Journal of Global Optimization **27:** 349–365, 2003.

An Application of Branch and Cut to Open Pit Mine Scheduling

LOUIS CACCETTA and STEPHEN P. HILL
School of Mathematics and Statistics, Curtin University of Technology, GPO Box U1987, PERTH, Western Australia, 6845 (E-mail: caccetta@maths.curtin.edu.au, hillsp@maths.curtin.edu.au)

Abstract. The economic viability of the modern day mine is highly dependent upon careful planning and management. Declining trends in average ore grades, increasing mining costs and environmental considerations will ensure that this situation will remain in the foreseeable future. The operation and management of a large open pit mine having a life of several years is an enormous and complex task. Though a number of optimization techniques have been successfully applied to resolve some important problems, the problem of determining an optimal production schedule over the life of the deposit is still very much unresolved. In this paper we will critically examine the techniques that are being used in the mining industry for production scheduling indicating their limitations. In addition, we present a mixed integer linear programming model for the scheduling problems along with a Branch and Cut solution strategy. Computational results for practical sized problems are discussed.

Key words: Branch and Cut, Mixed Integer Linear Programming, Mine Scheduling, Optimization

1. Introduction

The operation and management of a large open pit mine is an enormous and complex task, particularly for mines having a life of many years. Optimization techniques can be successfully applied to resolve a number of important problems that arise in the planning and management of a mine. These applications include: ore-body modelling and ore reserve estimation; the design of optimum pits; the determination of optimal production schedules; the determination of optimal operating layouts; the determination of optimal blends; the determination of equipment maintenance and replacement policies; and many more (Caccetta and Giannini, [7–9]).

A fundamental problem in mine planning is that of determining the optimum ultimate pit limit of a mine. The optimum ultimate pit limit of a mine is defined to be that contour which is the result of extracting the volume of material which provides the total maximum profit whilst satisfying the operational requirement of safe wall slopes. The ultimate pit limit gives the shape of the mine at the end of its life. Usually this contour is smoothed to produce the final pit outline.

Optimum pit design plays a major role in all stages of the life of an open pit: at the *feasibility study stage* when there is a need to produce a whole-of-life pit design; at the *operating phase* when pits need to be developed to respond to changes in metal prices, costs, ore reserves, and wall slopes; and towards the *end of a mine's life* where the final pit design may allow the economic termination of a project. At all stages there is a need for constant monitoring of the optimum pit, to facilitate the best long-term, medium-term and short-term mine planning and subsequent exploitation of the reserve. The optimum pit and mine planning are dynamic concepts requiring constant review. Thus the pit optimization technique should be regarded as a powerful and necessary management tool. Further, the pit optimization method must be highly efficient to allow for an effective sensitivity analysis. In practice one needs to construct a whole spectrum of pits, each corresponding to a specific set of parameters.

The ultimate pit limit problem has been efficiently solved using the Lerchs-Grossmann [28] graph theoretic algorithm or Picard's [33] network flow method (see also Caccetta and Giannini [7,8]). These methods are based on the "block model" of an orebody; the block model is detailed in the next section. A comparative analysis of the two methods is given by Caccetta et al [10]. Optimum pit design plays an important role in mine scheduling.

The open pit mine production scheduling problem can be defined as specifying the sequence in which "blocks" should be removed from the mine in order to maximise the total discounted profit from the mine subject to a variety of physical and economic constraints. Typically, the constraints relate to: the mining extraction sequence; mining, milling and refining capacities; grades of mill feed and concentrates; and various operational requirements such as minimum pit bottom width. The scheduling problem can be formulated as a mixed integer linear program (MILP). However, in real applications this formulation is too large, in terms of both the number of variables and the number of constraints, to solve by any available MILP software.

Several approaches to the scheduling problem have appeared in the literature including: heuristics (Caccetta et al [14] and Gershon [23]); Lagrangian relaxation (Caccetta et al [14]); parametric methods (Dagdelen and Johnson [17], Francois-Bongarcon and Guibal [6], Matheron [29,30] and Whittle [37,38]); dynamic programming techniques (Tolwinski and Underwood [36]); mixed integer linear programming (Caccetta et al [11,14]; Dagdelen and Johnson [17] and Gershon [22]); and the application of artificial intelligence algorithms such as simulated annealing, genetic algorithms (Denby and Schofield [18]) and neural networks (Denby et al [19]). Because of the complexity and size of the problem all these approaches suffer from one or more of the following limitations: cannot cater for most of the constraints that arise; yield only suboptimal solutions and in most cases without a quality measure; can only handle small sized problems.

In this paper we will present the results of our efforts to produce a computational method that incorporates all the constraints in the optimization and yields provably good solutions for reasonably large size problems. Given that mining represents some 4% of the worlds net GDP, our more accurate modelling and solution of practical mining problems has significant economic impact. We give a MILP formulation of the scheduling problem and present a Branch and Cut procedure for its solution. This is done in Section 3. Computational results are discussed in Section 4. The next section provides details of the block model used as well as a critical account of the various techniques that are used in the mining industry. We only discuss the more promising methods that have been applied to real mines.

