stockpile at the end of the given period by the total stockpile level.

$$LM_{mpct} = \frac{LM^{live}_{mpc,t-1} + LM^{out}_{mpct} + IOG_{mpct}IOP_{mpt} + BM^{in}_{mpct} - \sum_{f\in F}\sum_{d\in D_{mp}}\sum_{s\in S_{mp}} RG^{rail}_{mpcfdst}}{s_{mpt}},$$
$$\forall m \in M, p \in P_m, c \in C, t \in T \quad (3.32)$$

where

$$LM^{live}_{mpct} = LM_{mpct}s_{mpt}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T \quad (3.33)$$
$$LM^{out}_{mpct} = LM_{mpct}y^{out}_{mpt}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T \quad (3.34)$$
$$BM^{in}_{mpct} = BM_{mpct}y^{in}_{mpt}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T \quad (3.35)$$

and

$$RG^{rail}_{mpcfdst} = RG_{mpct}TS_{mpft}x_{mpfdst},$$
$$\forall m \in M, p \in P_m, c \in C, f \in F, d \in D_{mp}, s \in S_{mp}, t \in T \quad (3.36)$$

LM_{mpct} is the mine live grade quality of component c in product p at mine m in period t, BM_{mpct} is the mine bulk grade quality of component c in product p at mine m in period t, IOG_{mpct} is the grade quality of component c in product p produced at mine m in period t and RG_{mpct} is the grade quality of component c in product p transported by trains from mine m in period t.

Bulk stockpile grades at mines

The iron ore grades at bulk stockpile in a mine are calculated by dividing the total component of products left at the mine bulk stockpile at the end of the given period by the total stockpile level.

$$BM_{mpct} = \frac{BM^{bulk}_{mpc,t-1} - BM^{in}_{mpct} + LM^{out}_{mpct}}{b_{mpt}},$$
$$\forall m \in M, p \in P_m, c \in C, t \in T \quad (3.37)$$

where

$$BM^{bulk}_{mpct} = BM_{mpct}b_{mpct}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T \quad (3.38)$$

Railed grades

The iron ore grades in a train are calculated by finding the fraction of the total component of products in the train after loading process in the given period.

$$RG_{mpct} = \frac{IOG_{mpct}IOT_{mpt} + LM_{mpc,t-1}LT_{mpt} + BM_{mpc,t-1}BT_{mpt}}{\sum_{f\in F}\sum_{d\in D_{mrp}}\sum_{s\in S_{mp}} TS_{mpft}x_{mpfdst}},$$
$$\forall m \in M, p \in P_m, c \in C, t \in T \quad (3.39)$$

where IOT_{mpt}, LT_{mpt}, and BT_{mpt} are the total amount of product p taken by trains from the main production line, the live stockpile, and the bulk stockpile respectively at mine m in period t.

The order of train loading process depends on the mine's regime. If the mine's regime is LIFO (last in first out), the train will prioritise on loading the material produced from that mine first and the remaining amount will be taken from the live stockpile. If there is still space in the train, the ore from bulk stockpile will also be loaded until the train reaches its capacity.

Hence, if the regime is LIFO, the following applies:

$$IOT_{mpt} = \min\big\{\sum_{f\in F} TS_{mpft} \sum_{d\in D_{mp}}\sum_{s\in S_{mp}} x_{mpfdst}, IOP_{mpt}\big\},$$
$$\forall m \in M^{LF}, p \in P_m, t \in T \quad (3.40)$$

$$LTM_{mpt} = \max\big\{0, \sum_{f\in F} TS_{mpft} \sum_{d\in D_{mp}}\sum_{s\in S_{mp}} x_{mpfdst} - IOP_{mpt}\big\},$$
$$\forall m \in M^{LF}, p \in P_m, t \in T \quad (3.41)$$

$$LT_{mpt} = \min\big\{s_{mp,t-1}, LTM_{mpt}\big\}, \qquad \forall m \in M^{LF}, p \in P_m, t \in T \quad (3.42)$$

$$BT_{mpt} = \max\big\{0, \sum_{f\in F} TS_{mpft} \sum_{d\in D_{mrp}}\sum_{s\in S_{mp}} x_{mpfdst} - IOP_{mpt} - s_{mp,t-1}\big\},$$
$$\forall m \in M^{LF}, p \in P_m, t \in T \quad (3.43)$$

If the mine's regime is FIFO (first in first out), the train will load the material from the live stockpiles before taking the produced material from the mine. If there is still space in the train, the ore from bulk stockpile will also be loaded until the train reaches its capacity.

Hence, if the mine's regime is FIFO, we have:

$$LT_{mpt} = \min\big\{\sum_{f\in F} TS_{mpft} \sum_{d\in D_{mp}} \sum_{s\in S_{mp}} x_{mpfdst}, s_{mp,t-1}\big\},$$
$$\forall m \in M^{FF}, p \in P_m, t \in T \quad (3.44)$$

$$IOTM_{mpt} = \max\big\{0, \sum_{f\in F} TS_{mpft} \sum_{d\in D_{mp}} \sum_{s\in S_{mp}} x_{mpfdst} - s_{mp,t-1}\big\}$$
$$\forall m \in M^{FF}, p \in P_m, t \in T \quad (3.45)$$

$$IOT_{mpt} = \min\big\{IOP_{mpt}, IOTM_{mpt}\big\}, \ \forall m \in M^{FF}, p \in P_m, t \in T \quad (3.46)$$

$$BT_{mpt} = \max\big\{0, \sum_{f\in F} TS_{mpft} \sum_{d\in D_{mp}} \sum_{s\in S_{mp}} x_{mpfdst} - IOP_{mpt} - s_{mp,t-1}\big\},$$
$$\forall m \in M^{FF}, p \in P_m, t \in T \quad (3.47)$$

The railed grade calculations involve bilinear terms and max or min functions in the equations, and thus, non-linear. While it is not easy to linearise the bilinear terms, the max and min functions can be easily linearised by applying the MIP generalisation described in Section 3.2. For some variables which do not have minimum and/or maximum limits, we use 0 and high arbitrary numbers, such as 10,000 or 20,000, as their minimum and maximum limits respectively.

Following the procedure, we replace equation (3.40) with these inequalities:

$$IOT_{mpt} \le IOP_{mpt}, \qquad \forall m \in M^{LF}, p \in P_m, t \in T \quad (3.48)$$
$$IOT_{mpt} \le \sum_{f\in F} TS_{mpft} \sum_{d\in D_{mp}} \sum_{s\in S} x_{mpfdst}, \ \forall m \in M^{LF}, p \in P_m, t \in T \quad (3.49)$$

$$IOT_{mpt} \ge \sum_{f\in F} TS_{mpft} \sum_{d\in D_{mp}} \sum_{s\in S} x_{mpfdst} - (10,000 - IOP_{mpt}) IOT^{bin}_{mpt},$$
$$\forall m \in M^{LF}, p \in P_m, t \in T \quad (3.50)$$

$$IOT_{mpt} \ge IOP_{mpt} - IOP_{mpt}(1 - IOT^{bin}_{mpt}),$$
$$\forall m \in M^{LF}, p \in P_m, t \in T \quad (3.51)$$

where IOT^{bin}_{mpt} is a binary variable which will return 1 if IOP_{mpt} has a lower value than $\sum_{f\in F} TS_{mpft} \sum_{d\in D_{mp}} \sum_{s\in S} x_{mpfdst}$ at mine m for product p in period t, or 0 otherwise.

We replace equation (3.41) with the inequalities below:

$$LTM_{mpt} \geq 0, \qquad \forall m \in M^{LF}, p \in P_m, t \in T \quad (3.52)$$

$$LTM_{mpt} \geq \sum_{f\in F} TS_{mpft} \sum_{d\in D_{mp}} \sum_{s\in S} x_{mpfdst} - IOP_{mpt}, \qquad \forall m \in M^{LF}, p \in P_m, t \in T \quad (3.53)$$

$$LTM_{mpt} \leq \sum_{f\in F} TS_{mpft} \sum_{d\in D_{mp}} \sum_{s\in S} x_{mpfdst} - IOP_{mpt} + 10,000 LTM^{bin}_{mpt}, \qquad \forall m \in M^{LF}, p \in P_m, t \in T \quad (3.54)$$

$$LTM_{mpt} \leq 10,000(1 - LTM^{bin}_{mpt}), \qquad \forall m \in M^{LF}, p \in P_m, t \in T \quad (3.55)$$

where LTM^{bin}_{mpt} is a binary variable that will return 1 if $\sum_{f\in F} \sum_{d\in D_{mp}} \sum_{s\in S_{mp}} TS_{mpft} x_{mpfdst} - IOP_{mpt}$ is less than 0 at mine m for product p in period t, or 0 otherwise.

We replace equation (3.42) with the inequalities below:

$$LT_{mpt} \leq s_{mp,t-1}, \qquad \forall m \in M^{LF}, p \in P_m, t \in T \quad (3.56)$$

$$LT_{mpt} \leq LTM_{mpt}, \qquad \forall m \in M^{LF}, p \in P_m, t \in T \quad (3.57)$$

$$LT_{mpt} \geq s_{mp,t-1} - IOP_{mpt} - 20,000 LT^{bin}_{mpt}, \qquad \forall m \in M^{LF}, p \in P_m, t \in T \quad (3.58)$$

$$LT_{mpt} \geq LTM_{mpt} - 20,000(1 - LT^{bin}_{mpt}), \qquad \forall m \in M^{LF}, p \in P_m, t \in T \quad (3.59)$$

where LT^{bin}_{mpt} is a binary variable that will return 1 if LTM_{mpt} has a lower value than $s_{mp,t-1}$ at mine m for product p in period t, or 0 otherwise.

We replace equation (3.44) with the inequalities below:

$$LT_{mpt} \leq s_{mp,t-1}, \qquad \forall m \in M^{FF}, p \in P_m, t \in T \quad (3.60)$$

$$LT_{mpt} \leq \sum_{f\in F} TS_{mpft} \sum_{d\in D_{mp}} \sum_{s\in S} x_{mpfdst}, \qquad \forall m \in M^{FF}, p \in P_m, t \in T \quad (3.61)$$

$$LT_{mpt} \geq \sum_{f\in F} TS_{mpft} \sum_{d\in D_{mp}} \sum_{s\in S} x_{mpfdst} - 10,000 LT^{bin}_{mpt},$$
$$\forall m \in M^{FF}, p \in P_m, t \in T \quad (3.62)$$

$$LT_{mpt} \geq s_{mp,t-1} - 20,000(1 - LT^{bin}_{mpt}),$$
$$\forall m \in M^{FF}, p \in P_m, t \in T \quad (3.63)$$

where LT^{bin}_{mpt} in this case will return 1 if $s_{mp,t-1}$ has a lower value than $\sum_{f\in F} TS_{mpft} \sum_{d\in D_{mp}} \sum_{s\in S} x_{mpfdst}$ at mine m for product p in period t, or 0 otherwise.

We replace equation (3.45) with the inequalities below:

$$IOTM_{mpt} \geq 0, \qquad \forall m \in M^{FF}, p \in P_m, t \in T \quad (3.64)$$

$$IOTM_{mpt} \geq \sum_{f\in F} TS_{mpft} \sum_{d\in D_{mp}} \sum_{s\in S_{mp}} x_{mpfdst} - s_{mp,t-1},$$
$$\forall m \in M^{FF}, p \in P_m, t \in T \quad (3.65)$$

$$IOTM_{mpt} \leq \sum_{f\in F} TS_{mpft} \sum_{d\in D_{mp}} \sum_{s\in S_{mp}} x_{mpfdst} - s_{mp,t-1} + 10,000 IOTM^{bin}_{mpt},$$
$$\forall m \in M^{FF}, p \in P_m, t \in T \quad (3.66)$$

$$IOTM_{mpt} \leq 10,000(1 - IOTM^{bin}_{mpt}),$$
$$\forall m \in M^{FF}, p \in P_m, t \in T \quad (3.67)$$

where $IOTM^{bin}_{mpt}$ is a binary variable that will return 1 if $\sum_{f\in F} TS_{mpft} \sum_{d\in D_{mp}} \sum_{s\in S_{mp}} x_{mpfdst} - s_{mp,t-1}$ is lower than 0 at mine m for product p in period t, or 0 otherwise.

We replace equation (3.46) with the inequalities below:

$$IOT_{mpt} \leq IOP_{mpt}, \qquad \forall m \in M^{FF}, p \in P_m, t \in T \quad (3.68)$$
$$IOT_{mpt} \leq IOTM_{mpt}, \qquad \forall m \in M^{FF}, p \in P_m, t \in T \quad (3.69)$$

$$IOT_{mpt} \geq IOP_{mpt} - (IOP_{mpt} + 10,000) IOT^{bin}_{mpt},$$
$$\forall m \in M^{FF}, p \in P_m, t \in T \quad (3.70)$$

$$IOT_{mpt} \geq IOTM_{mpt} - (10,000 - IOP_{mpt})(1 - IOT^{bin}_{mpt}), \quad \forall m \in M^{FF}, p \in P_m, t \in T \quad (3.71)$$

where IOT^{bin}_{mpt} in this case will return 1 if $IOTM_{mpt}$ has a lower value than IOP_{mpt} at mine m for product p in period t, or 0 otherwise.

It is evident that the equations for calculating BT_{mpt} for $m \in M^{LF}$ and $m \in M^{FF}$ are the same equation. Therefore equations (3.43) and (3.47) can be combined into one equation. As $M^{LF} \cup M^{FF} = M$, we have

$$BT_{mpt} = \max\{0, \sum_{f \in F} TS_{mpft} \sum_{d \in D_{mp}} \sum_{s \in S_{mp}} x_{mpfdst} - IOP_{mpt} - s_{mp,t-1}\}, \quad \forall m \in M, p \in P_m, t \in T \quad (3.72)$$

Finally we replace this equation with the inequalities below:

$$BT_{mpt} \geq 0, \quad \forall m \in M, p \in P_m, t \in T \quad (3.73)$$

$$BT_{mpt} \geq \sum_{f \in F} TS_{mpft} \sum_{d \in D_{mp}} \sum_{s \in S_{mp}} x_{mpfdst} - IOP_{mpt} - s_{mp,t-1}, \quad \forall m \in M, p \in P_m, t \in T \quad (3.74)$$

$$BT_{mpt} \leq \sum_{f \in F} TS_{mpft} \sum_{d \in D_{mp}} \sum_{s \in S_{mp}} x_{mpfdst} - IOP_{mpt} + 10,000 BT^{bin}_{mpt}, \quad \forall m \in M, p \in P_m, t \in T \quad (3.75)$$

$$BT_{mpt} \leq 10,000(1 - BT^{bin}_{mpt}), \quad \forall m \in M, p \in P_m, t \in T \quad (3.76)$$

where BT^{bin}_{mpt} will return 1 if $\sum_{f \in F} TS_{mpft} \sum_{d \in D_{mp}} \sum_{s \in S_{mp}} x_{mpfdst} - IOP_{mpt}$ is less than 0 at mine m for product p in period t, or 0 otherwise.

Live stockpile grades at ports

The iron ore grades at live stockpile in a port are calculated by dividing the total component of products left at the port live stockpile in the given period by the total stockpile level. The lump screening process also affects the calculation of the grades. The constraints for live stockpile grades at ports for lump product are as follows:

$$LP_{rsct} = \frac{LP^{live}_{rsc,t-1} - LP^{out}_{rsct} + BP^{in}_{rsct} + \sum_{m \in M} \sum_{p \in P_m} \sum_{f \in F} \sum_{d \in D_{mrp}} RG^{rail}_{mpcfdst}}{w^{\#}_{rst}}, \quad \forall r \in R, s \in S^L, c \in C, t \in T \quad (3.77)$$

where

$$LP^{live}_{rsct} = LP_{rsct} w_{rst}, \qquad \forall r \in R, s \in S_r, c \in C, t \in T \quad (3.78)$$
$$LP^{out}_{rsct} = LP_{rsct} u^{out}_{rst}, \qquad \forall r \in R, s \in S_r, c \in C, t \in T \quad (3.79)$$
$$BP^{in}_{rsct} = BP_{rsct} u^{in}_{rst}, \qquad \forall r \in R, s \in S_r, c \in C, t \in T \quad (3.80)$$

and

$$w^{\#}_{rst} = w_{rs,t-1} - u^{out}_{rst} + u^{in}_{rst} + \sum_{m \in M} \sum_{p \in P_m} \sum_{f \in F} TS_{mpft} \sum_{d \in D_{mrp}} x_{mpdst}, \qquad \forall r \in R, s \in S_r, t \in T \quad (3.81)$$

LP_{rsct} is the port live grade quality of component c in shipped product s at port r in period t and BP_{rsct} is the port bulk grade.

For fines product, the live stockpile grades also depends on the grades of return fines from the associated lump product.

$$LP_{rsct} = \frac{ZG^{\#}_{rsct} + LP^{rf}_{rsct}}{w^{\#}_{rst} + \frac{z^*_{rst} RF^*_{rst}}{1 - RF^*_{rst}}}, \qquad \forall r \in R, s \in S^F, c \in C, t \in T \quad (3.82)$$

where

$$ZG^{\#}_{rsct} = ZG_{rsct} w^{\#}_{rst}, \qquad \forall r \in R, s \in S^F, t \in T \quad (3.83)$$
$$LP^{rf}_{rsct} = LP^*_{rsct} \frac{z^*_{rst} RF^*_{rst}}{1 - RF^*_{rst}}, \qquad \forall r \in R, s \in S^F, t \in T \quad (3.84)$$

and ZG_{rsct} is the grade quality of component c in shipped product s at port r in period t.

Bulk stockpile grades at ports

The iron ore grades at bulk stockpile in a port are calculated by dividing the total component of products left at the port bulk stockpile in the given period by the total stockpile level.

$$BP_{rsct} = \frac{BP^{bulk}_{rsc,t-1} - BP^{in}_{rsct} + LP^{out}_{rsct}}{v_{rst}}, \qquad \forall r \in R, s \in S_r, c \in C, t \in T \quad (3.85)$$

where

$$BP^{bulk}_{rsct} = BP_{rsct} v_{rsct}, \qquad \forall r \in R, s \in S_r, c \in C, t \in T \quad (3.86)$$

Shipped grades

The materials to be shipped are taken directly from the live stockpiles at the ports before any return fines material is added. For all lump and fines products, we define ZG_{rsct} as below.

$$ZG_{rsct} = \frac{\begin{array}{c} LP^{live}_{rsc,t-1} - LP^{out}_{rsct} + BP^{in}_{rsct} \\ + \sum_{m \in M} \sum_{p \in P_m} \sum_{f \in F} \sum_{d \in D_{mrp}} RG^{rail}_{mpcfdst} \end{array}}{w^{\#}_{rst}}, \quad \forall r \in R, s \in S_r, c \in C, t \in T \quad (3.87)$$

Note that for lump product, the shipped grades are equal to its live stockpile grades at ports since there is no additional material from the return fines process.

Grade deviations

The shipped grades should fall within the target range. These grade deviations are formulated as soft constraints and any violation will incur some penalties as shown in (3.7)

$$(TG_{rsct} - TG^{tol}_{rsc}) z_{rst} - si_{rsct} \leq ZG^{shipped}_{rsct} \leq (TG_{rsct} + TG^{tol}_{rsc}) z_{rst} + ei_{rsct} \quad \forall r \in R, s \in S_r, c \in C, t \in T \quad (3.88)$$

where TG_{rsct} is the target grade for component c of shipped product s at port r, TG^{tol}_{rsc} is the target grade tolerance, and

$$ZG^{shipped}_{rsct} = ZG_{rsct} z_{rst}, \quad \forall r \in R, s \in S_r, c \in C, t \in T \quad (3.89)$$

3.4.6 Summary of notations

Sets

C	The set of all components of a product.
D	The set of all car dumpers.
D_{mrp}	The set of all car dumpers in port $r \in R$ receiving mined product $p \in P$ from mine $m \in M$.
D^{WC}	The set of all car dumpers that serve the Cape Lambert-Western Creek region, $D^{WC} \subseteq D$.
F	The set of all train fleets.
G	The set of all regions.
M	The set of all mines.
M_f	The set of all mines serviced by the fleet $f \in F$, $M_f \subset M$.
M_g	The set of all mines belonging to region $g \in G$, $M_g \subset M$.
M^{FF}	The set of all mines whose regime is FIFO, $M^{FF} \subset M$.
M^{JV}	The set of all mines that have to comply with the joint ventures obligations, $M^{JV} \subseteq M$.
M^{LF}	The set of all mines whose regime is LIFO, $M^{LF} \subset M$.
P	The set of all mined products.
P_m	The set of all mined products for mine $m \in M$, $P_m \subseteq P$.
R	The set of all ports.
S	The set of all shipped products.
S_r	The set of all shipped products from port $r \in R$, $S_r \subseteq S$.
S_{mp}	The set of all shipped products sent from mine $m \in M$ and product $p \in P$, $S_{mp} \subseteq S$.
S^F	The set of all fines shipped products, $S^F \subset S$
S^L	The set of all lump shipped products, $S^L \subset S$
T	The set of planning periods.

Model parameters

π	Fraction of sales price and number of trains for incentive.
B^U_{mpt}	Maximum bulk stockpile level at mine $m \in M$ for product $p \in P$ in period $t \in T$.
CP_f	Cost for exceeding maximum train cycle time of fleet $f \in F$.
CT_{mpt}	Cycle time of a train used at mine $m \in M$ carrying mined product $p \in P_m$ in period $t \in T$.
DC_{dt}	Maximum capacity of car dumper $d \in D$ in period $t \in T$.
DP_{md}	Cost of dumping products coming from mine $m \in M$ in a specific car dumper $d \in D_m$.
GP_{rsc}	Cost of grade deviation for component $c \in C$ in shipped product $s \in S_r$ at port $r \in R$.
I	Discount factor.
IOG_{mpt}	Grade of mined product $p \in P_m$ produced at mine $m \in M$ in period $t \in T$.
IOP_{mpt}	Amount of product $p \in P_m$ produced at mine $m \in M$ in period $t \in T$.
JV^L_{mt}	Minimum cumulative number of trains to be sent from mine $m \in M^v$ constrained by joint ventures in period $t \in T$.
JV^U_{mt}	Maximum cumulative number of trains to be sent from mine $m \in M^v$ constrained by joint ventures in period $t \in T$.
MB^{in}_{mp}	Cost of transferring mined product $p \in P_m$ at mine $m \in M$ from bulk to live stockpiles.
MB^{out}_{mp}	Cost of transferring mined product $p \in P_m$ at mine $m \in M$ from live to bulk stockpiles.
MF_{ft}	Available number of trains in fleet $f \in F$ in period $t \in T$.
MS^{BE}_{mp}	Cost for violating maximum bulk stockpile limit for product $p \in P_m$ at mine $m \in M$.
MS^{LE}_{mp}	Cost for violating maximum live stockpile limit for product $p \in P_m$ at mine $m \in M$.
MS^{LS}_{mp}	Cost for violating minimum live stockpile limit for product $p \in P_m$ at mine $m \in M$.
MT_{gt}	Maximum capacity of trains in region $g \in G$ at period $t \in T$.
PB^{in}_{rs}	Cost of transferring shipped product $s \in S_r$ at port $r \in R$ from bulk to live stockpiles.
PB^{out}_{rs}	Cost of transferring shipped product $s \in S_r$ at port $r \in R$ from live to to bulk stockpiles.
PF_{ft}	Available pooled hours of fleet $f \in F$ in period $t \in T$.
PS^{BE}_{rs}	Cost for violating maximum bulk stockpile limit for product $s \in S_r$ at port $r \in R$.

PS^{LE}_{rs}	Cost for violating maximum live stockpile limit for product $s \in S_r$ at port $r \in R$.
PS^{LS}_{rs}	Cost for violating minimum live stockpile limit for product $s \in S_r$ at port $r \in R$.
RF_{rst}	Percentage of lump product $s \in S^L$ returned to fines stockpile at port $r \in R$ in period T.
RF^{*}_{rst}	Percentage of the associated lump product of fines product $s \in S^F$ at port $r \in R$ in period T.
S^L_{mpt}	Minimum live stockpile level at mine $m \in M$ for product $p \in P_m$ in period $t \in T$.
S^U_{mpt}	Maximum live stockpile level at mine $m \in M$ for product $p \in P_m$ in period $t \in T$.
SP_s	Sale price per tonne of shipped product $s \in S$.
TG_{rsct}	Target quality for component $c \in C$ of shipped product $s \in S_r$ at port $r \in R$ in period $t \in T$.
TG^{tol}_{rsc}	Target quality tolerance for component $c \in C$ of shipped product $s \in S_r$ at port $r \in R$.
TS_{mpft}	Capacity of a train in tonnes used at mine $m \in M$ transporting mined product $p \in P_m$ belonging to fleet $f \in F$ in period $t \in T$.
$u^{in,U}_{rs}$	Maximum amount of product $s \in S_r$ that can be inloaded at port $r \in R$.
$u^{out,U}_{rs}$	Maximum amount of product $s \in S_r$ that can be outloaded at port $r \in R$.
V^U_{rst}	Maximum bulk stockpile level at port $r \in R$ for product $s \in S_r$.
W^L_{rst}	Minimum live stockpile level at port $r \in R$ for product $s \in S_r$.
W^U_{rst}	Maximum live stockpile level at port $r \in R$ for product $s \in S_r$.
WC_t	Maximum capacity of car dumpers in Cape Lambert-Western Creek region in period $t \in T$.
x^U_{mt}	Maximum number of trains allowed at mine $m \in M$ in period $t \in T$.
$y^{in,U}_{mp}$	Maximum tonnes of product $p \in P_m$ that can be inloaded at mine $m \in M$.
$y^{out,U}_{mp}$	Maximum tonnes of product $p \in P_m$ that can be outloaded at mine $m \in M$.
YM_{mpt}	Yard capacity limit of live stockpile at mine $m \in M$ for product $p \in P_m$ in period $t \in T$.
YP_{rst}	Yard capacity limit of live stockpile at port $r \in R$ for product $s \in S_r$ in period $t \in T$.
Z^U_{rt}	Maximum capacity of amount shipped at port $r \in R$ in period $t \in T$.

Decision variables

α^{BE}_{mpt}	Amount by which maximum bulk stockpile limits were violated at mine $m \in M$ for mined product $p \in P_m$ in period $t \in T$.
α^{LE}_{mpt}	Amount by which maximum live stockpile limits were violated at mine $m \in M$ for mined product $p \in P_m$ in period $t \in T$.
α^{LS}_{mpt}	Amount by which minimum live stockpile limits were violated at mine $m \in M$ for mined product $p \in P_m$ in period $t \in T$.
β^{BE}_{rst}	Amount by which maximum bulk stockpile limits were violated at port $r \in R$ for shipped product $s \in S_r$ in period $t \in T$.
β^{LE}_{rst}	Amount by which maximum live stockpile limits were violated at port $r \in R$ for shipped product $s \in S_r$ in period $t \in T$.
β^{LS}_{rst}	Amount by which minimum live stockpile limits were violated at port $r \in R$ for shipped product $s \in S_r$ in period $t \in T$.
μ_{ft}	Additional cycle time of fleet $f \in F$ required at period $t \in T$.
b_{mpt}	Bulk stockpile level at mine $m \in M$ for mined product $p \in P_m$ in period $t \in T$.
BM_{mpct}	Bulk stockpile grade of component $c \in C$ in product $p \in P_m$ at mine $m \in M$ in period $t \in T$.
BM^{bulk}_{mpct}	Product of BM_{mpct} and b_{mpct}.
BM^{in}_{mpct}	Product of BM_{mpct} and y^{in}_{mpct}.
BP_{rsct}	Bulk stockpile grade of component $c \in C$ in product $s \in S_r$ at port $r \in R$ in period $t \in T$.
BP^{bulk}_{rsct}	Product of BP_{rsct} and v_{rsct}.
BP^{in}_{rsct}	Product of BP_{rsct} and u^{in}_{rsct}.
BT_{mpct}	Amount of component $c \in C$ of product $p \in P$ transported from the bulk stockpile at mine $m \in M$ in period $t \in T$.
BT^{bin}_{mpct}	A binary variable that will return 1 if $\sum_{f \in F} TS_{mpft} \sum_{d \in D_{mp}} \sum_{s \in S_{mp}} x_{mpfdst} - IOP_{mpt}$ is less than 0 at mine $m \in M$ for product $p \in P_m$ in period $t \in T$, or 0 otherwise.
ei_{rsct}	Excess variable to penalise when grade of component $c \in C$ of shipped product $s \in S_r$ from port $r \in R$ at time $t \in T$ is off target.
IOT_{mpct}	Amount of component $c \in C$ in mined product $p \in P$ produced from mine $m \in M$ that is transported by trains in period $t \in T$.

IOT^{bin}_{mpt}	If the mine regime is LIFO, it is a binary variable that will return 1 if IOP_{mpt} is lower than $\sum_{f\in F} TS_{mpft} \sum_{d\in D_{mp}} \sum_{s\in S}$ x_{mpfdst} at mine $m \in M^{LF}$ for product $p \in P_m$ in period $t \in T$, or 0 otherwise. If the mine regime is FIFO, it is a binary variable that will return 1 if $IOTM_{mpt}$ has a lower value than IOP_{mpt} at mine $m \in M^{FF}$ for product $p \in P_m$ in period $t \in T$, or 0 otherwise.
$IOTM_{mpt}$	A variable representing $\max\{0, \sum_{f\in F} \sum_{d\in D_{mp}} \sum_{s\in S_{mp}} TS_{mpft}$ $\mathrm{x}_{mpfdst} - s_{mp,t-1}\}$.
$IOTM^{bin}_{mpt}$	A binary variable that will return 1 if $\sum_{f\in F} TS_{mpft} \sum_{d\in D_{mp}}$ $\sum_{s\in S_{mp}} x_{mpfdst} - s_{mp,t-1}$ is lower than 0 at mine $m \in M^{FF}$ for product $p \in P_m$ in period $t \in T$, or 0 otherwise.component $c \in C$ in product $p \in P_m$ at mine $m \in M$ in period $t \in T$.
LM^{live}_{mpct}	Product of LM_{mpct} and s_{mpct}.
LM^{out}_{mpct}	Product of LM_{mpct} and y^{out}_{mpct}.
LP_{rsct}	Live stockpile grade of component $c \in C$ in product $s \in S_r$ at port $r \in R$ in period $t \in T$.
LP^{live}_{rsct}	Product of LP_{rsct} and w_{rsct}.
LP^{rf}_{rsct}	Product of LP^*_{rsct} and $z^*_{rst} RF^*_{rst}$.
LP^{out}_{rsct}	Product of LP_{rsct} and u^{out}_{rsct}.
LP^*_{rst}	Live stockpile grade of component $c \in C$ in the associated lump product of fines product $s \in S^F$ at port $r \in R$ in period $t \in T$.
LT_{mpct}	Amount of component $c \in C$ of product $p \in P$ transported from the live stockpile at mine $m \in M$ in period $t \in T$.
LT^{bin}_{mpt}	If the mine regime is LIFO, it is a binary variable that will return 1 if LTM_{mpt} has a lower value than $s_{mp,t-1}$ at mine $m \in M^{LF}$ for product $p \in P_m$ in period $t \in T$, or 0 otherwise. If the mine regime is FIFO, it is a binary variable that will return 1 if $s_{mp,t-1}$ has a lower value than $\sum_{f\in F} TS_{mpft} \sum_{d\in D_{mp}} \sum_{s\in S} x_{mpfdst}$ at mine $m \in M^{FF}$ for product $p \in P_m$ in period $t \in T$, or 0 otherwise.
LTM_{mpct}	A variable representing $\max\{0, \sum_{f\in F} \sum_{d\in D_{mp}} \sum_{s\in S_{mp}} TS_{mpft}$ $\mathrm{x}_{mpfdst} - IOP_{mpt}\}$.
LTM^{bin}_{mpt}	A binary variable that will return 1 if $\sum_{f\in F} \sum_{d\in D_{mp}}$ $\sum_{s\in S_{mp}} TS_{mpft} x_{mpfdst} - IOP_{mpt}$ is less than 0 at mine $m \in M^{LF}$ for product $p \in P_m$ in period $t \in T$, or 0 otherwise.
RG_{mpct}	Railed grade of component $c \in C$ in product $p \in P_m$ from mine $m \in M$ in period $t \in T$.

$RG^{rail}_{mpcfdst}$	Product of RG_{mpct} and $TS_{mpft}x_{mpfdst}$.
s_{mpt}	Live stockpile level at mine $m \in M$ for mined product $p \in P_m$ in period $t \in T$.
si_{rsct}	Slack variable to penalise when grade of component $c \in C$ of shipped product $s \in S_r$ from port $r \in R$ at time $t \in T$ is off target.
u^{in}_{rst}	Amount transferred from bulk to live stockpiles at port $r \in R$ for shipped product $s \in S_r$ in period $t \in T$.
u^{out}_{rst}	Amount transferred from live to bulk stockpiles at port $r \in R$ for shipped product $s \in S_r$ in period $t \in T$.
v_{rst}	Bulk stockpile level at port $r \in R$ for shipped product $s \in S_r$ in period $t \in T$.
w_{rst}	Live stockpile level at port $r \in R$ for shipped product $s \in S_r$ in period $t \in T$.
$w^{\#}_{rst}$	Live stockpile level at port $r \in R$ for shipped product $s \in S_r$ before lump screening and return fines process in period $t \in T$.
x_{mpfdst}	Number of trains used at mine $m \in M$ for mined product $p \in P_m$ of fleet f at car dumper $d \in D_{mrp}$ for shipped product $s \in S_{mp}$ in period $t \in T$.
y^{in}_{rst}	Amount transferred from bulk to live stockpiles at mine $m \in M$ for mined product $p \in P_m$ in period $t \in T$.
y^{out}_{rst}	Amount transferred from live to bulk stockpiles at mine $m \in M$ for mined product $p \in P_m$ in period $t \in T$.
z_{rst}	Amount of product $s \in S$ shipped from port $r \in R$ in period $t \in T$.
z^{*}_{rst}	Amount of product from the associated lump stockpile of fines product $s \in S^F$ shipped from port $r \in R$ in period $t \in T$.
ZG_{rsct}	Shipped grade of component $c \in C$ in product $s \in S_r$ from port $r \in R$ in period $t \in T$.
$ZG^{shipped}_{rsct}$	Product of ZG_{rsct} and z_{rst}.
$ZG^{\#}_{rsct}$	Product of ZG_{rsct} and $w^{\#}_{rst}$.

3.4.7 Complete formulation

Maximise:

$$
\begin{aligned}
&\sum_{s\in S_r} SP_s \sum_{t\in T}\sum_{r\in R} z_{rst}(1+I)^{1-t} \\
&- \sum_{m\in M}\sum_{p\in P_m}\left[MS^{LE}_{mp}\sum_{t\in T}\alpha^{LE}_{mpt} + MS^{LS}_{mp}\sum_{t\in T}\alpha^{LS}_{mpt}\right] \\
&- \sum_{r\in R}\sum_{s\in S_r}\left[PS^{LE}_{rs}\sum_{t\in T}\beta^{LE}_{rst} + PS^{LS}_{rs}\sum_{t\in T}\beta^{LS}_{rst}\right] \\
&- \sum_{m\in M}\sum_{p\in P_m} MS^{BE}_{mp}\sum_{t\in T}\alpha^{BE}_{mpt} + \sum_{r\in R}\sum_{s\in S_r} PS^{BE}_{rs}\sum_{t\in T}\beta^{BE}_{rst} \\
&- \sum_{m\in M}\sum_{p\in P_m}\left[MB^{out}_{mp}\sum_{t\in T} y^{out}_{mpt} + MB^{in}_{mp}\sum_{t\in T} y^{in}_{mpt}\right] \\
&- \sum_{r\in R}\sum_{s\in S_r}\left[PB^{out}_{rs}\sum_{t\in T} u^{out}_{rst} + PB^{in}_{rs}\sum_{t\in T} u^{in}_{rst}\right] \\
&- \sum_{f\in F} CP_f \sum_{t\in T}\mu_{ft} \\
&- \sum_{m\in M}\sum_{d\in D_m} DP_{md}\sum_{p\in P_m}\sum_{f\in F}\sum_{t\in T} TS_{mpft}x_{mpfdst} \\
&- \sum_{c\in C}\sum_{r\in R}\sum_{s\in S_r} GP_{rsc}\sum_{t\in T}(si_{rsct}+ei_{rsct}) \\
&+ \pi\sum_{s\in S} SP_s \sum_{m\in M}\sum_{p\in P_m}\sum_{f\in F}\sum_{d\in D}\sum_{t\in T} TS_{mpft}x_{mpfdst}
\end{aligned}
$$

subject to:

$$
\begin{aligned}
S^L_{mpt} - \alpha^{LS}_{mpt} \le s_{mpt} \le S^U_{mpt} + \alpha^{LE}_{mpt}, &\qquad \forall m\in M, p\in P_m, t\in T \\
W^L_{rst} - \beta^{LS}_{rst} \le w_{rst} \le W^U_{rst} + \beta^{LE}_{rst}, &\qquad \forall r\in R, s\in S_r, t\in T \\
0 \le b_{mpt} \le B^U_{mpt} + \alpha^{BE}_{mpt}, &\qquad \forall m\in M, p\in P_m, t\in T \\
0 \le v_{rst} \le V^U_{rst} + \beta^{BE}_{rst}, &\qquad \forall r\in R, s\in S_r, t\in T \\
S^U_{mpt} + \alpha^{LE}_{mpt} \le YM_{mpt}, &\qquad \forall m\in M, p\in P_m, t\in T \\
W^U_{rst} + \beta^{LE}_{rst} \le YP_{rst}, &\qquad \forall r\in R, s\in S_r, t\in T
\end{aligned}
$$

$$0 \leq y_{mpt}^{out} \leq y_{mp}^{out,U}, \qquad \forall m \in M, p \in P_m, t \in T$$
$$0 \leq y_{mpt}^{in} \leq y_{mp}^{in,U}, \qquad \forall m \in M, p \in P_m, t \in T$$
$$0 \leq u_{rst}^{out} \leq u_{rs}^{out,U}, \qquad \forall r \in R, s \in S_r, t \in T$$
$$0 \leq u_{rst}^{in} \leq u_{rs}^{in,U}, \qquad \forall r \in R, s \in S_r, t \in T$$

$$\sum_{p \in P_m} \sum_{f \in F} \sum_{d \in D_{mp}} \sum_{s \in S} x_{mpfdst} \leq x_{mt}^U, \qquad \forall m \in M, t \in T$$
$$\sum_{m \in M_g} \sum_{p \in P_m} \sum_{f \in F} \sum_{d \in D_{mrp}} \sum_{s \in S} x_{mpfdst} \leq MT_{gt}, \qquad \forall g \in G, t \in T$$
$$\sum_{m \in M_f} \sum_{p \in P_m} \sum_{d \in D_{mrp}} \sum_{s \in S} x_{mpfdst} \leq MF_{ft}, \qquad \forall f \in F, t \in T$$
$$\sum_{m \in M_f} \sum_{p \in P_m} CT_{mpt} \sum_{d \in D_{mrp}} \sum_{s \in S} x_{mpfdst} \leq PF_{ft} + \mu_{ft}, \qquad \forall f \in F, t \in T$$
$$\sum_{t=1}^{n} JV_{mt}^L \leq \sum_{p \in P_m} \sum_{f \in F} \sum_{d \in D_{mp}} \sum_{s \in S} \sum_{t=1}^{n} x_{mpfdst} \leq \sum_{t=1}^{n} JV_{mt}^U, \qquad \forall m \in M^{JV}, t \in T$$
$$\sum_{m \in M} \sum_{p \in P_m} \sum_{f \in F} \sum_{s \in S} x_{mpfdst} \leq DC_{dt}, \qquad \forall d \in D, t \in T$$
$$\sum_{m \in M} \sum_{p \in P_m} \sum_{f \in F} \sum_{d \in D^{WC}} \sum_{s \in S} x_{mpfdst} \leq WC_t, \qquad \forall t \in T$$
$$0 \leq \sum_{s \in S_r} z_{rst} \leq Z_{rt}^U, \qquad \forall r \in R, t \in T$$

$$s_{mpt} = s_{mp,t-1} + IOP_{mpt} + y_{mpt}^{in} - y_{mpt}^{out} - \sum_{f \in F} TS_{mpft} \sum_{d \in D_{mp}} \sum_{s \in S_{mp}} x_{mpfdst}, \qquad \forall m \in M, p \in P_m, t \in T$$

$$w_{rst} = w_{rs,t-1} + \sum_{m \in M} \sum_{p \in P_m} \sum_{f \in F} TS_{mpft} \sum_{d \in D_{mrp}} x_{mpfdst} + u_{rst}^{in} - u_{rst}^{out} - \frac{z_{rst}}{1 - RF_{rst}}, \qquad \forall r \in R, s \in S^L, t \in T$$

$$w_{rst} = w_{rs,t-1} + \sum_{m \in M} \sum_{p \in P_m} \sum_{f \in F} TS_{mpft} \sum_{d \in D_{mrp}} x_{mpfdst} + u_{rst}^{in} - u_{rst}^{out} - z_{rst} + \frac{z_{rst}^* RF_{rst}^*}{1 - RF_{rst}^*}, \qquad \forall r \in R, s \in S^F, t \in T$$

$$w^{\#}_{rst} = w_{rs,t-1} - u^{out}_{rst} + u^{in}_{rst} + \sum_{m\in M}\sum_{p\in P_m}\sum_{f\in F} TS_{mpft} \sum_{d\in D_{mrp}} x_{mpdst}, \qquad \forall r \in R, s \in S_r, t \in T$$

$$b_{mpt} = b_{mp,t-1} + y^{out}_{mpt} - y^{in}_{mpt}, \qquad \forall m \in M, p \in P_m, t \in T$$

$$v_{rst} = v_{rs,t-1} + u^{out}_{rst} - u^{in}_{rst}, \qquad \forall r \in R, s \in S_r, t \in T$$

$$LM_{mpct} = \frac{LM^{live}_{mpc,t-1} + LM^{out}_{mpct} + IOG_{mpct}IOP_{mpt} + BM^{in}_{mpct} - \sum_{f\in F}\sum_{d\in D_{mp}}\sum_{s\in S_{mp}} RG^{rail}_{mpcfdst}}{s_{mpt}}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$

$$BM_{mpct} = \frac{BM^{bulk}_{mpc,t-1} - BM^{in}_{mpct} + LM^{out}_{mpct}}{b_{mpt}}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$

$$RG_{mpct} = \frac{IOG_{mpct}IOT_{mpt} + LM_{mpc,t-1}LT_{mpt} + BM_{mpct}BT_{mpt}}{\sum_{f\in F}\sum_{d\in D_{mrp}}\sum_{s\in S_{mp}} TS_{mpft}x_{mpfdst}}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$

$$LP_{rsct} = \frac{LP^{live}_{rsc,t-1} - LP^{out}_{rsct} + BP^{in}_{rsct} + \sum_{m\in M}\sum_{p\in P_m}\sum_{f\in F}\sum_{d\in D_{mrp}} RG^{rail}_{mpcfdst}}{w^{\#}_{rst}}, \qquad \forall r \in R, s \in S_r, c \in C, t \in T$$

$$LP_{rsct} = \frac{ZG^{\#}_{rsct} + LP^{rf}_{rsct}}{w^{\#}_{rst} + \frac{z^*_{rst}RF^*_{rst}}{1-RF^*_{rst}}}, \qquad \forall r \in R, s \in S_r, c \in C, t \in T$$

$$BP_{rsct} = \frac{BP^{bulk}_{rsc,t-1} - BP^{in}_{rsct} + LP^{out}_{rsct}}{v_{rst}}, \qquad \forall r \in R, s \in S_r, c \in C, t \in T$$

$$ZG_{rsct} = \frac{LP^{live}_{rsc,t-1} - LP^{out}_{rsct} + BP^{in}_{rsct} + \sum_{m\in M}\sum_{p\in P_m}\sum_{f\in F}\sum_{d\in D_{mrp}} RG^{rail}_{mpcfdst}}{w^{\#}_{rst}}, \qquad \forall r \in R, s \in S_r, c \in C, t \in T$$

$$LM^{live}_{mpct} = LM_{mpct} s_{mpt}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$
$$LM^{out}_{mpct} = LM_{mpct} y^{out}_{mpt}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$
$$BM^{in}_{mpct} = BM_{mpct} y^{in}_{mpt}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$
$$BM^{bulk}_{mpct} = BM_{mpct} b_{mpct}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$
$$LP^{live}_{rsct} = LP_{rsct} w_{rst}, \qquad \forall r \in R, s \in S_r, c \in C, t \in T$$
$$LP^{out}_{rsct} = LP_{rsct} u^{out}_{rst}, \qquad \forall r \in R, s \in S_r, c \in C, t \in T$$
$$LP^{rf}_{rsct} = LP^{*}_{rsct} \frac{z^{*}_{rst} RF^{*}_{rst}}{1 - RF^{*}_{rst}}, \qquad \forall r \in R, s \in S_r, c \in C, t \in T$$
$$BP^{in}_{rsct} = BP_{rsct} u^{in}_{rst}, \qquad \forall r \in R, s \in S_r, c \in C, t \in T$$
$$BP^{bulk}_{rsct} = BP_{rsct} v_{rsct}, \qquad \forall r \in R, s \in S_r, c \in C, t \in T$$
$$ZG^{shipped}_{rsct} = ZG_{rsct} z_{rst}, \qquad \forall r \in R, s \in S_r, c \in C, t \in T$$
$$ZG^{\#}_{rsct} = ZG_{rsct} w^{\#}_{rst}, \qquad \forall r \in R, s \in S_r, c \in C, t \in T$$

$$RG^{rail}_{mpcfdst} = RG_{mpct} TS_{mpft} x_{mpfdst}, \qquad \forall m \in M, p \in P_m, c \in C, f \in F, d \in D_{mp}, s \in S_{mp}, t \in T$$

$$IOT_{mpt} \leq IOP_{mpt}, \qquad \forall m \in M^{LF}, p \in P_m, t \in T$$
$$IOT_{mpt} \leq \sum_{f \in F} TS_{mpft} \sum_{d \in D_{mp}} \sum_{s \in S} x_{mpfdst}, \qquad \forall m \in M^{LF}, p \in P_m, t \in T$$

$$IOT_{mpt} \geq \sum_{f \in F} TS_{mpft} \sum_{d \in D_{mp}} \sum_{s \in S} x_{mpfdst} - (10,000 - IOP_{mpt}) IOT^{bin}_{mpt}, \qquad \forall m \in M^{LF}, p \in P_m, t \in T$$
$$IOT_{mpt} \geq IOP_{mpt} - IOP_{mpt}(1 - IOT^{bin}_{mpt}), \qquad \forall m \in M^{LF}, p \in P_m, t \in T$$

$$LTM_{mpt} \geq 0, \qquad \forall m \in M^{LF}, p \in P_m, t \in T$$

$$LTM_{mpt} \geq \sum_{f \in F} TS_{mpft} \sum_{d \in D_{mp}} \sum_{s \in S} x_{mpfdst} - IOP_{mpt}, \qquad \forall m \in M^{LF}, p \in P_m, t \in T$$
$$LTM_{mpt} \leq \sum_{f \in F} TS_{mpft} \sum_{d \in D_{mp}} \sum_{s \in S} x_{mpfdst} - IOP_{mpt} + 10,000 LTM^{bin}_{mpt}, \qquad \forall m \in M^{LF}, p \in P_m, t \in T$$

$$LTM_{mpt} \le 10{,}000(1 - LTM^{bin}_{mpt}), \qquad \forall m \in M^{LF}, p \in P_m, t \in T$$
$$LT_{mpt} \le s_{mp,t-1}, \qquad \forall m \in M^{LF}, p \in P_m, t \in T$$
$$LT_{mpt} \le LTM_{mpt}, \qquad \forall m \in M^{LF}, p \in P_m, t \in T$$

$$LT_{mpt} \ge s_{mp,t-1} - IOP_{mpt} - 20{,}000 LT^{bin}_{mpt}, \qquad \forall m \in M^{LF}, p \in P_m, t \in T$$
$$LT_{mpt} \ge LTM_{mpt} - 20{,}000(1 - LT^{bin}_{mpt}), \qquad \forall m \in M^{LF}, p \in P_m, t \in T$$

$$IOTM_{mpt} \ge 0, \qquad \forall m \in M^{FF}, p \in P_m, t \in T$$
$$IOTM_{mpt} \le 10{,}000(1 - IOTM^{bin}_{mpt}), \qquad \forall m \in M^{FF}, p \in P_m, t \in T$$

$$IOTM_{mpt} \ge \sum_{f \in F} \sum_{d \in D_{mp}} \sum_{s \in S_{mp}} TS_{mpft} x_{mpfdst} - s_{mp,t-1}, \qquad \forall m \in M^{FF}, p \in P_m, t \in T$$
$$IOTM_{mpt} \le \sum_{f \in F} \sum_{d \in D_{mp}} \sum_{s \in S_{mp}} TS_{mpft} x_{mpfdst} - s_{mp,t-1} + 10{,}000 IOTM^{bin}_{mpt}, \qquad \forall m \in M^{FF}, p \in P_m, t \in T$$

$$IOT_{mpt} \le IOP_{mpt}, \qquad \forall m \in M^{FF}, p \in P_m, t \in T$$
$$IOT_{mpt} \le IOTM_{mpt}, \qquad \forall m \in M^{FF}, p \in P_m, t \in T$$

$$IOT_{mpt} \ge IOP_{mpt} - (IOP_{mpt} + 10{,}000) IOT^{bin}_{mpt}, \qquad \forall m \in M^{FF}, p \in P_m, t \in T$$
$$IOT_{mpt} \ge IOTM_{mpt} - (10{,}000 - IOP_{mpt})(1 - IOT^{bin}_{mpt}), \qquad \forall m \in M^{FF}, p \in P_m, t \in T$$

$$LT_{mpt} \le s_{mp,t-1}, \qquad \forall m \in M^{FF}, p \in P_m, t \in T$$
$$LT_{mpt} \le \sum_{f \in F} TS_{mpft} \sum_{d \in D_{mp}} \sum_{s \in S} x_{mpfdst}, \qquad \forall m \in M^{FF}, p \in P_m, t \in T$$
$$LT_{mpt} \ge s_{mp,t-1} - 20{,}000(1 - LT^{bin}_{mpt}), \qquad \forall m \in M^{FF}, p \in P_m, t \in T$$

$$LT_{mpt} \geq \sum_{f \in F} TS_{mpft} \sum_{d \in D_{mp}} \sum_{s \in S} x_{mpfdst} - 10,000 LT^{bin}_{mpt}, \qquad \forall m \in M^{FF}, p \in P_m, t \in T$$

$$BT_{mpt} \geq 0, \qquad \forall m \in M, p \in P_m, t \in T$$

$$BT_{mpt} \leq 10,000(1 - BT^{bin}_{mpt}), \qquad \forall m \in M, p \in P_m, t \in T$$

$$BT_{mpt} \geq \sum_{f \in F} \sum_{d \in D_{mp}} \sum_{s \in S_{mp}} TS_{mpft} x_{mpfdst} - IOP_{mpt} - s_{mp,t-1}, \qquad \forall m \in M, p \in P_m, t \in T$$

$$BT_{mpt} \leq \sum_{f \in F} \sum_{d \in D_{mp}} \sum_{s \in S_{mp}} TS_{mpft} x_{mpfdst} - IOP_{mpt} + 10,000 BT^{bin}_{mpt}, \qquad \forall m \in M, p \in P_m, t \in T$$

$$ZG^{shipped}_{rsct} \geq (TG_{rsct} - TG^{tol}_{rsc}) z_{rst} - si_{rsct}, \qquad \forall r \in R, s \in S_r, c \in C, t \in T$$

$$ZG^{shipped}_{rsct} \leq (TG_{rsct} + TG^{tol}_{rsc}) z_{rst} + ei_{rsct}, \qquad \forall r \in R, s \in S_r, c \in C, t \in T$$

3.5 Solution approach

Solution approaches have been developed to tackle the non-linearities in the model (Singh et al., 2014). These approaches include an iterative method which is based on the SLP method and two heuristics methods (see Section 2.3.4), based on sliding time window heuristic. In their current operation, RTIO completely relies on the iterative method, which will be discussed at length in this section and applied in our implementation for the case problems.

The iterative method involves multi-stage algorithm where the number of iterations is determined beforehand. Figure 3.4 describes the flowchart of the step-by-step iterative method.

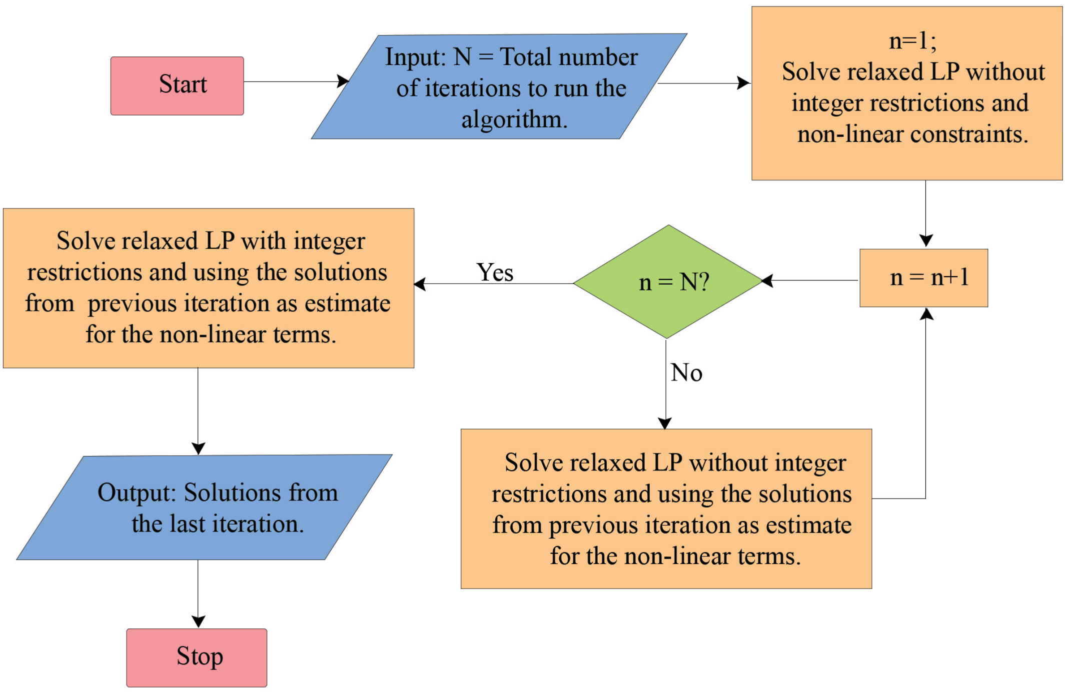

Figure 3.4: Flowchart of the iterative method

We firstly determine the number of iterations for which we run the algorithm. Note that the number of iterations N has to be greater than one. In the first iteration, we solve the problem where we relax the integer restrictions and grade constraints which are non-linear. In this stage, the problem is a simple linear programming.

In the next iterations, we include the grade constraints with the solutions from the previous iteration are used as estimates of the non-linear terms. This approach will linearise the non-linear constraints and simplify the model to linear programming. We repeat this step for a number of iterations and bring back the integer restrictions in the last iteration.

3.6 Test cases

In order to ensure the reliability of our implementation of the model, we run test cases using a real data set given by RTIO. Two small problems are considered; both are taken from the same data set. The first test case problem describes 5 periods of planning, while the second describes 11 periods. As each period represents few days to one week, the problems then consider approximately one and two months of planning respectively. Problems of larger size will be attempted as case studies in Chapter 5. We run our implementations in AIMMS 4.21 software with linkage to CPLEX 12.6.3 solver using 1% MIP relative tolerance gap as the termination criterion.

3.6.1 Preliminary

In this section, we define the phrases that will come up in the results analysis. Note that these definitions apply in the results analysis in Chapter 4 and 5 as well.

- **Solving time** is defined as the amount of time taken to run the implementation. We measure the solving time in seconds.
- **Gap** is the percentage of the MIP gap achieved.
- **Number of trains** is defined as the number of trains in total required over the full period.
- **Total amount of shipping** is the amount of material in kt shipped over the full period.
- **Total profit** is defined as the amount of profit in total over the full period. The total profit is also the objective function to be maximised in the problem.
- **Grade deviation cost** is defined as the amount of grade deviation cost in total over the full period.
- **Number of iterations** is defined as the number of iterations applied in the iterative method.

3.6.2 5-period case

We firstly solve the 5-period case without the grade quality requirements. In this case, the grade deviation cost (3.7) and the grade constraints (3.32)–(3.89) are omitted when running the optimisation tool. The results obtained are then used to calculate the grade deviation cost. As the problem involves no non-linear constraints, the iterative method is not needed. The results are displayed in the table below.

Solving time (seconds)	0.27
Gap (%)	0.7
Number of trains	873
Total amount of shipping (kt)	22,939.01
Total profit ($)	1,141,752.31
Grade deviation cost ($)	187,995.58

Table 3.1: Summary of results for the 5-period case without grade constraints

We then include the grade deviation cost in the objective function and all the grade constraints to the problem. This means, the full formulation of the model is considered. As the problem is MINLP, we apply the iterative algorithm outlined in Section 3.5 with up to 8 iterations.

As the size of the problem is very small, the solver found solutions within 1% of gap tolerance instantly, even with the highest number of iterations. This results can be seen in Table 3.2.

Table 3.3 displays the summary of results for the number of trains, total amount of shipping, total profit, and grade deviation cost for the 5-period case problem incorporating the grade constraints. The total number of trains needed to transport the iron ore from the mines to the ports remains unchanged across different number of iterations. Meanwhile, the total amount of shipping, total profit, and grade deviation cost fluctuate as different number of iterations are applied.

3.6.3 11-period case

The second test case problem is taken from the same data set with 11 periods and the same start of planning period. This problem then is an extension to the the previous 5-period case. Similarly as the first test case, we start our implementations by solving the problem without the grade quality constraints and the grade deviation cost in the objective function. Again, the

Number of iterations	Solving time (seconds)	Gap (%)
2	0.48	0.83
3	0.53	0.89
4	0.61	0.90
5	0.72	0.84
6	0.77	0.92
7	0.90	0.92
8	0.93	0.92

Table 3.2: Solving time and solution gap for the 5-period case using iterative method

Number of iterations	Number of trains	Total shipping (kt)	Total profit ($)	Grade deviation cost ($)
2	873	22,939.01	1,116,346.80	213,401.10
3	873	22,962.15	1,106,949.13	222,289.41
4	873	22,962.15	1,105,820.45	223,418.09
5	873	22,939.01	1,101,512.85	228,234.63
6	873	22,954.91	1,105,522.72	223,418.34
7	873	22,954.91	1,105,522.90	223,418.16
8	873	22,954.91	1,105,522.90	223,418.16

Table 3.3: Number of trains, total amount of shipping, total profit, and grade deviation cost for the 5-period case using iterative method

iterative method is not needed as the problem is a regular MILP problem. The summary of the results is shown in the table below.

Solving time (seconds)	3.60
Gap (%)	0.27
Number of trains	1,842
Total amount of shipping (kt)	45,754.34
Total profit ($)	1,969,437.50
Grade deviation cost ($)	663,727.37

Table 3.4: Summary of results for the 11-period case without grade constraints

We then bring the grade deviation cost and grade constraints back to the problem and apply the iterative algorithm outlined in Section 3.5 with up to 8 iterations. The solving times and gaps are listed in Table 3.5 below.

Number of iterations	Solving time (seconds)	Gap (%)
2	4.21	0.32
3	4.87	0.36
4	5.01	0.36
5	5.04	0.26
6	5.39	0.40
7	6.00	0.47
8	5.93	0.23

Table 3.5: Solving time and solution gap for the 11-period case using iterative method

The number of trains, shipping amount, total profit, and grade deviation cost for the full problem of 11-period test case are summarised in Table 3.6. Similarly as the previous case problem, the results show random fluctuation as different number of iterations are used.

3.6.4 Discussion

Solving time and solution gap

The solving times and solution gaps for the first and second test cases are summarised in the table below:

Number of iterations	Number of trains	Total shipping (kt)	Total profit ($)	Grade deviation cost ($)
2	1,841	45,725.14	2,011,251.97	622,513.45
3	1,842	45,753.84	1,991,286.42	641,707.59
4	1,842	45,753.84	2,017,672.63	615,321.39
5	1,841	45,724.90	2,023,004.68	611,922.11
6	1,840	45,697.84	1,983,886.56	648,300.45
7	1,840	45,699.69	2,018,881.57	611,922.21
8	1,842	45,756.45	2,023,559.72	611,922.08

Table 3.6: Number of trains, total amount of shipping, total profits, and grade deviation costs for the 11-period case using iterative method

	Number of	Solving time (s)		Gap (%)	
	iterations	5 periods	11 periods	5 periods	11 periods
Without grades	-	0.27	3.60	0.7	0.27
With grades	2	0.48	4.21	0.83	0.32
	3	0.53	4.87	0.89	0.36
	4	0.61	5.01	0.90	0.36
	5	0.72	5.04	0.84	0.26
	6	0.77	5.39	0.92	0.40
	7	0.90	6.00	0.92	0.47
	8	0.93	5.93	0.92	0.23

Table 3.7: Summary of solving times and solution gaps for the 5-period and 11-period case problems

The problem size of both test cases are considered small, and hence it does not take long for the AIMMS to find solutions within small solution gaps. In this case, both models without and with the grade constraints are reliable in terms of finding solutions within reasonable times.

Number of trains and total shipping

As shown in Table 3.8 below, it is not clear whether the grade constraints have any effect on the total number of trains and the total amount of shipping. On average, solving the full problems with the iterative method leads to very small changes (in some cases, no change) in the number of trains and amount of shipping, compared to the same problems without the grade constraints.

	Number of iterations	Number of trains		Total shipping (kt)	
		5 periods	11 periods	5 periods	11 periods
Without grades	-	873	1,842	22,939.01	45,754.34
With grades	2	873	1,841	22,939.01	45,725.14
	3	873	1,842	22,962.15	45,753.84
	4	873	1,842	22,962.15	45,753.84
	5	873	1,841	22,939.01	45,724.90
	6	873	1,840	22,954.91	45,697.84
	7	873	1,840	22,954.91	45,699.69
	8	873	1,842	22,954.91	45,756.45

Table 3.8: Summary of number of trains and total shipping amount for the 5-period and 11-period case problems

Total profit

We put more attention to the total profit when comparing our results since it is our objective function in the model's formulation. The results for the total profit for the 5-period and 11-period problems (both without and with the grade constraints) are summarised in Figures 3.5 and 3.6.

Figure 3.5 shows that adding the grade constraints to the 5-period problem reduces the objective function by the average of 3%. On the contrary, Figure 3.6, indicates an increase of objective function by the average of 2% when the grade constraints are added to the 11-period problem. It is interesting to see the results of implementing much bigger problems in our case studies in Chapter 5.

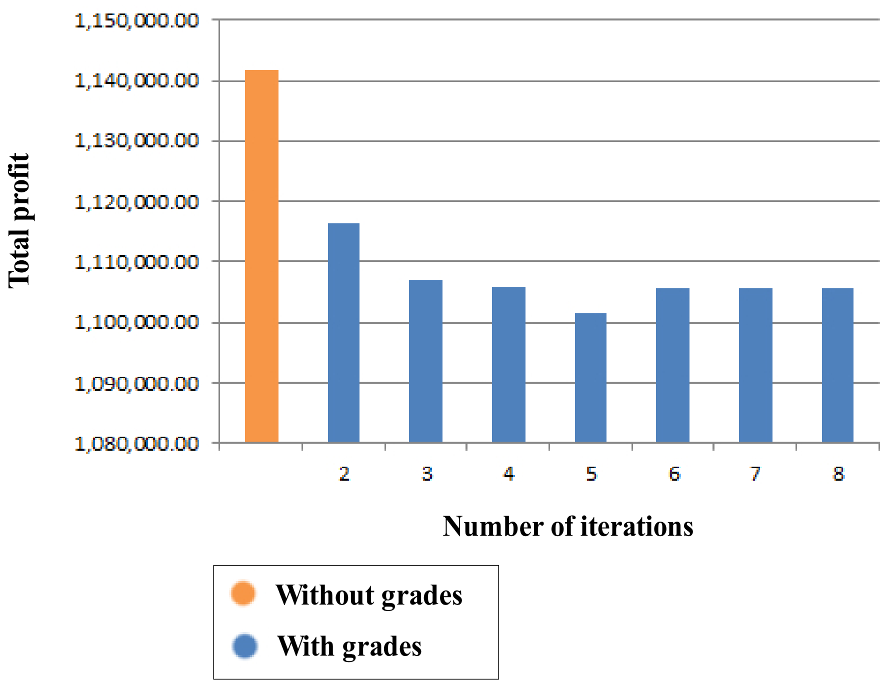

Figure 3.5: Total profit for the 5-period case using iterative method

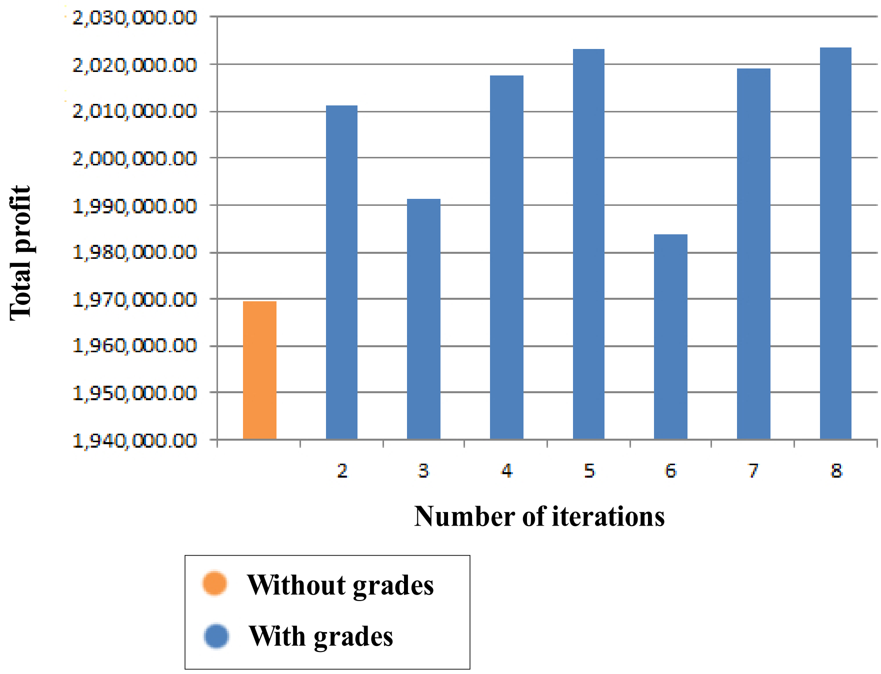

Figure 3.6: Total profit for the 11-period case using iterative method

The solutions achieved by applying different number of iterations for both case problems are also observed. The results fluctuate and do not indicate any convergence.

Grade deviation cost

Apart from maximising the total throughput and sales with minimum operational costs, meeting the target demands of the iron ore grades is also important in RTIO's operations. This is reflected in the grade deviation cost as part of the objective function. Figures 3.7 and 3.8 shows the comparison of results of the grade deviation costs without and with the grade constraints for the 5 and 11-period problems.

We can see that adding the grade constraints results in higher grade deviation costs for the 5-period problem. On average, the increase is 18% of the cost generated by the model without the grade constraints. As the problem size gets larger, however, lower grade deviation costs are attained by the problem with the grade constraints. This is indicated by the results for the 11-period problem where the average of 6% decrease in the grade deviation cost is achieved by incorporating the grade constraints. Moreover, the solutions do not appear to converge for both case problems, as more iterations are used.

3.7 Conclusions

In this chapter, we have extended the optimisation model and solution approach presented in (Garcıa-Flores et al., 2011) based on RTIO's current operation. We have implemented the model in AIMMS optimisation modelling software with CPLEX solver to solve the MILP problems.

This model involves allocating trains to mines such that operational costs are minimised and total throughput is maximised, taking into account the grade quality demands of the products. In addition, the incentives in the objective function encourages higher number of trains, hence increasing the total throughput.

We have run our own implementation on two test case problems of small size as validation tests. Both cases are extracted from a real data set given by RTIO. We apply the iterative method presented in Section 3.5 since it has been solely relied on by RTIO as their approach of tackling the non-linear constraints in the formulation. The method relaxes the problem so that the original MINLP problem can be run in CPLEX solver as some MILP problems.

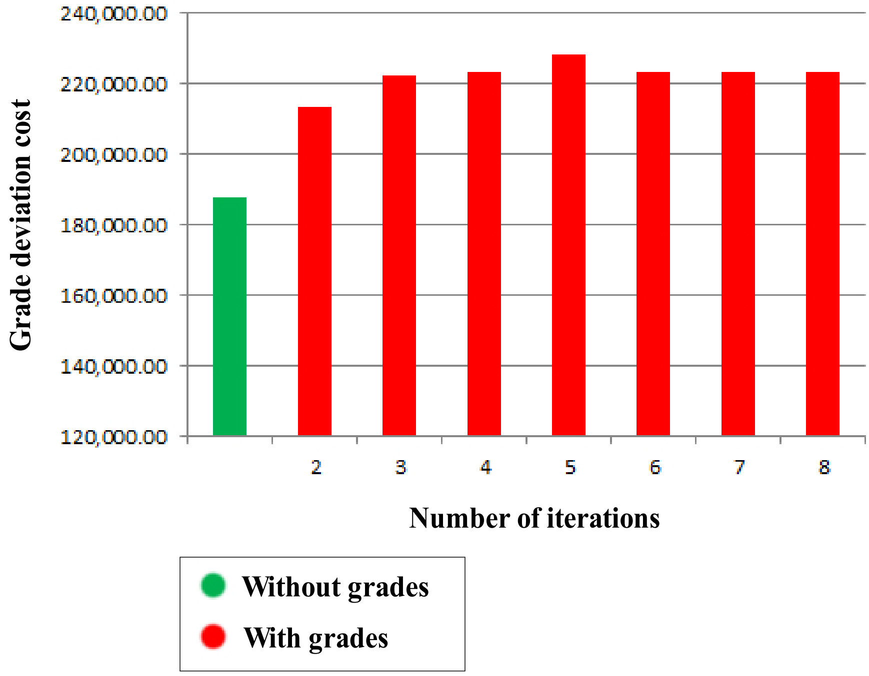

Figure 3.7: Total cost of grade deviations for the 5-period case using iterative method

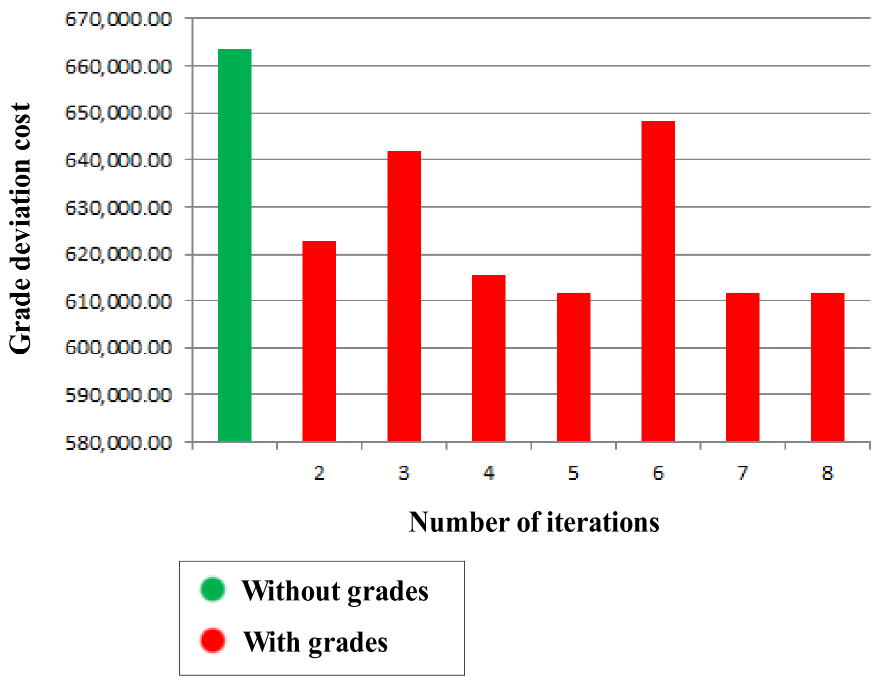

Figure 3.8: Total cost of grade deviations for the 11-period case using iterative method

For each test case, two different models were implemented; one without the grade constraints, and the other one with the grade constraints. We solved the problem without the grade constraints prior to solving the full problem in order to analyse the impacts of adding the non-linear constraints. For the full problems, we considered up to eight iterations and 1% gap of tolerance. The results have been shown in Tables 3.1–3.6.

All solutions achieved are reasonable, and hence the model is validated. Including the grade constraints presents very small impacts on the solving time, the number of trains, and the total amount of shipping. In regards to the total profit and the grade deviation cost, however, the model shows some inconsistencies. Adding the grade constraints generates worse solutions for the 5-period problem, in comparison to the same problem without the grade constraints; but performs better for the 11-period problem. Furthermore, the results do not show any convergence as more iterations are used; making it difficult to decide the number of iterations that should be used.

In the forthcoming chapter, we will present a new model of the same mining operation, along with some implementations.

Chapter 4

Convex Relaxation for Global Optimisation

The optimisation tool developed by Singh et al. (2014) and presented in the previous chapter produces plans that find the allocation of trains to mines and assist in medium to long term planning of logistics operations. It has performed consistently better than the former computer-based approach in terms of the quality of the results obtained and the time taken to solve the problem. The solution approach to tackle the non-linear constraints, however, is not reliable as the results of our implementation showed some inconsistencies in terms of the significance of including the grade constraints. Furthermore, there is no rigorous way of determining the number of iterations that should be used in applying the iterative method. As the company's plan to increase their capacity takes place, there is a need to develop a new solution approach, primarily on how it handles the non-linearity occurring in the formulation. A more effective approach with a more solid and rigorous theory behind it is needed to solve the optimisation models representing medium to long term logistics planning in iron ore mining. In this chapter, we introduce a different approach to relax the MINLP problem and present the implementations.

4.1 Introduction

The problem formulation of the optimisation model for logistics planning in mining described in Chapter 3 uses a mixed integer non-linear programming approach. A solution approach which uses iterations has been developed to approximate the non-linearity arising in the model. In this chapter, we present an alternative approach by solving a convex MILP relaxation of the original MINLP model. The relaxed model is obtained by using the concept of convex and concave envelopes (see McCormick, 1976) to replace the non-linear terms appearing in the formulation.

The convex envelope of a function over a non-empty convex set is the highest convex underestimator of the function. In a similar way, the concave envelope of a function over a non-empty convex set is the lowest concave overestimator of the function.

In Section 4.2, we present a procedure for obtaining the convex underestimators and concave overestimators of factorable functions. We outline the derivation of the convex and concave envelopes for bilinear terms and use one of the bilinear terms in our problem formulation to provide an example.

The new problem formulation is presented in Section 4.3. This problem formulation is a MILP relaxed version of the formulation in the previous chapter. As non-linearity only appears in grade quality constraints, only such constraints are replaced. In brief, this section includes the assumptions (Section 4.3.1), the objective function (Section 4.3.2), operational constraints (Section 4.3.3), constraints on iron ore grades (Section 4.3.4), the summary of notations (Section 4.3.5), and the complete summary of the problem formulation (Section 4.3.6).

Some test cases have been implemented for validation purposes. We use the same cases as in Section 3.6. We also use AIMMS optimisation software to build the model and link the software to CPLEX 12.6.3 solver. The results will be outlined and discussed in Section 4.4. The conclusion of the chapter is presented in Section 4.5.

4.2 Convex relaxation

The arising of non-linearity in the model formulation described in Chapter 3 is due to the appearance of bilinear terms in iron ore grade quality constraints. A MILP relaxation can be obtained by replacing those non-linear terms with individual terms and adding the convex functions that underestimate and concave functions that overestimate the corresponding terms everywhere in the domain of interest. This method was initiated by McCormick (1976).

4.2.1 Procedure

In this section, we outline the procedure of obtaining the underestimating convex and overestimating concave functions for factorable functions, as also summarised in McCormick (1976).

Suppose we have a continuous function of a single variable $F\big(f(x)\big)$ where $f(x)$ is a continuous function of one or multi-variables. We assume that for all x in a convex set S, there exist a convex function $c_f(x)$ and a concave function $C_f(x)$ such that

$$c_f(x) \leq f(x) \leq C_f(x)$$

and also

$$f^L \leq f(x) \leq f^U$$

where f^L and f^U are some scalars.

It is also assumed that the generation of the convex envelope and concave envelopes for the function $F\big(f(x)\big)$ in the interval $[f^L, f^U]$, denoted by e_F and E_F respectively, is always possible. The convex envelope of a function is defined as the highest underestimating convex function over a closed convex set, while the concave envelope is the lowest overestimating concave function over a closed convex set.

Next, it is required to find points $y^F_{\min}$ and $y^F_{\max}$ which are the points at which the infimum and supremum of function $F(y)$ are achieved. That is, we want to compute $y^F_{\min}$ and $y^F_{\max}$ such that

$$\begin{aligned} F(y^F_{\min}) &= \inf_{f^L \leq y \leq f^U} F(y) \\ F(y^F_{\max}) &= \sup_{f^L \leq y \leq f^U} F(y). \end{aligned}$$

Finally the convex underestimating function of $F[f(x)]$ is given by

$$F[f(x)] \geq e_F\big(\text{mid}\{c_f(x), C_f(x), y^F_{\min}\}\big), \text{ for } x \in S \cap \{x | f^L \leq f(x) \leq f^U\}$$

where the mid function returns the middle value of the specified scalars.

Similarly, the concave overestimating function of $F[f(x)]$ is given by

$$F[f(x)] \leq E_F\big(\text{mid}\{c_f(x), C_f(x), y^F_{\max}\}\big), \text{ for } x \in S \cap \{x | f^L \leq f(x) \leq f^U\}.$$

Using the same procedure, we can obtain the convex underestimating and concave overestimating functions for a bilinear term. Suppose our function is in the following form:

$$G\big(g(x)\big)H\big(h(x)\big)$$

where G and H are continuous functions of a single variable and g and h are continuous functions of multi-variables.

We can assume that for all x in a convex set S, there exist convex functions $c_g(x)$ and $c_h(x)$, and concave functions $C_g(x)$ and $C_h(x)$ such that

$$
\begin{aligned}
c_g(x) &\le g(x) \le C_g(x) \\
c_h(x) &\le h(x) \le C_h(x)
\end{aligned}
$$

and also

$$
\begin{aligned}
g^L &\le g(x) \le g^U \\
h^L &\le h(x) \le h^U
\end{aligned}
$$

where g^L, h^L, g^U, and h^U are some scalars.

For the intervals $G^L \le G \le G^U$ and $H^L \le H \le H^U$, the respective convex and concave envelopes for the product term are given by

$$
\begin{aligned}
&\max\left\{H^U G + G^U H - G^U H^U, H^L G + G^L H - G^L H^L\right\} \\
&\min\left\{H^L G + G^U H - G^U H^L, H^U G + G^L H - G^L H^U\right\}.
\end{aligned}
$$

The computation of these envelopes is detailed in the subsequent section.

We compute $y^G_{\min}$ and $y^G_{\max}$, as well as $y^H_{\min}$ and $y^H_{\max}$ such that

$$
\begin{aligned}
G(y^G_{\min}) &= \inf_{g^L \le y \le g^U} G(y) \\
G(y^G_{\max}) &= \sup_{g^L \le y \le g^U} G(y) \\
H(y^H_{\min}) &= \inf_{h^L \le y \le h^U} H(y) \\
H(y^H_{\max}) &= \sup_{h^L \le y \le h^U} H(y).
\end{aligned}
$$

Therefore, it follows that the convex underestimating function of the product term $G\big(g(x)\big)H\big(h(x)\big)$ is given by

$$
\begin{aligned}
G\big(g(x)\big)H\big(h(x)\big) \ge \max\big\{&H^U e_G\big(\mathrm{mid}\{c_g(x), C_g(x), y^G_{\min}\}\big)+ \\
&G^U e_H\big(\mathrm{mid}\{c_h(x), C_h(x), y^H_{\min}\}\big) - G^U H^U, \\
&H^L e_G\big(\mathrm{mid}\{c_g(x), C_g(x), y^G_{\min}\}\big)+ \\
&G^L e_H\big(\mathrm{mid}\{c_h(x), C_h(x), y^H_{\min}\}\big) - G^L H^L\big\}
\end{aligned}
$$

for $x \in S \cap \{x | g^L \le g(x) \le g^U, h^L \le h(x) \le h^U\}$.