2. Preliminaries

In this section we outline some of the methods that have been proposed for various mine development problems. We begin with the basic block model of an ore body, then we present a mixed integer linear programming formulation of the scheduling problem and discuss a number of algorithms that have been proposed for its solution. We focus mainly on methods which have proved useful in the mining industry. For a discussion of other methods we refer to Kim [27] and Thomas [34,35].

2.1. BLOCK MODEL

An early task in mine management is the establishment of an accurate model for the deposit. Though a number of models are available, the regular 3D fixed-block model is the most commonly used and is the best suited to the application of computerized optimization techniques (Gignac [24] and Kim [27]). This model is based on the ore body being divided into fixed-size blocks. The block dimensions are dependent on the physical characteristics of the mine, such as pit slopes, dip of deposit and grade variability as well as the equipment used. The centre of each block is assigned, based on drill hole data and a numerical technique, a grade representation of the whole block. The numerical technique used is some grade extension method such as: distance weighted interpolations, repression analysis, weighted moving averages and kriging (Gignac [24]). Using the financial and metallurgical data the net profit of each block is determined.

The wall slope requirements for each block are described by a set (typically 4 to 8) of azimuth-dip pairs. From these we can identify for each block x the set of blocks S_x which must be removed before block x can be mined. This collection of blocks, $x \cup S_x$, is usually referred to as a “cone”.

The key assumptions in the block model are: the cost of mining each block does not depend on the sequence of mining; and the desired wall slopes and pit shape can be approximated by the removed blocks.

2.2. THE SCHEDULING PROBLEM

The open pit mine production scheduling problem can be defined as specifying the sequence in which blocks should be removed from the mine in order to maximize the total discounted profit from the mine subject to a variety of constraints. The constraints may involve the following:

- mill throughput (mill feed and mill capacity)
- volume of material extracted per period
- blending constraints
- stockpile related constraints
- logistic constraints

We now present a simple mixed integer linear programming (MILP) formulation that incorporates the mill throughput and volume of material extracted constraints. We begin with some notation. Let

T is the number of periods over which the mine is being scheduled.
N is the total number of blocks in the ore body.
c_i^t is the profit (in NPV sense) resulting from the mining of block i in period t.
O is the set of ore blocks.
W is the set of waste blocks.
t_i is the tonnage of block i.
m^t is the tonnage of ore milled in period t.
S_i the set of blocks that must be removed prior to the mining of block i.

$$x_i^t = \begin{cases} 1, & \text{if block } i \text{ is mined in periods to } t \\ 0, & \text{otherwise.} \end{cases}$$

ℓ_0^t lower bound on the amount of ore that is milled in period t.
u_0^t upper bound on the amount of ore that is milled in period t.
u_w^t upper bound on the amount of waste that is mined in period t.
Then the MILP formulation is:

$$\text{Maximize } Z = \sum_{t=2}^{T} \sum_{i=1}^{N} \left(c_i^{t-1} - c_i^t\right) x_i^{t-1} + \sum_{i=1}^{N} c_i^T x_i^T \tag{2.1}$$

$$\text{subject to} \quad \sum_{i \in O} t_i x_i^1 - m^1 = 0 \tag{2.2}$$

$$\sum_{i \in O} t_i \left(x_i^t - x_i^{t-1}\right) - m^t = 0, \quad t = 2, 3, ..., T. \tag{2.3}$$

$$\sum_{i \in W} t_i x_i^1 \leqslant u_w^1 \tag{2.4}$$

$$\sum_{i \in W} t_i \left(x_i^t - x_i^{t-1}\right) \leqslant u_w^t, \quad t = 2, 3, ..., T. \tag{2.5}$$

$$x_i^{t-1} \leqslant x_i^t, \quad t = 2, 3, ..., T. \tag{2.6}$$

$$x_i^t \leqslant x_j^t, \; t = 1, 2, ..., T, \; j \in S_i; \; i = 1, 2, ..., N. \tag{2.7}$$

$$\ell_0^t \leqslant m^t \leqslant u_0^t, \quad t = 1, 2, ..., T. \tag{2.8}$$

$$x_i^t = 0, 1, \quad \text{for all } i, t. \tag{2.9}$$

Constraints (2.2), (2.3) and (2.8) ensure that the milling capacities hold. Constraints (2.4) and (2.5) ensure that the tonnage of waste removed does not exceed the prescribed upper bounds. Constraints (2.6) ensures that a block is removed in one period only. Constraints (2.7) are the wall slope restrictions.

The above formulation has NT $0-1$ variables, and $(N+2)T + N(d-1)$ linear constraints, where d is the average number of elements in a cone. Typically T is around 10, N is 100,000 for a small pit and over 1,000,000 for a larger pit. Consequently the MILP's that arise are much too large for direct application of commercial packages. However, as we demonstrate in this paper, the structure of the problem can be exploited to develop computational strategies that produce provably good solutions.

Solving MILP's such as (2.1) – (2.9) is a difficult and challenging task. Indeed, in the mining context, the lack of an immediate optimization technique has led the mining industry to focus on easy subproblems. Consequently the schedules that are generated, usually manually, are often outside the specified operating range and certainly far from optimal. The usual approach is to first determine the final pit outline and then through a series of refinements mining schedules are generated.

The final pit outline is determined by smoothing the contour produced by solving the ultimate pit limit problem. The ultimate pit limit is the maximum value pit resulting from the mining of ore and waste blocks under the assumption that all mining could be done in one period. That is, the solution to the problem (2.1) subject to (2.7) with $T = 1$ and (2.9).

The ultimate pit limit problem can be solved using the Lerchs-Grossmann graph theoretic algorithm [28] or by solving Picard's [33] network flow formulation. Over the past 10 to 15 years efficient packages have become available for solving this problem (Caccetta et al. [10,11]; Whittle [37,38]). Prior to this, the Moving Cone Technique was widely used because of its simplicity. This technique basically selects a block x for mining providing the total profit from blocks contained in the cone $x \cup S_x$ is positive. Whilst the method is extremely simple, it is easy to show that solutions far from optimal can be obtained. A number of refinements to the technique have been proposed (Yamatomi et al. [40]).

The Lerchs-Grossmann Algorithm (LGA) provided an important tool for mine design. However, as time is not an input parameter, its use for scheduling is restricted to mines having a very short life (up to 3 years). In the following we detail various approaches to mine scheduling.

2.3. ALGORITHMS

We begin by showing that the strategy outlined above for obtaining the production schedule of a mine by first determining the final pit outline and then generating the schedule is sound.

Let

$\mathcal{C}_u$: contour generated by the application of the Lerchs-Grossman algorithm (LGA).

$\mathcal{C}_s$: final contour generated by an optimal schedule.

We note that the LGA generates a contour $\mathcal{C}_u$ with minimal number of blocks. We assume that the contour $\mathcal{C}_s$ also has a minimal number of blocks. Note that the $\mathcal{C}_s$ contains all blocks mined over the T time periods, that is:

$$\mathcal{C}_s = \bigcup_{i=1}^{T} \{P_i : P_i \text{ set of blocks mined in period } i\}.$$

THEOREM. *Consider an open pit mine in which all constraints have a non-negative upper bound and a zero lower bound. Then $\mathcal{C}_s \subseteq \mathcal{C}_u$.*

Proof. Suppose to the contrary that $\mathcal{C}_s \nsubseteq \mathcal{C}_u$. Let $\mathcal{C}_1 = \mathcal{C}_s \backslash \mathcal{C}_u$ be the set of blocks mined under the optimal schedule that do not lie in $\mathcal{C}_u$. Consider

$$\mathcal{C}_2 = \mathcal{C}_s \backslash \mathcal{C}_1 = \mathcal{C}_s \cap \mathcal{C}_u,$$

the set of blocks mined under the optimal schedule that lie in $\mathcal{C}_u$. Observe that $\mathcal{C}_2$ is a feasible pit. Further,

$$Z(\mathcal{C}_s) = Z(\mathcal{C}_1) + Z(\mathcal{C}_2),$$

where $Z(\mathcal{C}_i)$ is the total value of the blocks in contour $\mathcal{C}_i$, $i = 1, 2$. The minimality of $\mathcal{C}_s$ implies that $Z(\mathcal{C}_i) > 0$ for each i. Now consider the contour

$$\mathcal{C}' = \mathcal{C}_u \cup \mathcal{C}_1.$$

Observe that $\mathcal{C}'$ satisfies the wall slope restrictions and consequently is a feasible contour in terms of the ultimate pit limit problem. Further, since $\mathcal{C}_u \cap \mathcal{C}_1 = \phi$, contour $\mathcal{C}'$ has a total value of

$$Z(\mathcal{C}') = Z(\mathcal{C}_u) + Z(\mathcal{C}_1) > Z(\mathcal{C}_u),$$

a contradiction. This proves that $\mathcal{C}_s \subseteq \mathcal{C}_u$.

As mentioned in the previous subsection, good commercial packages are available for obtaining the contour $\mathcal{C}_u$. The proof of the above result allows us to reduce the block model that needs to be considered. This leads to a reduction in the number of variables in (2.1) – (2.9). Below we detail approaches that attempt to determine the production schedule.

2.3.1. *Parameterization Method*

In their paper Lerchs and Grossmann [28] introduced the concept of parametric analysis in order to generate an extraction sequence. They considered the undiscounted model and varied the economic value of each block i from c_i to $(c_i - \lambda)$ for varying $\lambda \geqslant 0$. An increasing sequence of λ values gives rise to a set of nested pits. These pits can be used to produce a production schedule. Since this early work a number of authors have considered the implementation aspects of this method and its variations (Francois-Bongarcon and Guibal [6], Caccetta et al. [11,14], Caléou [15], Dagdelen and Johnson [17], Matheron [29,30], and Whittle [37,38]).

The mostly widely used scheduling software package that is based on parameterization, is Whittle's Four-D and Four-X [37,38]; the latter allows for multiple ore types in the calculation of block costs. The parameter used in theses packages is referred to as the "metal cost of mining" which is defined as: extraction cost ($/ton)/selling price ($/gm). This quantity provides an indication of the amount of product that must be sold to cover the cost of extracting one ton of material. The rationale for using this parameter is that the three components for calculating the block values (selling price ($/unit), processing cost ($/ton) and extraction cost ($/ton)) reduce to one factor under the assumption that the ratio of processing and extraction costs is constant. Whittle [37,38] suggests that a "best" mining schedule comes from extracting each of the nested pits in turn and a "worst" schedule comes from extracting the ore bench by bench.