Similarly, the concave overestimating function of $G[g(x)]H[h(x)]$ is given by

$$
\begin{aligned}
G[g(x)]H[h(x)] \le \min\big\{&H^L E_G\big(\mathrm{mid}\{c_g(x), C_g(x), y^G_{\max}\}\big)+ \\
&G^U E_H\big(\mathrm{mid}\{c_h(x), C_h(x), y^H_{\max}\}\big) - G^U H^L, \\
&H^U E_G\big(\mathrm{mid}\{c_g(x), C_g(x), y^G_{\max}\}\big)+ \\
&G^L E_H\big(\mathrm{mid}\{c_h(x), C_h(x), y^H_{\max}\}\big) - G^L H^U\big\}
\end{aligned}
$$

for $x \in S \cap \{x | g^L \leq g(x) \leq g^U, h^L \leq h(x) \leq h^U\}$.

4.2.2 Envelopes for bilinear terms

The convex and concave envelopes for a bilinear term is derived in this section. Suppose we have a bilinear term $z = xy$. Variables x and y have x^L, x^U, y^L, and y^U as their lower and upper bound values respectively such that

$$\begin{aligned} x^L &\leq x \leq x^U \\ y^L &\leq y \leq y^U. \end{aligned}$$

Let p and q be the non-negative difference between the variables and their lower bounds. We have

$$\begin{aligned} p &= (x - x^L) \\ q &= (y - y^L) \end{aligned}$$

As p and q are non-negative, the product of p and q is also non-negative. That is, $pq \geq 0$. Thus, we have

$$\begin{aligned} pq &= (x - x^L)(y - y^L) \\ &= xy - x^L y - y^L x + x^L y^L \geq 0 \\ xy &\geq x^L y + y^L x - x^L y^L \end{aligned}$$

Now we let r and s be the non-negative difference between the variables and their upper bounds. Therefore, we have

$$\begin{aligned} r &= (x^U - x) \\ s &= (y^U - y) \end{aligned}$$

We then have the following equation for the product of r and s:

$$\begin{aligned} rs &= (x^U - x)(y^U - y) \\ &= xy - x^U y - y^U x + x^U y^U \geq 0 \\ xy &\geq x^U y + y^U x - x^U y^U. \end{aligned}$$

Consequently, we obtain the convex envelopes as follows:

$$\begin{aligned} z &\geq x^L y + y^L x - x^L y^L \\ z &\geq x^U y + y^U x - x^U y^U. \end{aligned}$$

By following the same steps for the product terms rq and ps, we obtain the following concave envelopes:

$$z \leq x^U y + y^L x - x^U y^L$$
$$z \leq x^L y + y^U x - x^L y^U.$$

4.2.3 Example

In this section, we provide an example of creating the convex underestimating and concave overestimating functions using a non-linear term arising in our model formulation. For interested readers, more examples can be found in Horst and Tuy (1993) and Tawarmalani and Sahinidis (2002).

We choose the non-linear term in equation (3.33) as an example. The term is bilinear as it involves multiplication of two decision variables, namely the live stockpile grade at a mine and the stockpile level:

$$LM^{live}_{mpct} = LM_{mpct} s_{mpt}.$$

To create the underestimating and overestimating functions, we firstly need to introduce new parameters, namely the lower and upper bounds for the grade composition variables. The lower and upper bounds for the live stockpile grade quality of component c in product p at mine m are given by LM^L_{mpct} and LM^U_{mpct} respectively. The live stockpile level s_{mpt} has existing bounds, namely zero as the lower bound and the stockpile yard limit as its upper bound. Both LM_{mpct} and s_{mpt} must lie within their specified bounds, that is:

$$LM^L_{mpct} \leq LM_{mpct} \leq LM^U_{mpct},$$
$$0 \leq s_{mpt} \leq YLM_{mpt}.$$

By following the procedure in Section 4.2.1, we obtain the following inequalities for convex underestimators:

$$LM^{live}_{mpct} \geq LM^L_{mpct} s_{mpt},$$
$$LM^{live}_{mpct} \geq YLM_{mpt} LM_{mpct} + LM^U_{mpct} s_{mpt} - YLM_{mpt} LM^U_{mpct}.$$

Similarly, we also obtain the following inequalities for concave overestimators:

$$LM^{live}_{mpct} \leq LM^L_{mpct} s_{mpt} + YLM_{mpt} LM_{mpct} - LM^L_{mpct} YLM_{mpt},$$
$$LM^{live}_{mpct} \leq LM^U_{mpct} s_{mpt}.$$

In the new formulation, the equation (3.33) is removed and the corresponding inequalities describing convex and concave envelopes are added.

We apply the same procedure to every bilinear term in iron ore grade constraints. As the convex and concave envelope functions are all linear, this procedure will relax non-linearities in the formulation.

4.3 Problem formulation

This section presents a modified version of the problem formulation described in the previous chapter. The nature of the problem is still the same. The objective is to maximise the total profit which is expressed by total revenue less total penalties subject to operational capacity constraints, contractual obligations, and grade quality requirements.

Following the convex relaxation procedure outlined in Section 4.2.1, all non-linear terms are replaced w single variables, making the problem of convex MILP. Consequently, the solution approaches developed in Chapter 3 are not needed for this model.

4.3.1 Assumptions

All assumptions made in the previous model still apply in this model. In addition, the lower and upper bounds for iron ore grades are estimated based on the grades of products coming in to each stockpile. In summary, the assumptions made for this model are listed below:

Bounds for stockpile grades

The iron ore grade at each stockpile cannot be lower than the minimum of the stockpile grade in the previous period and the grades of products coming into that stockpile. A similar assumption also applies for the maximum limits of the grades.

Bounds for railed grades

We assumed that the lower and upper bounds for iron ore grades in the trains are to be equal to the lower and upper bounds for the live stockpile grades respectively at the associated mines.

Bounds for shipped grades

We assumed that the lower and upper bounds for iron ore grades for shipping are to be equal to the lower and upper bounds for the live stockpile grades respectively at the associated ports.

Mine production amount is determined

The exploration process is scheduled separately and the amount of production from each mine is estimated.

No rail operational cost

To make the best use of the available trains, we apply no operational cost for running the trains. Nevertheless, we consider penalties for exceeding the total allowed number of trains.

Estimated lump screening proportion

The parameters describing the proportion of lump products to be removed and added to the fines stockpile at each port in each period are estimated.

Uniformity of grades

At each stockpile in the mines and ports, we assume that the grade composition of the product is uniform across the stockpile, that is, partial mixing that occurs after transferring new materials is ignored.

4.3.2 Objective function

As the objective function in the previous model is linear, we consider the same objective function in this model. Therefore, the objective of the problem is to maximise the total profit, described as follows:

Total profit = Total revenue - Cost of live stockpile violations - Cost of bulk stockpile violations - Cost of bulk handling - Cost of cycle time violations - Cost of dumping materials - Cost of grade non-compliance + Incentive.

The total revenue, all the penalties and the incentive are defined in equations (3.1)–(3.8).

4.3.3 Operational constraints

None of the operational constraints in the previous model are non-linear. For the two models to be comparable, we keep all the linear constraints unchanged. Thus, we keep equations (3.9)–(3.31) in the new formulation.

4.3.4 Iron ore grades

In the new formulation, we introduce lower and upper bounds for all the grade quality variables. The value for these bounds are estimated based on the grades at the previous period and of products coming in. All non-linear terms appearing in the previous model are substituted by single linear terms. This section also outlines the convex and concave envelopes added into the new formulation.

Live stockpile grades at mines

Suppose LM^{L}_{mpct} and LM^{U}_{mpct} are lower and upper bounds respectively for live stockpile grades of component c of mined product p at mine m in period t. Thus, we have

$$LM^{L}_{mpct} \leq LM_{mpct} \leq LM^{U}_{mpct}, \quad \forall m \in M, p \in P_m, c \in C, t \in T. \qquad (4.1)$$

LM^{L}_{mpct} is estimated by determining the minimum value between the live stockpile grades at mines in the previous period and the iron ore grades of products coming in to that stockpile. In similar fashion, LM^{U}_{mpct} is estimated by determining the maximum value between the live stockpile grades at mines in the previous period and the iron ore grades of products coming in to that stockpile.

We replace equation (3.32) with the following equation:

$$\begin{aligned} LM^{live}_{mpct} = LM^{live}_{mpc,t-1} &+ LM^{out}_{mpct} + IOG_{mpct}IOP_{mpt} \\ &+ BM^{in}_{mpct} - \sum_{f \in F} \sum_{d \in D_{mrp}} \sum_{s \in S_{mp}} RG^{rail}_{mpcfdst}, \\ &\forall m \in M, p \in P_m, c \in C, t \in T \qquad (4.2) \end{aligned}$$

and remove equations (3.33)–(3.36) in the new formulation.

Bulk stockpile grades at mines

Suppose BM^{L}_{mpct} and BM^{U}_{mpct} are lower and upper bounds respectively for bulk stockpile grades of component c of mined product p at mine m in period t. Thus, we have

$$BM^{L}_{mpct} \leq BM_{mpct} \leq BM^{U}_{mpct}, \quad \forall m \in M, p \in P_m, c \in C, t \in T. \qquad (4.3)$$

BM^{L}_{mpct} is estimated by determining the minimum value between the bulk stockpile grades at mines in the previous period and the iron ore grades of products coming in to that stockpile. In similar fashion, BM^U_{mpct} is estimated by determining the maximum value between the bulk stockpile grades at mines in the previous period and the iron ore grades of products coming in to that stockpile.

We replace equation (3.37) with the following equation:

$$BM^{bulk}_{mpct} = BM^{bulk}_{mpc,t-1} - BM^{in}_{mpct} + LM^{out}_{mpct}, \quad \forall m \in M, p \in P_m, c \in C, t \in T \tag{4.4}$$

and remove equation (3.38) in the new formulation.

Railed grades

The lower and upper bounds for the railed grades are assumed to be equal to the lower and upper bounds for the live stockpile grades respectively at the corresponding mines. Thus, we have

$$LM^L_{mpct} \leq RG_{mpct} \leq LM^U_{mpct}, \quad \forall m \in M, p \in P_m, c \in C, t \in T. \tag{4.5}$$

We then replace equation (3.39) with the following equation:

$$\sum_{f \in F} \sum_{d \in D_{mrp}} \sum_{s \in S_{mp}} RG^{rail}_{mpcfdst} = IOT_{mpt} IOG_{mpct} + LM^{rail}_{mpct} + BM^{rail}_{mpct}, \quad \forall m \in M, p \in P_m, c \in C, t \in T \tag{4.6}$$

where LM^{rail}_{mpct} replaces $LM_{mpc,t-1}LT_{mpt}$ and BM^{rail}_{mpct} replaces $BM_{mpc,t-1}BT_{mpt}$ in equation (3.39).

The two different mine regimes, namely LIFO and FIFO, are still relevant in the new formulation. If the mine's regime is LIFO (last in first out), the trains will load the material produced in that mine first and the remaining amount will be taken from the live stockpiles. If there is still space in the train, the product from the bulk stockpile will also be loaded.Hence, if the regime is LIFO, equations (3.40)–(3.43) apply.

If the mine's regime is FIFO (first in first out), the trains will load the material from the live stockpiles before taking the produced material from the mine. If there is still space in the train, the product from the bulk stockpile will also be loaded. Hence, if the mine's regime is FIFO, equations (3.44)–(3.47) apply.

Live stockpile grades at ports

Suppose LP^L_{rsct} and LP^U_{rsct} are lower and upper bounds for live stockpile grades at ports respectively. Thus, we have

$$LP^L_{rsct} \leq LP_{rsct} \leq LP^U_{rsct}, \quad \forall r \in R, s \in S_r, c \in C, t \in T. \tag{4.7}$$

LP^L_{rsct} is estimated by determining the minimum value between the live stockpile grades at ports in the previous period and the iron ore grades of products coming in to that stockpile. In similar fashion, LP^U_{rsct} is estimated by determining the maximum value between the live stockpile grades at ports in the previous period and the iron ore grades of products coming in to that stockpile.

The following equation replaces the calculation of the live stockpile grades for lump products at ports described as equation (3.77):

$$LP^{\#}_{rsct} = LP^{live}_{rsc,t-1} - LP^{out}_{rsct} + BP^{in}_{rsct} + \sum_{m\in M}\sum_{p\in P_m}\sum_{f\in F}\sum_{d\in D_{mrp}} RG^{rail}_{mpcfdst},$$
$$\forall r \in R, s \in S^L, c \in C, t \in T \quad (4.8)$$

where $LP^{\#}_{rsct}$ replaces the non-linear term $LP_{rsct}w^{\#}_{rst}$ in the original equation.

Meanwhile equation (3.82) which describes the live stockpile grades for fine products at ports is substituted with:

$$LP^{\#\#}_{rsct} = ZG^{\#}_{rsct} + LP^{rf}_{rsct}, \quad \forall r \in R, s \in S^F, c \in C, t \in T \quad (4.9)$$

where $LP^{\#\#}_{rsct}$ replaces the non-linear term $LP_{rsct}(w^{\#}_{rst} + \frac{z^*_{rst}RF^*_{rst}}{1-RF^*_{rst}})$ in the original equation. We remove constraints (3.78)–(3.80) and (3.83)–(3.84) in the new formulation.

Bulk stockpile grades at ports

Suppose BP^L_{rsct} and BP^U_{rsct} are lower and upper bounds for bulk stockpile grades at ports respectively. Thus, we have

$$BP^L_{rsct} \leq BP_{rsct} \leq BP^U_{rsct}, \quad \forall r \in R, s \in S_r, c \in C, t \in T. \quad (4.10)$$

BP^L_{rsct} is estimated by determining the minimum value between the bulk stockpile grades at ports in the previous period and the iron ore grades of products coming in to that stockpile. In similar fashion, BP^U_{rsct} is estimated by determining the maximum value between the bulk stockpile grades at ports in the previous period and the iron ore grades of products coming in to that stockpile.

We replace equation (3.85) with the following equation:

$$BP^{bulk}_{rsct} = BP^{bulk}_{rsc,t-1} - BP^{in}_{rsct} + LP^{out}_{rsct}, \quad \forall r \in R, s \in S_r, c \in C, t \in T \quad (4.11)$$

and remove equation (3.86) in the new formulation.

Shipped grades

The lower and upper bounds for the shipped grades are assumed to be equal to the lower and upper bounds for the live stockpile grades respectively at the corresponding ports. Thus, we have

$$LP^{L}_{rsct} \leq ZG_{rsct} \leq LP^{U}_{rsct}, \quad \forall r \in R, s \in S_r, c \in C, t \in T. \tag{4.12}$$

The shipped grades are equal to its live stockpile grades at ports. Thus, we replace equation (3.87) with the following equation:

$$ZG^{\#}_{rsct} = LP^{live}_{rsc,t-1} - LP^{out}_{rsct} + BP^{in}_{rsct} + \sum_{m \in M} \sum_{p \in P_m} \sum_{f \in F} \sum_{d \in D_{mrp}} RG^{rail}_{mpcfdst} \quad \forall r \in R, s \in S_r, c \in C, t \in T. \tag{4.13}$$

Grade deviations

The grades of the shipped products must fall within the target range. This constraint was described in inequality (3.88). We remove equation (3.89) from the problem formulation.

Convex and concave envelopes

In the new formulation, for every substitute variable that replaces a non-linear term, we add four constraints containing two convex and two concave envelopes.

Consequently, for all $m \in M$, $p \in P_m$, $c \in C$, and $t \in T$, equation (3.33) is replaced by the following inequalities:

$$LM^{live}_{mpct} \geq LM^{L}_{mpct} s_{mpt}, \tag{4.14}$$
$$LM^{live}_{mpct} \geq YM_{mpt} LM_{mpct} + LM^{U}_{mpct} s_{mpt} - YM_{mpt} LM^{U}_{mpct}, \tag{4.15}$$
$$LM^{live}_{mpct} \leq LM^{L}_{mpct} s_{mpt} + YM_{mpt} LM_{mpct} - LM^{L}_{mpct} YM_{mpt}, \tag{4.16}$$
$$LM^{live}_{mpct} \leq LM^{U}_{mpct} s_{mpt}. \tag{4.17}$$

For all $m \in M$, $p \in P_m$, $c \in C$, and $t \in T$, equation (3.34) is replaced by the following inequalities:

$$LM^{out}_{mpct} \geq LM^{L}_{mpct} y^{out}_{mpt}, \tag{4.18}$$
$$LM^{out}_{mpct} \geq y^{out,U}_{mp} LM_{mpct} + LM^{U}_{mpct} y^{out}_{mpt} - y^{out,U}_{mp} LM^{U}_{mpct}, \tag{4.19}$$
$$LM^{out}_{mpct} \leq LM^{L}_{mpct} y^{out}_{mpt} + y^{out,U}_{mp} LM_{mpct} - LM^{L}_{mpct} y^{out,U}_{mp}, \tag{4.20}$$
$$LM^{out}_{mpct} \leq LM^{U}_{mpct} y^{out}_{mpt}. \tag{4.21}$$

For all $m \in M$, $p \in P_m$, $c \in C$, and $t \in T$, equation (3.38) is replaced by the following inequalities:

$$BM^{bulk}_{mpct} \geq BM^{L}_{mpct} b_{mpt}, \tag{4.22}$$

$$BM^{bulk}_{mpct} \geq B^{U}_{mpt} BM_{mpct} + BM^{U}_{mpct} b_{mpt} - B^{U}_{mpt} BM^{U}_{mpct}, \tag{4.23}$$

$$BM^{bulk}_{mpct} \leq BM^{L}_{mpct} b_{mpt} + B^{U}_{mpt} BM_{mpct} - BM^{L}_{mpct} B^{U}_{mpt}, \tag{4.24}$$

$$BM^{bulk}_{mpct} \leq BM^{U}_{mpct} b_{mpt}. \tag{4.25}$$

For all $m \in M$, $p \in P_m$, $c \in C$, and $t \in T$, equation (3.35) is replaced by the following inequalities:

$$BM^{in}_{mpct} \geq BM^{L}_{mpct} y^{in}_{mpt}, \tag{4.26}$$

$$BM^{in}_{mpct} \geq y^{in,U}_{mp} BM_{mpct} + BM^{U}_{mpct} y^{in}_{mpt} - y^{in,U}_{mp} BM^{U}_{mpc}, \tag{4.27}$$

$$BM^{in}_{mpct} \leq BM^{L}_{mpc} y^{in}_{mpt} + y^{in,U}_{mp} BM_{mpct} - BM^{L}_{mpc} y^{in,U}_{mp}, \tag{4.28}$$

$$BM^{in}_{mpct} \leq BM^{U}_{mpc} y^{in}_{mpt}. \tag{4.29}$$

For all $m \in M$, $p \in P_m$, $c \in C$, $f \in F$, $d \in D_{mp}$, $s \in S_{mp}$, and $t \in T$, equation (3.36) is replaced by the following inequalities:

$$RG^{rail}_{mpcfdst} \geq LM^{L}_{mpct} TS_{mpft} x_{mpfdst}, \tag{4.30}$$

$$RG^{rail}_{mpcfdst} \geq \begin{array}{l} TS_{mpft} x^{U}_{mt} RG_{mpct} + LM^{U}_{mpct} TS_{mpft} x_{mpfdst} \\ \quad - TS_{mpft} x^{U}_{mt} LM^{U}_{mpct}, \end{array} \tag{4.31}$$

$$RG^{rail}_{mpcfdst} \leq \begin{array}{l} LM^{L}_{mpct} TS_{mpft} x_{mpfdst} + TS_{mpft} x^{U}_{mt} RG_{mpct} \\ \quad - LM^{L}_{mpct} TS_{mpft} x^{U}_{mt}, \end{array} \tag{4.32}$$

$$RG^{rail}_{mpcfdst} \leq LM^{U}_{mpct} TS_{mpft} x_{mpfdst}. \tag{4.33}$$

For all $m \in M$, $p \in P_m$, $c \in C$, and $t \in T$, we obtain the following inequalities for the equation LM^{rail}_{mpct}:

$$LM^{rail}_{mpct} \geq LM^{L}_{mpct} LT_{mpt}, \tag{4.34}$$

$$LM^{rail}_{mpct} \geq YM_{mpt} LM_{mpct} + LM^{U}_{mpct} LT_{mpt} - YM_{mpt} LM^{U}_{mpct}, \tag{4.35}$$

$$LM^{rail}_{mpct} \leq LM^{L}_{mpct} LT_{mpt} + YM_{mpt} LM_{mpct} - LM^{L}_{mpct} YM_{mpt}, \tag{4.36}$$

$$LM^{rail}_{mpct} \leq LM^{U}_{mpct} LT_{mpt}. \tag{4.37}$$

For all $m \in M$, $p \in P_m$, $c \in C$, and $t \in T$, we obtain the following inequalities for the equation BM^{rail}_{mpct}:

$$BM^{rail}_{mpct} \geq BM^{L}_{mpct} BT_{mpt}, \tag{4.38}$$

$$BM^{rail}_{mpct} \geq B^{U}_{mpt} BM_{mpct} + BM^{U}_{mpct} BM_{mpt} - B^{U}_{mpt} BM^{U}_{mpct}, \tag{4.39}$$

$$BM^{rail}_{mpct} \leq BM^{L}_{mpct} BT_{mpt} + B^{U}_{mpt} BM_{mpct} - BM^{L}_{mpct} B^{U}_{mpt}, \tag{4.40}$$

$$BM^{rail}_{mpct} \leq BM^{U}_{mpct} BT_{mpt}. \tag{4.41}$$

For all $r \in R$, $s \in S_r$, $c \in C$, and $t \in T$, equation (3.78) is replaced by the following inequalities:

$$LP^{live}_{rsct} \geq LP^{L}_{rsct} w_{rst}, \tag{4.42}$$

$$LP^{live}_{rsct} \geq YP_{rst} LP_{rsct} + LP^{U}_{rsct} w_{rst} - YP_{rst} LP^{U}_{rsct}, \tag{4.43}$$

$$LP^{live}_{rsct} \leq LP^{L}_{rsct} w_{rst} + YP_{rst} LP_{rsct} - LP^{L}_{rsct} YP_{rst}, \tag{4.44}$$

$$LP^{live}_{rsct} \leq LP^{U}_{rsct} w_{rst}. \tag{4.45}$$

For all $r \in R$, $s \in S^F$, $c \in C$, and $t \in T$, equation (3.84) is replaced by the following inequalities:

$$LP^{rf}_{rsct} \geq LP^{L}_{rsct} \frac{z^*_{rst} RF^*_{rst}}{1 - RF^*_{rst}}, \tag{4.46}$$

$$LP^{rf}_{rsct} \geq \frac{Z^U_{rt} RF^*_{rst}}{1 - RF^*_{rst}} LP_{rsct} + LP^{U}_{rsct} \frac{z^*_{rst} RF^*_{rst}}{1 - RF^*_{rst}} - \frac{Z^U_{rt} RF^*_{rst}}{1 - RF^*_{rst}} LP^{U}_{rsct}, \tag{4.47}$$

$$LP^{rf}_{rsct} \leq LP^{L}_{rsct} \frac{z^*_{rst} RF^*_{rst}}{1 - RF^*_{rst}} + \frac{Z^U_{rt} RF^*_{rst}}{1 - RF^*_{rst}} LP_{rsct} - LP^{L}_{rsct} \frac{Z^U_{rt} RF^*_{rst}}{1 - RF^*_{rst}}, \tag{4.48}$$

$$LP^{rf}_{rsct} \leq LP^{U}_{rsct} \frac{z^*_{rst} RF^*_{rst}}{1 - RF^*_{rst}}. \tag{4.49}$$

For all $r \in R$, $s \in S_r$, $c \in C$, and $t \in T$, equation (3.79) is replaced by the following inequalities:

$$LP^{out}_{rsct} \geq LP^{L}_{rsct} u^{out}_{rst}, \tag{4.50}$$

$$LP^{out}_{rsct} \geq u^{out,U}_{rs} LP_{rsct} + LP^{U}_{rsct} u^{out}_{rst} - u^{out,U}_{rs} LP^{U}_{rsct}, \tag{4.51}$$

$$LP^{out}_{rsct} \leq LP^{L}_{rsct} u^{out}_{rst} + u^{out,U}_{rs} LP_{rsct} - LP^{L}_{rsct} u^{out,U}_{rs}, \tag{4.52}$$

$$LP^{out}_{rsct} \leq LP^{U}_{rsct} u^{out}_{rst}. \tag{4.53}$$

For all $r \in R$, $s \in S^F$, $c \in C$, and $t \in T$, we obtain the following inequalities for $LP^{\#\#}_{rst}$:

$$LP^{\#\#}_{rsct} \geq LP^{L}_{rsct} \big(w^{\#}_{rst} + \frac{z^*_{rst} RF^*_{rst}}{1 - RF^*_{rst}} \big), \tag{4.54}$$

$$LP^{\#\#}_{rsct} \geq \begin{array}{l} (YP_{rst} + Z^U_{rst}) LP_{rsct} + LP^{U}_{rsct} \big(w^{\#}_{rst} + \frac{z^*_{rst} RF^*_{rst}}{1 - RF^*_{rst}} \big) \\ \quad - (YP_{rst} + Z^U_{rst}) LP^{U}_{rsct}, \end{array} \tag{4.55}$$

$$LP^{\#\#}_{rsct} \leq \begin{array}{l} LP^{L}_{rsct} \big(w^{\#}_{rst} + \frac{z^*_{rst} RF^*_{rst}}{1 - RF^*_{rst}} \big) + (YP_{rst} + Z^U_{rst}) LP_{rsct} \\ \quad - LP^{L}_{rsct} (YP_{rst} + Z^U_{rst}), \end{array} \tag{4.56}$$

$$LP^{\#\#}_{rsct} \leq LP^{U}_{rsct} \big(w^{\#}_{rst} + \frac{z^*_{rst} RF^*_{rst}}{1 - RF^*_{rst}} \big). \tag{4.57}$$

For all $r \in R$, $s \in S_r$, $c \in C$, and $t \in T$, equation (3.86) is replaced by the following inequalities:

$$BP^{bulk}_{rsct} \geq BP^{L}_{rsct} v_{rst}, \tag{4.58}$$
$$BP^{bulk}_{rsct} \geq V^{U}_{rst} BP_{rsct} + BP^{U}_{rsct} v_{rst} - V^{U}_{rst} BP^{U}_{rsct}, \tag{4.59}$$
$$BP^{bulk}_{rsct} \leq BP^{L}_{rsct} v_{rst} + V^{U}_{rst} BP_{rsct} - BP^{L}_{rsct} V^{U}_{rst}, \tag{4.60}$$
$$BP^{bulk}_{rsct} \leq BP^{U}_{rsct} v_{rst}. \tag{4.61}$$

For all $r \in R$, $s \in S_r$, $c \in C$, and $t \in T$, equation (3.80) is replaced by the following inequalities:

$$BP^{in}_{rsct} \geq BP^{L}_{rsct} u^{in}_{rst}, \tag{4.62}$$
$$BP^{in}_{rsct} \geq u^{in,U}_{rs} BP_{rsct} + BP^{U}_{rsct} u^{in}_{rst} - u^{in,U}_{rs} BP^{U}_{rsct}, \tag{4.63}$$
$$BP^{in}_{rsct} \leq BP^{L}_{rsct} u^{in}_{rst} + u^{in,U}_{rs} BP_{rsct} - BP^{L}_{rsc} u^{in,U}_{rs}, \tag{4.64}$$
$$BP^{in}_{rsct} \leq BP^{U}_{rsc} u^{in}_{rst}. \tag{4.65}$$

For all $r \in R$, $s \in S_r$, $c \in C$, and $t \in T$, equation (3.89) is replaced by the following inequalities:

$$ZG^{shipped}_{rsct} \geq LP^{L}_{rsct} z_{rst}, \tag{4.66}$$
$$ZG^{shipped}_{rsct} \geq Z^{U}_{rt} ZG_{rsct} + LP^{U}_{rsct} z_{rst} - Z^{U}_{rt} LP^{U}_{rsct}, \tag{4.67}$$
$$ZG^{shipped}_{rsct} \leq LP^{L}_{rsct} z_{rst} + Z^{U}_{rt} ZG_{rsct} - LP^{L}_{rsct} Z^{U}_{rt}, \tag{4.68}$$
$$ZG^{shipped}_{rsct} \leq LP^{U}_{rsct} z_{rst}. \tag{4.69}$$

For all $r \in R$, $s \in S_r$, $c \in C$, and $t \in T$, equation (3.83) is replaced by the following inequalities:

$$ZG^{\#}_{rsct} \geq LP^{L}_{rsct} w^{\#}_{rst}, \tag{4.70}$$
$$ZG^{\#}_{rsct} \geq \begin{array}{l} (YP_{rst} + Z^{U}_{rst}) ZG_{rsct} + LP^{U}_{rsct} w^{\#}_{rst} \\ \quad - (YP_{rst} + Z^{U}_{rst}) LP^{U}_{rsct}, \end{array} \tag{4.71}$$
$$ZG^{\#}_{rsct} \leq \begin{array}{l} LP^{L}_{rsct} w^{\#}_{rst} + (YP_{rst} + Z^{U}_{rst}) ZG_{rsct} \\ \quad - LP^{L}_{rsct} (YP_{rst} + Z^{U}_{rst}), \end{array} \tag{4.72}$$
$$ZG^{\#}_{rsct} \leq LP^{U}_{rsct} w^{\#}_{rst}. \tag{4.73}$$

4.3.5 Summary of notations

Sets

C	The set of all components of a product.
D	The set of all car dumpers.
D_{mrp}	The set of all car dumpers in port $r \in R$ receiving mined product $p \in P$ from mine $m \in M$.
D^{WC}	The set of all car dumpers that serve the Cape Lambert-Western Creek region, $D^{WC} \subseteq D$.
F	The set of all train fleets.
G	The set of all regions.
M	The set of all mines.
M_f	The set of all mines serviced by the fleet $f \in F$, $M_f \subset M$.
M_g	The set of all mines belonging to region $g \in G$, $M_g \subset M$.
M^{FF}	The set of all mines whose regime is FIFO, $M^{FF} \subset M$.
M^{JV}	The set of all mines that have to comply with the joint ventures obligations, $M^{JV} \subseteq M$.
M^{LF}	The set of all mines whose regime is LIFO, $M^{LF} \subset M$.
P	The set of all mined products.
P_m	The set of all mined products for mine $m \in M$, $P_m \subseteq P$.
R	The set of all ports.
S	The set of all shipped products.
S_r	The set of all shipped products from port $r \in R$, $S_r \subseteq S$.
S_{mp}	The set of all shipped products sent from mine $m \in M$ and product $p \in P$, $S_{mp} \subseteq S$.
S^F	The set of all fines shipped products, $S^F \subset S$
S^L	The set of all lump shipped products, $S^L \subset S$
T	The set of planning periods.

Model parameters

B^U_{mpt}	Maximum bulk stockpile level at mine $m \in M$ for product $p \in P$ in period $t \in T$.
BM^L_{mpct}	Minimum bulk stockpile grade of component $c \in C$ in product $p \in P_m$ at mine $m \in M$ in period $t \in T$.
BM^U_{mpct}	Maximum bulk stockpile grade of component $c \in C$ in product $p \in P_m$ at mine $m \in M$ in period $t \in T$.
BP^L_{rsct}	Minimum bulk stockpile grade of component $c \in C$ in product $s \in S_r$ at port $r \in R$ in period $t \in T$.
BP^U_{rsct}	Maximum bulk stockpile grade of component $c \in C$ in product $s \in S_r$ at port $r \in R$ in period $t \in T$.
IOG_{mpt}	Grade of mined product $p \in P_m$ produced at mine $m \in M$ in period $t \in T$.
IOP_{mpt}	Amount of product $p \in P_m$ produced at mine $m \in M$ in period $t \in T$.
LM^L_{mpct}	Minimum live stockpile grade of component $c \in C$ in product $p \in P_m$ at mine $m \in M$ in period $t \in T$.
LM^U_{mpct}	Maximum live stockpile grade of component $c \in C$ in product $p \in P_m$ at mine $m \in M$ in period $t \in T$.
LP^L_{rsct}	Minimum live stockpile grade of component $c \in C$ in product $s \in S_r$ at port $r \in R$ in period $t \in T$.
LP^U_{rsct}	Maximum live stockpile grade of component $c \in C$ in product $s \in S_r$ at port $r \in R$ in period $t \in T$.
RF_{rst}	Percentage of lump product $s \in S^L$ returned to fines stockpile at port $r \in R$ in period T.
RF^*_{rst}	Percentage of the associated lump product of fines product $s \in S^F$ at port $r \in R$ in period T.
TS_{mpft}	Capacity of a train in tonnes sent to mine $m \in M$ transporting mined product $p \in P_m$ belonging to fleet $f \in F$ in period $t \in T$.
$u^{in,U}_{rs}$	Maximum amount of product $s \in S_r$ that can be inloaded at port $r \in R$.
$u^{out,U}_{rs}$	Maximum amount of product $s \in S_r$ that can be outloaded at port $r \in R$.
V^U_{rst}	Maximum bulk stockpile level at port $r \in R$ for product $s \in S_r$.
x^U_{mt}	Maximum number of trains allowed at mine $m \in M$ in period $t \in T$.

$y^{in,U}_{mp}$	Maximum tonnes of product $p \in P_m$ that can be inloaded at mine $m \in M$.
$y^{out,U}_{mp}$	Maximum tonnes of product $p \in P_m$ that can be outloaded at mine $m \in M$.
YM_{mpt}	Yard capacity limit of live stockpile at mine $m \in M$ for product $p \in P_m$ in period $t \in T$.
YP_{rst}	Yard capacity limit of live stockpile at port $r \in R$ for product $s \in S_r$ in period $t \in T$.
Z^U_{rt}	Maximum capacity of amount shipped at port $r \in R$ in period $t \in T$.

Decision variables

b_{mpt}	Bulk stockpile level at mine $m \in M$ for mined product $p \in P_m$ in period $t \in T$.
BM_{mpct}	Bulk stockpile grade of component $c \in C$ in product $p \in P_m$ at mine $m \in M$ in period $t \in T$.
BM_{mpct}^{bulk}	Substitute variable for non-linear term $BM_{mpct}b_{mpt}$.
BM_{mpct}^{in}	Substitute variable for non-linear term $BM_{mpct}y_{mpt}^{in}$.
BM_{mpct}^{rail}	Substitute variable for non-linear term $BM_{mpct}BT_{mpt}$.
BP_{rsct}	Bulk stockpile grade of component $c \in C$ in product $s \in S_r$ at port $r \in R$ in period $t \in T$.
BP_{rsct}^{bulk}	Substitute variable for non-linear term $BP_{rsct}v_{rsct}$.
BP_{rsct}^{in}	Substitute variable for non-linear term $BP_{rsct}u_{rst}^{in}$.
BT_{mpct}	Amount of component $c \in C$ of product $p \in P$ transported from the bulk stockpile at mine $m \in M$ in period $t \in T$.
IOT_{mpct}	Amount of component $c \in C$ in mined product $p \in P$ produced from mine $m \in M$ that is transported by trains in period $t \in T$.
LM_{mpct}	Live stockpile grade of component $c \in C$ in product $p \in P_m$ at mine $m \in M$ in period $t \in T$.
LM_{mpct}^{live}	Substitute variable for non-linear term $LM_{mpct}s_{mpt}$.
LM_{mpct}^{out}	Substitute variable for non-linear term $LM_{mpct}y_{mpt}^{out}$.
LM_{mpct}^{rail}	Substitute variable for non-linear term $LM_{mpct}LT_{mpt}$.
LP_{rsct}	Live stockpile grade of component $c \in C$ in product $s \in S_r$ at port $r \in R$ in period $t \in T$.
LP_{rsct}^{live}	Substitute variable for non-linear term $LP_{rsct}w_{rst}$.
LP_{rsct}^{rf}	Substitute variable for non-linear term $LP_{rsct}^{*}z_{rst}^{*}RF_{rst}^{*}$.
LP_{rsct}^{out}	Substitute variable for non-linear term $LP_{rsct}u_{rsct}^{out}$.
LP_{rst}^{*}	Live stockpile grade of component $c \in C$ in the associated lump product of fines product $s \in S^F$ at port $r \in R$ in period $t \in T$.
LT_{mpct}	Amount of component $c \in C$ of product $p \in P$ transported from the live stockpile at mine $m \in M$ in period $t \in T$.
$LP_{rsct}^{\#}$	Substitute variable for non-linear term $LM_{mpct}w_{rst}^{\#}$.
$LP_{rsct}^{\#\#}$	Substitute variable for non-linear term $LP_{rsct}\left(w_{rst}^{\#}+\frac{z_{rst}^{*}RF_{rst}^{*}}{1-RF_{rst}^{*}}\right)$.
RG_{mpct}	Railed grade of component $c \in C$ in product $p \in P_m$ from mine $m \in M$ in period $t \in T$.
$RG_{mpcfdst}^{rail}$	Substitute variable for non-linear term $RG_{mpct}TS_{mpft}x_{mpfdst}$.

s_{mpt}	Live stockpile level at mine $m \in M$ for mined product $p \in P_m$ in period $t \in T$.
u^{in}_{rst}	Amount transferred from bulk to live stockpiles at port $r \in R$ for shipped product $s \in S_r$ in period $t \in T$.
u^{out}_{rst}	Amount transferred from live to bulk stockpiles at port $r \in R$ for shipped product $s \in S_r$ in period $t \in T$.
v_{rst}	Bulk stockpile level at port $r \in R$ for shipped product $s \in S_r$ in period $t \in T$.
w_{rst}	Live stockpile level at port $r \in R$ for shipped product $s \in S_r$ in period $t \in T$.
$w^{\#}_{rst}$	Live stockpile level at port $r \in R$ for shipped product $s \in S_r$ before lump screening and return fines process in period $t \in T$.
x_{mpfdst}	Number of trains used at mine $m \in M$ for mined product $p \in P_m$ of fleet f at car dumper $d \in D_{mrp}$ for shipped product $s \in S_{mp}$ in period $t \in T$.
y^{in}_{rst}	Amount transferred from bulk to live stockpiles at mine $m \in M$ for mined product $p \in P_m$ in period $t \in T$.
y^{out}_{rst}	Amount transferred from live to bulk stockpiles at mine $m \in M$ for mined product $p \in P_m$ in period $t \in T$.
z_{rst}	Amount of product $s \in S$ shipped from port $r \in R$ in period $t \in T$.
z^{*}_{rst}	Amount of product from the associated lump stockpile of fines product $s \in S^F$ shipped from port $r \in R$ in period $t \in T$.
ZG_{rsct}	Shipped grade of component $c \in C$ in product $s \in S_r$ from port $r \in R$ in period $t \in T$.
$ZG^{shipped}_{rsct}$	Substitute variable for non-linear term $ZG_{rsct} z_{rst}$.
$ZG^{\#}_{rsct}$	Substitute variable for non-linear term $ZG_{rsct} w^{\#}_{rst}$.