The advantages of the Whittle approach include:

- the nested pits can be determined efficiently as each requires the solution of an ultimate pit limit problem.
- the identification of clusters of high grade ore in the model.
- a measure in the design of the final pit contour subject to a change in price and thus some sensitivity analysis can be performed.

The disadvantages of the Whittle approach includes:

- time and other variable factors (for example, extraction rate, different ore types, blending, etc.) are only implicitly included in the optimization through modifying the cost function.
- the possibility of a large increment in the size of the pit from one nested pit to the next. This is referred to as the "gapping problem" and it arises because there is no clear method for choosing the values of λ.
- optimality is not guaranteed. Indeed the "best" schedule may not even provide an upper bound for the NPV of the mine. This is easily seen by noting the nested pits produced by the LGA cannot have waste at the bottom whereas those produced by an optimal schedule can.
- the validity of the underlying assumption that the rate of processing and extractor costs are constant. Indeed, Whittle [37] states that variation of $\pm$ 20% over 5 years is typical.

- the necessity of reducing time costs to a cost per ton basis. Making assumptions about the production rates, these costs are determined iteratively until a "reasonable" solution is found.
- structural constrains such as mining to a maximum vertical depth or a minimum mining width, cannot be incorporated into the parameterization model.
- the extraction sequence may not satisfy the production requirements of the mine.

Recently, in an attempt to address the spacial requirements, Whittle [38] introduced the Milawa Algorithm which given a set of nested pits produces a revised schedule with the aim of improving the NPV.

Another commercially available package which extends the nested pit approach is the Earthworks NPV Scheduler [20]. This package first generates the nested pits and then using these, the pushbacks are defined heuristically. The criteria for the pushbacks is to keep them as close as possible to the extraction sequence suggested by the nested pits taking into consideration equipment access. Finally, a restricted tree search procedure is used to resequence the pushback removal to increase the NPV. A major advantage of this package is that it may produce schedules that are more likely to be acceptable to mining engineers because practical spacial constraints are taken into acccount when defining the pushbacks.

Unfortunately, all methods that use the above nested pits approach in a sequential optimization procedure may produce a schedule that varies considerably from the optimum. Indeed, even a feasible solution cannot be guaranteed.

2.3.2. *MILP Approach*

A number of authors have proposed MILP formulations to various mine scheduling problems (Caccetta et al. [11,14], Dagdelen and Johnson [17], Gershon [21,22] and Kim [27]). The major computational difficulty has been the size of the problem. Typically MILP approaches are developed "in house" for short term schedules. Below we outline two approaches for solving the MILP formulations.

Recently, Combinatorics Pty Ltd has released the package MineMax [31] for long term mine scheduling. The MILP is solved using a commercial package (for example, CPLEX). Our understanding is that if the problem is too large for the MILP solver, or if a solution is not obtained within a prescribed time period, then a second option is offered. This option is to solve each MILP formulation for free variables on a period by period basis.

The advantages of MineMax include:

- all periods are addressed concurrently in the global case.
- the optimization incorporates all constraints including grade; structural considerations; etc. Thus even if larger block sizes are used (to reduce the number of variables) the solution obtained may be better than that obtained using the nested pit approach.

The disadvantages of MineMax include:

- capable of solving only very small size problems due to the large number of integer variables and constraints. This is true for both options as will be illustrated in the computational section.
- since reblocking is often required, the wall slope requirements are poorly approximated as is the block data.
- in the period by period option, the schedule obtained may be far from optimum.

Caccetta et al [14] proposed a Lagrangian relaxation method for solving the MILP. At each step a problem similar to the ultimate pit limit problem is solved using the LGA with additional constraints dualized. Subgradient optimization is used to reduce the duality gaps. The method is tested on a real ore body with 20,979 blocks and 6 time periods. The schedules obtained are within 5% of the theoretical optimum. The main problem with the method is resolving the duality gaps. However, the subproblems are useful in producing solutions using a heuristic. In fact, the heuristic solution obtained for the real ore body is within 2% of the theoretical optimum.

2.3.3. *Other Methods*

We conclude this section by briefly mentioning two other approaches. Runge Mining Pty Ltd have developed the XPAC Autoscheduler package for mine scheduling [5]. Their heuristic approach is based on the method proposed by Gershon [23] which iteratively selects blocks to be extracted on a period by period basis. A weighted function is used to determine the removal sequence. At each step only blocks whose predecessors have been mined are considered. The advantage of the method is its speed. It's main use is an interactive tool where the user can see a large number of scenarios by fixing in and out blocks and running the heuristic. The main disadvantages are: the search is myopic; no guarantee of finding a feasible solution; the obtained solution may be far from optimal. The method has been applied to models with up to 100,000 blocks.

Tolwinski and Underwood [36] proposed a method which combines concepts from stochastic optimization and artificial neural networks with heuristics exploiting the structure of the mine. The method works by modelling the development of the mine as a sequence of pits (states) where each pit differs from the previous pit by the removal of one block (state change). A probability distribution based on the frequency with which particular states occur is used to determine the state changes. Heuristic rules are incorporated to learn these characteristics of the sequence of pits which produce a good, or poor, result.

The main advantages of the method are:

- effects of time and other factors can be explicitly included in the optimization.
- the structural constraints are incorporated.
- applicable to mines of realistic size (tested on models of up to 88,000 blocks).

The main disadvantages are:

- only a small number of all possible sequences can be explored. The method suffers from "combinatorial explosion" of the number of states.
- no guarantee of finding a feasible solution if one exists.
- no measure of the quality of the solution.

3. A New Branch and Cut Method

In this section we outline our Branch and Cut procedure for solving the MILP (2.1) - (2.9). Our work is motivated by the recent success of this approach to various large combinatorial optimization problems including: the travelling salesman problem (Applegate et al. [3] and Padberg and Rinaldi 32]); the vehicle routing problem (Achuthan et al. [1,2] and Augerat et al. [4]); airline scheduling (Hoffman and Padberg [25]); and various constrained spanning tree problems (Caccetta and Hill [12,13]).

The objective of our work is to produce a method which explicitly incorporates all constraints in the optimization and is capable of producing provably good solutions for reasonably large problems. The quality bound is important as it provides mining engineers confidence with the results produced.

We now detail some of the important features of our method which exploits the structure of the problem. Our algorithm has been implemented in C++ and involves some 17,000 lines of code. The code has been tested on operating mines. Because of the commercialization of the software and confidentiality agreements we are unable to provide full details of all aspects of our work. However, we do summarize below some important features.

3.1. KEY FEATURES

1. The block model is reduced to only include blocks inside the final pit design developed from the ultimate pit (Note: Theorem in previous section). Further reductions are made through consideration of (2.2) - (2.5) and (2.7).
2. The MILP has strong branching variables due to the dependencies between variables ((2.6) and (2.7)). Note that setting a variable to 0 or 1 will fix a potentially large number of other variables. Consequently the subsequent LP relaxations are significantly smaller in size. This motivates more branching compared to typical Branch and Cut methods.
3. Cutting planes involving Knapsack constraints are identified using the capacity upper bounds ((2.2) - (2.5)) and the block removal dependencies (2.7). Also cuts are identified through material removal dependencies between benches.
4. Our search strategy involves a combination of best first search and depth first search. The motivation for this is to achieve a "good spread" of possible pit schedules (best first search) whilst benefiting from using depth first search where successive LP's are closely related from one child node to the next.

For large problems this often results in provably good solutions being found earlier than a search method geared to establishing an optimal solution.

5. Good lower bounds are generated through the use of an LP-Heuristic. The method works by considering each period in turn and fixing in and out sets of free variables. Cutting planes are then generated for the period, further LP's are solved and further fixing occurs. Throughout the fixing of variables feasibility checks are used. If the heuristic succeeds, or fails due to an inferior lower bound being found, then periods are considered in the same direction, otherwise the direction is reversed. The heuristic is called for the first five levels of the search tree and every eighth node created thereafter.
6. Standard fixing of non basic variables using reduced costs is carried out. Because of the block dependencies this may lead to the LP solution losing its optimality. In this case we call the LP solver and re-enter the cutting plane generation phase without branching.
7. Many branching rules were tested and the following proved to be the best. The free variables are considered and a subset of these is chosen on the basis of closeness to the value of 0.5. For each variable in the subset we calculate the sum change in the fractional values of all variables dependent on the inclusion and exclusion of the branching variable. Choose the one with the highest minimal sum change in both directions of branching. Strong branching is used if the gap between the lower and upper bound is sufficiently small.
8. If the LP subproblem is not solved within a prescribed maximum time (2 minutes), then the LP optimization is terminated and branching is performed using the rules in 7 above. An attempt is then made to solve the resulting LP's within the specified time. This process is repeated as long as necessary.
9. The cutting plane phase is terminated early if tailing-off is detected or if the LP subproblem is solved optimally in more than a prescribed time (1 minute). Note that adding further cutting planes, even with purging of ineffective constraints, tends to increase the solution times for successive calls to the LP solver.
10. When branching we probe a random subset of variables having the same time index as the branching one. Bounds on variables may also be updated during this process.
11. All our LP subproblems are solved using CPLEX Version 6.6 [16]. We only use CPLEX in solving the relaxed LP's. The preprocessor and aggregator options are switched off and all other defaults are used.

Before discussing computational results in the next section we conclude this section with brief discussion on some extensions to the MILP formulation (2.1) - (2.9) to incorporate further practical mining restrictions.