4.3.6 Complete formulation

Maximise:

$$
\begin{aligned}
&\sum_{s\in S_r} SP_s \sum_{t\in T}\sum_{r\in R} z_{rst}(1+I)^{1-t} \\
&-\sum_{m\in M}\sum_{p\in P_m}\left[MS^{LE}_{mp}\sum_{t\in T}\alpha^{LE}_{mpt} + MS^{LS}_{mp}\sum_{t\in T}\alpha^{LS}_{mpt}\right] \\
&-\sum_{r\in R}\sum_{s\in S_r}\left[PS^{LE}_{rs}\sum_{t\in T}\beta^{LE}_{rst} + PS^{LS}_{rs}\sum_{t\in T}\beta^{LS}_{rst}\right] \\
&-\sum_{m\in M}\sum_{p\in P_m} MS^{BE}_{mp}\sum_{t\in T}\alpha^{BE}_{mpt} + \sum_{r\in R}\sum_{s\in S_r} PS^{BE}_{rs}\sum_{t\in T}\beta^{BE}_{rst} \\
&-\sum_{m\in M}\sum_{p\in P_m}\left[MB^{out}_{mp}\sum_{t\in T} y^{out}_{mpt} + MB^{in}_{mp}\sum_{t\in T} y^{in}_{mpt}\right] \\
&-\sum_{r\in R}\sum_{s\in S_r}\left[PB^{out}_{rs}\sum_{t\in T} u^{out}_{rst} + PB^{in}_{rs}\sum_{t\in T} u^{in}_{rst}\right] \\
&-\sum_{f\in F} CP_f \sum_{t\in T}\mu_{ft} \\
&-\sum_{m\in M}\sum_{d\in D_m} DP_{md}\sum_{p\in P_m}\sum_{f\in F}\sum_{t\in T} TS_{mpft}x_{mpfdst} \\
&-\sum_{c\in C}\sum_{r\in R}\sum_{s\in S_r} GP_{rsc}\sum_{t\in T}(si_{rsct} + ei_{rsct}) \\
&+\pi\sum_{s\in S} SP_s \sum_{m\in M}\sum_{p\in P_m}\sum_{f\in F}\sum_{d\in D}\sum_{t\in T} TS_{mpft}x_{mpfdst}
\end{aligned}
$$

subject to:

$$
\begin{aligned}
S^L_{mpt} - \alpha^{LS}_{mpt} \le s_{mpt} \le S^U_{mpt} + \alpha^{LE}_{mpt}, &\qquad \forall m\in M, p\in P_m, t\in T \\
W^L_{rst} - \beta^{LS}_{rst} \le w_{rst} \le W^U_{rst} + \beta^{LE}_{rst}, &\qquad \forall r\in R, s\in S_r, t\in T \\
0 \le b_{mpt} \le B^U_{mpt} + \alpha^{BE}_{mpt}, &\qquad \forall m\in M, p\in P_m, t\in T \\
0 \le v_{rst} \le V^U_{rst} + \beta^{BE}_{rst}, &\qquad \forall r\in R, s\in S_r, t\in T \\
S^U_{mpt} + \alpha^{LE}_{mpt} \le YM_{mpt}, &\qquad \forall m\in M, p\in P_m, t\in T \\
W^U_{rst} + \beta^{LE}_{rst} \le YP_{rst}, &\qquad \forall r\in R, s\in S_r, t\in T
\end{aligned}
$$

$$0 \leq y_{mpt}^{out} \leq y_{mp}^{out,U}, \qquad \forall m \in M, p \in P_m, t \in T$$
$$0 \leq y_{mpt}^{in} \leq y_{mp}^{in,U}, \qquad \forall m \in M, p \in P_m, t \in T$$
$$0 \leq u_{rst}^{out} \leq u_{rs}^{out,U}, \qquad \forall r \in R, s \in S_r, t \in T$$
$$0 \leq u_{rst}^{in} \leq u_{rs}^{in,U}, \qquad \forall r \in R, s \in S_r, t \in T$$

$$\sum_{p \in P_m} \sum_{f \in F} \sum_{d \in D_{mp}} \sum_{s \in S} x_{mpfdst} \leq x_{mt}^{U}, \qquad \forall m \in M, t \in T$$
$$\sum_{m \in M_g} \sum_{p \in P_m} \sum_{f \in F} \sum_{d \in D_{mrp}} \sum_{s \in S} x_{mpfdst} \leq MT_{gt}, \qquad \forall g \in G, t \in T$$
$$\sum_{m \in M_f} \sum_{p \in P_m} \sum_{d \in D_{mrp}} \sum_{s \in S} x_{mpfdst} \leq MF_{ft}, \qquad \forall f \in F, t \in T$$
$$\sum_{m \in M_f} \sum_{p \in P_m} CT_{mpt} \sum_{d \in D_{mrp}} \sum_{s \in S} x_{mpfdst} \leq PF_{ft} + \mu_{ft}, \qquad \forall f \in F, t \in T$$
$$\sum_{t=1}^{n} JV_{mt}^{L} \leq \sum_{p \in P_m} \sum_{f \in F} \sum_{d \in D_{mp}} \sum_{s \in S} \sum_{t=1}^{n} x_{mpfdst} \leq \sum_{t=1}^{n} JV_{mt}^{U}, \qquad \forall m \in M^{JV}, t \in T$$
$$\sum_{m \in M} \sum_{p \in P_m} \sum_{f \in F} \sum_{s \in S} x_{mpfdst} \leq DC_{dt}, \qquad \forall d \in D, t \in T$$
$$\sum_{m \in M} \sum_{p \in P_m} \sum_{f \in F} \sum_{d \in D^{WC}} \sum_{s \in S} x_{mpfdst} \leq WC_{t}, \qquad \forall t \in T$$
$$0 \leq \sum_{s \in S_r} z_{rst} \leq Z_{rt}^{U}, \qquad \forall r \in R, t \in T$$

$$s_{mpt} = s_{mp,t-1} + IOP_{mpt} + y_{mpt}^{in} - y_{mpt}^{out} - \sum_{f \in F} TS_{mpft} \sum_{d \in D_{mp}} \sum_{s \in S_{mp}} x_{mpfdst}, \qquad \forall m \in M, p \in P_m, t \in T$$

$$w_{rst} = w_{rs,t-1} + \sum_{m \in M} \sum_{p \in P_m} \sum_{f \in F} TS_{mpft} \sum_{d \in D_{mrp}} x_{mpfdst} + u_{rst}^{in} - u_{rst}^{out} - \frac{z_{rst}}{1 - RF_{rst}}, \qquad \forall r \in R, s \in S^{L}, t \in T$$

$$w_{rst} = w_{rs,t-1} + \sum_{m \in M} \sum_{p \in P_m} \sum_{f \in F} TS_{mpft} \sum_{d \in D_{mrp}} x_{mpfdst} + u_{rst}^{in} - u_{rst}^{out} - z_{rst} + \frac{z_{rst}^{*} RF_{rst}^{*}}{1 - RF_{rst}^{*}}, \qquad \forall r \in R, s \in S^{F}, t \in T$$

$$w^{\#}_{rst} = w_{rs,t-1} - u^{out}_{rst} + u^{in}_{rst} + \sum_{m \in M} \sum_{p \in P_m} \sum_{f \in F} TS_{mpft} \sum_{d \in D_{mrp}} x_{mpdst}, \quad \forall r \in R, s \in S_r, t \in T$$

$$b_{mpt} = b_{mp,t-1} + y^{out}_{mpt} - y^{in}_{mpt}, \quad \forall m \in M, p \in P_m, t \in T$$
$$v_{rst} = v_{rs,t-1} + u^{out}_{rst} - u^{in}_{rst}, \quad \forall r \in R, s \in S_r, t \in T$$
$$LM^{L}_{mpct} \leq LM_{mpct} \leq LM^{U}_{mpct}, \quad \forall m \in M, p \in P_m, c \in C, t \in T$$
$$BM^{L}_{mpct} \leq BM_{mpct} \leq BM^{U}_{mpct}, \quad \forall m \in M, p \in P_m, c \in C, t \in T$$
$$LM^{L}_{mpct} \leq RG_{mpct} \leq LM^{U}_{mpct}, \quad \forall m \in M, p \in P_m, c \in C, t \in T$$
$$LP^{L}_{rsct} \leq LP_{rsct} \leq LP^{U}_{rsct}, \quad \forall r \in R, s \in S_r, c \in C, t \in T$$
$$BP^{L}_{rsct} \leq BP_{rsct} \leq BP^{U}_{rsct}, \quad \forall r \in R, s \in S_r, c \in C, t \in T$$
$$LP^{L}_{rsct} \leq ZG_{rsct} \leq LP^{U}_{rsct}, \quad \forall r \in R, s \in S_r, c \in C, t \in T$$

$$LM^{live}_{mpct} = LM^{live}_{mpc,t-1} + LM^{out}_{mpct} + IOG_{mpct} IOP_{mpt} + BM^{in}_{mpct} - \sum_{f \in F} \sum_{d \in D_{mrp}} \sum_{s \in S_{mp}} RG^{rail}_{mpcfdst}, \quad \forall m \in M, p \in P_m, c \in C, t \in T$$

$$BM^{bulk}_{mpct} = BM^{bulk}_{mpc,t-1} - BM^{in}_{mpct} + LM^{out}_{mpct}, \quad \forall m \in M, p \in P_m, c \in C, t \in T$$

$$\sum_{f \in F} \sum_{d \in D_{mrp}} \sum_{s \in S_{mp}} RG^{rail}_{mpcfdst} = IOT_{mpt} IOG_{mpct} + LM^{rail}_{mpct} + BM^{rail}_{mpct}, \quad \forall m \in M, p \in P_m, c \in C, t \in T$$

$$LP^{\#}_{rsct} = LP^{live}_{rsc,t-1} - LP^{out}_{rsct} + BP^{in}_{rsct} + \sum_{m \in M} \sum_{p \in P_m} \sum_{f \in F} \sum_{d \in D_{mrp}} RG^{rail}_{mpcfdst}, \quad \forall r \in R, s \in S^{L}, c \in C, t \in T$$

$$ZG^{\#}_{rsct} = LP^{live}_{rsc,t-1} - ZG^{out}_{rsct} + BP^{in}_{rsct} + \sum_{m \in M} \sum_{p \in P_m} \sum_{f \in F} \sum_{d \in D_{mrp}} RG^{rail}_{mpcfdst}, \quad \forall r \in R, s \in S_r, c \in C, t \in T$$

$$LP^{\#\#}_{rsct} = ZG^{\#}_{rsct} + LP^{rf}_{rsct}, \quad \forall r \in R, s \in S^{F}, c \in C, t \in T$$
$$BP^{bulk}_{rsct} = BP^{bulk}_{rsc,t-1} - BP^{in}_{rsct} + LP^{out}_{rsct}, \quad \forall r \in R, s \in S_r, c \in C, t \in T$$

$$IOT_{mpt} \leq IOP_{mpt}, \qquad \forall m \in M^{LF}, p \in P_m, t \in T$$

$$IOT_{mpt} \leq \sum_{f \in F} TS_{mpft} \sum_{d \in D_{mp}} \sum_{s \in S} x_{mpfdst}, \qquad \forall m \in M^{LF}, p \in P_m, t \in T$$

$$IOT_{mpt} \geq \sum_{f \in F} TS_{mpft} \sum_{d \in D_{mp}} \sum_{s \in S} x_{mpfdst} - (10,000 - IOP_{mpt}) IOT^{bin}_{mpt}, \qquad \forall m \in M^{LF}, p \in P_m, t \in T$$

$$IOT_{mpt} \geq IOP_{mpt} - IOP_{mpt}(1 - IOT^{bin}_{mpt}), \qquad \forall m \in M^{LF}, p \in P_m, t \in T$$

$$LTM_{mpt} \geq 0, \qquad \forall m \in M^{LF}, p \in P_m, t \in T$$

$$LTM_{mpt} \geq \sum_{f \in F} TS_{mpft} \sum_{d \in D_{mp}} \sum_{s \in S} x_{mpfdst} - IOP_{mpt}, \qquad \forall m \in M^{LF}, p \in P_m, t \in T$$

$$LTM_{mpt} \leq \sum_{f \in F} TS_{mpft} \sum_{d \in D_{mp}} \sum_{s \in S} x_{mpfdst} - IOP_{mpt} + 10,000 LTM^{bin}_{mpt}, \qquad \forall m \in M^{LF}, p \in P_m, t \in T$$

$$LTM_{mpt} \leq 10,000(1 - LTM^{bin}_{mpt}), \qquad \forall m \in M^{LF}, p \in P_m, t \in T$$

$$LT_{mpt} \leq s_{mp,t-1}, \qquad \forall m \in M^{LF}, p \in P_m, t \in T$$

$$LT_{mpt} \leq LTM_{mpt}, \qquad \forall m \in M^{LF}, p \in P_m, t \in T$$

$$LT_{mpt} \geq s_{mp,t-1} - IOP_{mpt} - 20,000 LT^{bin}_{mpt}, \qquad \forall m \in M^{LF}, p \in P_m, t \in T$$

$$LT_{mpt} \geq LTM_{mpt} - 20,000(1 - LT^{bin}_{mpt}), \qquad \forall m \in M^{LF}, p \in P_m, t \in T$$

$$IOTM_{mpt} \geq 0, \qquad \forall m \in M^{FF}, p \in P_m, t \in T$$

$$IOTM_{mpt} \geq \sum_{f \in F} \sum_{d \in D_{mp}} \sum_{s \in S_{mp}} TS_{mpft} x_{mpfdst} - s_{mp,t-1}, \qquad \forall m \in M^{FF}, p \in P_m, t \in T$$

$$IOTM_{mpt} \leq \sum_{f\in F}\sum_{d\in D_{mp}}\sum_{s\in S_{mp}} TS_{mpft}x_{mpfdst} - s_{mp,t-1} + 10,000IOTM^{bin}_{mpt}, \quad \forall m \in M^{FF}, p \in P_m, t \in T$$

$$IOTM_{mpt} \leq 10,000(1 - IOTM^{bin}_{mpt}), \quad \forall m \in M^{FF}, p \in P_m, t \in T$$

$$IOT_{mpt} \leq IOP_{mpt}, \quad \forall m \in M^{FF}, p \in P_m, t \in T$$

$$IOT_{mpt} \leq IOTM_{mpt}, \quad \forall m \in M^{FF}, p \in P_m, t \in T$$

$$IOT_{mpt} \geq IOP_{mpt} - (IOP_{mpt} + 10,000)IOT^{bin}_{mpt}, \quad \forall m \in M^{FF}, p \in P_m, t \in T$$

$$IOT_{mpt} \geq IOTM_{mpt} - (10,000 - IOP_{mpt})(1 - IOT^{bin}_{mpt}), \quad \forall m \in M^{FF}, p \in P_m, t \in T$$

$$LT_{mpt} \leq s_{mp,t-1}, \quad \forall m \in M^{FF}, p \in P_m, t \in T$$

$$LT_{mpt} \leq \sum_{f\in F} TS_{mpft} \sum_{d\in D_{mp}}\sum_{s\in S} x_{mpfdst}, \quad \forall m \in M^{FF}, p \in P_m, t \in T$$

$$LT_{mpt} \geq \sum_{f\in F} TS_{mpft} \sum_{d\in D_{mp}}\sum_{s\in S} x_{mpfdst} - 10,000LT^{bin}_{mpt}, \quad \forall m \in M^{FF}, p \in P_m, t \in T$$

$$LT_{mpt} \geq s_{mp,t-1} - 20,000(1 - LT^{bin}_{mpt}), \quad \forall m \in M^{FF}, p \in P_m, t \in T$$

$$BT_{mpt} \geq 0, \quad \forall m \in M, p \in P_m, t \in T$$

$$BT_{mpt} \geq \sum_{f\in F}\sum_{d\in D_{mp}}\sum_{s\in S_{mp}} TS_{mpft}x_{mpfdst} - IOP_{mpt} - s_{mp,t-1}, \quad \forall m \in M, p \in P_m, t \in T$$

$$BT_{mpt} \leq \sum_{f\in F}\sum_{d\in D_{mp}}\sum_{s\in S_{mp}} TS_{mpft}x_{mpfdst} - IOP_{mpt} + 10,000BT^{bin}_{mpt}, \quad \forall m \in M, p \in P_m, t \in T$$

$$BT_{mpt} \leq 10,000(1 - BT^{bin}_{mpt}), \qquad \forall m \in M, p \in P_m, t \in T$$

$$LM^{live}_{mpct} \geq LM^{L}_{mpct} s_{mpt}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$
$$LM^{live}_{mpct} \leq LM^{U}_{mpct} s_{mpt}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$

$$LM^{live}_{mpct} \geq YM_{mpt} LM_{mpct} + LM^{U}_{mpct} s_{mpt} - YM_{mpt} LM^{U}_{mpct}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$
$$LM^{live}_{mpct} \leq LM^{L}_{mpct} s_{mpt} + YM_{mpt} LM_{mpct} - LM^{L}_{mpct} YM_{mpt}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$

$$LM^{out}_{mpct} \geq LM^{L}_{mpct} y^{out}_{mpt}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$
$$LM^{out}_{mpct} \leq LM^{U}_{mpct} y^{out}_{mpt}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$

$$LM^{out}_{mpct} \geq y^{out,U}_{mp} LM_{mpct} + LM^{U}_{mpct} y^{out}_{mpt} - y^{out,U}_{mp} LM^{U}_{mpct}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$
$$LM^{out}_{mpct} \leq LM^{L}_{mpct} y^{out}_{mpt} + y^{out,U}_{mp} LM_{mpct} - LM^{L}_{mpct} y^{out,U}_{mp}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$

$$BM^{bulk}_{mpct} \geq BM^{L}_{mpct} b_{mpt}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$
$$BM^{bulk}_{mpct} \leq BM^{U}_{mpct} b_{mpt}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$

$$BM^{bulk}_{mpct} \geq B^{U}_{mpt} BM_{mpct} + BM^{U}_{mpct} b_{mpt} - B^{U}_{mpt} BM^{U}_{mpct}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$
$$BM^{bulk}_{mpct} \leq BM^{L}_{mpct} b_{mpt} + B^{U}_{mpt} BM_{mpct} - BM^{L}_{mpct} B^{U}_{mpt}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$

$$BM^{in}_{mpct} \geq BM^{L}_{mpct} y^{in}_{mpt}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$
$$BM^{in}_{mpct} \leq BM^{U}_{mpc} y^{in}_{mpt}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$

$$BM^{in}_{mpct} \geq y^{in,U}_{mp} BM_{mpct} + BM^{U}_{mpct} y^{in}_{mpt} - y^{in,U}_{mp} BM^{U}_{mpc}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$
$$BM^{in}_{mpct} \leq BM^{L}_{mpc} y^{in}_{mpt} + y^{in,U}_{mp} BM_{mpct} - BM^{L}_{mpc} y^{in,U}_{mp}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$

$$RG^{rail}_{mpcfdst} \geq LM^{L}_{mpct} TS_{mpfft} x_{mpfdst}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$
$$RG^{rail}_{mpcfdst} \leq LM^{U}_{mpct} TS_{mpfft} x_{mpfdst}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$

$$\begin{aligned} RG^{rail}_{mpcfdst} \geq\ & TS_{mpfft} x^{U}_{mt} RG_{mpct} + LM^{U}_{mpct} TS_{mpfft} x_{mpfdst} \\ & - TS_{mpfft} x^{U}_{mt} LM^{U}_{mpct}, \\ & \qquad \forall m \in M, p \in P_m, c \in C, t \in T \end{aligned}$$
$$\begin{aligned} RG^{rail}_{mpcfdst} \leq\ & LM^{L}_{mpct} TS_{mpfft} x_{mpfdst} + TS_{mpfft} x^{U}_{mt} RG_{mpct} \\ & - LM^{L}_{mpct} TS_{mpfft} x^{U}_{mt}, \\ & \qquad \forall m \in M, p \in P_m, c \in C, t \in T \end{aligned}$$

$$LM^{rail}_{mpct} \geq LM^{L}_{mpct} LT_{mpt}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$
$$LM^{rail}_{mpct} \leq LM^{U}_{mpct} LT_{mpt}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$

$$\begin{aligned} LM^{rail}_{mpct} \geq\ & YM_{mpt} LM_{mpct} + LM^{U}_{mpct} LT_{mpt} - YM_{mpt} LM^{U}_{mpct}, \\ & \qquad \forall m \in M, p \in P_m, c \in C, t \in T \end{aligned}$$
$$\begin{aligned} LM^{rail}_{mpct} \leq\ & LM^{L}_{mpct} LT_{mpt} + YM_{mpt} LM_{mpct} - LM^{L}_{mpct} YM_{mpt}, \\ & \qquad \forall m \in M, p \in P_m, c \in C, t \in T \end{aligned}$$

$$BM^{rail}_{mpct} \geq BM^{L}_{mpct} BT_{mpt}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$
$$BM^{rail}_{mpct} \leq BM^{U}_{mpct} BT_{mpt}, \qquad \forall m \in M, p \in P_m, c \in C, t \in T$$

$$\begin{aligned} BM^{rail}_{mpct} \geq\ & B^{U}_{mpt} BM_{mpct} + BM^{U}_{mpct} BM_{mpt} - B^{U}_{mpt} BM^{U}_{mpct}, \\ & \qquad \forall m \in M, p \in P_m, c \in C, t \in T \end{aligned}$$
$$\begin{aligned} BM^{rail}_{mpct} \leq\ & BM^{L}_{mpct} BT_{mpt} + B^{U}_{mpt} BM_{mpct} - BM^{L}_{mpct} B^{U}_{mpt}, \\ & \qquad \forall m \in M, p \in P_m, c \in C, t \in T \end{aligned}$$

$$LP^{live}_{rsct} \geq LP^{L}_{rsct} w_{rst}, \qquad \forall r \in R, s \in S_r, c \in C, t \in T$$
$$LP^{live}_{rsct} \leq LP^{U}_{rsct} w_{rst}, \qquad \forall r \in R, s \in S_r, c \in C, t \in T$$

$$\begin{aligned} LP^{live}_{rsct} \geq\ & YP_{rst} LP_{rsct} + LP^{U}_{rsct} w_{rst} - YP_{rst} LP^{U}_{rsct}, \\ & \qquad \forall r \in R, s \in S_r, c \in C, t \in T \end{aligned}$$
$$\begin{aligned} LP^{live}_{rsct} \leq\ & LP^{L}_{rsct} w_{rst} + YP_{rst} LP_{rsct} - LP^{L}_{rsct} YP_{rst}, \\ & \qquad \forall r \in R, s \in S_r, c \in C, t \in T \end{aligned}$$

$$LP^{rf}_{rsct} \geq LP^{L}_{rsct}\frac{z^*_{rst}RF^*_{rst}}{1-RF^*_{rst}}, \qquad \forall r \in R, s \in S^F, c \in C, t \in T$$

$$LP^{rf}_{rsct} \leq LP^{U}_{rsct}\frac{z^*_{rst}RF^*_{rst}}{1-RF^*_{rst}}, \qquad \forall r \in R, s \in S^F, c \in C, t \in T$$

$$LP^{rf}_{rsct} \geq \frac{Z^U_{rt}RF^*_{rst}}{1-RF^*_{rst}}LP_{rsct} + LP^{U}_{rsct}\frac{z^*_{rst}RF^*_{rst}}{1-RF^*_{rst}} - \frac{Z^U_{rt}RF^*_{rst}}{1-RF^*_{rst}}LP^U_{rsct}, \qquad \forall r \in R, s \in S^F, c \in C, t \in T$$

$$LP^{rf}_{rsct} \leq LP^{L}_{rsct}\frac{z^*_{rst}RF^*_{rst}}{1-RF^*_{rst}} + \frac{Z^U_{rt}RF^*_{rst}}{1-RF^*_{rst}}LP_{rsct} - LP^L_{rsct}\frac{Z^U_{rt}RF^*_{rst}}{1-RF^*_{rst}}, \qquad \forall r \in R, s \in S^F, c \in C, t \in T$$

$$LP^{out}_{rsct} \geq LP^{L}_{rsct}u^{out}_{rst}, \qquad \forall r \in R, s \in S_r, c \in C, t \in T$$

$$LP^{out}_{rsct} \leq LP^{U}_{rsct}u^{out}_{rst}, \qquad \forall r \in R, s \in S_r, c \in C, t \in T$$

$$LP^{out}_{rsct} \geq u^{out,U}_{rs}LP_{rsct} + LP^U_{rsct}u^{out}_{rst} - u^{out,U}_{rs}LP^U_{rsct}, \qquad \forall r \in R, s \in S_r, c \in C, t \in T$$

$$LP^{out}_{rsct} \leq LP^L_{rsct}u^{out}_{rst} + u^{out,U}_{rs}LP_{rsct} - LP^L_{rsct}u^{out,U}_{rs}, \qquad \forall r \in R, s \in S_r, c \in C, t \in T$$

$$LP^{\#\#}_{rsct} \geq LP^L_{rsct}\big(w^{\#}_{rst} + \frac{z^*_{rst}RF^*_{rst}}{1-RF^*_{rst}}\big), \qquad \forall r \in R, s \in S^F, c \in C, t \in T$$

$$LP^{\#\#}_{rsct} \leq LP^U_{rsct}\big(w^{\#}_{rst} + \frac{z^*_{rst}RF^*_{rst}}{1-RF^*_{rst}}\big), \qquad \forall r \in R, s \in S^F, c \in C, t \in T$$

$$LP^{\#\#}_{rsct} \geq (YP_{rst} + Z^U_{rst})LP_{rsct} + LP^U_{rsct}\big(w^{\#}_{rst} + \frac{z^*_{rst}RF^*_{rst}}{1-RF^*_{rst}}\big) - (YP_{rst} + Z^U_{rst})LP^U_{rsct}, \qquad \forall r \in R, s \in S^F, c \in C, t \in T$$

$$LP^{\#\#}_{rsct} \leq LP^L_{rsct}\big(w^{\#}_{rst} + \frac{z^*_{rst}RF^*_{rst}}{1-RF^*_{rst}}\big) + (YP_{rst} + Z^U_{rst})LP_{rsct} - LP^L_{rsct}(YP_{rst} + Z^U_{rst}), \qquad \forall r \in R, s \in S^F, c \in C, t \in T$$

$$BP^{bulk}_{rsct} \geq BP^{L}_{rsct} v_{rst}, \quad \forall r \in R, s \in S_r, c \in C, t \in T$$
$$BP^{bulk}_{rsct} \leq BP^{U}_{rsct} v_{rst}, \quad \forall r \in R, s \in S_r, c \in C, t \in T$$

$$BP^{bulk}_{rsct} \geq V^{U}_{rst} BP_{rsct} + BP^{U}_{rsct} v_{rst} - V^{U}_{rst} BP^{U}_{rsct}, \quad \forall r \in R, s \in S_r, c \in C, t \in T$$
$$BP^{bulk}_{rsct} \leq BP^{L}_{rsct} v_{rst} + V^{U}_{rst} BP_{rsct} - BP^{L}_{rsct} V^{U}_{rst}, \quad \forall r \in R, s \in S_r, c \in C, t \in T$$

$$BP^{in}_{rsct} \geq BP^{L}_{rsct} u^{in}_{rst}, \quad \forall r \in R, s \in S_r, c \in C, t \in T$$
$$BP^{in}_{rsct} \leq BP^{U}_{rsc} u^{in}_{rst}, \quad \forall r \in R, s \in S_r, c \in C, t \in T$$

$$BP^{in}_{rsct} \geq u^{in,U}_{rs} BP_{rsct} + BP^{U}_{rsct} u^{in}_{rst} - u^{in,U}_{rs} BP^{U}_{rsct}, \quad \forall r \in R, s \in S_r, c \in C, t \in T$$
$$BP^{in}_{rsct} \leq BP^{L}_{rsct} u^{in}_{rst} + u^{in,U}_{rs} BP_{rsct} - BP^{L}_{rsc} u^{in,U}_{rs}, \quad \forall r \in R, s \in S_r, c \in C, t \in T$$

$$ZG^{shipped}_{rsct} \geq ZG^{L}_{rsct} z_{rst}, \quad \forall r \in R, s \in S_r, c \in C, t \in T$$
$$ZG^{shipped}_{rsct} \leq ZG^{U}_{rsct} z_{rst}, \quad \forall r \in R, s \in S_r, c \in C, t \in T$$

$$ZG^{shipped}_{rsct} \geq Z^{U}_{rt} ZG_{rsct} + ZG^{U}_{rsct} z_{rst} - Z^{U}_{rt} ZG^{U}_{rsct}, \quad \forall r \in R, s \in S_r, c \in C, t \in T$$
$$ZG^{shipped}_{rsct} \leq ZG^{L}_{rsct} z_{rst} + Z^{U}_{rt} ZG_{rsct} - ZG^{L}_{rsct} Z^{U}_{rt}, \quad \forall r \in R, s \in S_r, c \in C, t \in T$$

$$ZG^{\#}_{rsct} \geq LP^{L}_{rsct} w^{\#}_{rst}, \quad \forall r \in R, s \in S_r, c \in C, t \in T$$
$$ZG^{\#}_{rsct} \leq LP^{U}_{rsct} w^{\#}_{rst}, \quad \forall r \in R, s \in S_r, c \in C, t \in T$$

$$ZG^{\#}_{rsct} \geq (YP_{rst} + Z^{U}_{rst}) ZG_{rsct} + LP^{U}_{rsct} w^{\#}_{rst} - (YP_{rst} + Z^{U}_{rst}) LP^{U}_{rsct}, \quad \forall r \in R, s \in S_r, c \in C, t \in T$$
$$ZG^{\#}_{rsct} \leq LP^{L}_{rsct} w^{\#}_{rst} + (YP_{rst} + Z^{U}_{rst}) ZG_{rsct} - LP^{L}_{rsct} (YP_{rst} + Z^{U}_{rst}), \quad \forall r \in R, s \in S_r, c \in C, t \in T$$

$$ZG^{shipped}_{rsct} \geq (TG_{rsct} - TG^{tol}_{rsc}) z_{rst} - si_{rsct}, \quad \forall r \in R, s \in S_r, c \in C, t \in T$$
$$ZG^{shipped}_{rsct} \leq (TG_{rsct} + TG^{tol}_{rsc}) z_{rst} + ei_{rsct}, \quad \forall r \in R, s \in S_r, c \in C, t \in T$$

4.4 Test cases

In this section, we run test cases to validate the new model. The same 5-period and 11-period problems solved in Section 3.6 are considered in order to analyse the comparison of results. We have solved the two problems without and with the grade constraints previously. For the problems with the grade constraints, the iterative method was used to run the problem in AIMMS as MILP problems. In this section, we run the full problems using convex relaxation presented in this chapter.

Several new parameters are added, namely the lower and upper bounds for each of the grade composition variables. Similarly as before, we use 1% MIP relative tolerance gap as the termination criterion for all cases. Problems of larger size will be considered as case studies in Chapter 5.

4.4.1 5-period case

We run our implementations in AIMMS software with linkage to CPLEX 12.6.3 solver. The summary of results of the problem is displayed in the Table 4.1. This includes the time taken, the gap achieved, the total number of trains, the total amount of shipping, the total profit, and the grade deviation cost.

Solving time (seconds)	69.64
Gap (%)	0.82
Number of trains	873
Total amount of shipping (kt)	22,786.89
Total profit ($)	1,241,528.16
Grade deviation cost ($)	57,610.40

Table 4.1: Summary of results for the 5-period case using convex relaxation

The time taken for the model to find solutions within 1% of solution gap is approximately one minute, which is reasonable for the problem of small size. From the table above, the quality of the solutions clearly validate this model.

4.4.2 11-period case

The results of the full problem using the convex relaxation approach are displayed in Table 4.2. This includes the time taken, the gap achieved, the total number of trains, the total amount of shipping, the total profit, and the grade deviation cost.

Solving time (seconds)	696.62
Gap (%)	0.50
Number of trains	1,842
Total amount of shipping (kt)	45,477.48
Total profit ($)	2,269,630.05
Grade deviation cost ($)	294,210.95

Table 4.2: Summary of results for the 11-period case using convex relaxation

As expected, the time taken for the tool to solve the problem increases to over 11 minutes. The remaining results show that the model is still reliable to solve problems of larger size.

4.4.3 Discussion

Solving time and solution gap

Table 4.3 below provides comparisons of solving times and solution gaps generated by the test case problems without the grade constraints, using the iterative method, and using the convex relaxation.

	Number of iterations	Solving time (s)		Gap (%)	
		5 periods	11 periods	5 periods	11 periods
Without grades	-	0.27	3.60	0.7	0.27
Iterative method	2	0.48	4.21	0.83	0.32
	3	0.53	4.87	0.89	0.36
	4	0.61	5.01	0.90	0.36
	5	0.72	5.04	0.84	0.26
	6	0.77	5.39	0.92	0.40
	7	0.90	6.00	0.92	0.47
	8	0.93	5.93	0.92	0.23
Convex relaxation	-	69.64	696.62	0.82	0.50

Table 4.3: Summary of solving times and solution gaps for the 5-period and 11-period case problems

It has been shown that solving the problems using the convex relaxation approach takes much longer than the iterative approach, taking over 1 minute for the 5-period problem and 11 minutes for the 11-period problem. This is due to the increase of problem size caused by the additional variables and constraints. In reality, however, 11 minutes is still a reasonable solving time, albeit much longer compared to the time taken by the iterative method.

Number of trains and total shipping

The solutions for the total throughput, expressed by the number of trains and the total shipping amount, are summarised in Table 4.4.

	Number of iterations	Number of trains		Total shipping (kt)	
		5 periods	11 periods	5 periods	11 periods
Without grades	-	873	1,842	22,939.01	45,754.34
Iterative method	2	873	1,841	22,939.01	45,725.14
	3	873	1,842	22,962.15	45,753.84
	4	873	1,842	22,962.15	45,753.84
	5	873	1,841	22,939.01	45,724.90
	6	873	1,840	22,954.91	45,697.84
	7	873	1,840	22,954.91	45,699.69
	8	873	1,842	22,954.91	45,756.45
Convex relaxation	-	873	1,842	22,786.89	45,477.48

Table 4.4: Summary of number of trains and total shipping amount for the 5-period and 11-period case problems

The results obtained by the convex relaxation method for the number of trains are consistent with those obtained by the problems without grades and with grades using iterative method. Slight decrease can be seen in the total shipping, nevertheless. Based on the two case problems, the iterative method performs slightly better than the convex relaxation approach in regards to generating higher iron ore throughput.

Total profit

As mentioned previously in Chapter 3, we pay more attention to the total profit as the objective function is weighted to maximising it. We present Figures 4.1 and 4.2 to help with the analysis of the total profit.

The total profits for the 5 and 11-period problems produced by the convex relaxation method increase by approximately 9% and 15% in respective order;

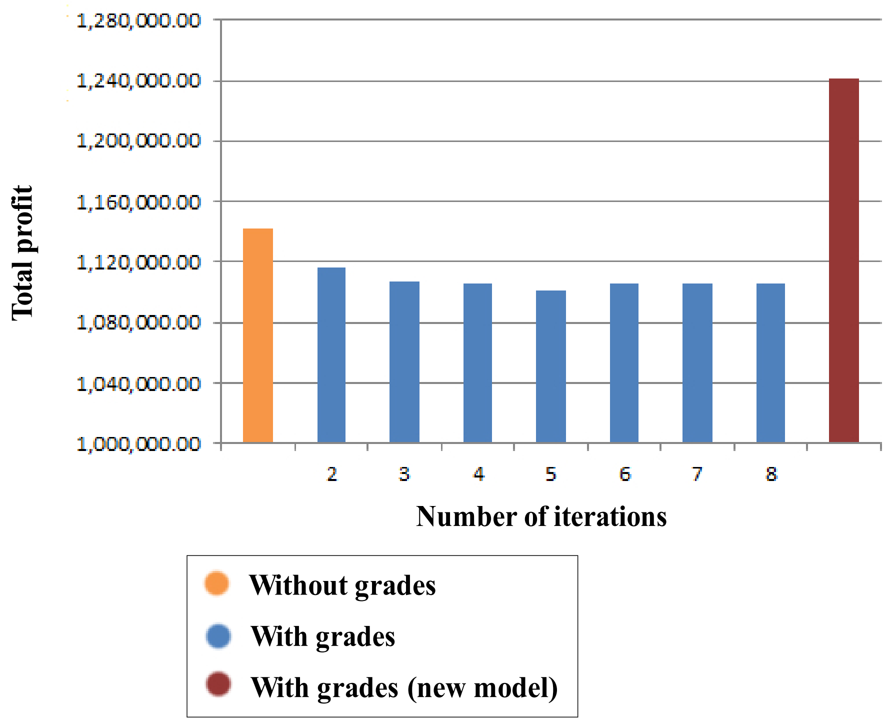

Figure 4.1: Total profit for the 5-period case

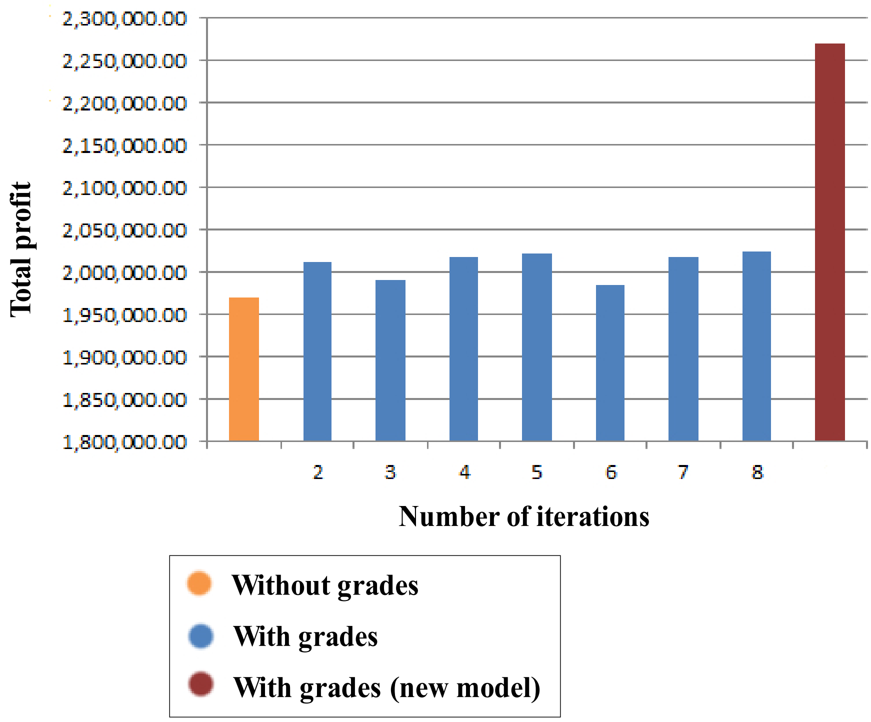

Figure 4.2: Total profit for the 11-period case

compared to -3% and 2% produced by the iterative method. Based on the two small case problems, we can conclude that the new model is more effective in producing better objective values.

Grade deviation cost

Since the two models differ in how they handle the grade constraints, it is necessary to note the difference in the results for the grade deviation cost. We present the summary of these results in Figure 4.3 and 4.4.

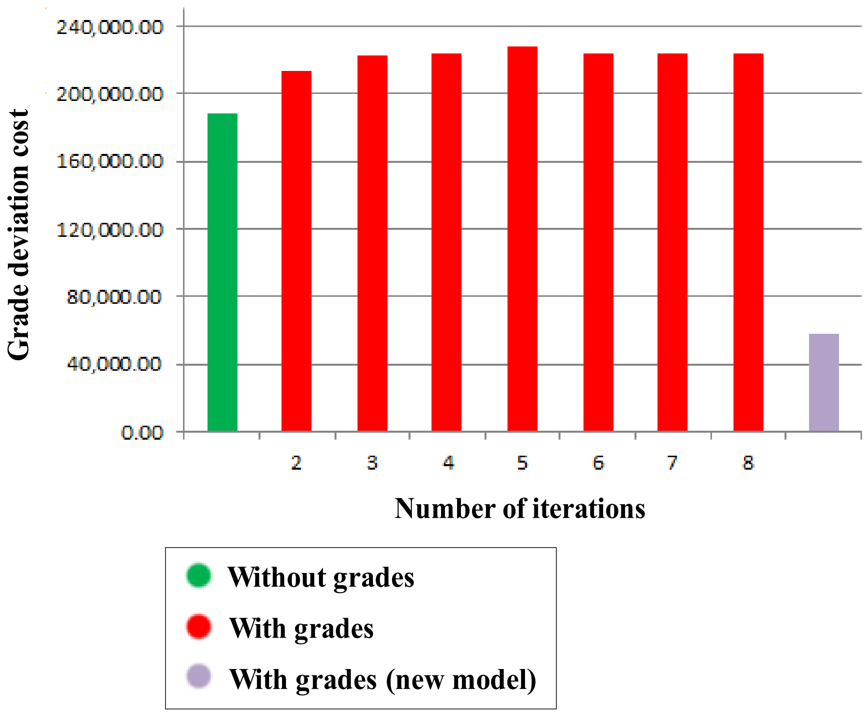

Figure 4.3: Total cost of grade deviations for the 5-period case

It is obvious that the new model is more consistent in producing minimum grade deviation cost, hence meeting the target quality demands. The new model decrease the grade deviation by 69% and 56% for the 5 and 11-period problems respectively; compared to -18% and 6% produced by the iterative method.