3.2. MODEL EXTENSIONS

Besides including constraints from the linear sum of attributes of blocks or the ratio of quantities and bounding them (for example, blending), the following are modelled:

- **Processing different ore types:**
  In many applications we have $K \geqslant 2$ ore types which need to be processed through the mill. The processing rate, in tons/hour, r_i for ore type i is known. The total processing time per period constraints can be written as:

$$\sum_{k=1}^{K} \frac{1}{r_k} \left(\sum_{i=1}^{N} t_{ik} x_i^1 \right) \leqslant Q,$$
$$\sum_{k=1}^{K} \frac{1}{r_k} \left(\sum_{i=1}^{N} t_{ik} \left(x_i^t - x_i^{t-1} \right) \right) \leqslant Q, \quad t = 2, 3, ..., T$$

  where
  t_{ik}: tonnage of ore type k in block i
  Q : expected time (in hours) the mill is available per period.
- **Maximum vertical depth:**
  To allow pit access (haul roads) it is desirable to restrict the maximum vertical depth D that can be mined in any one period. This restriction can be modelled as follows. Suppose blocks i and j have coordinates (x, y, z_1) and (x, y, z_2), respectively with $z_1 < z_2$ and $z_2 - z_1 > D$. Then blocks i and j must be mined in different time periods and so

$$x_\ell^1 = 0, \text{ for all blocks } \ell \text{ more than } D \text{ units below the surface.}$$

$$x_j^t \leqslant x_i^{t-1} \text{ , for } t = 2, 3, ..., T.$$

- **Minimum pit bottom width:**
  In order to facilitate equipment movement, a minimum pit bottom width for each period needs to be specified. We do not believe that linear constraints can be used to specify this. The usual approach is to manually "smooth" the base of each incremental pit; optimality is lost by this process. A feature we have noted of mining is that when a pit is developed with blocks extracted from several benches the wall slopes formed are rarely at their upper bounds, except for blocks located at the limits of the final pit. Consequently, our approach is to redefine the wall slope angles proportionally at all blocks except those located at the limits. This will have the effect that blocks are removed in clusters and pushbacks can be naturally defined.
- **Stockpiles:**
  Stockpiles are formed for a number of reasons including: blending; storage of excessive production; and storage of low grade ore for possible future processing. When placing an ore block on a stockpile the block characteristics

Table 1. Results for Smaller Problems

Problem Characteristics		CPU Time (hours) 0.25	0.5	1	2	4
Prob. 1	Gap(%)	0.2667	0.1929	0.1636	0.1538	0.1317
	Nodes	2248	4715	9121	15341	27908
6720 Blocks	Rows	771	643	641	639	702
Prob. 2	Gap(%)	0.7205	0.6616	0.5469	0.5190	0.4972
	Nodes	316	890	2049	4686	10775
13440 Blocks	Rows	1786	1787	1784	1728	1674
Prob. 3	Gap(%)	1.2507	1.2507	0.7901	0.7214	0.6787
	Nodes	19	54	286	873	2280
26208 Blocks	Rows	3988	4120	3781	3837	3812

(grade, tonnage, etc) are known. However, as blocks are mixed on the stockpile, the characteristics of materials removed from the stockpile to the mill need to be treated as variables. Since the amount of ore removed from the stockpile is unknown prior to the optimization this gives rise to some non-linear constraints. To overcome this, we define a variable for the grams of metal taken from the stockpile per time period as well as tonnage of material. Then, using these variables, the average grade of ore being removed from the stockpile is implied and can be rounded. Note that this formulation defines a valid upper bound for the problem. We use several different constraints to bound the average grade value as well as conservation of movement constraints to make the upper bound as tight as possible.

4. Computational Results

Our Branch and Cut algorithm has been implemented in C++ and tested on an SGI Origin 200 dual processor computer as well as an SGI Origin 2000 shared memory multiprocessor computer. The dual processor capability was only used to solve the relaxed LP's. The software has been extensively tested both on test data provided to us by our industry partner as well as data from producing mines. The models in our test data range from 6,720 to 209,664 blocks. In all cases $T = 10$. We ranged the constraints on the amount of material removed as well as the bounds on the milling requirements so as to cover the large number of cases that can actually occur.

Table 2. Results for Larger Problems

Problem		CPU Time (hours)									
Characteristics		0.25	0.5	1	2	4	8	12	14	16	20
Prob. 4	Gap(%)	4.5099	2.0274	1.9247	1.6918	1.1364	0.8187	0.7977	0.7880	0.7838	0.7766
	Nodes	1	7	14	47	247	818	1327	1631	1916	2513
52416	Rows	8411	8122	8251	8383	8003	8176	8254	8312	8253	8234
Blocks											
Prob. 5	Gap(%)	27.9331	21.5744	16.9511	14.6372	10.5561	7.2088	6.3193	6.1240	5.6889	5.3598
	Nodes	2	5	11	21	44	97	170	221	268	366
104832	Rows	15591	15769	15926	16271	16667	16916	17174	17212	17198	17278
Blocks											
Prob. 6	Gap(%)	58.0049	58.0049	56.8978	19.1737	17.6026	11.0502	9.7508	9.7508	9.7033	8.3265
	Nodes	1	16	30	53	104	202	305	355	395	480
209664	Rows	19437	25791	25252	22310	23365	24289	241187	24123	24351	24306
Blocks											

On the SGI Origin 200 computer we obtained for the 26,208 block model optimum solutions guaranteed to be within 0.4% of the optimum within 12 minutes. For the largest model, solutions guaranteed to be within 2.5% of the optimum were obtained within 4 hours. For these larger models we continue the computations for a further 16 hours and observed there was negligible change in the gap.

On the SGI Origin 2000 computer we ran 6 different sized problems ranging from 6720 to 209,664 blocks and obtained the results, which are typical of our output, given in Tables 1 and 2. For the smaller problems we used serial CPLEX. In our tables we use the following notation:

Gap(%): $\dfrac{(UB - LB)}{UB} \times 100$
Nodes: total number of child nodes generated
Rows: total number of constraints in the LP.

Our method generates tight bounds. However, establishing optimality (except, of course, for small problems) is difficult because once we achieve a near optimal solution there are no available cutting planes to remove fractional variables occurring in the same bench level. Note that (2.6) and (2.7) give dependencies between variables corresponding to block removal in time and the vertical dimension, but not horizontally.

Following the above extensive testing we applied our method to a producing gold mine. This mine was operating on a schedule generated by MineMax. In order to make a meaningful comparison with this schedule we simulated the same test conditions used by MineMax. This involved reblocking the original block model which contained 23 million blocks to one containing 1363 blocks. However, as reblocking was carried out with different packages, the total value of the undiscounted pit used in our model was 3.3% less. In this application $T = 6$ and the discount rate was 10%. The constraints involved material movement and an upper bound on the milling capacity (per period). Our software generated 7 good schedules within a total time of 10 minutes. Our best schedule was within 0.27% of the optimum and validated by mining engineers as being realistic. Our schedule yielded an increase of 13.1% in the NPV profit. In fact taking into account the differences in the block model our solution value was at least 15% higher.

The MineMax solution (which was supplied to the mining company by the software author) was obtained through a period by period optimization as the package could not solve globally within the prescribed time limit. An important difference between the two solutions is that ours generates a significantly higher cash flow in the first two periods. This is in fact consistent with the aim of mine planners.

5. Conclusions

The branch and cut method that we have described in this paper provides the first optimization technique that incorporates all the constraints that arise in mine production scheduling. We have demonstrated that provably good solutions can be obtained for practical sized problems. Our more accurate modelling coupled with fast solution techniques provides the mining industry with powerful mathematically based tools for managing its resources. As demonstrated significant economic benefits are attainable.

Acknowledgements

The authors gratefully acknowledge the Australian Research Council for financial support through a SPIRT Grant (No. C69804881) and our industry partner Optimum Planit for financial support and for assisting with all aspects of the project including the provision of data.

References

1. Achuthan, N.R., Caccetta, L. and Hill, S.P. (1995), A New Subtour Elimination Constraint for the Vehicle Routing Problem, *European Journal of Operations Research* 91, 573–586.
2. Achuthan, N.R., Caccetta, L. and Hill, S.P. (1998), Capacitated Vehicle Routing Problem: Some New Cutting Planes, *Asian-Pacific Journal of Operational Research* 15, 109–123.
3. Applegate, R., Bixby, R., Chvatal, V. and Cook, W. (1995), *Finding Cuts in the TSP (A Preliminary Report)*, DIMACS Technical Report, pp. 95–105.
4. Augerat, P., Belengeur, J.M., Benavent, E., Corberan, A., Naddef, N. and Rinaldi, G. (1995) *Computational Results with a Branch and Cut Code for the Capacitated Vehicle Routing Problem*, Research Report 949-M, Universite Joseph Fourier, Grenoble, France, .
5. Autoscheduler, Runge Mining Pty Ltd. (Australia). Web: http//www.runge.com/xpac.
6. Francois-Bongarcon, D.M. and Guibal, D. (1984), Parametization of Optimal Designs of an Open Pit Beginning of a New Phase of Research, *Trans. SME, AIME* 274, 1801–1805.
7. Caccetta, L. and Giannini, L.M. (1986), Optimization Techniques for the Open Pit Limit Problem, *Proc. Australas. Inst. Min. Metall.* 291, 57–63.
8. Caccetta, L. and Giannini, L.M. (1988), An Application of Discrete Mathematics in the Design of an Open Pit Mine, Discrete Applied Mathematics 21, 1–19.
9. Caccetta, L. and Giannini, L.M. (1990), *Application of Operations Research Techniques in Open Pit Mining*, in Byong-Hun Ahn (Ed.), Asian-Pacific Operations Research: APORS'88, Elsevier Science Publishers BV , pp. 707–724.
10. Caccetta, L., Giannini, L.M. and Kelsey, P. (1994), On the Implementation of Exact Optimization Techniques for Open Pit Design, *Asia-Pacific Journal of Operations Research* 11, 155–170.
11. Caccetta, L., Giannini, L.M. and Kelsey, P. (1998), *Application of Optimization Techniques in Open Pit Mining*, Proceedings of the Fourth International Conference on Optimization Techniques and Applications (ICOTA'98) (L. Caccetta et al. Editors.), Vol. 1, pp. 414–422. (Curtin University of Technology: Perth, Australia).
12. Caccetta, L. and Hill, S.P. (2001), Branch and Cut Methods for Network Optimization, *Mathematical and Computer Modelling* 33, 517–532.
13. Caccetta, L. and Hill, S.P. (2001), A Branch and Cut Method for the Degree Constrained Minimum Spanning Tree Problem, *Networks* 37, 74–83.
14. Caccetta, L., Kelsey, P. and Giannini, L.M. (1998), Open Pit Mine Production Scheduling, in A.J. Basu, N. Stockton and D. Spottiswood (Eds.), Computer Applications in the Minerals Industries International Symposium (3^{rd} Regional APCOM), Austral. Inst. Min. Metall. Publication Series 5, 65–72.
15. Coléou, T. (1988), *Technical Parameterization for Open Pit Design and Mine Planning*, in: Proc. 21^{st} APCOM Symposium of the Society of Mining Engineers (AIME) pp. 485–494.