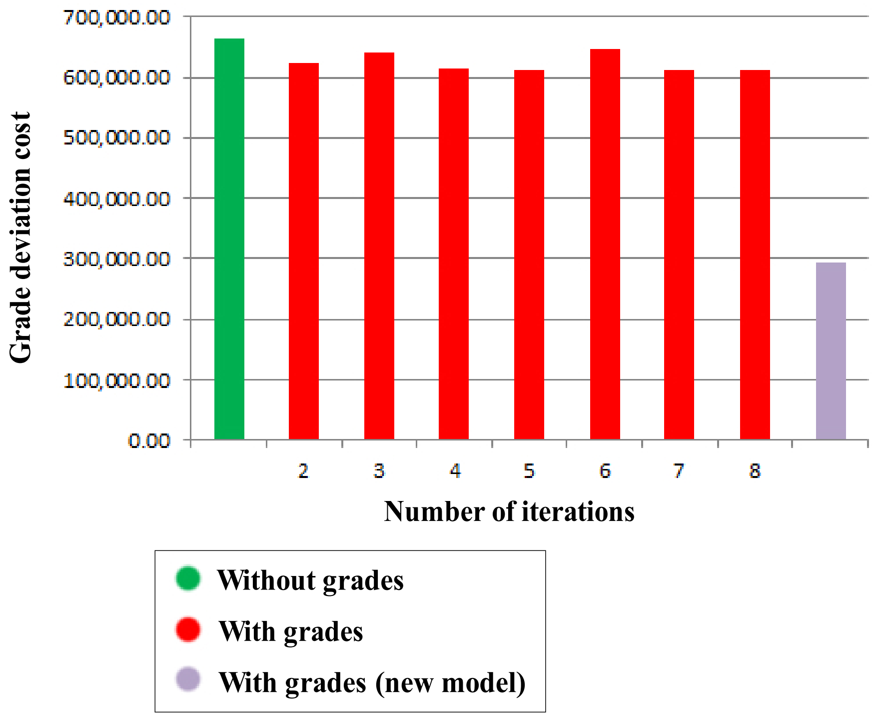

Figure 4.4: Total cost of grade deviations for the 11-period case

4.5 Conclusions

In the previous chapter, we considered an optimisation model for logistics planning in iron ore mining developed by CSIRO as well as the solution approach to tackle the non-linearity. In this chapter, we have introduced a new model which has had changes in the problem formulation but essentially portrays an equivalent problem.

The nature of the problem remains the same as the model described in Chapter 3, which is to allocate trains to mines such that operational and other costs are minimised and total throughput is maximised, taking into account the grade quality requirements. The incentives whose purpose are to encourage a higher number of trains still apply in this model.

We have implemented the new model using the same 5-period and 11-period test cases from the previous chapter in AIMMS 4.21 software for validation purpose. We solved the MILP problem using CPLEX 12.6.3 version. We have compared the results obtained by the model without the grade constraints, the model with iterative approach, and the new model with convex relaxation approach.

The new model requires a much longer time to solve the MILP problems compared to the previous model due to a higher number of decision variables and constraints. The new model, however, indicates a better performance in terms of producing a higher objective value and lower grade deviation cost for both the 5-period and 11-period cases.

In the following chapter, we will discuss some case studies of a larger size. The two models we have presented so far will be applied and the results will be analysed and compared.

Chapter 5

Case studies

Two different approaches have been presented to deal with the non-linear constraints occurring in the logistics planning problem in mining. They have also been implemented and validated using the same test cases with 5 and 11 periods. This chapter discusses two case studies based on real life problems in the mining industry. The two models have been implemented to solve each case and the results will be analysed.

5.1 Introduction

Our industry partner provided us with two different data sets of real life cases. The first data set involves a monthly logistics operation in iron ore mining over a one year time horizon, whereas the second data set is a 10-month problem in which the periods vary from one day to one week. Despite a shorter time horizon, we consider the second case study to be more difficult to solve due to a higher number of periods, and hence a higher number of constraints and variables in the mathematical formulation.

Both cases were solved in AIMMS 4.21 with linkage to CPLEX 12.6.3 solver using the two models we outlined in prior chapters. We used certain values of relative MIP gap tolerance as the termination criterion for the solver when implementing the model. A lower percentage of gap tolerance is preferable as it will lead to better performance and therefore closer to optimality. However, provided that the solution is reasonably good, a higher percentage of gap tolerance may practically be more advantageous to generate a solution more quickly.

For convenience, we introduce some notations as follows:

- We refer to the first and second case studies as problems **P1** and **P2** respectively.

- The model that omits iron ore grade constraints is referred to as model **MO**. This model involves maximising the objective function without the grade deviation penalty subject to constraints (3.9)–(3.31).

- The first model with iron ore grade constraints described in Chapter 3 is referred to as model **M1**. The complete formulation of this model has been presented in Section 3.4.7. We use the iterative method described in Section 3.5 as the solution approach.

- The second model with iron ore grade constraints described in Chapter 4 is referred to as model **M2**. The complete formulation of this model has been presented in Section 4.3.6.

In terms of numerical results, we discuss the quality of our solutions based on several aspects:

- For each implementation, we provide the solving time and the solution gap achieved. As we seek a model which is practical and applicable in real life industry problems, the time taken to implement the model is aimed at being reasonable. Furthermore, the MIP gap achieved will tell us how close our solutions are to optimality.

- RTIO seeks maximum utilisation from their trains. In order to do this, we have assumed that no cost for operating trains is involved and added incentives in the objective function to encourage a higher number of trains. We show the results by providing the total number of trains obtained from each implementation.

- As this model aims to maximise the total throughput of iron ore, we provide the total tonnes of shipping in our results.

- The normal way to measure the quality of solutions in an optimisation problem is by looking at the optimised objective value. In our model, the objective value represents the total profit obtained from the logistics plan during the given periods.

- The blending requirement constraints in this model were added to maintain the grade quality within target range and therefore minimise the grade deviations of the shipped products from the target grade quality. We indicate the total cost of grade deviation for each case in our results.

We present the first case study (problem **P1**) in Section 5.2. This section includes the main features of the problem, the computational results obtained from the implementations, and the discussion. In the computational results section, we will show our results obtained by implementing the model **M0**, then **M1** and **M2**.

We describe the second case study (problem **P2**) in Section 5.3. Similarly to problem **P1**, this section is divided into three subsections. They are the main features, the computational results, and the discussion sections. We have also modified problem **P2** in order to reduce the solving time. The description and the computational results are presented in Section 5.4. The concluding remarks of our case studies are presented in Section 5.5.

5.2 Case study 1

Problem **P1** looks at a monthly iron ore mining operation for a time horizon of one year. Although RTIO prefers a weekly schedule to monthly, this case study is implemented to show the flexibility of the optimisation models.

5.2.1 Main features

Some important characteristics of problem **P1** are outlined as follows:

- We consider 12 periods where each period represents one month.
- The logistics operation involves 15 mines in the Pilbara with 21 pits in total considered in the formulation.
- Most of the mines produce two different types of mined products, namely lump and fines, based on the size of the ore. In addition, we separate the high grade lump from the low grade lump product produced at the Tom Price mine.
- Most of the mines have a FIFO regime. The Paraburdoo mine is the only one with a LIFO regime.
- 5 mines are constrained by the joint venture contract obligations in which the number of trains must comply with the cumulative target.
- The mines are located within 7 mine regions.
- The rail network consists of 2 different fleets.
- The logistics operation involves 4 shipping terminals located at 2 different ports in the Pilbara; all of which have car dumping facilities.

- This case considers 9 different shipped products at the ports.
- Typically, each product type contains 10 different components, including iron (Fe), silicon dioxide (SiO_2), aluminium oxide (Al_2O_3), phosphorus (P), etc. Some product types contain only 6 different components.
- The return fines ratio at each port is fixed across the periods.
- A high percentage of incentive is used in the objective function to optimise the use of the trains, thus maximising the total throughput. This incentive is described in the constraint (3.8).

5.2.2 Computational results

We modeled the MILP formulations of problem **P1** in AIMMS 4.21 software. After some trials, we decided to use the CPLEX solver within AIMMS as it generates solutions considerably faster than Gurobi. We use the relative MIP gap tolerance of 1%, which means the solving procedure will terminate once the integer feasible solution is achieved within 1% of optimality.

Without grade constraints (M0)

We firstly solved a simpler version of problem **P1**, that is, without the grade constraints (model **M0**). The model generates 99,476 variables, including 7,284 integer variables, and 92,108 constraints in the AIMMS formulation. Table 5.1 shows the summary of the solutions of problem **P1** solved using model **M0**.

Solving time (seconds)	32.43
Gap (%)	0.01
Number of trains	13,119
Total amount of shipping (kt)	351,886.63
Total profit ($)	2,221,475,479.82
Grade deviation cost ($)	2,589,899.96

Table 5.1: Summary of results for problem **P1** using model **M0**

The solver found the optimal solution in 32.43 seconds. Using a 1% relative MIP gap tolerance, the feasible solution is obtained within 0.01% of optimality.

With grade constraints (M1)

Model **M1** is applied to problem **P1**. In this case, the model contains 119,997 variables; 7,284 of which are integer variables, and 115,349 constraints in the AIMMS formulation. We applied the solution algorithm outlined in Section 3.5 for up to 8 iterations. The results for problem **P1** solved using model **M1** are indicated in Tables 5.2 and 5.3 below.

Number of iterations	Solving time (seconds)	Gap (%)
2	39.11	0.01
3	38.77	0.01
4	14.05	0.08
5	39.33	0.07
6	41.14	0.08
7	49.79	0.01
8	34.28	0.01

Table 5.2: Solving times and solution gaps for problem **P1** using model **M1**

Number of iterations	Number of trains	Total shipping (kt)	Total profit ($)	Grade deviation cost ($)
2	13,121	351,881.37	2,221,690,532.07	2,404,653.05
3	13,125	351,959.05	2,221,702,427.48	2,432,709.99
4	13,039	349,919.51	2,219,626,892.31	2,938,064.75
5	13,040	349,997.30	2,220,001,367.53	2,621,906.36
6	13,042	350,050.11	2,220,024,872.55	2,529,651.17
7	13,120	351,875.90	2,221,951,441.04	2,161,661.07
8	13,122	351,924.28	2,221,564,201.58	2,552,459.32

Table 5.3: Number of trains, total amount of shipping, total profits, and grade deviation costs for problem **P1** using model **M1**

The results obtained indicate that as we use more iterations, the results do not appear to converge. Instead, the solving time, the MIP gap, the number of trains, the total shipping quantity, the total profit, and the grade deviation cost fluctuate as more iterations are used. Based on the final objective values and the grade deviation costs, we conclude that the model with 7 iterations produces the best logistics plan.

With grade constraints (M2)

We apply model **M2** which uses convex relaxation to solve problem **P1**. This model generates 247,773 variables; 10,644 of which are integer variables, and 368,941 constraints in the AIMMS formulation. Table 5.4 displays the summary of the results.

Solving time (seconds)	6,074.94
Gap (%)	0.02
Number of trains	13,121
Total amount of shipping (kt)	352,047.85
Total profit ($)	2,221,755,738.40
Grade deviation cost ($)	1,961,060.92

Table 5.4: Summary of results for problem **P1** using model **M2**

For this implementation, we found a good solution within 0.02% of optimality in approximately two hours. While the solving time taken is much longer, the quality of the solutions is reasonably good. We will discussed this further in the next section.

5.2.3 Discussion

In this case study, we have produced solutions from implementing models **M0**, **M1**, and **M2** derived from the previous chapters to problem **P1**. We aimed to produce an optimal plan that represents monthly logistics in iron ore mining operation over a year in the Pilbara.

Solving time and solution gap

We present Table 5.5 to show the comparison of solving times and solution gaps obtained for problem **P1** using three different models. We also include the number of variables and constraints in the table to compare the problem size.

To find solutions within the specified gap, we expected the time taken to implement model **M1** to increase as we used more iterations. It has been shown, nonetheless, that the associated solving times rise and fall irregularly across different numbers of iterations. Furthermore, the time achieved with

Model applied	No. of variables	No. of integers	No. of constraints	No. of iterations	Solving time (s)	Gap (%)
M0	99,476	7,284	92,108	-	32.43	0.01
M1	119,997	7,284	115,349	2	39.11	0.01
				3	38.77	0.01
				4	14.05	0.08
				5	39.33	0.07
				6	41.14	0.08
				7	49.79	0.01
				8	34.28	0.01
M2	247,773	10,644	368,941	-	6,074.94	0.02

Table 5.5: Summary of solving times and solution gaps for problem **P1**

4 iterations is less than half of the time taken by applying model **M0**, which is supposedly easier to solve. Despite the inconsistency, all of the solving times achieved by model **M1** are very good when it comes to solving a full mining logistics problem with one year time horizon.

We have seen that model **M2** generates around twice as many variables and 3 times as many constraints as model **M1**. Due to the great difference in size, we have expected the solving time to rise greatly. It is shown that **M2** solves problem **P1** in more than 6,074.94 seconds, or approximately 1 hour 40 minutes. This makes model **M2** less efficient in terms of the time taken to implement.

Number of trains and total shipping

The summary of results for number of trains and total amount of shipping for problem **P1** can be seen in Table 5.6.

The irregular pattern in the solutions for model **M1** also occurs in the total number of trains and total shipping amount. The results show that applying 4 iterations generates the lowest total throughput, while 3 iterations generate the highest. In general, model **M1** produces good solutions when the number of iterations is less than 4 or greater than 6. This again shows the instability of the results for model **M1**.

Based on the results, we conclude that model **M2** produces the best total throughput overall with small difference, despite model **M1** with 3 iterations showing a higher number of trains.

Model applied	Number of iterations	Number of trains	Total shipping (kt)
M0	-	13,119	351,886.63
	2	13,121	351,881.37
	3	13,125	351,959.05
	4	13,039	349,919.51
M1	5	13,040	349,997.30
	6	13,042	350,050.11
	7	13,120	351,875.90
	8	13,122	351,924.28
M2	-	13,121	352,047.85

Table 5.6: Summary of number of trains and total shipping amount for problem **P1**

Total profit

In analysing the total profit, we note that the objective function in our problem is weighted to maximising the total profit. The results are indicated in Figure 5.1.

We observed a similar trend from the previous analysis occuring in the total profits. The results for model **M1** show improvements when the number of iterations used is less than 4 or greater than 6. Based on the results for the total profits, model **M1** performs the best when 7 iterations are used. However, taking into account all the results of model **M1** with various numbers of iterations, the objective value has decreased by 0.02% overall.

Model **M2** performs better than model **M1** overall, with an increase of total profit by 0.01% from model **M0**. The highest total profit, nevertheless, is still achieved by model **M1** with 7 iterations, followed by model **M2**. Based on the proximity from the result of model **M0**, we accept the solutions produced by model **M1** with any number of iterations less than 4 or greater than 6, as well as model **M2**, as being good and reasonable solutions.

Grade deviation cost

Although the grade deviation cost is not the sole cost involved in the objective function, it is still an important aspect in our result analysis because the models differ in how we treat the grade requirement constraints. One way to assess the effectiveness of the methods is to seek the model which gives the biggest impact in minimising the grade deviation cost. The results for total

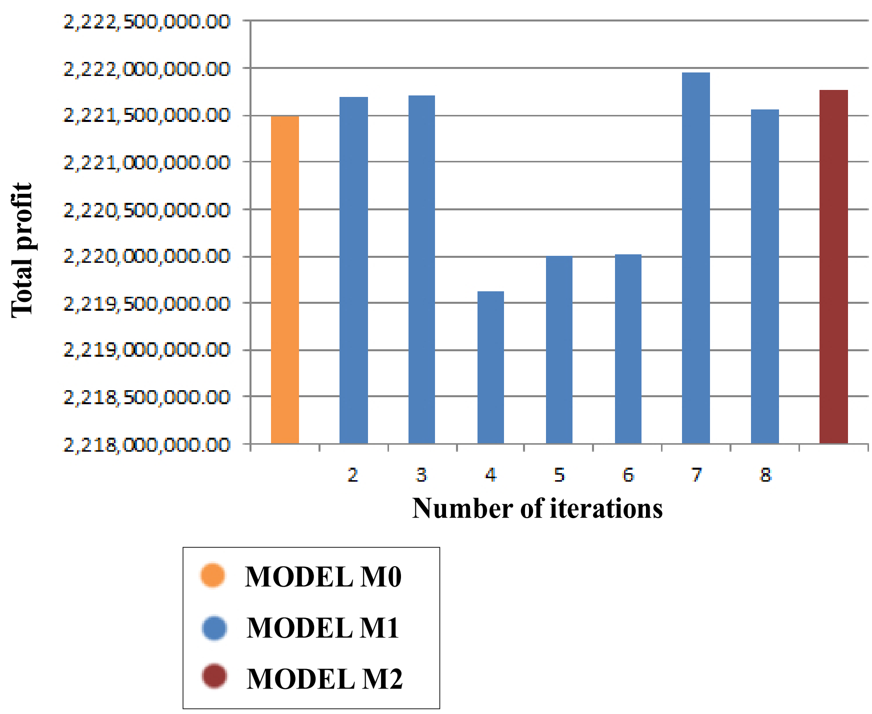

Figure 5.1: Total profits for problem **P1**

costs of grade deviations are compared side by side in Figure 5.2.

It is observed that two results, obtained by model **M1** with 4 and 5 iterations, generate higher grade deviation costs than model **M0**. In this case, it is pointless to include the grade constraints whose purpose it to satisfy grade quality requirements, and hence minimising the grade constraints. Overall, with 2 to 8 iterations, model **M1** has managed to decrease the grade deviation cost by 2.69%.

We can see that model **M2** performs substantially better than model **M1** in terms of minimising the grade deviations. It produces the lowest grade deviation cost among all implementations, with a decrease of 24.28% from the same cost associated with model **M0**.

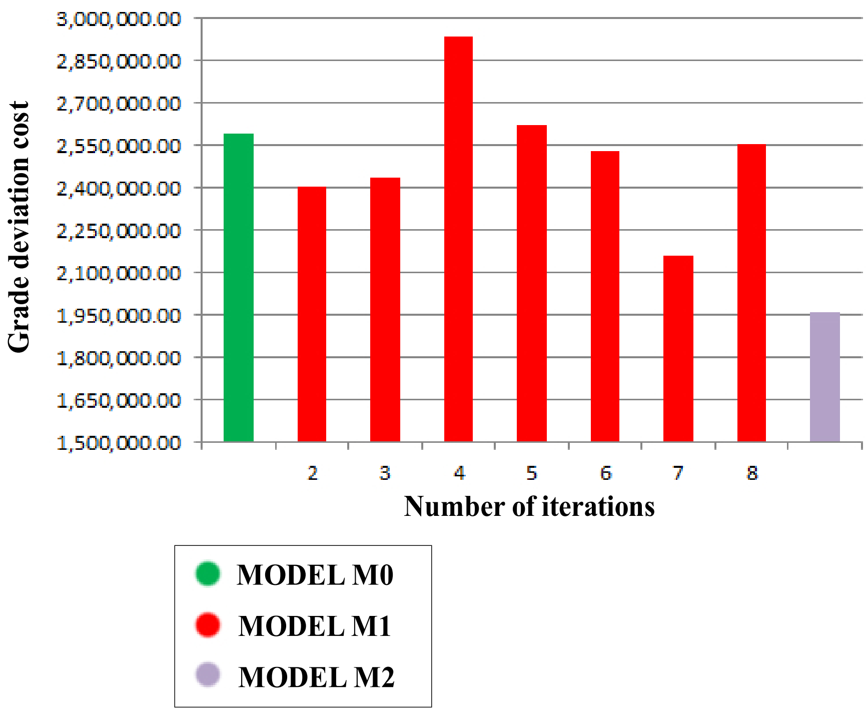

Figure 5.2: Total costs of grade deviation for problem **P1**

5.3 Case study 2

Case study 2 (**P2**) considers a real life mining logistics operation over ten months. Instead of looking at a monthly schedule, this problem examines shorter length periods, ranging from daily to weekly. This case study is the closest to mining schedule that is implemented in real life.

5.3.1 Main features

Some of the features of problem **P2** are the same as problem **P1**. Some differences have been observed since the data for this case study is less recent than the previous one. The main differences that problem **P2** has in comparison to problem **P1** are listed below:

- Each period in problem **P2** represents between one and seven days (one week). In total, problem **P2** considers 52 number of periods.

- Only 18 pits in total are considered in the problem formulation with the same number of mines.
- Only 2 mines are constrained by the joint venture contract obligations in which the number of trains must comply with the cumulative target.
- The problem only considers 4 mine regions at which all the mines are located.
- This problem only considers 7 different shipped products at the ports.
- A low percentage of incentive is used in the objective function. This incentive is described in the constraint (3.8).

5.3.2 Computational results

The CPLEX solver within AIMMS 4.21 optimisation software is again used to solve the MILP formulations of problem **P2**. Due to a larger size than problem **P1**, we use a higher relative MIP gap tolerance of 2%. This means the solving procedure will terminate once the integer feasible solution is achieved within 2% of optimality.

Without grade constraints (M0)

We apply model **M0** to problem **P2**. The AIMMS formulation of this model generates 32,312 variables, which include 9,580 integer variables, and 22,411 constraints. The summary of the solutions is shown in Table 5.7.

Solving time (seconds)	31.92
Gap (%)	0.55
Number of trains	9,639
Total amount of shipping (kt)	243,089.55
Total profit ($)	11,447,396.63
Grade deviation cost ($)	2,509,051.87

Table 5.7: Summary of results for problem **P2** using model **M0**

AIMMS found the solution with 0.55% gap from optimality within 31.92 seconds.

With grade constraints (M1)

Model **M1** is implemented to solve problem **P2**. For this case, the model generates 117,233 decision variables; 9,580 of which are integer, and 124,282 constraints in the AIMMS formulation. The summary of the results is shown in Tables 5.8 and 5.9.

Number of iterations	Solving time (seconds)	Gap (%)
2	39.22	1.73
3	39.12	1.56
4	29.19	1.68
5	44.99	1.30
6	76.59	1.02
7	144.91	1.35
8	63.18	0.89

Table 5.8: Solving time and solution gap for problem **P2** using model **M1**

Number of iterations	Number of trains	Total shipping (kt)	Total profit ($)	Grade deviation cost ($)
2	9,634	242,987.11	11,386,766.54	2,449,318.58
3	9,638	243,096.52	11,456,755.38	2,397,552.30
4	9,636	242,998.09	11,400,364.75	2,443,000.22
5	9,636	242,985.40	11,477,094.46	2,406,542.94
6	9,645	243,247.93	11,502,252.78	2,413,139.25
7	9,640	243,126.64	11,455,772.94	2,421,513.94
8	9,639	243,071.73	11,549,710.54	2,380,311.35

Table 5.9: Number of trains, total amount of shipping, total profits, and grade deviation costs for problem **P2** using model **M1**

Similar to problem **P1**, the solutions for problem **P2** show irregular patterns when model M1 is used with various numbers (up to 8) of iterations. The irregularity is reflected in the solving times, total number of trains, total shipping amount, total profits, and the grade deviation costs. The model with 6 iterations produces a plan with the most throughput, whereas, the model with 8 iterations produces the best objective value and minimum grade deviation cost.

With grade constraints (M2)

We solve problem **P2** using model **M2**. The model generates a MILP formulation in AIMMS with 414,189 variables, which include 21,956 integer variables, and 976,630 constraints. We display the summary of results in Table 5.10.

Solving time (seconds)	43,976.99
Gap (%)	1.87
Number of trains	9,639
Total amount of shipping (kt)	243,043.52
Total profit ($)	11,936,461.06
Grade deviation cost ($)	1,761,146.72

Table 5.10: Summary of results for problem **P2** using model **M2**

This model requires more than 12 hours to complete the solving procedure and obtains a solution within 1.87% of optimality. The quality of results will be discussed in the immediate section.

5.3.3 Discussion

We have utilised models **M0**, **M1** and **M2** to find solutions for problem **P2**. This problem is aimed to determine an optimal logistics plan for iron ore mining operations in the Pilbara over a 10 month time horizon on a weekly basis. Similarly as previous problem, we emphasis our analysis on the solving times, total number of trains, total amount of shipping, total profits, and grade deviation costs.

Solving time and solution gap

The summary of solving times and solution gaps for problem **P2** is shown in Table 5.11. This summary includes the number of variables, number of integer variables, number of constraints, number of iterations, solving times, and solution gaps obtained by models **M0**, **M1**, and **M2**.

Although the problem size of model **M1** increased greatly due to the inclusion of the grade constraints, it does not affect the solving times to a large

Model applied	No. of variables	No. of integers	No. of constraints	No. of iterations	Solving time (s)	Gap (%)
M0	32,312	9,580	22,411	-	31.92	0.55
M1	117,233	9,580	124,282	2	39.22	1.73
				3	39.12	1.56
				4	29.19	1.68
				5	44.99	1.30
				6	76.59	1.02
				7	144.91	1.35
				8	63.18	0.89
M2	414,189	21,956	976,630	-	43,976.99	1.87

Table 5.11: Summary of solving times and solution gaps for problem **P2**

extent. The longest time taken by model **M1** is 144.91 seconds, or approximately 2 minutes 25 seconds, obtained by applying 7 iterations. The solving times generally increase as more iterations are used, with the exceptions of 4 and 8 iterations where the solving times decline. The inconsistency occurs, and yet model **M1** remains efficient in terms of the solving time taken.

Model **M2** incorporates additional variables and constraints for the grade requirements, generating 3.5 times as many variables and almost 8 times as many constraints as model **M1**. While the problem is still solvable by AIMMS, the time taken substantially rise to 43,976.99 seconds, or over 12 hours.

Number of trains and total shipping

Table 5.12 displays the summary of results for the number of trains and total amount of shippig for problem **P2**.

The solutions obtained by model **M1** rise and fall irregularly with no convergence. The lowest and highest numbers of trains are achieved by 2 and 6 iterations respectively; while the lowest and highest shipping amounts are achieved by 5 and 6 iterations respectively.

Comparing the results of the three different models, we conclude that the inclusion of the grade constraints does not bring a large effect on the total throughput in this case study, as all models produce the similar total throughput overall.

Model applied	Number of iterations	Number of trains	Total shipping (kt)
M0	-	9,639	243,089.55
	2	9,634	242,987.11
	3	9,638	243,096.52
	4	9,636	242,998.09
M1	5	9,636	242,985.40
	6	9,645	243,247.93
	7	9,640	243,126.64
	8	9,639	243,071.73
M2	-	9,639	243,043.52

Table 5.12: Summary of number of trains and total shipping amount for problem **P2**

Total profit

We present the total profits obtained by models **M0**, **M1**, and **M2** side by side in Figure 5.3 for comparison and analysis.

An inconsistency has been again shown in the solutions for model **M1** where some results are higher than model **M0** and some are lower. Overall, this model produces a higher total profit by only 0.12% from model **M0**.

The improvement that model **M2** shows in terms of the objective value (total profit) in this case study is much more obvious. With the result shows the highest total profit obtained, model **M2** increase the total profit by 4.27%. We consider this a significant growth as it means a $500,000 increase in value.

Grade deviation cost

Figure 5.4 demonstrates the comparison of results for the grade deviation costs generated by models **M0**, **M1**, and **M2** for problem **P2**.

As predicted, the costs of grade deviations produced by models **M1** and **M2** are less in value than the cost produced by model **M0**. This indicates that our results are more consistent in this case study than previously.

When model **M1** is used, we analyse that 8 iterations gives the minimum grade deviation, albeit no convergence is observed. Overall, model **M1** improves the cost of grade deviation with a decrease of 3.71%, which is more than $90,000 in value.

It can be seen, nevertheless, that model **M2** produces the best grade

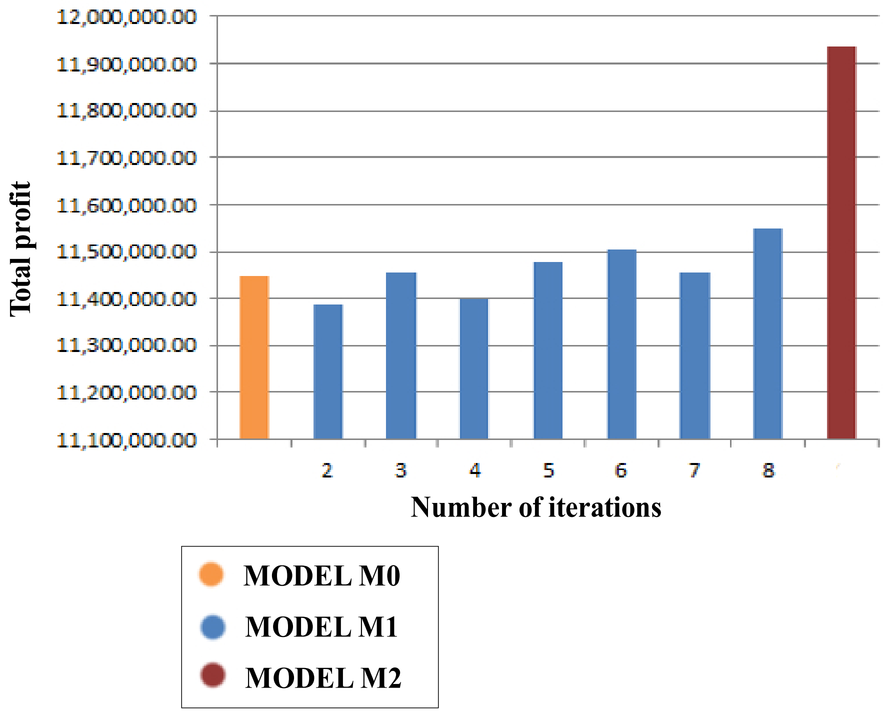

Figure 5.3: Total profits for problem **P2**

deviation cost in the results. The model drops the cost of grade deviation by 29.81%, which is equivalent to almost $750,000 in value. This improvement of results is significant in comparison to that shown by model **M1**.

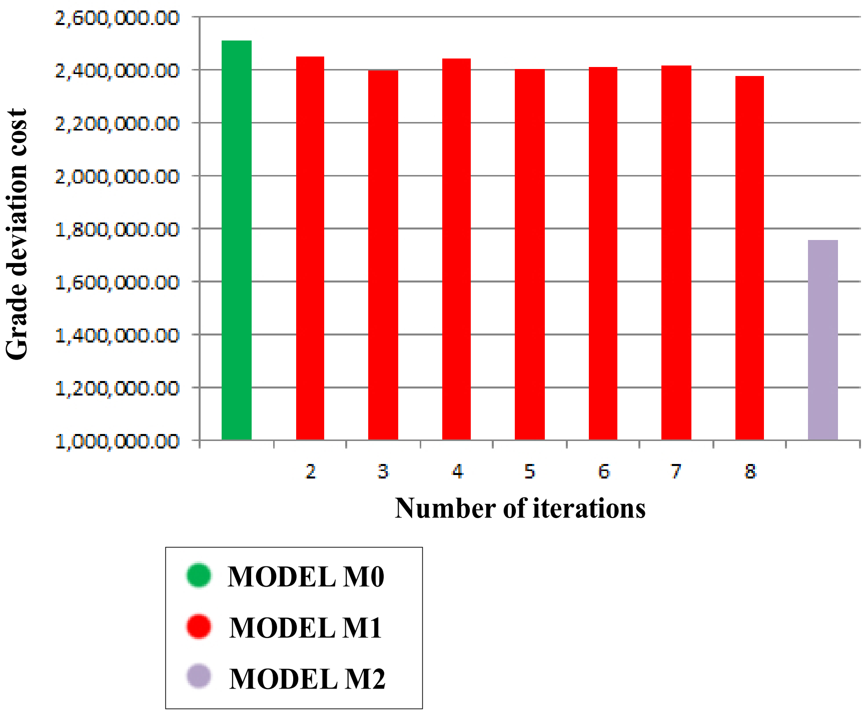

Figure 5.4: Total costs of grade deviation for problem **P2**

5.4 Case study 2 (aggregated)

Based on the results of the two case studies, we have seen that model **M2** has consistently performed better than model **M1** in generating optimal solutions. In terms of the time taken to implement the model, however, the model is less practical as it takes a significant amount of time to solve. In order to reduce the solving time, we amend the data set of problem **P2** given by RTIO in this section. We take this approach as the attempt to produce a logistics plan with similar quality of solutions as model **M2** but is more efficient and reasonable in terms of the solving time.

In this model, the planning periods in problem **P2** have been aggregated such that the mining operation is solved as per normal (between daily and weekly) for a certain number of periods and monthly for the remaining periods. The approach taken is based on the idea that the optimisation schedule for the later periods is not as critical as the immediate ones and hence can be implemented monthly. This approach will reduce the number of periods greatly, and hence reducing the problem size.

5.4.1 Main features

We provide 3 different aggregations to problem **P2**, referred to as problems **P2.1**, **P2.2**, and **P2.3**. These modifications reduce the problem size greatly, thus reducing the solving time. We describe each problem discussed in this section as below:

- The original problem **P2** contains 52 periods over a 10 month time horizon.
- In problem **P2.1**, we leave the data in problem **P2** unchanged for the first 27 periods, that is, 5 months. We then combine the data for the remaining into a 5 month set. The total number of periods in this problem is therefore 27 + 5 = 32.
- In problem **P2.2**, we leave the data in problem **P2** unchanged for the first 16 periods, that is, 3 months. We then combine the data for the remaining into a 7 month set. The total number of periods in this problem is therefore 16 + 7 = 23.
- In problem **P2.3**, we leave the data in problem **P2** unchanged for the first 6 periods, that is, 1 month. We then combine the data for the remaining into a 9 month set. The total number of periods in this problem is therefore 6 + 9 = 15.

All other features for problems **P2.1**, **P2.2**, and **P2.3** remain the same.

5.4.2 Computational results

We perform model **M2** on problems **P2.1**, **P2.2**, and **P2.3**, in addition to the results for problem **P2** obtained in the prior section. Each problem is solved in AIMMS 4.21 linked to CPLEX 12.6.3 solver with 2% MIP gap tolerance.

Table 5.13 below shows the solving times and the gaps achieved in solving the 4 different cases of problem **P2**, along with the number of variables, integer variables, and constraints.

Problem type	Number of variables	Number of integers	Number of constraints	Solving time (s)	Gap (%)
P2	414,189	21,956	976,630	43,976.99	1.87
P2.1	254,485	13,456	601,258	10,382.63	1.23
P2.2	182,961	9,679	432,355	4,097.72	0.69
P2.3	119,367	6,279	282,206	1,536.86	0.50

Table 5.13: Problem sizes, solving times and solution gaps for problems **P2**, **P2.1**, **P2.2**, and **P2.3** using model **M2**

As anticipated, the amendments of the data set reduce the solving time greatly due to the significant decrease in the problem size. It takes 2.88 hours, 1.14 hours, and 0.43 hours to solve problems **P2.1**, **P2.2**, and **P2.3** respectively; in comparison to the original problem **P2** which take more or less 12 hours.

Table 5.14 displays the summary of the number of trains for the 4 different problems. For each problem, we show the detail of the number of trains used in each period and month, and the total in the whole operation. We can see that the numbers of trains are similar in the earlier periods but slightly differ in the later periods. The total number of trains overall increases as more aggregations are implemented.

Month	Period	Number of trains			
		P2	P2.1	P2.2	P2.3
1	1	61	62	62	62
	2	177	177	177	177
	3	210	211	210	209
	4	207	206	207	206
	5	224	224	224	224
	6	30	30	30	30
	Total	909	910	910	908
2	7	182	182	182	932
	8	220	220	220	
	9	216	216	216	
	10	225	225	225	
	11	89	89	89	
	Total	932	932	932	932
3	12	109	109	109	928
	13	223	223	223	
	14	204	204	204	
	15	226	226	226	
	16	162	162	162	
	Total	924	924	924	928
4	17	33	32	946	946
	18	226	226		
	19	219	219		
	20	226	227		
	21	205	205		
	22	32	33		
	Total	941	942	946	946
5	23	194	194	978	976
	24	201	202		
	25	227	227		
	26	219	219		
	27	129	129		
	Total	970	971	978	976
6	28	97	968	968	968
	29	223			
	30	227			
	31	198			
	32	223			

	Total	968	968	968	968
7	33	209	946	946	946
	34	227			
	35	221			
	36	228			
	37	61			
	Total	946	946	946	946
8	38	161	989	988	989
	39	234			
	40	213			
	41	233			
	42	146			
	Total	987	989	988	989
9	43	69	1026	1026	1026
	44	241			
	45	232			
	46	243			
	47	234			
	Total	1019	1026	1026	1026
10	48	252	1048	1048	1048
	49	219			
	50	214			
	51	253			
	52	105			
	Total	1043	1048	1048	1048
Total number of trains		9639	9656	9666	9667

Table 5.14: Summary of number of trains for problems **P2**, **P2.1**, **P2.2**, and **P2.3** using model **M2**

The total profits and total costs of grade deviations are outlined in Table 5.15. We have noted, however, that the total profits and the costs across the 4 different problems are not comparable since altering the data set means that some of the costs for problems **P2.1**, **P2.2**, and **P2.3** are incurred monthly instead of weekly. The alteration will reduce the total costs incurred and increase the total profits, and hence not comparable with the original problem.

Model	Total profit ($)	Grade deviation cost ($)
P2	11,936,461.06	1,761,146.72
P2.1	11,822,554.74	2,659,115.67
P2.2	12,699,320.97	2,363,204.05
P2.3	13,559,373.30	2,163,877.71

Table 5.15: Total profits, and grade deviation costs for problems **P2**, **P2.1**, **P2.2**, and **P2.3** using model **M2**

5.4.3 Discussion

As the solving procedure of model **M2** takes too long to conclude, we have aggregated the planning periods in the data set of problem **P2**. The alteration has reduced the problem size down to 29% of the original size. This causes the solving time to reduce from 12 hours down to 26 minutes.

The results show the impact of aggregating the problem periods on the total number of trains over the time horizon. Some interesting results have been observed. We initially predicted the total number of trains to decrease or remain the same in relation to the original problem. The results, however, have shown the opposite. As the problem size decreases, the total number of trains increases.

To explain this phenomena, we need to observe the models in a broader perspective. As the periods get aggregated, the problem becomes less restricted. Subsequently, this lesser amount of restriction encourage a higher number of trains.

Although a higher number of trains is preferred, using too many trains may be wasteful. To analyse how well the trains are utilised, we observe the number of trains in each period, as well as the total for each month. We can see that, in spite of the big difference in the total, the numbers of trains used per period do not differ significantly, especially in the earlier periods.

We have also shown the results for the total profits. We observed that the different amounts of restrictions also impact the total profits. As we aggregate the problem periods, some of the costs are incurred monthly instead of weekly. This greatly declines the total costs and consequently increase the total profits. We conclude then that the total profits for the three aggregated problems and the original problem are not comparable. We also observe that the quality of solutions for the aggregated models is still reasonably good, albeit not being comparable.

A disadvantage of the aggregated versions is that they do not produce weekly logistics schedules, as preferred by the company. Nonetheless, it does well in maintaining good solutions that model **M2** achieves.

5.5 Conclusions

In this chapter, we have considered case studies based on two different data sets provided by RTIO. Both case studies describe real life mining logistics problems; in which the first one (problem **P1**) looks at a monthly plan for one year time horizon, and the second (problem **P2**) looks at an, approximately, weekly plan for a 10 month time horizon. Although the monthly plan has managed to reflect the reliability and flexibility of the optimisation tools provided, a weekly plan is still preferred in real life for practical reasons.

Based on the results from the two case studies, we analyse that the iterative method (model **M1**) does not necessarily solve the minimum grade deviation problem. Therefore, the objective values, as well as the grade deviation costs generated, do not differ much from those generated by the model without the grade constraints (model **M0**).

Moreover, we observed some inconsistencies in results of model **M1**. In our implementations, 4 iterations perform the best for both problems in case of the time efficiency. However, our result analysis shows that 3 and 6 iterations produce the most total throughput for problem **P1** and **P2** respectively; and 7 and 8 iterations perform the best in producing the highest total profit and the lowest grade deviation cost for problem **P1** and **P2** respectively. These inconsistencies in results indicate that there is not a way to determine which number of iterations should be used.

The model with the convex relaxation approach (model **M2**) works consistently better in producing results with good quality. The total throughput produced is similar to other models, while the total profit and grade deviation cost show significant improvements.

Despite being more effective in optimising the logistics plan, this model is less efficient in terms of the time taken to a large extent. We altered the data set of problem **P2** as an attempt to reduce the problem size greatly, and hence reducing the solving times. The model reduction is undertaken by aggregating the planning periods. This approach has effectively solved the problems in reasonably good solving times while maintaining the solution quality of the total throughput.

Chapter 6

Conclusions and Recommendations

In this thesis, we have addressed the optimisation problem of medium to long term logistics planning in iron ore mining. This chapter provides the summary of our findings and some recommendations for future research.

6.1 Conclusions

The overview of this thesis has been as follows:

- In Chapter 1, we described the problem and the key issues that we addressed in this research. We provided some preliminary background and context necessary to understand the problem.

- In Chapter 2, we collated a literature review that includes substantive findings and theories in the area of surface mining. We briefly started with open-pit mining problems in general, then focused on logistics and transportation scheduling problems as well as the use of mixed-integer linear and non-linear programming in mining applications.

- In Chapter 3, we presented the first model which is an extended work of the current model done in Garcıa-Flores et al. (2011) and Singh et al. (2014). This model produces an optimisation plan of logistics and train scheduling in the iron ore mining operation of the Pilbara region. We contributed by putting the formulation together, making some changes based on the current operation, implementing the model in the optimisation solver software, and analysing the results, thus identifying the issues.

- In Chapter 4, we proposed a new model that represents the same logistics problem as in Chapter 3. We outlined the changes in the formulation, completed some implementations, and analysed the results.

- In Chapter 5, we considered two case studies based on real-life problems provided by our industry partner, RTIO. The two models we presented in the prior chapters were applied and the results from both models were analysed and compared.

This section presents our concluding remarks of the whole thesis. It comprises of two parts; namely our contributions to mining industry and the comparative analysis of the models presented in this thesis.

6.1.1 Our contribution

In this thesis, we have observed the use of mixed-integer programming in one of the applications of surface mining, namely logistics planning. As the material is being transported, a blending process takes place in order to comply with grade quality requirements. The logistics operation, therefore, is not only concerned with the quantity, but also the quality of the materials being produced. The problem formulation seeks to maximise total profit and involves both integer variables and non-linear constraints; making the problem mixed-integer non-linear programming.

The models we presented and utilised in this thesis incorporate both transportation scheduling and blending problems into one logistics planning problem. While the model in Chapter 3 (model **M1**) was developed by CSIRO, our main contribution includes the model in Chapter 4 (model **M2**); as well as implementation and result analysis of case studies in Chapter 5.

Model **M1**, which was developed by CSIRO and presented in Chapter 3, was proven to perform more reliably than the traditional Excel-based approach. An iterative method is undertaken to avoid the non-linearity in the formulation. The procedure of this method involves ignoring the non-linear constraints in the first iteration and approximating the non-linear terms in the next iterations using the solutions gained in the preceding iteration. While the implementation of test cases validates the model, the approximation technique is derived to avoid the non-linearity in the formulation without any underlying theory behind it. Moreover, the solutions obtained have become more inconsistent as the components of the mining operation expand.

As a response to the need of an improved model, we have developed a new model (model **M2**) in Chapter 4. Unlike model **M1**, this model deals with the non-linearity by using a convex relaxation approach to linearise the non-linear constraints in the formulation. Global optimal solutions can be obtained by solving the convex MILP reformulation of the problem.

Our contribution also includes the implementation of both models to solve case studies and the comparative analysis of the results in Chapter 5. In addition, we have performed a model reduction by aggregating the planning periods in order to reduce the problem size, thus cutting the solving time. The summary of our comparative analysis will be outlined in the subsequent section.

6.1.2 Comparative analysis

We have presented our result analysis of the case studies in terms of the solving time, the total throughput, the total profit, and the total cost of grade deviation. In all cases, we analyse the results obtained by models **M1** and **M2** by comparing them to the results of model **M0**, which is defined as the model without the complication of grade requirements. This analysis can be summarised as follows:

- In terms of the solving times, model **M1** is clearly more practical in real life than model **M2** as it can generate solutions in much faster solving time. This can be seen in Table 6.1 below, where all solving times displayed are in seconds. Note that the overall times for model **M1** are calculated by taking the average of the times taken by the model with 2 to 8 iterations.

Model	Problem **P1**	Problem **P2**
M0	32.43	31.92
M1	36.64	62.46
M2	6,074.94	43,976.99

Table 6.1: The summary of solving times of models **M0**, **M1**, and **M2**

- Although model **M2** generates the greatest amount of throughput overall, the difference is insignificant. We conclude, therefore, that both models have no effect on the quantity of the material shipped.

- The summary of the models' impacts on the total profits and grade deviation costs is described in Tables 6.2 and 6.3 respectively. Positive values represent an increase in percentage from the results obtained by model **M0**, while negative values represent a decrease.

Model	Problem **P1**	Problem **P2**
M1	-0.02%	+0.12%
M2	+0.01%	+4.27%

Table 6.2: The summary of total profit improvement for models **M1** and **M2**

Model	Problem **P1**	Problem **P2**
M1	-2.69%	-3.71%
M2	-24.28%	-29.81%

Table 6.3: The summary of grade deviation cost improvement for models **M1** and **M2**

The more desirable total profits and grade deviations costs in all our implementations are achieved by model **M2**; implying that model **M2** performs significantly better than model **M1** in terms of the overall results.

- An aggregated approach has been utilised to the planning periods in order to reduce the problem size. By applying this approach, we have obtained good quality solutions in more suitable time frames. The summary of the solving times for the modified problems is as follows.

Problem type	Solving time (s)
P2	43,976.99
P2.1	10,382.63
P2.2	4,097.72
P2.3	1,536.86

Table 6.4: Summary of solving times for the problems with aggregated periods

Based on the results analysis above, we summarise our findings about the two models as follow:

Model M1

Model **M1** is relatively fast when it comes to implementation. Small gaps of MIP tolerance were easily achieved in all our case studies. However, some issues have been identified. An inconsistency in solutions was clearly seen as we progressed with different numbers of iterations. As a result, it is difficult to choose the number of iterations to be applied as there is no way of determining which number of iterations gives the best solutions.

The main issue of model **M1**, nonetheless, lies in the quality of solutions. This model does not significantly affect the total profits and the grade deviation costs generated. As a result of altering the grade constraints to avoid the non-linearity, this model fails to restrict the grade requirements and increase the objective value. We therefore question the reliability of applying the iterative method to handle the grade constraints.

Model M2

Model **M2**, which we have developed, has been more consistent in producing a better quality of numerical results. All of the solutions we obtained showed a substantially higher profit and lower grade deviation cost. We therefore recommend this model to be applied if better solutions are sought.

An evident disadvantage of the model is the great increase of the problem size due to additional variables and constraints. This causes a significant jump in the time required to implement the model, thus making the model less practical when solving real life industry problems.

Aggregation technique

To address the slow implementation issue, we have performed a model reduction by aggregating the planning periods in the data set to reduce the problem size. The same model was then applied to the modified problem and a much shorter solving time was obtained (see Table 6.4).

Implementing the aggregation technique will reduce the number or periods and will not produce a weekly schedule, as preferred by RTIO. The company, nevertheless, has the flexibility to decide on the number of periods to be aggregated in applying this technique. We have provided three different applications of the technique with different numbers of aggregated periods. Furthermore, we suggest aggregating later periods in order to provide a weekly schedule for immediate periods.

6.2 Recommendations

This research has met its objectives which are to identify the issues in the current logistics model at the Pilbara mining operation and develop a new model to solve the problem. We realise, however, that there are some issues arising from this work which can potentially be of interest to extend the research. Our recommendations for further research are listed as follows:

- Although the optimisation tool that we developed generates a better set of solutions in terms of the objective function, we still question the practicality of the tool due to the big jump in the time required to run the solving procedure. We have seen that the increase in solution time is simply due to a large number of additional variables and constraints. We believe that there are ways to speed up the solution process while maintaining the quality of the solutions, such as the aggregation technique that we implemented in one of our case studies. Other approaches may be considered for future research. One may look at possibilities of applying a decomposition method to divide the problem into subproblems or adding cutting planes to reduce the solution time. Due to the large size of the problem, utilising these approaches may require a great amount of time and effort, but is still possible.

- We applied our optimisation tool to real iron ore mining operations in the Pilbara region. Further research may consider applying this model to other case studies from other different mining operations to show the flexibility of the model.

Bibliography

Adams, W. P. and H. D. Sherali (1993). Mixed-integer bilinear programming problems. *Mathematical Programming 59*(1), 279–305.

Audet, C., J. Brimberg, P. Hansen, S. L. Digabel, and N. Mladenović (2004). Pooling problem: Alternate formulations and solution methods. *Management Science 50*(6), 761–776.

Audet, C., P. Hansen, B. Jaumard, and G. Savard (2000). A branch and cut algorithm for nonconvex quadratically constrained quadratic programming. *Mathematical Programming 87*(1), 131–152.

Australian Bureau of Statistics (2012). Year book australia. `http://www.abs.gov.au/ausstats/abs@.nsf/mf/1301.0`.

Australian Mining (2015). Australian mine map 2015. `https://www.australianmining.com.au/mine-map/`.

Bilgen, B. and I. Ozkarahan (2007). A mixed-integer linear programming model for bulk grain blending and shipping. *International Journal of Production Economics 107*(2), 555–571.

Boland, N., I. Dumitrescu, and G. Froyland (2008). A multistage stochastic programming approach to open pit mine production scheduling with uncertain geology. *Optimization Online*.

Burer, S. and A. N. Letchford (2012). Non-convex mixed-integer nonlinear programming: a survey. *Survey in Operations Research and Management Science 17*(2), 97–106.

Busnach, E., A. Mehrez, and Z. Sinuany-Stern (1985). A production problem in phosphate mining. *Journal of the Operational Research Society 36*(4), 285–288.

Bussieck, M. R. and A. Pruessner (2003). Mixed-integer nonlinear programming. *SIAG/OPT Newsletter: Views & News 14*(1), 19–22.

Caccetta, L. (2007). Application of optimisation techniques in open pit mining. In *Handbook of operations research in natural resources*, pp. 547–559. Springer.

Caccetta, L. and L. Giannini (1986). Optimization techniques for the open pit limit problem. In *Proceedings of the Australalaian Institute of Mining and Metallurgy*, Volume 291, pp. 57–63.

Caccetta, L. and L. Giannini (1988). An application of discrete mathematics in the design of an open pit mine. *Discrete Applied Mathematics 21*(1), 1–19.

Caccetta, L. and L. Giannini (1990). Application of operations research techniques in open pit mining. *Asian-Pacific Operations Research: APORS 88*, 707–724.

Caccetta, L. and S. P. Hill (2003). An application of branch and cut to open pit mine scheduling. *Journal of Global Optimization 27*(2-3), 349–365.

Caccetta, L., P. Kelsey, and L. Giannini (1998). Open pit mine production scheduling. In *APCOM 98: Computer Applications in the Mineral Industries International Symposium*, pp. 65–72.

Caris, A., C. Macharis, and G. K. Janssens (2008). Planning problems in intermodal freight transport: accomplishments and prospects. *Transportation Planning and Technology 31*(3), 277–302.

Christiansen, M., K. Fagerholt, and D. Ronen (2004). Ship routing and scheduling: Status and perspectives. *Transportation Science 38*(1), 1–18.

Christopher, M. (2011). *Logistics & supply chain management* (fourth ed.). Financial Times Prentice Hall, Harlow.

Cordeau, J.-F., P. Toth, and D. Vigo (1998). A survey of optimization models for train routing and scheduling. *Transportation Science 32*(4), 380–404.

Cullenbine, C., R. K. Wood, and A. Newman (2011). A sliding time window heuristic for open pit mine block sequencing. *Optimization Letters 5*(3), 365–377.

Duran, M. A. and I. E. Grossmann (1986). An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming 36*(3), 307–339.

Eidenbenz, L. A. S., M. Gantenbein, C. Stamm, D. S. Taylor, B. Weber, and P. Widmayer (2003). Train routing algorithms: Concepts, design choices, and practical considerations. In *Proceedings of the Fifth Workshop on Algorithm Engineering and Experiments*, Volume 111, pp. 106. SIAM.

Everett, J. (2001). Iron ore production scheduling to improve product quality. *European Journal of Operational Research 129*(2), 355–361.

Everett, J. (2007). Computer aids for production systems management in iron ore mining. *International Journal of Production Economics 110*(1), 213–223.

Fishman, F. and P. McInnes (2005). Sampling and blending to optimise revenue – a minerals industry. In *Proceedings of the Second World Conference on Sampling and Blending 2005, Number 4 in Australasian Institute of Mining and Metallurgy Publication Series*, pp. 95–99. Australasian Institute of Mining and Metallurgy.

Fletcher, R. and S. Leyffer (1994). Solving mixed integer nonlinear programs by outer approximation. *Mathematical programming 66*(1), 327–349.

Foulds, L., D. Haugland, and K. Jörnsten (1992). A bilinear approach to the pooling problem. *Optimization 24*(1-2), 165–180.

Fröhling, M., F. Schwaderer, H. Bartusch, and O. Rentz (2010). Integrated planning of transportation and recycling for multiple plants based on process simulation. *European Journal of Operational Research 207*(2), 958–970.

Fügenschuh, A., H. Homfeld, H. Schülldorf, and S. Vigerske (2010). Mixed-integer nonlinear problems in transportation applications. In *Proceedings of the 2nd International Conference on Engineering Optimization (CD-ROM)*, pp. 14. Instituto Superior Técnico Lisboa.

Fytas, K., J. Hadjigeorgiou, and J. Collins (1993). Production scheduling optimization in open pit mines. *International Journal of Surface Mining and Reclamation 7*(1), 1–9.

Garcıa-Flores, R., G. Singh, A. Ernst, and P. Welgama (2011, December). Medium-term rail planning at rio tinto iron ore. In F. Chan and D. Marinova (Eds.), *19th International Congress on Modelling and Simulation*, pp. 311–317. Modelling and Simulation Society of Australia and New Zealand.

Geoscience Australia (2015). Iron fact sheet. http://www.australianminesatlas.gov.au/education/fact_sheets/iron.html.

Gershon, M. (1987). Heuristic approaches for mine planning and production scheduling. *International Journal of Mining and Geological Engineering 5*(1), 1–13.

Gershon, M. E. (1983a). Mine scheduling optimization with mixed integer programming. *Mining Engineering 35*(4), 351–354.

Gershon, M. E. (1983b). Optimal mine production scheduling: evaluation of large scale mathematical programming approaches. *International Journal of Mining Engineering 1*(4), 315–329.

Greenberg, H. J. (1995). Analyzing the pooling problem. *ORSA Journal on Computing 7*(2), 205–217.

Grossmann, I. E. (2002). Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and Engineering 3*(3), 227–252.

Gupte, A., S. Ahmed, S. S. Dey, and M. S. Cheon (2013). Pooling problems: relaxations and discretizations. *School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA. and ExxonMobil Research and Engineering Company, Annandale, NJ.*

Haverly, C. (1978). Studies of the behavior of recursion for the pooling problem. *ACM SIGMAP Bulletin* (25), 19–28.

Haverly, C. (1979). Behavior of recursion model-more studies. *ACM SIGMAP Bulletin* (26), 22–28.

Haverly, C. (1980). Recursion model behavior: more studies. *ACM SIGMAP Bulletin* (28), 39–41.

Higgins, A., E. Kozan, and L. Ferreira (1996). Optimal scheduling of trains on a single line track. *Transportation Research Part B: Methodological 30*(2), 147–161.

Horst, R. and H. Tuy (1993). *Global optimization: Deterministic approaches* (second ed.). Springer-Verlag, Berlin.

Ivanchev, M., I. Mitev, and N. Yanev (1976). Realization of the branch and bound method for solving the general mixed integer programming problem. *USSR Computational Mathematics and Mathematical Physics 16*(3), 241–246.

Johnson, T. B. (1968). Optimum open pit mine production scheduling. Technical report, DTIC Document.

Karuppiah, R., K. C. Furman, and I. E. Grossmann (2008). Global optimization for scheduling refinery crude oil operations. *Computers & Chemical Engineering 32*(11), 2745–2766.

Kennedy, B. A. (1990). *Surface mining* (second ed.). Society for Mining, Metallurgy, and Exploration.

Kilic, O. A., R. Akkerman, D. P. van Donk, and M. Grunow (2013). Intermediate product selection and blending in the food processing industry. *International Journal of Production Research 51*(1), 26–42.

Kozan, E. and S. Q. Liu (2011). Operations research for mining: a classification and literature review. *ASOR Bulletin 30*(1), 2–23.

Kumral, M. (2011). Incorporating geo-metallurgical information into mine production scheduling. *Journal of the Operational Research Society 62*(1), 60–68.

Kuo, A., E. Miller-Hooks, and H. S. Mahmassani (2010). Freight train scheduling with elastic demand. *Transportation Research Part E: Logistics and Transportation Review 46*(6), 1057–1070.

Land, A. H. and A. G. Doig (2010). An automatic method for solving discrete programming problems. In *50 Years of Integer Programming 1958-2008*, pp. 105–132. Springer.

Lee, H., J. M. Pinto, I. E. Grossmann, and S. Park (1996). Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management. *Industrial & Engineering Chemistry Research 35*(5), 1630–1641.

Lerchs, H. and I. F. Grossmann (1965). Optimum design of open pit mines. *Canadian Institute of Mining Mettalurgical Bulletin 58*, 47–54.

Leyffer, S., J. Linderoth, J. Luedtke, A. Miller, and T. Munson (2009). Applications and algorithms for mixed integer nonlinear programming. In *Journal of Physics: Conference Series*, Volume 180, pp. 012014. IOP Publishing.

Li, L. and S. Tayur (2005). Medium-term pricing and operations planning in intermodal transportation. *Transportation Science 39*(1), 73–86.

Liu, C.-M. and H. D. Sherali (2000). A coal shipping and blending problem for an electric utility company. *Omega 28*(4), 443–444.

Luenberger, D. G. and Y. Ye (2008). *Linear and nonlinear programming* (third ed.), Volume 116 of *International Series in Operations Research & Management Science*. Springer, New York.

Macharis, C. and Y. M. Bontekoning (2004). Opportunities for or in intermodal freight transport research: A review. *European Journal of Operational Research 153*(2), 400–416.

McCormick, G. P. (1976). Computability of global solutions to factorable nonconvex programs. I. Convex underestimating problems. *Mathematical Programming 10*(2), 147–175.

McKay, B., I. Lambert, and S. Miyazaki (2000). *The Australian mining industry: from settlement to 2000.* Australian Bureau of Statistics, Canberra.

Mehrez, A., M. S. Hung, and B. H. Ahn (1995). An industrial ocean-cargo shipping problem. *Decision Sciences 26*(3), 395.

Mendez, C. A., I. E. Grossmann, I. Harjunkoski, and P. Kaboré (2006). A simultaneous optimization approach for off-line blending and scheduling of oil-refinery operations. *Computers & ChemicalEengineering 30*(4), 614–634.

Mining-Technology (2016a). Australia's carbon tax sustainable sense or economic deadweight? `http://www.mining-technology.com/features/featureaustralias-carbon-tax-sustainable-sense-or-economic-deadweight/featureaustralias-carbon-tax-sustainable-sense-or-economic-deadweight-4.html`.

Mining-Technology (2016b). Paraburdoo iron ore mine, pilbara, australia. `http://www.mining-technology.com/projects/paraburdoo-iron-ore-mine-pilbara/paraburdoo-iron-ore-mine-pilbara1.html`.

Misener, R. and C. A. Floudas (2009). Advances for the pooling problem: Modeling, global optimization, and computational studies. *Applied and Computational Mathematics 8*(1), 3–22.

Morlok, E. K., E. Thomas, et al. (1970). *Final report on the development of a geographic transportation network generation and evaluation model.* Transportation center Northwestern university.

Newman, A. M., E. Rubio, R. Caro, A. Weintraub, and K. Eurek (2010). A review of operations research in mine planning. *Interfaces 40*(3), 222–245.

Newman, A. M. and C. A. Yano (2000). Centralized and decentralized train scheduling for intermodal operations. *IIE Transactions 32*(8), 743–754.

Osanloo, M., J. Gholamnejad, and B. Karimi (2008). Long-term open pit mine production planning: a review of models and algorithms. *International Journal of Mining, Reclamation and Environment 22*(1), 3–35.

Ramazan, S. (2007). The new fundamental tree algorithm for production scheduling of open pit mines. *European Journal of Operational Research 177*(2), 1153–1166.

Ramazan, S. and R. Dimitrakopoulos (2004). Recent applications of operations research and efficient mip formulations in open pit mining. *SME Transactions 316*.

Ramazan, S. and R. Dimitrakopoulos (2007). Stochastic optimisation of long-term production scheduling for open pit mines with a new integer programming formulation. *Orebody Modelling and Strategic Mine Planning, Spectrum Series 14*, 359–365.

Rehman, S. U. and M. W. A. Asad (2010). A mixed-integer linear programming (MILP) model for short-range production scheduling of cement quarry operations. *Asia-Pacific Journal of Operational Research 27*, 315.

Rio Tinto (2009). Iron ore production in the pilbara. `http://www.riotinto.com/documents/_Iron Ore/Iron_ore_production_in_the_Pilbara.pdf`.

Rio Tinto (2013). Site visit to iron ore operations in western australia. `http://www.riotinto.com/documents/Investors/130902_Presentation_Site_visit_to_iron_ore_operations_in_Western_Australia.pdf`.

Rio Tinto (2014). Mine of the future. `http://www.riotinto.com/australia/pilbara/mine-of-the-future-9603.aspx`.

Rio Tinto (2016). New trains, new locomotives and wagons operating in the pilbara region of western australia. `http://www.riotinto.com/media/photo-library-263.aspx?y=1713#`.

Ronen, D. (1983). Cargo ships routing and scheduling: Survey of models and problems. *European Journal of Operational Research 12*(2), 119–126.

Sandeman, T., C. Fricke, P. Bodon, and C. Stanford (2010). Integrating optimization and simulation-a comparison of two case studies in mine planning. In *Simulation Conference (WSC), Proceedings of the 2010 Winter*, pp. 1898–1910. IEEE.

Shih, L.-H. (1997). Planning of fuel coal imports using a mixed integer programming method. *International Journal of Production Economics 51*(3), 243–249.

Singh, G., R. Garcia-Flores, A. Ernst, P. Welgama, M. Zhang, and K. Munday (2014). Medium-term rail scheduling for an iron ore mining company. *Interfaces 44*(2), 222–240.

Taha, H. A. (1975). *Integer programming: Theory, applications and computations, Operations Research and Industrial Engineering.* Academic Press [A subsidiary of Harcourt Brace Jovanovich, Publishers], New York-London.

Tawarmalani, M. and N. V. Sahinidis (2002). *Convexification and global optimization in continuous and mixed-integer nonlinear programming: Theory, algorithms, software, and applications*, Volume 65 of *Nonconvex Optimization and its Applications.* Kluwer Academic Publishers, Dordrecht.

Topuz, E. and C. Duan (1989). A survey of operations research applications in the mining industry. *CIM Bulletin 82*(925), 48–50.

Ulstein, N. L., B. Nygreen, and J. R. Sagli (2007). Tactical planning of offshore petroleum production. *European Journal of Operational Research 176*(1), 550–564.

Walser, G. (2000, October). Economic impact of world mining. In *International Symposium on the Uranium Production Cycle and the Environment*, pp. 263–264. International Atomic Energy Agency.

Weiss, A. (1979). *Computer Methods for the 80's in the Mineral Industry.* Society of Mining Engineers of the American Institute of Mining, Metallurgical, and Petroleum Engineers.

Wenkai, L., C.-W. Hui, B. Hua, and Z. Tong (2002). Scheduling crude oil unloading, storage, and processing. *Industrial & Engineering Chemistry Research 41*(26), 6723–6734.

Williams, N. (1967). *Linear and non-linear programming in industry.* Topics in Operational Research. Pitman.