16. CPLEX 6.0 User Manual, ILOG Inc., CPLEX Division, 889 Alder Avenue, Suite 200, Incline Village, NV 89451, U.S.A.
17. Dagdelen, K. and Johnson, T.B. (1986), *Optimum Open Pit Mine Production Scheduling by Lagrangian Parameterization,* in: Proc. 19^{th} APCOM Symposium of the Society of Mining Engineers (AIME) pp. 127–142.
18. Denby, B. and Schofield, D. (1995), *The Use of Genetic Algorithms in Underground Mine Scheduling,* in: Proc. 25^{th} APCOM Symposium of the Society of Mining Engineers (AIME, New York), pp. 389–394.
19. Denby, B., Schofield, D. and Bradford, S. (1991), *Neural Network Applications in Mining Engineering,* Department of Mineral Resources Engineering Magazine, University of Nottingham, pp. 13–23.
20. Earthworks, NPV Scheduler, Web: http://www.earthworks.com.au.
21. Gershon, M. (1982), *A Linear Programming Approach to Mine Scheduling Optimization,* in: Proc. 17^{th} APCOM Symposium of the Society of Mining Engineers (AIME), pp. 483–493.
22. Gershon, M. (1983), Mine Scheduling Optimization with Mixed Integer Programming, *Mining Engineering* 35, 351–354.
23. Gershon, M. (1987), Heuristic Approaches for Mine Planning and Production Scheduling, *Int. Journal of Mining and Geological Engineering* 5, 1–13.
24. Gignac, L. (1975), *Computerized Ore Evaluation and Open Pit Design,* in: Proc. 36^{th} Annual Mining Symposium of the Society of Mining Engineers (AIME), pp. 45–53.
25. Hoffman, K.L. and Padberg, M.W. (1993), Solving Airline Crew Scheduling Problems by Branch and Cut, *Management Science* 39, 657–682.
26. Kim, Y.C. (1979), *Open-Pit Limits Analysis: Technical Overview,* in: A. Weiss (Ed.), Computer Methods for the 80’s (AIME, New York), pp. 297-303.
27. Kim, Y.C. (1979), *Production Scheduling: Technical Overview,* in: A. Weiss, (Ed.), Computer Methods for the 80’s (AIME, New York) pp. 610-614.
28. Lerchs, H. and Grossmann, I.F. (1965), Optimum Design of Open Pit Mines, *Canad. Inst. Mining Bull.* 58, 47–54.
29. Matheron, G. (1975), *Le Parametrage des Contours Optimaux,* Technical Report No. 403, Centre de Geostatistiques, Fontainebleau, France.
30. Matheron, G. (1975), *Le Parametrage Technique des Reseues*, Technical Report No. 453, Centre de Geostatistiques, Fontainebleau, France.
31. MineMax, Combinatorics Pty Ltd, Unit 4b, R&D Centre, 1 Sarich Way, Bentley, Western Australia, 6102.
32. Padberg, M. and Rinaldi, G. (1991), A Branch and Cut Algorithm for the Resolution of Large Scale Travelling Salesman Problems, *SIAM Review* 33(1), 60–100.
33. Picard, J.C. (1976), Maximum Closure of a Graph and Applications to Combinatorial Problems, *Management Sc*. 22, 1268-1272.
34. Thomas, G. (1996), *Optimization of Mine Production Scheduling – the State of the Art,* in Proceedings IIR Dollar Driven Planning Conference.
35. Thomas, G. *Pit Optimization and Mine Production Scheduling – The Way Ahead,* in: Proceed. 26^{th} APCOM Symposium of the Society of Mining Engineers (AIME) (1976), pp. 221–228.
36. Tolwinski, B. and Underwood, R. (1996), *A Scheduling Algorithm for Open Pit Mines,* IMA Journal of Mathematics Applied in Business and Industry 7, 247–270.
37. Whittle, J. (1993), *Four-D User Manual*, Whittle Programming Pty Ltd., Melbourne, Australia.
38. Whittle, J. (1998), *Four-X User Manual*, Whittle Programming Pty Ltd., Melbourne, Australia.
39. Whittle, J. (1990), *Open Pit Optimization,* Surface Mining (2^{nd} Edition), AMIE pp. 470–475.
40. Yamatomi, J., Mogi, G., Akaike, A. and Yamaguchi, U. (1995), *Selection Extraction Dynamic Cone Algorithm for Three-Dimensional Open Pit Designs,* in: Proceed. 25^{th} APCOM Symposium of the Society of Mining Engineers (AIME) pp. 267-274.