Appendix

The following is the AIMMS text representation for our models described in Chapter 3 and 4. This text representation excludes the codes for excel linkage to retreive the input data from the data file.

```
Model Main_AIMMS {
    DeclarationSection sets {
        Set Periods {
            SubsetOf: Integers;
            Index: t; }
        Set Mines {
            Index: m; }
        Set MinedProducts {
            Index: p; }
        Set MineToProducts {
            IndexDomain: m;
            SubsetOf: MinedProducts; }
        Set Ports {
            Index: r; }
        Set ShippedProducts {
            Index: s, ss; }
        Set LumpProducts {
            SubsetOf: ShippedProducts; }
        Set LumpToFinePairs {
            IndexDomain: s;
            SubsetOf: ShippedProducts; }
        Set PortToProducts {
            IndexDomain: r;
            SubsetOf: ShippedProducts; }
        Set MinePortToProducts {
            IndexDomain: (m,p)|p in MineToProducts(m);
            SubsetOf: ShippedProducts; }
        Set PortToMine {
```

```
        IndexDomain: r;
        SubsetOf: Mines; }
    Set ShippedProductToProduct {
        IndexDomain: (m,s);
        SubsetOf: MinedProducts; }
    Set Components {
        Index: c; }
    Set Fleets {
        Index: f; }
    Set FleetToMines {
        IndexDomain: f;
        SubsetOf: Mines; }
    Set MinesToFleet {
        IndexDomain: m;
        SubsetOf: Fleets; }
    Set RobeValleyMines {
        SubsetOf: Mines; }
    Set BrockmanLoopMines {
        SubsetOf: Mines; }
    Set Regions {
        Index: g; }
    Set RegionToMines {
        IndexDomain: g;
        SubsetOf: Mines; }
    Set JVMines {
        SubsetOf: Mines; }
    Set CarDumpers {
        Index: d; }
    Set PortToDumpers {
        IndexDomain: r;
        SubsetOf: CarDumpers; }
    Set MineToDumpers {
        IndexDomain: (m,p)|p in MineToProducts(m);
        SubsetOf: CarDumpers; }
    Set DumpersInWC {
        SubsetOf: CarDumpers; }
}
DeclarationSection Penalties {
    Parameter SP {
        IndexDomain: s; }
    Parameter MSVLMax {
```

```
        IndexDomain: (m,p)|p in MineToProducts(m); }
    Parameter MSVLMin {
        IndexDomain: (m,p)|p in MineToProducts(m); }
    Parameter PSVLMax {
        IndexDomain: (r,s)|s in PortToProducts(r); }
    Parameter PSVLMin {
        IndexDomain: (r,s)|s in PortToProducts(r); }
    Parameter MSVBMax {
        IndexDomain: (m,p)|p in MineToProducts(m); }
    Parameter PSVBMax {
        IndexDomain: (r,s)|s in PortToProducts(r); }
    Parameter BPMToBulk {
        IndexDomain: (m,p)|p in MineToProducts(m); }
    Parameter BPMFromBulk {
        IndexDomain: (m,p)|p in MineToProducts(m); }
    Parameter BPRToBulk {
        IndexDomain: (r,s)|s in PortToProducts(r); }
    Parameter BPRFromBulk {
        IndexDomain: (r,s)|s in PortToProducts(r); }
    Parameter CTP;
    Parameter PP {
        IndexDomain: (m,d); }
    Parameter GPI {
        IndexDomain: (s,c); }
}
DeclarationSection Objective_Function {
    Variable Revenue {
        Range: free;
        Definition: {
            sum((r,s,t)|s in PortToProducts(r),SP(s)*zPlus
            RF(r,s,t)*(1-PercentLump(r,s,t))/(1+Discount)^
            (t-FirstPeriod)); } }
    Parameter Discount { }
    Variable TotalLiveStockPenalty {
        Range: nonnegative;
        Definition: {
            sum((m,p,t)|p in MineToProducts(m),MSVLMax(m,p
            )*AlphaMax(m,p,t)+MSVLMin(m,p)*AlphaMin(m,p,t)
            )+sum((r,s,t)|s in PortToProducts(r),PSVLMax(r
            ,s)*BetaMax(r,s,t)+PSVLMin(r,s)*BetaMin(r,s,t)
            ); } }
```

```
Variable TotalBulkStockPenalty {
    Range: nonnegative;
    Definition: {
        sum((m,p,t)|p in MineToProducts(m),MSVBMax(m,p
        )*AlphaBulk(m,p,t))+sum((r,s,t)|s in PortToPro
        ducts(r),PSVBMax(r,s)*BetaBulk(r,s,t)); } }
Variable TotalBulkHandlingCost {
    Range: free;
    Definition: {
        sum((m,p,t)|p in MineToProducts(m),BPMtoBulk(m
        ,p)*yToBulk(m,p,t)+BPMfromBulk(m,p)*yFromBulk(
        m,p,t))+sum((r,s,t)|s in PortToProducts(r),BPR
        toBulk(r,s)*uToBulk(r,s,t)+BPRfromBulk(r,s)*uF
        romBulk(r,s,t)); } }
Variable CycleTimePenalty {
    Range: free;
    Definition: {
        sum((f,t),CTP*Mu(f,t)); } }
Variable DumperPreference {
    Range: free;
    Definition: {
        sum((m,f,p,d,s,t),TS(m,f,p,t)*x(m,f,p,d,s,t)*P
        P(m,d)); } }
Variable GradeDeviationsPenalty {
    Range: free;
    Definition: {
        sum((r,s,c,t)|s in PortToProducts(r),GPI(s,c)*
        (EI(r,s,c,t)+SI(r,s,c,t))); } }
Variable Incentives {
    Range: free;
    Definition: {
        2*sum((m,f,p,d,s,t),SP(s)*TS(m,f,p,t)*x(m,f,p,
        d,s,t)); } }
Variable Profit {
    Range: free;
    Definition: {
        Revenue-TotalLiveStockPenalty-TotalBulkStockPe
        nalty-TotalBulkHandlingCost-CycleTimePenalty-D
        umperPreference-GradeDeviationsPenalty+Incenti
        ves; } }
Set WithoutGradeVariables {
```

```
        SubsetOf: AllVariables;
        Definition: {
            data { Revenue, TotalLiveStockPenalty, TotalBu
            lkStockPenalty, TotalBulkHandlingCost, CycleTi
            mePenalty, DumperPreference, Incentives, Profi
            t, sLevel, bLevel, wLevel, wBeforeZ, wBeforeZP
            lusLump, vLevel, yToBulk, yFromBulk, uToBulk,
            uFromBulk, zPlusRF, x,AlphaMin, AlphaMax, Alph
            aBulk, BetaMin, BetaMax,BetaBulk, FleetX, Mu,
            xCumulative } } }
    Set SLPGradeVariables {
        SubsetOf: AllVariables;
        Definition: {
            data { Revenue, TotalLiveStockPenalty, TotalBu
            lkStockPenalty, TotalBulkHandlingCost, CycleTi
            mePenalty, DumperPreference, GradeDeviationsPe
            nalty, Incentives, Profit, sLevel, bLevel, wLe
            vel, wBeforeZ, wBeforeZPlusLump, vLevel, yToBu
            lk, yFromBulk, uToBulk, uFromBulk, zPlusRF, x,
            AlphaMin, AlphaMax, AlphaBulk, BetaMin, BetaMa
            x, BetaBulk, FleetX, Mu, xCumulative, LMGrade,
            BMGrade, LPGrade, BPGrade, RGrade, ZG, SI, EI
            } } }
    Set GradeVariables {
        SubsetOf: AllVariables;
        Definition: {
            data { Revenue, TotalLiveStockPenalty, TotalBu
            lkStockPenalty, TotalBulkHandlingCost, CycleTi
            mePenalty, DumperPreference, GradeDeviationsPe
            nalty, Incentives, Profit, sLevel, bLevel, wLe
            vel, wBeforeZ, wBeforeZPlusLump, vLevel, yToBu
            lk, yFromBulk, uToBulk, uFromBulk, zPlusRF, x,
            RFromIOP, RFromL, RFromB, RFromBBinary, RFromI
            OPFIFO, RFromIOPFIFOBinary, TSxMinusSPos, TSxM
            inusSPosBinary, RFromLFIFO, RFromLFIFOBinary,
            RFromIOPLIFO, RFromIOPLIFOBinary, TSxMinusIOPP
            os, TSxMinusIOPPosBinary, RFromLLIFO, RFromLLI
            FOBinary, AlphaMin, AlphaMax, AlphaBulk, BetaM
            in, BetaMax, BetaBulk, FleetX, Mu, xCumulative
            , LMGrade, BMGrade, LPGrade, BPGrade, RGrade,
            ZG, LMxS, LMxYToBulk, BMxB, BMxYFromBulk, RGxX
```

```
        , LMxRL, BMxRB, LPxW, ZGxWBeforeZ, LPxWBeforeZ
        PlusLump, ZGxUToBulk, BPxV, BPxUFromBulk, ZGxZ
        } } }
Set RealGradesVars {
    SubsetOf: AllVariables;
    Definition: data { RealLM, RealBM, RealRG, RealLP,
        RealBP, RealZG, RealSI, RealEI, RealGradePenal
        ty }; }
Set WithoutGradeConstraints {
    SubsetOf: AllConstraints;
    Definition: {
        data { Revenue, TotalLiveStockPenalty, TotalBu
        lkStockPenalty, TotalBulkHandlingCost, CycleTi
        mePenalty, DumperPreference, Incentives, Profi
        t, sLevel, bLevel, wLevel, wBeforeZ, wBeforeZP
        lusLump, vLevel, MineLiveStockpileLevelMin, Mi
        neLiveStockpileLevelMax, MineBulkStockpileLeve
        l, MineYardLimit, RegionBulkInMax, RegionBulkO
        utMax, PortLiveStockpileLevelMin, PortLiveStoc
        kpileLevelMax, PortBulkStockpileLevel, PortYar
        dLimit, PortBulkInMax, PortBulkOutMax, MaxTrai
        ns, MaxTrainsInRegion, FleetX, FleetCapacity,
        FleetHourCapacity, DumperCapacity, CLWCCapacit
        y, ShipCapacity, xCumulative, JVCummulativeMin
        , JVCummulativeMax } } }
Set SLPGradeConstraints {
    SubsetOf: AllConstraints;
    Definition: {
        data { Revenue, TotalLiveStockPenalty, TotalBu
        lkStockPenalty, TotalBulkHandlingCost, CycleTi
        mePenalty, DumperPreference, GradeDeviationsPe
        nalty, Incentives, Profit, sLevel, bLevel, wLe
        vel, wBeforeZ, wBeforeZPlusLump, vLevel, MineL
        iveStockpileLevelMin, MineLiveStockpileLevelMa
        x, MineBulkStockpileLevel, MineYardLimit, Regi
        onBulkInMax, RegionBulkOutMax, PortLiveStockpi
        leLevelMin, PortLiveStockpileLevelMax, PortBul
        kStockpileLevel, PortYardLimit, PortBulkInMax,
        PortBulkOutMax, MaxTrains, MaxTrainsInRegion,
        FleetX, FleetCapacity, FleetHourCapacity, Dump
        erCapacity, CLWCCapacity, ShipCapacity, xCumul
```

```
        ative, JVCummulativeMin, JVCummulativeMax, SLP
        MineLiveGrades, SLPMineBulkGrades, SLPRailedGr
        ades, SLPPortLiveGrades, SLPPortLiveGradesForF
        ines, SLPPortBulkGrades, SLPGradeDeviationsMin
        , SLPGradeDeviationsMax } } }
Set GradeConstraints {
    SubsetOf: AllConstraints;
    Definition: {
        data { Revenue, TotalLiveStockPenalty, TotalBu
        lkStockPenalty, TotalBulkHandlingCost, CycleTi
        mePenalty, DumperPreference, GradeDeviationsPe
        nalty, Incentives, Profit, sLevel, bLevel, wLe
        vel, wBeforeZ, wBeforeZPlusLump, vLevel, Raile
        dQuantity, RFromIOP, RFromL, RFromBInq1, RFrom
        BInq2, RFromBInq3, RFromIOPFIFOInq1, RFromIOPF
        IFOInq2, RFromIOPFIFOInq3, TSxMinusSPosInq1, T
        SxMinusSPosInq2, TSxMinusSPosInq3, RFromLFIFOI
        nq1, RFromLFIFOInq2, RFromLFIFOInq3, RFromLFIF
        OInq4, RFromIOPLIFOInq1, RFromIOPLIFOInq2, RFr
        omIOPLIFOInq3, TSxMinusIOPPosInq1, TSxMinusIOP
        PosInq2, TSxMinusIOPPosInq3, RFromLLIFOInq1, R
        FromLLIFOInq2, RFromLLIFOInq3, RFromLLIFOInq,
        MineLiveStockpileLevelMin, MineLiveStockpileLe
        velMax, MineBulkStockpileLevel, MineYardLimit,
        RegionBulkInMax, RegionBulkOutMax, PortLiveSto
        ckpileLevelMin, PortLiveStockpileLevelMax, Por
        tBulkStockpileLevel, PortYardLimit, PortBulkIn
        Max, PortBulkOutMax, MaxTrains, MaxTrainsInReg
        ion, FleetX, FleetCapacity, FleetHourCapacity,
        DumperCapacity, CLWCCapacity, ShipCapacity, xC
        umulative, JVCummulativeMin, JVCummulativeMax,
        BMGradeAndQuantity, LPGrade, BPGradeAndQuantit
        y, LMxS, LMxSConvex1, LMxSConvex2, LMxSConcave
        1, LMxSConcave2, LMxYToBulkConvex1, LMxYToBulk
        Convex2, LMxYToBulkConcave1, LMxYToBulkConav2,
        BMxB, BMxBConvex1, BMxBConvex2, BMxBConcave1,
        BMxBConcave2, BMxYFromBulkConvex1, BMxYFromBul
        kConvex2, BMxYFromBulkConcave1, BMxYFromBulkCo
        ncave2, RailedGrades, RGxXConvex1, RGxXConvex2
        , RGxXConcave1, RGxXConcave2, LMxRL, LMxRLConv
        ex1, LMxRLConvex2, LMxRLConcave1, LMxRLConcave
```

```
        2, BMxRB, BMxRBConvex1, BMxRBConvex2, BMxRBCon
        cave1, BMxRBConcave2, LPxW, LPxWConvex1, LPxWC
        onvex2, LPxWConcave1, LPxWConcave2, ZGxWBefore
        Z, ZGxWBeforeZConvex1, ZGxWBeforeZConvex2, ZGx
        WBeforeZConcave1, ZGxWBeforeZConcave2, LPxWBef
        oreZPlusLump, LPxWBeforeZPlusLumpConvex1, LPxW
        BeforeZPlusLumpConvex2, LPxWBeforeZPlusLumpCon
        cave1, LPxWBeforeZPlusLumpConcave2, ZGxUToBulk
        Convex1, ZGxUToBulkConvex2, ZGxUToBulkConcave1
        , ZGxUToBulkConcave2, BPxV, BPxVConvex1, BPxVC
        onvex2, BPxVConcave1, BPxVConcave2, BPxUFromBu
        lkConvex1, BPxUFromBulkConvex2, BPxUFromBulkCo
        ncave1, BPxUFromBulkConcave2, ZGxZConvex1, ZGx
        ZConvex2, ZGxZConcave1, ZGxZConcave2, LPxZLump
        Convex1, LPxZLumpConvex2, LPxZLumpConcave1, LP
        xZLumpConcave2, GradeDeviationsMin, GradeDevia
        tionsMax } } }
Set RealGradesCons {
    SubsetOf: AllConstraints;
    Definition: {
        data { LMConstraint, BMConstraint, RGConstrain
        t, RealLP, LPConstraint, BPConstraint, ZGConst
        raint, RealGradeDevMin, RealGradeDevMax, RealG
        radePenalty } } }
Parameter relative_optimality_gap {  }
MathematicalProgram Model_WithoutGrades {
    Objective: Profit;
    Direction: maximize;
    Constraints: WithoutGradeConstraints;
    Variables: WithoutGradeVariables;
    Type: MIP; }
MathematicalProgram Model1_WithoutGradesNoInteger {
    Objective: Profit;
    Direction: maximize;
    Constraints: WithoutGradeConstraints;
    Variables: WithoutGradeVariables;
    Type: RMIP; }
MathematicalProgram Model1_WithGradesNoInteger {
    Objective: Profit;
    Direction: maximize;
    Constraints: SLPGradeConstraints;
```

```
        Variables: SLPGradeVariables;
        Type: RMIP; }
    MathematicalProgram Model1_WithGrades {
        Objective: Profit;
        Direction: maximize;
        Constraints: SLPGradeConstraints;
        Variables: SLPGradeVariables;
        Type: MIP; }
    MathematicalProgram Model2_WithGrades1 {
        Objective: Profit;
        Direction: maximize;
        Constraints: GradeConstraints;
        Variables: GradeVariables;
        Type: MIP; }
    MathematicalProgram Model2_WithGrades2 {
        Objective: Profit;
        Direction: maximize;
        Constraints: RealGradesCons;
        Variables: RealGradesVars;
        Type: LP; }
}
DeclarationSection Mass_Variables_and_Initials {
    Variable sLevel {
        IndexDomain: (m,p,t)|p in MineToProducts(m);
        Range: nonnegative;
        Definition: {
            IF t=FirstPeriod THEN sInitial(m,p)+IOP(m,p,t)
            +yFromBulk(m,p,t)-yToBulk(m,p,t)-sum((f,d,s),T
            S(m,f,p,t)*x(m,f,p,d,s,t))
            ELSE slevel(m,p,t-1)+IOP(m,p,t)+yFromBulk(m,p,
            t)-yToBulk(m,p,t)-sum((f,d,s),TS(m,f,p,t)*x(m,
            f,p,d,s,t))
            ENDIF; }  }
    Parameter sInitial {
        IndexDomain: (m,p)|p in MineToProducts(m); }
    Variable bLevel {
        IndexDomain: (m,p,t)|p in MineToProducts(m);
        Range: nonnegative;
        Definition: {
            IF t=FirstPeriod THEN bInitial(m,p)+yToBulk(m,
            p,t)-yFromBulk(m,p,t)
```

```
        ELSE blevel(m,p,t-1)+yToBulk(m,p,t)-yFromBulk(
        m,p,t) ENDIF; } }
Parameter bInitial {
    IndexDomain: (m,p)|p in MineToProducts(m); }
Variable wLevel {
    IndexDomain: (r,s,t)|s in PortToProducts(r);
    Range: nonnegative;
    Definition: {
        wBeforeZPlusLump(r,s,t)-zPlusRF(r,s,t); } }
Parameter wInitial {
    IndexDomain: (r,s)|s in PortToProducts(r); }
Variable wBeforeZ {
    IndexDomain: (r,s,t)|s in PortToProducts(r);
    Range: nonnegative;
    Definition: {
        IF t=FirstPeriod THEN wInitial(r,s)+uFromBulk(
        r,s,t)-uToBulk(r,s,t)+sum((m,f,p in MineToProd
        ucts(m),d in PortToDumpers(r)),TS(m,f,p,t)*x(m
        ,f,p,d,s,t))
        ELSE wlevel(r,s,t-1)+uFromBulk(r,s,t)-uToBulk(
        r,s,t)+sum((m,f,p in MineToProducts(m),d in Po
        rtToDumpers(r)),TS(m,f,p,t)*x(m,f,p,d,s,t))
        ENDIF; } }
Variable wBeforeZPlusLump {
    IndexDomain: (r,s,t)|s in PortToProducts(r);
    Range: free;
    Definition: {
        wBeforeZ(r,s,t)+sum(ss in LumpToFinePairs(s),
        PercentLump(r,ss,t)*zPlusRF(r,ss,t)); } }
Variable vLevel {
    IndexDomain: (r,s,t)|s in PortToProducts(r);
    Range: nonnegative;
    Definition: {
        IF t=FirstPeriod THEN vInitial(r,s)+uToBulk(r,
        s,t)-uFromBulk(r,s,t)
        ELSE vlevel(r,s,t-1)+uToBulk(r,s,t)-uFromBulk(
        r,s,t) ENDIF; } }
Parameter vInitial {
    IndexDomain: (r,s)|s in PortToProducts(r); }
Variable yToBulk {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
```

```
        Range: [0, MineBulkOutRates(m, p, t)]; }
    Variable yFromBulk {
        IndexDomain: (m,p,t)|p in MineToProducts(m);
        Range: [0, MineBulkInRates(m, p, t)]; }
    Variable uToBulk {
        IndexDomain: (r,s,t)|s in PortToProducts(r);
        Range: nonnegative; }
    Variable uFromBulk {
        IndexDomain: (r,s,t)|s in PortToProducts(r);
        Range: nonnegative; }
    Parameter IOP {
        IndexDomain: (m,p,t)|p in MineToProducts(m); }
    Variable zPlusRF {
        IndexDomain: (r,s,t)|s in PortToProducts(r);
        Range: nonnegative; }
    Parameter PercentLump {
        IndexDomain: (r,s,t)|s in PortToProducts(r);
        Range: (0, 1); }
}
DeclarationSection Railed_Quantity {
    Parameter TrainTonnesPerCar {
        IndexDomain: (m,p,t)|p in MineToProducts(m); }
    Parameter CarsPerTrain {
        IndexDomain: (f,t); }
    Parameter TS {
        IndexDomain: (m,f,p,t)|m in FleetToMines(f) and p
            in MineToProducts(m);
        Definition: {
            TrainTonnesPerCar(m,p,t)/1000*CarsPerTrain(f,t
            ); } }
    Variable x {
        IndexDomain: (m,f,p,d,s,t)|m in FleetToMines(f) an
            d p in MineToProducts(m) and s in MinePortToPr
            oducts(m,p) and d in MineToDumpers(m,p);
        Range: {
            {xMin(m, t)..xMax(m, t)} }
        RelaxStatus: RelaxX(m, t); }
    Parameter RelaxX {
        IndexDomain: (m,t);
        Range: binary;
        Definition: {
```

```
        IF m in RobeValleyMines THEN 1 ELSE 0 ENDIF; }
}
Constraint RailedQuantity {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Definition: {
        sum((f,d,s),TS(m,f,p,t)*x(m,f,p,d,s,t))=RFromI
        OP(m,p,t)+RFromL(m,p,t)+RFromB(m,p,t); } }
Parameter MineRegime {
    IndexDomain: m;
    Range: binary; }
Variable RFromIOP {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Range: nonnegative;
    Definition: {
        IF MineRegime(m)=1 THEN RFromIOPFIFO(m,p,t)
        ELSE RFromIOPLIFO(m,p,t) ENDIF; } }
Variable RFromL {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Range: nonnegative;
    Definition: {
        IF MineRegime(m)=1 THEN RFromLFIFO(m,p,t)
        ELSE RFromLLIFO(m,p,t) ENDIF; } }
Variable RFromB {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Range: nonnegative; }
Variable RFromBBinary {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Range: binary; }
Constraint RFromBInq1 {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Definition: {
        IF t=FirstPeriod THEN RFromB(m,p,t)>=sum((f,d,
        s),TS(m,f,p,t)*x(m,f,p,d,s,t))-IOP(m,p,t)-sIni
        tial(m,p)
        ELSE RFromB(m,p,t)>=sum((f,d,s),TS(m,f,p,t)*x(
        m,f,p,d,s,t))-IOP(m,p,t)-sLevel(m,p,t-1)
        ENDIF; } }
Constraint RFromBInq2 {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Definition: {
        RFromB(m,p,t)<=sum((f,d,s),TS(m,f,p,t)*x(m,f,p
```

```
                ,d,s,t))-IOP(m,p,t)+10000*RFromBBinary(m,p,t);
            } }
        Constraint RFromBInq3 {
            IndexDomain: (m,p,t)|p in MineToProducts(m);
            Definition: {
                RFromB(m,p,t)<=10000*(1-RFromBBinary(m,p,t));
            } }
        Parameter SLP_RFromIOP {
            IndexDomain: (m,p,t)|p in MineToProducts(m);
            Range: nonnegative;
            Definition: {
                IF MineRegime(m)=1 THEN SLP_RFromIOPFIFO(m,p,t
                ) ELSE SLP_RFromIOPLIFO(m,p,t) ENDIF; } }
        Parameter SLP_RFromL {
            IndexDomain: (m,p,t)|p in MineToProducts(m);
            Range: nonnegative;
            Definition: {
                IF MineRegime(m)=1 THEN SLP_RFromLFIFO(m,p,t)
                ELSE SLP_RFromLLIFO(m,p,t) ENDIF; } }
        Parameter SLP_RFromB {
            IndexDomain: (m,p,t)|p in MineToProducts(m);
            Range: nonnegative;
            Definition: {
                IF t=FirstPeriod THEN max(0,sum((f,d,s),TS(m,f
                ,p,t)*OT_x(m,f,p,d,s,t))-IOP(m,p,t)-sInitial(m
                ,p))
                ELSE max(0,sum((f,d,s),TS(m,f,p,t)*OT_x(m,f,p,
                d,s,t))-IOP(m,p,t)-OT_sLevel(m,p,t-1))
ENDIF;} }
}
    DeclarationSection Railed_Quantity_FIFO {
        Variable RFromIOPFIFO {
            IndexDomain: (m,p,t)|p in MineToProducts(m);
            Range: [0, IOP(m, p, t)]; }
        Variable RFromIOPFIFOBinary {
            IndexDomain: (m,p,t)|p in MineToProducts(m);
            Range: binary; }
        Constraint RFromIOPFIFOInq1 {
            IndexDomain: (m,p,t)|p in MineToProducts(m);
            Definition: {
                RFromIOPFIFO(m,p,t)<=TSxMinusSPos(m,p,t); } }
```

```
Constraint RFromIOPFIFOInq2 {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Definition: {
        RFromIOPFIFO(m,p,t)>=IOP(m,p,t)-(IOP(m,p,t)+10
        000)*RFromIOPFIFOBinary(m,p,t); } }
Constraint RFromIOPFIFOInq3 {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Definition: {
        RFromIOPFIFO(m,p,t)>=TSxMinusSPos(m,p,t)-(1000
        0-IOP(m,p,t))*(1-RFromIOPFIFOBinary(m,p,t));}}
Variable TSxMinusSPos {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Range: nonnegative; }
Variable TSxMinusSPosBinary {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Range: binary; }
Constraint TSxMinusSPosInq1 {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Definition: {
        IF t=FirstPeriod THEN TSxMinusSPos(m,p,t)>=sum
        ((f,d,s),TS(m,f,p,t)*x(m,f,p,d,s,t))-sInitial(
        m,p)
        ELSE TSxMinusSPos(m,p,t)>=sum((f,d,s),TS(m,f,p
        ,t)*x(m,f,p,d,s,t))-sLevel(m,p,t-1) ENDIF; } }
Constraint TSxMinusSPosInq2 {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Definition: {
        IF t=FirstPeriod THEN TSxMinusSPos(m,p,t)<=sum
        ((f,d,s),TS(m,f,p,t)*x(m,f,p,d,s,t))-sInitial(
        m,p)+10000*TSxMinusSPosBinary(m,p,t)
        ELSE TSxMinusSPos(m,p,t)<=sum((f,d,s),TS(m,f,p
        ,t)*x(m,f,p,d,s,t))-sLevel(m,p,t-1)+10000*TSxM
        inusSPosBinary(m,p,t) ENDIF; } }
Constraint TSxMinusSPosInq3 {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Definition: {
        TSxMinusSPos(m,p,t)<=10000*(1-TSxMinusSPosBina
        ry(m,p,t)); } }
Variable RFromLFIFO {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Range: nonnegative; }
```

```
Variable RFromLFIFOBinary {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Range: binary; }
Constraint RFromLFIFOInq1 {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Definition: {
        RFromLFIFO(m,p,t)<=sum((f,d,s),TS(m,f,p,t)*x(m
        ,f,p,d,s,t)); } }
Constraint RFromLFIFOInq2 {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Definition: {
        IF t=FirstPeriod THEN RFromLFIFO(m,p,t)<=sInit
        ial(m,p) ELSE RFromLFIFO(m,p,t)<=sLevel(m,p,t-
        1) ENDIF; } }
Constraint RFromLFIFOInq3 {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Definition: {
        RFromLFIFO(m,p,t)>=sum((f,d,s),TS(m,f,p,t)*x(m
        ,f,p,d,s,t))-10000*RFromLFIFOBinary(m,p,t);} }
Constraint RFromLFIFOInq4 {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Definition: {
        IF t=FirstPeriod THEN RFromLFIFO(m,p,t)>=sInit
        ial(m,p)-20000*(1-RFromLFIFOBinary(m,p,t))
        ELSE RFromLFIFO(m,p,t)>=sLevel(m,p,t-1)-20000*
        (1-RFromLFIFOBinary(m,p,t)) ENDIF; } }
Parameter SLP_RFromIOPFIFO {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Range: nonnegative;
    Definition: {
        IF t=FirstPeriod THEN min(IOP(m,p,t),max(0,sum
        ((f,d,s),TS(m,f,p,t)*OT_x(m,f,p,d,s,t))-sIniti
        al(m,p)))
        ELSE min(IOP(m,p,t),max(0,sum((f,d,s),TS(m,f,p
        ,t)*OT_x(m,f,p,d,s,t))-OT_sLevel(m,p,t-1)))
        ENDIF; } }
Parameter SLP_RFromLFIFO {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Range: nonnegative;
    Definition: {
        IF t=FirstPeriod THEN min(sum((f,d,s),TS(m,f,p
```

```
            ,t)*OT_x(m,f,p,d,s,t)),sInitial(m,p))
            ELSE min(sum((f,d,s),TS(m,f,p,t)*OT_x(m,f,p,d,
            s,t)),OT_sLevel(m,p,t-1)) ENDIF; } }
}
DeclarationSection Railed_Quantity_LIFO {
    Variable RFromIOPLIFO {
        IndexDomain: (m,p,t)|p in MineToProducts(m);
        Range: [0, IOP(m, p, t)]; }
    Variable RFromIOPLIFOBinary {
        IndexDomain: (m,p,t)|p in MineToProducts(m);
        Range: binary; }
    Constraint RFromIOPLIFOInq1 {
        IndexDomain: (m,p,t)|p in MineToProducts(m);
        Definition: {
            RFromIOPLIFO(m,p,t)<=sum((f,d,s), TS(m,f,p,t)*
            x(m,f,p,d,s,t)); } }
    Constraint RFromIOPLIFOInq2 {
        IndexDomain: (m,p,t)|p in MineToProducts(m);
        Definition: {
            RFromIOPLIFO(m,p,t)>=sum((f,d,s), TS(m,f,p,t)*
            x(m,f,p,d,s,t))-(10000-IOP(m,p,t))*RFromIOPLIF
            OBinary(m,p,t); } }
    Constraint RFromIOPLIFOInq3 {
        IndexDomain: (m,p,t)|p in MineToProducts(m);
        Definition: {
            RFromIOPLIFO(m,p,t)>=IOP(m,p,t)-IOP(m,p,t)*(1-
            RFromIOPLIFOBinary(m,p,t)); } }
    Variable TSxMinusIOPPos {
        IndexDomain: (m,p,t)|p in MineToProducts(m);
        Range: nonnegative; }
    Variable TSxMinusIOPPosBinary {
        IndexDomain: (m,p,t)|p in MineToProducts(m);
        Range: binary; }
    Constraint TSxMinusIOPPosInq1 {
        IndexDomain: (m,p,t)|p in MineToProducts(m);
        Definition: {
            TSxMinusIOPPos(m,p,t)>=sum((f,d,s), TS(m,f,p,t
            )*x(m,f,p,d,s,t))-IOP(m,p,t); }  }
    Constraint TSxMinusIOPPosInq2 {
        IndexDomain: (m,p,t)|p in MineToProducts(m);
        Definition: {
```

```
        TSxMinusIOPPos(m,p,t)<=sum((f,d,s), TS(m,f,p,t
        )*x(m,f,p,d,s,t))-IOP(m,p,t)+10000*TSxMinusIOP
        PosBinary(m,p,t); } }
Constraint TSxMinusIOPPosInq3 {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Definition: {
        TSxMinusIOPPos(m,p,t)<=10000*(1-TSxMinusIOPPos
        Binary(m,p,t)); } }
Variable RFromLLIFO {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Range: nonnegative; }
Variable RFromLLIFOBinary {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Range: binary; }
Constraint RFromLLIFOInq1 {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Definition: {
        IF t=FirstPeriod THEN RFromLLIFO(m,p,t)<=sInit
        ial(m,p) ELSE RFromLLIFO(m,p,t)<=sLevel(m,p,t-
        1) ENDIF; } }
Constraint RFromLLIFOInq2 {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Definition: {
        RFromLLIFO(m,p,t)<=TSxMinusIOPPos(m,p,t); } }
Constraint RFromLLIFOInq3 {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Definition: {
        IF t=FirstPeriod THEN RFromLLIFO(m,p,t)>=sInit
        ial(m,p)-20000*RFromLLIFOBinary(m,p,t)
        ELSE RFromLLIFO(m,p,t)>=sLevel(m,p,t-1)-20000*
        RFromLLIFOBinary(m,p,t) ENDIF; } }
Constraint RFromLLIFOInq4 {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Definition: {
        RFromLLIFO(m,p,t)>=TSxMinusIOPPos(m,p,t)-20000
        *(1-RFromLLIFOBinary(m,p,t)); } }
Parameter SLP_RFromIOPLIFO {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Range: nonnegative;
    Definition: {
        min(sum((f,d,s),TS(m,f,p,t)*OT_x(m,f,p,d,s,t))
```

```
            ,IOP(m,p,t)); } }
    Parameter SLP_RFromLLIFO {
        IndexDomain: (m,p,t)|p in MineToProducts(m);
        Range: nonnegative;
        Definition: {
            IF t=FirstPeriod THEN min(sInitial(m,p),max(0,
            sum((f,d,s),TS(m,f,p,t)*OT_x(m,f,p,d,s,t))-IOP
            (m,p,t)))
            ELSE min(OT_sLevel(m,p,t-1),max(0,sum((f,d,s),
            TS(m,f,p,t)*OT_x(m,f,p,d,s,t))-IOP(m,p,t)))
            ENDIF; } }
}
DeclarationSection Mine_Stockpile_Capacities {
    Parameter SMin {
        IndexDomain: (m,p,t)|p in MineToProducts(m); }
    Parameter SMax {
        IndexDomain: (m,p,t)|p in MineToProducts(m); }
    Variable AlphaMin {
        IndexDomain: (m,p,t)|p in MineToProducts(m);
        Range: nonnegative; }
    Variable AlphaMax {
        IndexDomain: (m,p,t)|p in MineToProducts(m);
        Range: nonnegative; }
    Constraint MineLiveStockpileLevelMin {
        IndexDomain: (m,p,t)|p in MineToProducts(m);
        Definition: {
            SMin(m,p,t)-AlphaMin(m,p,t)<=sLevel(m,p,t);} }
    Constraint MineLiveStockpileLevelMax {
        IndexDomain: (m,p,t)|p in MineToProducts(m);
        Definition: {
            sLevel(m,p,t)<=SMax(m,p,t)+AlphaMax(m,p,t);} }
    Parameter BMax {
        IndexDomain: (m,p,t)|p in MineToProducts(m); }
    Variable AlphaBulk {
        IndexDomain: (m,p,t)|p in MineToProducts(m);
        Range: nonnegative; }
    Constraint MineBulkStockpileLevel {
        IndexDomain: (m,p,t)|p in MineToProducts(m);
        Definition: {
            bLevel(m,p,t)<=BMax(m,p,t)+AlphaBulk(m,p,t);}}
    Parameter YLM {
```

```
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Range: nonnegative; }
Parameter YLB {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Definition: {
        max(bInitial(m,p),BMax(m,p,t)); } }
Constraint MineYardLimit {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Definition: {
        SMax(m,p,t)+AlphaMax(m,p,t)<=YLM(m,p,t); } }
Parameter MineBulkInRatesPerDay {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Definition: {
        IF m in BrockmanLoopMines THEN 20 ELSE MineBul
        kInRatesDummy(m,p,t) ENDIF; } }
Parameter MineBulkInRatesDummy {
    IndexDomain: (m,p,t)|p in MineToProducts(m); }
Parameter MineBulkInRates {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Definition: {
        days(t)*MineBulkInRatesPerDay(m,p,t); } }
Parameter RegionBulkInRatesPerDay {
    IndexDomain: (g,t);
    Definition: IF RegionBulkInDummy(g,t) <> 0 THEN Re
        gionBulkInDummy(g,t) ELSE 100 ENDIF; }
Parameter RegionBulkInDummy {
    IndexDomain: (g,t); }
Parameter RegionBulkInRates {
    IndexDomain: (g,t);
    Definition: {
        days(t)*RegionBulkInRatesPerDay(g,t); } }
Constraint RegionBulkInMax {
    IndexDomain: (g,t);
    Definition: {
        sum((m in RegionToMines(g), p in MineToProduct
        s(m)), yFromBulk(m,p,t))<=RegionBulkInRates(g,
        t); } }
Parameter MineBulkOutRatesPerDay {
    IndexDomain: (m,p,t)|p in MineToProducts(m);
    Definition: {
        IF m in BrockmanLoopMines THEN 20 ELSE MineBul
```

```
            kOutRatesDummy(m,p,t) ENDIF; } }
    Parameter MineBulkOutRatesDummy {
        IndexDomain: (m,p,t)|p in MineToProducts(m); }
    Parameter MineBulkOutRates {
        IndexDomain: (m,p,t)|p in MineToProducts(m);
        Definition: {
            days(t)*MineBulkOutRatesPerDay(m,p,t); } }
    Parameter RegionBulkOutRatesPerDay {
        IndexDomain: (g,t);
        Definition: IF RegionBulkOutDummy(g,t) <> 0 THEN R
            egionBulkOutDummy(g,t) ELSE 100 ENDIF; }
    Parameter RegionBulkOutDummy {
        IndexDomain: (g,t); }
    Parameter RegionBulkOutRates {
        IndexDomain: (g,t);
        Definition: {
            days(t)*RegionBulkOutRatesPerDay(g,t); } }
    Constraint RegionBulkOutMax {
        IndexDomain: (g,t);
        Definition: {
            sum((m in RegionToMines(g), p in MineToProduct
            s(m)), yToBulk(m,p,t))<=RegionBulkOutRates(g,t
            ); } }
}
DeclarationSection Port_Stockpile_Capacities {
    Parameter WMin {
        IndexDomain: (r,s,t)|s in PortToProducts(r); }
    Parameter WMax {
        IndexDomain: (r,s,t)|s in PortToProducts(r); }
    Variable BetaMin {
        IndexDomain: (r,s,t)|s in PortToProducts(r);
        Range: nonnegative; }
    Variable BetaMax {
        IndexDomain: (r,s,t)|s in PortToProducts(r);
        Range: nonnegative; }
    Constraint PortLiveStockpileLevelMin {
        IndexDomain: (r,s,t)|s in PortToProducts(r);
        Definition: {
            WMin(r,s,t)-BetaMin(r,s,t)<=wLevel(r,s,t); } }
    Constraint PortLiveStockpileLevelMax {
        IndexDomain: (r,s,t)|s in PortToProducts(r);
```

```
    Definition: {
        wLevel(r,s,t)<=WMax(r,s,t)+BetaMax(r,s,t); } }
Parameter VMax {
    IndexDomain: (r,s,t)|s in PortToProducts(r); }
Variable BetaBulk {
    IndexDomain: (r,s,t)|s in PortToProducts(r);
    Range: nonnegative; }
Constraint PortBulkStockpileLevel {
    IndexDomain: (r,s,t)|s in PortToProducts(r);
    Definition: {
        vLevel(r,s,t)<=VMax(r,s,t)+BetaBulk(r,s,t);} }
Parameter YLP {
    IndexDomain: (r,s,t)|s in PortToProducts(r);
    Range: nonnegative; }
Constraint PortYardLimit {
    IndexDomain: (r,s,t)|s in PortToProducts(r);
    Definition: {
        WMax(r,s,t)+BetaMax(r,s,t)<=YLP(r,s,t); } }
Parameter PortBulkInTotalRatesPerDay {
    IndexDomain: (r,t); }
Parameter PortBulkInTotalRates {
    IndexDomain: (r,t);
    Definition: {
        days(t)*PortBulkInTotalRatesPerDay(r,t); } }
Parameter PortBulkInRates {
    IndexDomain: (r,s,t);
    Definition: {
        PortBulkInTotalRates(r,t); } }
Constraint PortBulkInMax {
    IndexDomain: (r,t);
    Definition: {
        sum(s,uFromBulk(r,s,t))<=PortBulkInTotalRates(
        r,t); } }
Parameter PortBulkOutTotalRatesPerDay {
    IndexDomain: (r,t); }
Parameter PortBulkOutTotalRates {
    IndexDomain: (r,t);
    Definition: {
        days(t)*PortBulkOutTotalRatesPerDay(r,t); } }
Parameter PortBulkOutRates {
    IndexDomain: (r,s,t);
```

```
        Definition: {
            PortBulkOutTotalRates(r,t); } }
    Constraint PortBulkOutMax {
        IndexDomain: (r,t);
        Definition: {
            sum(s,uToBulk(r,s,t))<=PortBulkOutTotalRates(r
            ,t); } }
}
DeclarationSection Train_and_Fleet_Capacities {
    Parameter xMin {
        IndexDomain: (m,t);
        Range: integer; }
    Parameter xMax {
        IndexDomain: (m,t);
        Range: integer;
        Definition: {
            IF m in RobeValleyMines THEN sum(f in MinesToF
            leet(m), MFT(f,t)) ELSE xMaxDummy(m,t) ENDIF;
        } }
    Parameter xMaxDummy {
        IndexDomain: (m,t);
        Range: integer; }
    Parameter MT {
        IndexDomain: (g,t);
        Range: integer; }
    Constraint MaxTrains {
        IndexDomain: (m,t);
        Definition: {
            sum((f,p,d,s),x(m,f,p,d,s,t))<=xMax(m,t); } }
    Constraint MaxTrainsInRegion {
        IndexDomain: (g,t);
        Definition: {
            sum((m in RegionToMines(g),f,p in  MineToProdu
            cts(m),d,s),x(m,f,p,d,s,t))<=MT(g,t); } }
    Parameter MFT {
        IndexDomain: (f,t);
        Range: integer; }
    Variable FleetX {
        IndexDomain: (f,t);
        Range: integer;
        Definition: {
```

```
            sum((m in FleetToMines(f),p in MineToProducts(
            m),d,s),x(m,f,p,d,s,t)); } }
    Constraint FleetCapacity {
        IndexDomain: (f,t);
        Definition: {
            sum((m in FleetToMines(f),p in MineToProducts(
            m),d,s),x(m,f,p,d,s,t))<=MFT(f,t); } }
    Parameter CT {
        IndexDomain: (m,p,t)|p in MineToProducts(m); }
    Parameter PFT {
        IndexDomain: (f,t); }
    Variable Mu {
        IndexDomain: (f,t);
        Range: nonnegative; }
    Constraint FleetHourCapacity {
        IndexDomain: (f,t);
        Definition: {
            sum((m in FleetToMines(f),p in MineToProducts(
            m),d,s), CT(m,p,t)*x(m,f,p,d,s,t))<=PFT(f,t)+M
            u(f,t); } }
}
DeclarationSection Car_Dumper_Capacities {
    Parameter DC {
        IndexDomain: (d,t);
        Range: integer; }
    Constraint DumperCapacity {
        IndexDomain: (d,t);
        Definition: {
            sum((m,f,p,s),x(m,f,p,d,s,t))<=DC(d,t); } }
    Parameter CLWC {
        IndexDomain: t; }
    Constraint CLWCCapacity {
        IndexDomain: t;
        Definition: {
            sum((m,f,p,d in DumpersInWC,s),x(m,f,p,d,s,t))
            <=CLWC(t); } }
}
DeclarationSection Shipping_Capacities {
    Parameter ZMax {
        IndexDomain: (r,t); }
    Constraint ShipCapacity {
```

```
        IndexDomain: (r,t);
        Definition: sum(s in PortToProducts(r),zPlusRF(r,s
            ,t)*(1-PercentLump(r,s,t)))<=ZMax(r,t); } }
DeclarationSection Joint_Ventures {
    Parameter JVTarget {
        IndexDomain: (m,t)|m in JVMines; }
    Parameter JVCummulative {
        IndexDomain: (m,t)|m in JVMines; }
    Parameter JVTolerance {
        IndexDomain: (m,t)|m in JVMines; }
    Variable xCumulative {
        IndexDomain: (m,f,p,d,s,t)|m in JVMines;
        Range: free;
        Definition: {
            IF t=FirstPeriod THEN x(m,f,p,d,s,t) ELSE xCum
            ulative(m,f,p,d,s,t-1)+x(m,f,p,d,s,t) ENDIF;}}
    Constraint JVCummulativeMin {
        IndexDomain: (m,t)|m in JVMines;
        Definition: {
            sum((f,p,d,s),xCumulative(m,f,p,d,s,t))>=JVCum
            mulative(m,t)-JVTolerance(m,t); } }
    Constraint JVCummulativeMax {
        IndexDomain: (m,t)|m in JVMines;
        Definition: {
            sum((f,p,d,s),xCumulative(m,f,p,d,s,t))<=JVCum
            mulative(m,t)+JVTolerance(m,t); } }
}
DeclarationSection Other_Parameters {
    Parameter days {
        IndexDomain: t; }
    Parameter FirstPeriod;
}
DeclarationSection Grade_Variables_and_Initials {
    Parameter IOPGrade {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m); }
    Variable LMGrade {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Range: [LMMin(m, p, c, t), LMMax(m, p, c, t)]; }
    Parameter LMInitial {
        IndexDomain: (m,p,c)|p in MineToProducts(m);
        Range: nonnegative; }
```

```
    Variable BMGrade {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Range: [0, BMMax(m, p, c, t)]; }
    Constraint BMGradeAndQuantity {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            BMGrade(m,p,c,t)<=10000*bLevel(m,p,t); } }
    Parameter BMInitial {
        IndexDomain: (m,p,c)|p in MineToProducts(m); }
    Variable LPGrade {
        IndexDomain: (r,s,c,t)| s in PortToProducts(r);
        Range: [0, LPMax(r, s, c, t)];
        Definition: {
            IF s in LumpProducts THEN ZG(r,s,c,t) ELSE LPG
            rade(r,s,c,t) ENDIF; } }
    Parameter LPInitial {
        IndexDomain: (r,s,c)|s in PortToProducts(r); }
    Variable BPGrade {
        IndexDomain: (r,s,c,t)|s in PortToProducts(r);
        Range: [0, BPMax(r, s, c, t)]; }
    Constraint BPGradeAndQuantity {
        IndexDomain: (r,s,c,t)|s in PortToProducts(r);
        Definition: {
            BPGrade(r,s,c,t)<=10000*vLevel(r,s,t); } }
    Parameter BPInitial {
        IndexDomain: (r,s,c)|s in PortToProducts(r); }
    Variable RGrade {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Range: [0, RGMax(m, p, c, t)]; }
    Variable ZG {
        IndexDomain: (r,s,c,t)|s in PortToProducts(r);
        Range: [0, LPMax(r, s, c, t)]; } }
DeclarationSection Grade_Limits {
    Parameter LMMin {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            IF t=FirstPeriod THEN IF sInitial(m,p)=0 and I
            OP(m,p,t)=0 and bInitial(m,p)=0 THEN 0 ELSE mi
            n(IF sInitial(m,p)<>0 THEN LMInitial(m,p,c) EL
            SE 100 ENDIF,IF IOP(m,p,t)<>0 THEN IOPGrade(m,
            p,c,t) ELSE 100 ENDIF, IF bInitial(m,p)<>0 THE
```

```
            N BMInitial(m,p,c) ELSE 100 ENDIF) ENDIF
            ELSE min(IF IOP(m,p,t)=0 THEN LMMin(m,p,c,t-1)
            ELSE IOPGrade(m,p,c,t) ENDIF,LMMin(m,p,c,t-1))
            ENDIF; } }
    Parameter LMMax {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            IF t=FirstPeriod THEN max(LMInitial(m,p,c),IOP
            Grade(m,p,c,t),BMInitial(m,p,c))
            ELSE max(IOPGrade(m,p,c,t),LMMax(m,p,c,t-1))
            ENDIF; } }
    Parameter BMMin {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            IF BMax(m,p,t)=0 or bInitial(m,p)=0 THEN 0
            ELSE LMMin(m,p,c,t) ENDIF; } }
    Parameter BMMax {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            IF BMax(m,p,t)=0 THEN 0
            ELSE LMMax(m,p,c,t) ENDIF; } }
    Parameter LPMin {
        IndexDomain: (r,s,c,t)|s in PortToProducts(r);
        Definition: {
            IF t=FirstPeriod THEN min(IF wInitial(r,s)<>0
            THEN LPInitial(r,s,c) ELSE 100 ENDIF, IF vInit
            ial(r,s)<>0 THEN BPInitial(r,s,c) ELSE 100 END
            IF,min((m in PortToMine(r),p in ShippedProduct
            ToProduct(m,s)),IF sInitial(m,p)=0 and IOP(m,p
            ,t)=0 and bInitial(m,p)=0 THEN 0 ELSE LMMin(m,
            p,c,t) ENDIF))
            ELSE min(min((m in PortToMine(r),p in ShippedP
            roductToProduct(m,s)),IF sInitial(m,p)=0 and I
            OP(m,p,t)=0 and bInitial(m,p)=0 THEN 0
            ELSE LMMin(m,p,c,t) ENDIF),LPMin(r,s,c,t-1))
            ENDIF; } }
    Parameter LPMax {
        IndexDomain: (r,s,c,t)|s in PortToProducts(r);
        Definition: {
            IF t=FirstPeriod THEN max(LPInitial(r,s,c), BP
            Initial(r,s,c),max((m in PortToMine(r),p in Sh
```

```
            ippedProductToProduct(m,s)),LMMax(m,p,c,t)))
            ELSE max(max((m in PortToMine(r),p in ShippedP
            roductToProduct(m,s)),LMMax(m,p,c,t)),LPMax(r,
            s,c,t-1)) ENDIF; } }
    Parameter BPMin {
        IndexDomain: (r,s,c,t)|s in PortToProducts(r);
        Definition: {
            IF VMax(r,s,t)=0 or vInitial(r,s)=0 THEN 0
            ELSE LPMin(r,s,c,t) ENDIF; } }
    Parameter BPMax {
        IndexDomain: (r,s,c,t)|s in PortToProducts(r);
        Definition: {
            IF VMax(r,s,t)=0 THEN 0
            ELSE LPMax(r,s,c,t) ENDIF; } }
    Parameter RGMin {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            LMMin(m,p,c,t); } }
    Parameter RGMax {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            LMMax(m,p,c,t); } }
}
DeclarationSection Mine_Grade_Constraints {
    Variable LMxS {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Range: nonnegative;
        Definition: {
            IF t=FirstPeriod THEN LMInitial(m,p,c)*sInitia
            l(m,p)-LMxYToBulk(m,p,c,t)+IOP(m,p,t)*IOPGrade
            (m,p,c,t)+BMxYFromBulk(m,p,c,t)-sum((f,d,s),RG
            xX(m,f,p,d,s,c,t))
            ELSE LMxS(m,p,c,t-1)-LMxYToBulk(m,p,c,t)+IOP(m
            ,p,t)*IOPGrade(m,p,c,t)+BMxYFromBulk(m,p,c,t)-
            sum((f,d,s),RGxX(m,f,p,d,s,c,t)) ENDIF; } }
    Constraint LMxSConvex1 {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            LMxS(m,p,c,t)>=LMMax(m,p,c,t)*sLevel(m,p,t)+YL
            M(m,p,t)*LMGrade(m,p,c,t)-LMMax(m,p,c,t)*YLM(m
            ,p,t); } }
```

```
Constraint LMxSConvex2 {
    IndexDomain: (m,p,c,t)|p in MineToProducts(m);
    Definition: {
        LMxS(m,p,c,t)>=LMMin(m,p,c,t)*sLevel(m,p,t);}}
Constraint LMxSConcave1 {
    IndexDomain: (m,p,c,t)|p in MineToProducts(m);
    Definition: {
        LMxS(m,p,c,t)<=LMMax(m,p,c,t)*sLevel(m,p,t);}}
Constraint LMxSConcave2 {
    IndexDomain: (m,p,c,t)|p in MineToProducts(m);
    Definition: {
        LMxS(m,p,c,t)<=LMMin(m,p,c,t)*sLevel(m,p,t)+YL
        M(m,p,t)*LMGrade(m,p,c,t)-LMMin(m,p,c,t)*YLM(m
        ,p,t); } }
Variable LMxYToBulk {
    IndexDomain: (m,p,c,t)|p in MineToProducts(m);
    Range: nonnegative; }
Constraint LMxYToBulkConvex1 {
    IndexDomain: (m,p,c,t)|p in MineToProducts(m);
    Definition: {
        IF m in RobeValleyMines THEN LMxYToBulk(m,p,c,
        t)>=IOPGrade(m,p,c,t)*yToBulk(m,p,t)
        ELSE LMxYToBulk(m,p,c,t)>=MineBulkOutRates(m,p
        ,t)*LMGrade(m,p,c,t)+LMMax(m,p,c,t)*yToBulk(m,
        p,t)-MineBulkOutRates(m,p,t)*LMMax(m,p,c,t)
        ENDIF; } }
Constraint LMxYToBulkConvex2 {
    IndexDomain: (m,p,c,t)|p in MineToProducts(m);
    Definition: {
        IF m in RobeValleyMines THEN LMxYToBulk(m,p,c,
        t)>=IOPGrade(m,p,c,t)*yToBulk(m,p,t)
        ELSE LMxYToBulk(m,p,c,t)>=LMMin(m,p,c,t)*yToBu
        lk(m,p,t) ENDIF; } }
Constraint LMxYToBulkConcave1 {
    IndexDomain: (m,p,c,t)|p in MineToProducts(m);
    Definition: {
        IF m in RobeValleyMines THEN LMxYToBulk(m,p,c,
        t)<=IOPGrade(m,p,c,t)*yToBulk(m,p,t)
        ELSE LMxYToBulk(m,p,c,t)<=MineBulkOutRates(m,p
        ,t)*LMGrade(m,p,c,t)+LMMin(m,p,c,t)*yToBulk(m,
        p,t)-MineBulkOutRates(m,p,t)*LMMin(m,p,c,t)
```

```
        ENDIF; } }
    Constraint LMxYToBulkConcave2 {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            IF m in RobeValleyMines THEN LMxYToBulk(m,p,c,
            t)<=IOPGrade(m,p,c,t)*yToBulk(m,p,t)
            ELSE LMxYToBulk(m,p,c,t)<=LMMax(m,p,c,t)*yToBu
            lk(m,p,t) ENDIF; } }
    Variable BMxB {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Range: nonnegative;
        Definition: {
            IF t=FirstPeriod THEN BMInitial(m,p,c)*bInitia
            l(m,p)-BMxYFromBulk(m,p,c,t)+LMxYToBulk(m,p,c,
            t) ELSE BMxB(m,p,c,t-1)-BMxYFromBulk(m,p,c,t)+
            LMxYToBulk(m,p,c,t) ENDIF; } }
    Constraint BMxBConvex1 {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            BMxB(m,p,c,t)>=BMMax(m,p,c,t)*bLevel(m,p,t)+YL
            B(m,p,t)*BMGrade(m,p,c,t)-BMMax(m,p,c,t)*YLB(m
            ,p,t); } }
    Constraint BMxBConvex2 {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            BMxB(m,p,c,t)>=BMMin(m,p,c,t)*bLevel(m,p,t);}}
    Constraint BMxBConcave1 {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            BMxB(m,p,c,t)<=BMMin(m,p,c,t)*bLevel(m,p,t)+YL
            B(m,p,t)*BMGrade(m,p,c,t)-BMMin(m,p,c,t)*YLB(m
            ,p,t); } }
    Constraint BMxBConcave2 {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            BMxB(m,p,c,t)<=BMMax(m,p,c,t)*bLevel(m,p,t);}}
    Variable BMxYFromBulk {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Range: nonnegative; }
    Constraint BMxYFromBulkConvex1 {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
```

```
        Definition: {
            BMxYFromBulk(m,p,c,t)>=BMMax(m,p,c,t)*yFromBul
            k(m,p,t)+MineBulkInRates(m,p,t)*BMGrade(m,p,c,
            t)-BMMax(m,p,c,t)*MineBulkInRates(m,p,t); } }
    Constraint BMxYFromBulkConvex2 {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            BMxYFromBulk(m,p,c,t)>=BMMin(m,p,c,t)*yFromBul
            k(m,p,t); } }
    Constraint BMxYFromBulkConcave1 {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            BMxYFromBulk(m,p,c,t)<=BMMin(m,p,c,t)*yFromBul
            k(m,p,t)+MineBulkInRates(m,p,t)*BMGrade(m,p,c,
            t)-BMMin(m,p,c,t)*MineBulkInRates(m,p,t); } }
    Constraint BMxYFromBulkConcave2 {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            BMxYFromBulk(m,p,c,t)<=BMMax(m,p,c,t)*yFromBul
            k(m,p,t); } }
}
DeclarationSection Railed_Grade_Constraints {
    Variable RGxX {
        IndexDomain: (m,f,p,d,s,c,t) | m in FleetToMines(f
            )and p in MineToProducts(m) and d in MineToDum
            pers(m,p) and s in MinePortToProducts(m,p);
        Range: nonnegative; }
    Constraint RailedGrades {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            IF t=FirstPeriod THEN sum((f,d,s),RGxX(m,f,p,d
            ,s,c,t))=RFromIOP(m,p,t)*IOPGrade(m,p,c,t)+LMI
            nitial(m,p,c)*RFromL(m,p,t)+BMinitial(m,p,c)*R
            FromB(m,p,t)
            ELSE sum((f,d,s),RGxX(m,f,p,d,s,c,t))=RFromIOP
            (m,p,t)*IOPGrade(m,p,c,t)+LMxRL(m,p,c,t)+BMxRB
            (m,p,c,t) ENDIF; } }
    Constraint RGxXConvex1 {
        IndexDomain: (m,f,p,c,t)|p in MineToProducts(m);
        Definition: {
            sum((d,s),RGxX(m,f,p,d,s,c,t))>=RGMax(m,p,c,t)
```

```
            *sum((d,s),TS(m,f,p,t)*x(m,f,p,d,s,t))+TS(m,f,
            p,t)*xMax(m,t)*RGrade(m,p,c,t)-TS(m,f,p,t)*xMa
            x(m,t)*RGMax(m,p,c,t); } }
    Constraint RGxXConvex2 {
        IndexDomain: (m,f,p,c,t)|p in MineToProducts(m);
        Definition: {
            sum((d,s),RGxX(m,f,p,d,s,c,t))>=RGMin(m,p,c,t)
            *sum((d,s),TS(m,f,p,t)*x(m,f,p,d,s,t))+TS(m,f,
            p,t)*xMin(m,t)*RGrade(m,p,c,t)-TS(m,f,p,t)*xMi
            n(m,t)*RGMin(m,p,c,t); } }
    Constraint RGxXConcave1 {
        IndexDomain: (m,f,p,c,t)|p in MineToProducts(m);
        Definition: {
            sum((d,s),RGxX(m,f,p,d,s,c,t))<=RGMax(m,p,c,t)
            *sum((d,s),TS(m,f,p,t)*x(m,f,p,d,s,t))+TS(m,f,
            p,t)*xMin(m,t)*RGrade(m,p,c,t)-TS(m,f,p,t)*xMi
            n(m,t)*RGMax(m,p,c,t); } }
    Constraint RGxXConcave2 {
        IndexDomain: (m,f,p,c,t)|p in MineToProducts(m);
        Definition: {
            sum((d,s),RGxX(m,f,p,d,s,c,t))<=RGMin(m,p,c,t)
            *sum((d,s),TS(m,f,p,t)*x(m,f,p,d,s,t))+TS(m,f,
            p,t)*xMax(m,t)*RGrade(m,p,c,t)-TS(m,f,p,t)*xMa
            x(m,t)*RGMin(m,p,c,t); } }
    Variable LMxRL {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Range: nonnegative;
        Definition: {
            IF t='1' THEN LMInitial(m,p,c)*RFromL(m,p,t)
            ELSE LMxRL(m,p,c,t) ENDIF; } }
    Constraint LMxRLConvex1 {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            IF t=FirstPeriod THEN LMxRL(m,p,c,t)>=LMInitia
            l(m,p,c)*RFromL(m,p,t)
            ELSE LMxRL(m,p,c,t)>=LMMax(m,p,c,t)*RFromL(m,p
            ,t)+YLM(m,p,t)*LMGrade(m,p,c,t-1)-LMMax(m,p,c,
            t)*YLM(m,p,t) ENDIF; } }
    Constraint LMxRLConvex2 {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
```

```
        IF t=FirstPeriod THEN LMxRL(m,p,c,t)>=LMInitia
        l(m,p,c)*RFromL(m,p,t)
        ELSE LMxRL(m,p,c,t)>=LMMin(m,p,c,t)*RFromL(m,p
        ,t) ENDIF; } }
Constraint LMxRLConcave1 {
    IndexDomain: (m,p,c,t)|p in MineToProducts(m);
    Definition: {
        IF t=FirstPeriod THEN LMxRL(m,p,c,t)<=LMInitia
        l(m,p,c)*RFromL(m,p,t)
        ELSE LMxRL(m,p,c,t)<=LMMax(m,p,c,t)*RFromL(m,p
        ,t) ENDIF; } }
Constraint LMxRLConcave2 {
    IndexDomain: (m,p,c,t)|p in MineToProducts(m);
    Definition: {
        IF t=FirstPeriod THEN LMxRL(m,p,c,t)<=LMInitia
        l(m,p,c)*RFromL(m,p,t)
        ELSE LMxRL(m,p,c,t)<=LMMin(m,p,c,t)*RFromL(m,p
        ,t)+YLM(m,p,t)*LMGrade(m,p,c,t-1)-LMMin(m,p,c,
        t)*YLM(m,p,t) ENDIF; } }
Variable BMxRB {
    IndexDomain: (m,p,c,t)|p in MineToProducts(m);
    Range: nonnegative;
    Definition: {
        IF t=FirstPeriod THEN BMInitial(m,p,c)*RFromB(
        m,p,t) ELSE BMxRB(m,p,c,t) ENDIF; } }
Constraint BMxRBConvex1 {
    IndexDomain: (m,p,c,t)|p in MineToProducts(m);
    Definition: {
        IF t=FirstPeriod THEN BMxRB(m,p,c,t)>=BMInitia
        l(m,p,c)*RFromB(m,p,t)
        ELSE BMxRB(m,p,c,t)>=BMMax(m,p,c,t)*RFromB(m,p
        ,t)+YLB(m,p,t)*BMGrade(m,p,c,t-1)-BMMax(m,p,c,
        t)*YLB(m,p,t) ENDIF; } }
Constraint BMxRBConvex2 {
    IndexDomain: (m,p,c,t)|p in MineToProducts(m);
    Definition: {
        IF t=FirstPeriod THEN BMxRB(m,p,c,t)>=BMInitia
        l(m,p,c)*RFromB(m,p,t)
        ELSE BMxRB(m,p,c,t)>=BMMin(m,p,c,t)*RFromB(m,p
        ,t) ENDIF; } }
Constraint BMxRBConcave1 {
```

```
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            IF t=FirstPeriod THEN BMxRB(m,p,c,t)<=BMInitia
            l(m,p,c)*RFromB(m,p,t)
            ELSE BMxRB(m,p,c,t)<=BMMin(m,p,c,t)*RFromB(m,p
            ,t)+YLB(m,p,t)*BMGrade(m,p,c,t-1)-BMMin(m,p,c,
            t)*YLB(m,p,t) ENDIF; } }
    Constraint BMxRBConcave2 {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            IF t=FirstPeriod THEN BMxRB(m,p,c,t)<=BMInitia
            l(m,p,c)*RFromB(m,p,t)
            ELSE BMxRB(m,p,c,t)<=BMMax(m,p,c,t)*RFromB(m,p
            ,t) ENDIF; } }
}
DeclarationSection Port_Grade_Constraints {
    Variable LPxW {
        IndexDomain: (r,s,c,t)|s in PortToProducts(r);
        Range: nonnegative;
        Definition: {
            IF s in LumpProducts THEN ZGxWBeforeZ(r,s,c,t)
            -ZGxZ(r,s,c,t)-LPxZLump(r,s,c,t) ELSE LPxW(r,s
            ,c,t) ENDIF; } }
    Constraint LPxWConvex1 {
        IndexDomain: (r,s,c,t)|s in PortToProducts(r);
        Definition: {
            LPxW(r,s,c,t)>=LPMax(r,s,c,t)*wLevel(r,s,t)+YL
            P(r,s,t)*LPGrade(r,s,c,t)-LPMax(r,s,c,t)*YLP(r
            ,s,t); } }
    Constraint LPxWConvex2 {
        IndexDomain: (r,s,c,t)|s in PortToProducts(r);
        Definition: {
            LPxW(r,s,c,t)>=LPMin(r,s,c,t)*wLevel(r,s,t);}}
    Constraint LPxWConcave1 {
        IndexDomain: (r,s,c,t)|s in PortToProducts(r);
        Definition: {
            LPxW(r,s,c,t)<=LPMax(r,s,c,t)*wLevel(r,s,t);}}
    Constraint LPxWConcave2 {
        IndexDomain: (r,s,c,t)|s in PortToProducts(r);
        Definition: {
            LPxW(r,s,c,t)<=LPMin(r,s,c,t)*wLevel(r,s,t)+YL
```

```
        P(r,s,t)*LPGrade(r,s,c,t)-LPMin(r,s,c,t)*YLP(r
        ,s,t); } }
Variable ZGxWBeforeZ {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Range: nonnegative;
    Definition: {
        IF t=FirstPeriod THEN LPInitial(r,s,c)*wInitia
        l(r,s)-ZGxUToBulk(r,s,c,t)+BPxUFromBulk(r,s,c,
        t)+sum((m,f,p,d in PortToDumpers(r)),RGxX(m,f,
        p,d,s,c,t))
        ELSE LPxW(r,s,c,t-1)-ZGxUToBulk(r,s,c,t)+BPxUF
        romBulk(r,s,c,t)+sum((m,f,p,d in PortToDumpers
        (r)),RGxX(m,f,p,d,s,c,t)) ENDIF; } }
Constraint ZGxWBeforeZConvex1 {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        ZGxWBeforeZ(r,s,c,t)>=LPMax(r,s,c,t)*wBeforeZ(
        r,s,t)+(YLP(r,s,t)+ZMax(r,t))*ZG(r,s,c,t)-LPMa
        x(r,s,c,t)*(YLP(r,s,t)+ZMax(r,t)); } }
Constraint ZGxWBeforeZConvex2 {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        ZGxWBeforeZ(r,s,c,t)>=LPMin(r,s,c,t)*wBeforeZ(
        r,s,t); } }
Constraint ZGxWBeforeZConcave1 {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        ZGxWBeforeZ(r,s,c,t)<=LPMax(r,s,c,t)*wBeforeZ(
        r,s,t); } }
Constraint ZGxWBeforeZConcave2 {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        ZGxWBeforeZ(r,s,c,t)<=LPMin(r,s,c,t)*wBeforeZ(
        r,s,t)+(YLP(r,s,t)+ZMax(r,t))*ZG(r,s,c,t)-LPMi
        n(r,s,c,t)*(YLP(r,s,t)+ZMax(r,t)); } }
Variable LPxWBeforeZPlusLump {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Range: nonnegative;
    Definition: {
        ZGxWBeforeZ(r,s,c,t)+sum(ss in LumpToFinePairs
        (s),LPxZLump(r,ss,c,t)); } }
```

```
Constraint LPxWBeforeZPlusLumpConvex1 {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        LPxWBeforeZPlusLump(r,s,c,t)>=LPMax(r,s,c,t)*w
        BeforeZPlusLump(r,s,t)+(YLP(r,s,t)+ZMax(r,t))*
        LPGrade(r,s,c,t)-LPMax(r,s,c,t)*(YLP(r,s,t)+ZM
        ax(r,t)); } }
Constraint LPxWBeforeZPlusLumpConvex2 {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        LPxWBeforeZPlusLump(r,s,c,t)>=LPMin(r,s,c,t)*w
        BeforeZPlusLump(r,s,t); } }
Constraint LPxWBeforeZPlusLumpConcave1 {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        LPxWBeforeZPlusLump(r,s,c,t)<=LPMax(r,s,c,t)*w
        BeforeZPlusLump(r,s,t); } }
Constraint LPxWBeforeZPlusLumpConcave2 {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        LPxWBeforeZPlusLump(r,s,c,t)<=LPMin(r,s,c,t)*w
        BeforeZPlusLump(r,s,t)+(YLP(r,s,t)+ZMax(r,t))*
        LPGrade(r,s,c,t)-LPMin(r,s,c,t)*(YLP(r,s,t)+ZM
        ax(r,t)); } }
Variable ZGxUToBulk {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Range: nonnegative; }
Constraint ZGxUToBulkConvex1 {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        ZGxUToBulk(r,s,c,t)>=LPMax(r,s,c,t)*uToBulk(r,
        s,t)+PortBulkOutTotalRates(r,t)*ZG(r,s,c,t)-LP
        Max(r,s,c,t)*PortBulkOutTotalRates(r,t); } }
Constraint ZGxUToBulkConvex2 {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        ZGxUToBulk(r,s,c,t)>=LPMin(r,s,c,t)*uToBulk(r,
        s,t); } }
Constraint ZGxUToBulkConcave1 {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
```

```
        ZGxUToBulk(r,s,c,t)<=LPMin(r,s,c,t)*uToBulk(r,
        s,t)+PortBulkOutTotalRates(r,t)*ZG(r,s,c,t)-LP
        Min(r,s,c,t)*PortBulkOutTotalRates(r,t); } }
Constraint ZGxUToBulkConcave2 {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        ZGxUToBulk(r,s,c,t)<=LPMax(r,s,c,t)*uToBulk(r,
        s,t); } }
Variable BPxV {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Range: nonnegative;
    Definition: {
        IF t=FirstPeriod THEN BPInitial(r,s,c)*vInitia
        l(r,s)-BPxUFromBulk(r,s,c,t)+ZGxUToBulk(r,s,c,
        t) ELSE BPxV(r,s,c,t-1)-BPxUFromBulk(r,s,c,t)+
        ZGxUToBulk(r,s,c,t) ENDIF; } }
Constraint BPxVConvex1 {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        BPxV(r,s,c,t)>=BPMax(r,s,c,t)*vLevel(r,s,t)+VM
        ax(r,s,t)*BPGrade(r,s,c,t)-BPMax(r,s,c,t)*VMax
        (r,s,t); } }
Constraint BPxVConvex2 {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        BPxV(r,s,c,t)>=BPMin(r,s,c,t)*vLevel(r,s,t);}}
Constraint BPxVConcave1 {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        BPxV(r,s,c,t)<=BPMin(r,s,c,t)*vLevel(r,s,t)+VM
        ax(r,s,t)*BPGrade(r,s,c,t)-BPMin(r,s,c,t)*VMax
        (r,s,t); } }
Constraint BPxVConcave2 {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        BPxV(r,s,c,t)<=BPMax(r,s,c,t)*vLevel(r,s,t);}}
Variable BPxUFromBulk {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Range: nonnegative; }
Constraint BPxUFromBulkConvex1 {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
```

```
    Definition: {
        BPxUFromBulk(r,s,c,t)>=BPMax(r,s,c,t)*uFromBul
        k(r,s,t)+PortBulkInTotalRates(r,t)*BPGrade(r,s
        ,c,t)-BPMax(r,s,c,t)*PortBulkInTotalRates(r,t)
        ; } }
Constraint BPxUFromBulkConvex2 {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        BPxUFromBulk(r,s,c,t)>=BPMin(r,s,c,t)*uFromBul
        k(r,s,t); } }
Constraint BPxUFromBulkConcave1 {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        BPxUFromBulk(r,s,c,t)<=BPMin(r,s,c,t)*uFromBul
        k(r,s,t)+PortBulkInTotalRates(r,t)*BPGrade(r,s
        ,c,t)-BPMin(r,s,c,t)*PortBulkInTotalRates(r,t)
        ; } }
Constraint BPxUFromBulkConcave2 {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        BPxUFromBulk(r,s,c,t)<=BPMax(r,s,c,t)*uFromBul
        k(r,s,t); } }
Variable ZGxZ {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Range: nonnegative; }
Constraint ZGxZConvex1 {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        ZGxZ(r,s,c,t)>=ZMax(r,t)*ZG(r,s,c,t)+LPMax(r,s
        ,c,t)*zPlusRF(r,s,t)*(1-PercentLump(r,s,t))-ZM
        ax(r,t)*LPMax(r,s,c,t); } }
Constraint ZGxZConvex2 {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        ZGxZ(r,s,c,t)>=LPMin(r,s,c,t)*zPlusRF(r,s,t)*(
        1-PercentLump(r,s,t)); } }
Constraint ZGxZConcave1 {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        ZGxZ(r,s,c,t)<=ZMax(r,t)*ZG(r,s,c,t)+LPMin(r,s
        ,c,t)*zPlusRF(r,s,t)*(1-PercentLump(r,s,t))-ZM
```

```
            ax(r,t)*LPMin(r,s,c,t); } }
        Constraint ZGxZConcave2 {
            IndexDomain: (r,s,c,t)|s in PortToProducts(r);
            Definition: {
                ZGxZ(r,s,c,t)<=LPMax(r,s,c,t)*zPlusRF(r,s,t)*(
                1-PercentLump(r,s,t)); } }
        Variable LPxZLump {
            IndexDomain: (r,s,c,t)|s in PortToProducts(r);
            Range: nonnegative; }
        Constraint LPxZLumpConvex1 {
            IndexDomain: (r,s,c,t)|s in PortToProducts(r);
            Definition: {
                LPxZLump(r,s,c,t)>=PercentLump(r,s,t)*ZMax(r,t
                )*LPGrade(r,s,c,t)+LPMax(r,s,c,t)*PercentLump(
                r,s,t)*zPlusRF(r,s,t)-PercentLump(r,s,t)*ZMax(
                r,t)*LPMax(r,s,c,t); } }
        Constraint LPxZLumpConvex2 {
            IndexDomain: (r,s,c,t)|s in PortToProducts(r);
            Definition: {
                LPxZLump(r,s,c,t)>=LPMin(r,s,c,t)*PercentLump(
                r,s,t)*zPlusRF(r,s,t); } }
        Constraint LPxZLumpConcave1 {
            IndexDomain: (r,s,c,t)|s in PortToProducts(r);
            Definition: {
                LPxZLump(r,s,c,t)<=PercentLump(r,s,t)*ZMax(r,t
                )*LPGrade(r,s,c,t)+LPMin(r,s,c,t)*PercentLump(
                r,s,t)*zPlusRF(r,s,t)-PercentLump(r,s,t)*ZMax(
                r,t)*LPMin(r,s,c,t); } }
        Constraint LPxZLumpConcave2 {
            IndexDomain: (r,s,c,t)|s in PortToProducts(r);
            Definition: {
                LPxZLump(r,s,c,t)<=LPMax(r,s,c,t)*PercentLump(
                r,s,t)*zPlusRF(r,s,t); } }
    }
    DeclarationSection Grade_Deviations {
        Parameter TG {
            IndexDomain: (s,c,t); }
        Parameter GradeTolerance {
            IndexDomain: (s,c); }
        Variable SI {
            IndexDomain: (r,s,c,t)|s in PortToProducts(r);
```

```
        Range: nonnegative; }
    Variable EI {
        IndexDomain: (r,s,c,t)|s in PortToProducts(r);
        Range: nonnegative; }
    Constraint GradeDeviationsMin {
        IndexDomain: (r,s,c,t)|s in PortToProducts(r);
        Definition: {
            (TG(s,c,t)-GradeTolerance(s,c))*zPlusRF(r,s,t)
            *(1-PercentLump(r,s,t))-SI(r,s,c,t)<=ZGxZ(r,s,
            c,t); } }
    Constraint GradeDeviationsMax {
        IndexDomain: (r,s,c,t)|s in PortToProducts(r);
        Definition: {
            ZGxZ(r,s,c,t)<=(TG(s,c,t)+GradeTolerance(s,c))
            *zPlusRF(r,s,t)*(1-PercentLump(r,s,t))+EI(r,s,
            c,t); } }
}
DeclarationSection SLP_Grade_Constraints {
    Constraint SLPMineLiveGrades {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            IF t=FirstPeriod THEN LMGrade(m,p,c,t)*OT_sLev
            el(m,p,t)=LMInitial(m,p,c)*sInitial(m,p)-LMGra
            de(m,p,c,t)*OT_MineBulkOut(m,p,t)+IOP(m,p,t)*I
            OPGrade(m,p,c,t)+BMGrade(m,p,c,t)*OT_MineBulkI
            n(m,p,t)-sum((f,d,s),RGrade(m,p,c,t)*TS(m,f,p,
            t)*OT_x(m,f,p,d,s,t))
            ELSE LMGrade(m,p,c,t)*OT_sLevel(m,p,t)=LMGrade
            (m,p,c,t-1)*OT_sLevel(m,p,t-1)-LMGrade(m,p,c,t
            )*OT_MineBulkOut(m,p,t)+IOP(m,p,t)*IOPGrade(m,
            p,c,t)+BMGrade(m,p,c,t)*OT_MineBulkIn(m,p,t)-s
            um((f,d,s),RGrade(m,p,c,t)*TS(m,f,p,t)*OT_x(m,
            f,p,d,s,t)) ENDIF; } }
    Constraint SLPMineBulkGrades {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            IF t=FirstPeriod THEN BMGrade(m,p,c,t)*OT_bLev
            el(m,p,t)=BMInitial(m,p,c)*bInitial(m,p)-BMGra
            de(m,p,c,t)*OT_MineBulkIn(m,p,t)+LMGrade(m,p,c
            ,t)*OT_MineBulkOut(m,p,t)
            ELSE BMGrade(m,p,c,t)*OT_bLevel(m,p,t)=BMGrade
```

```
        (m,p,c,t-1)*OT_bLevel(m,p,t-1)-BMGrade(m,p,c,t
        )*OT_MineBulkIn(m,p,t)+LMGrade(m,p,c,t)*OT_Min
        eBulkOut(m,p,t) ENDIF; } }
Constraint SLPRailedGrades {
    IndexDomain: (m,p,c,t)|p in MineToProducts(m);
    Definition: {
        IF t=FirstPeriod THEN sum((f,d,s),RGrade(m,p,c
        ,t)*TS(m,f,p,t)*OT_x(m,f,p,d,s,t))=SLP_RFromIO
        P(m,p,t)*IOPGrade(m,p,c,t)+LMInitial(m,p,c)*SL
        P_RFromL(m,p,t)+BMinitial(m,p,c)*SLP_RFromB(m,
        p,t)
        ELSE sum((f,d,s),RGrade(m,p,c,t)*TS(m,f,p,t)*O
        T_x(m,f,p,d,s,t))=SLP_RFromIOP(m,p,t)*IOPGrade
        (m,p,c,t)+LMGrade(m,p,c,t-1)*SLP_RFromL(m,p,t)
        +BMGrade(m,p,c,t-1)*SLP_RFromB(m,p,t) ENDIF;}}
Constraint SLPPortLiveGrades {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        IF t=FirstPeriod THEN ZG(r,s,c,t)*OT_wBeforeZ(
        r,s,t)=LPInitial(r,s,c)*wInitial(r,s)-ZG(r,s,c
        ,t)*OT_PortBulkOut(r,s,t)+BPGrade(r,s,c,t)*OT_
        PortBulkIn(r,s,t)+sum((m,f,p,d in PortToDumper
        s(r)),RGrade(m,p,c,t)*TS(m,f,p,t)*OT_x(m,f,p,d
        ,s,t))
        ELSE ZG(r,s,c,t)*OT_wBeforeZ(r,s,t)=LPGrade(r,
        s,c,t-1)*OT_wLevel(r,s,t-1)-ZG(r,s,c,t)*OT_Por
        tBulkOut(r,s,t)+BPGrade(r,s,c,t)*OT_PortBulkIn
        (r,s,t)+sum((m,f,p,d in PortToDumpers(r)),RGra
        de(m,p,c,t)*TS(m,f,p,t)*OT_x(m,f,p,d,s,t))
        ENDIF; } }
Constraint SLPPortLiveGradesForFines {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        LPGrade(r,s,c,t)*OT_wBeforeZPlusLump(r,s,t)=ZG
        (r,s,c,t)*OT_wBeforeZ(r,s,t)+sum(ss in LumpToF
        inePairs(s),LPGrade(r,ss,c,t)*PercentLump(r,ss
        ,t)*OT_zPlusRF(r,ss,t)); } }
Constraint SLPPortBulkGrades {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        IF t=FirstPeriod THEN BPGrade(r,s,c,t)*OT_vLev
```

```
            el(r,s,t)=BPInitial(r,s,c)*vInitial(r,s)-BPGra
            de(r,s,c,t)*OT_PortBulkIn(r,s,t)+ZG(r,s,c,t)*O
            T_PortBulkOut(r,s,t)
            ELSE BPGrade(r,s,c,t)*OT_vLevel(r,s,t)=BPGrade
            (r,s,c,t-1)*OT_vLevel(r,s,t-1)-BPGrade(r,s,c,t
            )*OT_PortBulkIn(r,s,t)+ZG(r,s,c,t)*OT_PortBulk
            Out(r,s,t) ENDIF;  }
    Constraint SLPGradeDeviationsMin {
        IndexDomain: (r,s,c,t)|s in PortToProducts(r);
        Definition: {
            (TG(s,c,t)-GradeTolerance(s,c))*OT_zLevel(r,s,
            t)-SI(r,s,c,t)<=ZG(r,s,c,t)*OT_zLevel(r,s,t);
        } }
    Constraint SLPGradeDeviationsMax {
        IndexDomain: (r,s,c,t)|s in PortToProducts(r);
        Definition: {
            ZG(r,s,c,t)*OT_zLevel(r,s,t)<=(TG(s,c,t)+Grade
            Tolerance(s,c))*OT_zLevel(r,s,t)+EI(r,s,c,t);
        } }
}
DeclarationSection Mine_Output_Variables {
    Parameter OT_sLevel {
        IndexDomain: (m,p,t)|p in MineToProducts(m);
        Definition: {
            sLevel(m,p,t); } }
    Parameter OT_bLevel {
        IndexDomain: (m,p,t)|p in MineToProducts(m);
        Definition: {
            bLevel(m,p,t); } }
    Parameter OT_MineBulkIn {
        IndexDomain: (m,p,t);
        Definition: {
            yFromBulk(m,p,t); } }
    Parameter OT_MineBulkOut {
        IndexDomain: (m,p,t);
        Definition: {
            yToBulk(m,p,t); } }
}
DeclarationSection Port_Output_Variables {
    Parameter OT_wLevel {
        IndexDomain: (r,s,t)|s in PortToProducts(r);
```

```
        Definition: {
            wLevel(r,s,t); } }
    Parameter OT_wBeforeZ {
        IndexDomain: (r,s,t);
        Definition: {
            wBeforeZ(r,s,t); } }
    Parameter OT_wBeforeZPlusLump {
        IndexDomain: (r,s,t);
        Definition: {
            wBeforeZPlusLump(r,s,t); } }
    Parameter OT_vLevel {
        IndexDomain: (r,s,t)|s in PortToProducts(r);
        Definition: {
            vLevel(r,s,t); } }
    Parameter OT_PortBulkIn {
        IndexDomain: (r,s,t);
        Definition: {
            uFromBulk(r,s,t); } }
    Parameter OT_PortBulkOut {
        IndexDomain: (r,s,t);
        Definition: {
            uToBulk(r,s,t); } }
    Parameter OT_zLevel {
        IndexDomain: (r,s,t)|s in PortToProducts(r);
        Definition: {
            zPlusRF(r,s,t)*(1-PercentLump(r,s,t)); } }
    Parameter OT_zPlusRF {
        IndexDomain: (r,s,t);
        Definition: {
            zPlusRF(r,s,t); } }
    Parameter OT_ReturnFines {
        IndexDomain: (r,s,t);
        Definition: {
            PercentLump(r,s,t)*OT_zPlusRF(r,s,t); } }
}
DeclarationSection Rail_Output_Variables {
    Parameter OT_x {
        IndexDomain: (m,f,p,d,s,t)|p in MineToProducts(m)
            and s in MinePortToProducts(m,p) and d in Mine
            ToDumpers(m,p);
        Definition: {
```

```
            x(m,f,p,d,s,t); } }
    Parameter OT_RFromIOP {
        IndexDomain: (m,p,t);
        Definition: {
            RFromIOP(m,p,t); } }
    Parameter OT_RFromL {
        IndexDomain: (m,p,t);
        Definition: {
            RFromL(m,p,t); } }
    Parameter OT_RFromB {
        IndexDomain: (m,p,t);
        Definition: {
            RFromB(m,p,t); } }
}
DeclarationSection Grade_Output_Variables {
    Parameter OT_LMGrade {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            LMGrade(m,p,c,t); } }
    Parameter OT_sGrade {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            IF OT_sLevel(m,p,t)=0 THEN 0 ELSE RealLM(m,p,c
            ,t) ENDIF; } }
    Parameter OT_MineBulkOutGrade {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            IF OT_MineBulkOut(m,p,t)=0 THEN 0 ELSE RealLM(
            m,p,c,t) ENDIF; } }
    Parameter OT_BMGrade {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            BMGrade(m,p,c,t); } }
    Parameter OT_bGrade {
        IndexDomain: (m,p,c,t);
        Definition: {
            RealBM(m,p,c,t); } }
    Parameter OT_MineBulkInGrade {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Definition: {
            IF OT_MineBulkIn(m,p,t)=0 THEN 0 ELSE RealBM(m
```

```
        ,p,c,t) ENDIF; } }
Parameter OT_LPGrade {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        LPGrade(r,s,c,t); } }
Parameter OT_wGrade {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        IF OT_wLevel(r,s,t)=0 THEN 0 ELSE RealLP(r,s,c
        ,t) ENDIF; } }
Parameter OT_PortBulkOutGrade {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        IF OT_PortBulkOut(r,s,t)=0 THEN 0 ELSE RealLP(
        r,s,c,t) ENDIF; } }
Parameter OT_BPGrade {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        BPGrade(r,s,c,t); } }
Parameter OT_vGrade {
    IndexDomain: (r,s,c,t);
    Definition: {
        RealBP(r,s,c,t); } }
Parameter OT_PortBulkInGrade {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        IF OT_PortBulkIn(r,s,t)=0 THEN 0 ELSE RealBP(r
        ,s,c,t) ENDIF; } }
Parameter OT_RGrade {
    IndexDomain: (m,p,c,t)|p in MineToProducts(m);
    Definition: {
        RGrade(m,p,c,t); } }
Parameter OT_xGrade {
    IndexDomain: (m,p,c,t)|p in MineToProducts(m);
    Definition: {
        IF sum((f,d,s),OT_x(m,f,p,d,s,t))=0 THEN 0
        ELSE RealRG(m,p,c,t) ENDIF; } }
Parameter OT_zGrade {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        RealZG(r,s,c,t); } }
```

```
    Parameter OT_GradeDev {
        IndexDomain: (r,s,c,t);
        Definition: {
            IF OT_zLevel(r,s,t)=0 THEN 0 ELSE TG(s,c,t)-OT
            _zGrade(r,s,c,t) ENDIF; } }
    Parameter OT_ZGxZ {
        IndexDomain: (r,s,c,t);
        Definition: {
            ZGxZ(r,s,c,t); } }
}
DeclarationSection Objective_Function_Output {
    Parameter OT_GradeDeviationsPenalty {
        Definition: {
            GradeDeviationsPenalty ; } }
}
DeclarationSection Real_Grades {
    Variable RealLM {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Range: free; }
    Constraint LMConstraint {
        IndexDomain: (m,p,c,t);
        Definition: {
            IF t=FirstPeriod THEN RealLM(m,p,c,t)*OT_sLeve
            l(m,p,t)=LMInitial(m,p,c)*sInitial(m,p)-RealLM
            (m,p,c,t)*OT_MineBulkOut(m,p,t)+IOP(m,p,t)*IOP
            Grade(m,p,c,t)+RealBM(m,p,c,t)*OT_MineBulkIn(m
            ,p,t)-sum((f,d,s),RealRG(m,p,c,t)*TS(m,f,p,t)*
            OT_x(m,f,p,d,s,t))
            ELSE RealLM(m,p,c,t)*OT_sLevel(m,p,t)=RealLM(m
            ,p,c,t-1)*OT_sLevel(m,p,t-1)-RealLM(m,p,c,t)*O
            T_MineBulkOut(m,p,t)+IOP(m,p,t)*IOPGrade(m,p,c
            ,t)+RealBM(m,p,c,t)*OT_MineBulkIn(m,p,t)-sum((
            f,d,s),RealRG(m,p,c,t)*TS(m,f,p,t)*OT_x(m,f,p,
            d,s,t)) ENDIF;  }
    Variable RealBM {
        IndexDomain: (m,p,c,t)|p in MineToProducts(m);
        Range: free; }
    Constraint BMConstraint {
        IndexDomain: (m,p,c,t);
        Definition: {
            IF t=FirstPeriod THEN RealBM(m,p,c,t)*OT_bLeve
```

```
        l(m,p,t)=BMInitial(m,p,c)*bInitial(m,p)-RealBM
        (m,p,c,t)*OT_MineBulkIn(m,p,t)+RealLM(m,p,c,t)
        *OT_MineBulkOut(m,p,t)
        ELSE RealBM(m,p,c,t)*OT_bLevel(m,p,t)=RealBM(m
        ,p,c,t-1)*OT_bLevel(m,p,t-1)-RealBM(m,p,c,t)*O
        T_MineBulkIn(m,p,t)+RealLM(m,p,c,t)*OT_MineBul
        kOut(m,p,t) ENDIF; } }
Variable RealRG {
    IndexDomain: (m,p,c,t)|p in MineToProducts(m);
    Range: free; }
Constraint RGConstraint {
    IndexDomain: (m,p,c,t);
    Definition: {
        IF t=FirstPeriod THEN sum((f,d,s),RealRG(m,p,c
        ,t)*TS(m,f,p,t)*OT_x(m,f,p,d,s,t))=OT_RFromIOP
        (m,p,t)*IOPGrade(m,p,c,t)+LMInitial(m,p,c)*OT_
        RFromL(m,p,t)+BMinitial(m,p,c)*OT_RFromB(m,p,t
        ) ELSE sum((f,d,s),RealRG(m,p,c,t)*TS(m,f,p,t)
        *OT_x(m,f,p,d,s,t))=OT_RFromIOP(m,p,t)*IOPGrad
        e(m,p,c,t)+RealLM(m,p,c,t-1)*OT_RFromL(m,p,t)+
        RealBM(m,p,c,t-1)*OT_RFromB(m,p,t) ENDIF; } }
Variable RealLP {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Range: free;
    Definition: {
        IF s in LumpProducts THEN RealZG(r,s,c,t)
        ELSE RealLP(r,s,c,t) ENDIF; } }
Constraint LPConstraint {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        RealLP(r,s,c,t)*OT_wBeforeZPlusLump(r,s,t)=Rea
        lZG(r,s,c,t)*OT_wBeforeZ(r,s,t)+sum(ss in Lump
        ToFinePairs(s),RealLP(r,ss,c,t)*PercentLump(r,
        ss,t)*OT_zPlusRF(r,ss,t)); } }
Variable RealBP {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Range: free; }
Constraint BPConstraint {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        IF t=FirstPeriod THEN RealBP(r,s,c,t)*OT_vLeve
```

```
        l(r,s,t)=BPInitial(r,s,c)*vInitial(r,s)-RealBP
        (r,s,c,t)*OT_PortBulkIn(r,s,t)+RealZG(r,s,c,t)
        *OT_PortBulkOut(r,s,t)
        ELSE RealBP(r,s,c,t)*OT_vLevel(r,s,t)=RealBP(r
        ,s,c,t-1)*OT_vLevel(r,s,t-1)-RealBP(r,s,c,t)*O
        T_PortBulkIn(r,s,t)+RealZG(r,s,c,t)*OT_PortBul
        kOut(r,s,t) ENDIF; } }
Variable RealZG {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Range: free; }
Constraint ZGConstraint {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        IF t=FirstPeriod THEN RealZG(r,s,c,t)*OT_wBefo
        reZ(r,s,t)=LPInitial(r,s,c)*wInitial(r,s)-Real
        ZG(r,s,c,t)*OT_PortBulkOut(r,s,t)+RealBP(r,s,c
        ,t)*OT_PortBulkIn(r,s,t)+sum((m,f,p,d in PortT
        oDumpers(r)),RealRG(m,p,c,t)*TS(m,f,p,t)*OT_x(
        m,f,p,d,s,t))
        ELSE RealZG(r,s,c,t)*OT_wBeforeZ(r,s,t)=RealLP
        (r,s,c,t-1)*OT_wLevel(r,s,t-1)-RealZG(r,s,c,t)
        *OT_PortBulkOut(r,s,t)+RealBP(r,s,c,t)*OT_Port
        BulkIn(r,s,t)+sum((m,f,p,d in PortToDumpers(r)
        ),RealRG(m,p,c,t)*TS(m,f,p,t)*OT_x(m,f,p,d,s,t
        )) ENDIF; } }
Variable RealSI {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Range: nonnegative; }
Constraint RealGradeDevMin {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        (TG(s,c,t)-GradeTolerance(s,c))*OT_zLevel(r,s,
        t)-RealSI(r,s,c,t)<=RealZG(r,s,c,t)*OT_zLevel(
        r,s,t); } }
Variable RealEI {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Range: nonnegative; }
Constraint RealGradeDevMax {
    IndexDomain: (r,s,c,t)|s in PortToProducts(r);
    Definition: {
        RealZG(r,s,c,t)*OT_zLevel(r,s,t)<=(TG(s,c,t)+
```

```
                GradeTolerance(s,c))*OT_zLevel(r,s,t)+RealEI(r
                ,s,c,t); } }
        Variable RealGradePenalty {
            Range: free;
            Definition: {
                sum((r,s,c,t)|s in PortToProducts(r),GPI(s,c)*
                (RealEI(r,s,c,t)+RealSI(r,s,c,t))); } }
        Parameter OT_RealGradePenalty {
            Definition: {
                RealGradePenalty; } }
        Parameter RealProfit {
            Definition: {
                Profit-RealGradePenalty+OT_GradeDeviationsPen
                alty; } }
    }

    Procedure MainInitialization;
    Procedure M0_WithoutGrades {
        Body: {
            solve Model_WithoutGrades where MIP_Relative_Optim
            ality_Tolerance:=(relative_optimality_gap);; } }
    Procedure M1_WithoutGradesNoInteger {
        Body: {
            solve Model1_WithoutGradesNoInteger where MIP_Rela
            tive_Optimality_Tolerance:=(relative_optimality_ga
            p);; } }
    Procedure M1_WithGradesNoInteger {
        Body: {
            solve Model1_WithGradesNoInteger where MIP_Relativ
            e_Optimality_Tolerance:=(relative_optimality_gap);
            ; } }
    Procedure M1_WithGrades {
        Body: {
            solve Model1_WithGrades where MIP_Relative_Optimal
            ity_Tolerance:=(relative_optimality_gap);; } }
    Procedure M2_WithGrades1 {
        Body: {
            solve Model2_WithGrades1 where MIP_Relative_Optima
            lity_Tolerance:=(relative_optimality_gap);; } }
    Procedure M2_WithGrades2 {
        Body: {
```

```
            solve Model2_WithGrades2 where MIP_Relative_Optima
            lity_Tolerance:=(relative_optimality_gap);; } }
        Procedure MainTermination {
        Body: {
            return DataManagementExit(); } }
}
```

Index