Is openpit production scheduling “easier” than its underground counterpart?

by D. O’Sullivan, A. Brickey and A. Newman

Abstract ■ Although some of the prevalence of openpit production scheduling software can be explained by the predominance of openpit mining throughout the world, other factors have led to a lag in corresponding underground software. We explain mathematically why underground production scheduling is more difficult than its openpit counterpart and provide directions for research in the underground scheduling arena.

Mining Engineering, 2015, Vol. 67, No. 4, pp. 68-73.
Official publication of the Society for Mining, Metallurgy & Exploration Inc.

Introduction

Academic research over the past three decades has focused on using the Lerchs-Grossmann algorithm as a mechanism for solving more general openpit mine scheduling problems. This knowledge base has transferred to industry and, at present, a number of software firms, including Geovia, Maptek and MineMax, offer production scheduling optimization solutions for openpit mining operations. Because openpit mines comprise the vast majority of mining operations in the United States (Hartman, 2007), the commercial focus on optimizing their production schedules is understandable. However, while companies strive to produce next-generation tools for openpit mines—such as optimization of integrated mine design and production problems—scheduling software for underground mines remains limited to supporting manual production planning, a complex and difficult task that can require months to complete.

D. O’Sullivan, A. Brickey, member SME, and A. Newman are consultant and Ph.D. graduate (Mineral and Energy Resources), Ph.D. candidate (Department of Mining Engineering) and professor (Department of Mechanical Engineering), respectively, at the Colorado School of Mines, Golden, CO. Paper number TP-14-027. Original manuscript submitted April 2014. Revised manuscript accepted for publication December 2014. Discussion of this peer-reviewed and approved paper is invited and must be submitted to SME Publications by July 31, 2015.

Integer programming review

Johnson’s (1968) integer paradigm for modeling openpit production scheduling continues to be the predominate approach. Optimal solutions to integer programs (whether they also contain continuous variables representing the amount of material extracted, or in a stockpile, for example) can be identified by executing the branch-and-bound algorithm (Rardin, 1998). A process of intelligent enumeration, this algorithm systematically examines a tree of potential solutions, eliminating those that are clearly dominated based on their objective function values. Commercial solvers like CPLEX (IBM) implement variations of this algorithm in combination with other advanced optimization techniques, that is, heuristic methods and cutting planes (Klotz and Newman, 2013b). In theory, as the size of an integer program grows, the time required for the branch-and-bound algorithm to solve the problem increases exponentially. For this reason, despite phenomenal increases in computing power, large and complex mine scheduling models continue to challenge researchers and practitioners alike.

Mine design and scheduling models

Before examining the complexity of production scheduling problems, we first review two common mathematical formulations as they appear in Espinoza et al. (2013). The mathematical formulations are as follows:

Indices and sets:

$t \in \mathcal{T}$: set of time periods t in the horizon
$b \in \mathcal{B}$: set of blocks b
$b' \in \widehat{\mathcal{B}}_b$: set of blocks b' that are predecessor blocks for block b
$r \in \mathcal{R}$: set of operational resource types r

Parameters:

p_b $(\hat{p}_{bt})$: profit obtained from extracting (and processing) block b (at time period t)($)
q_{br} : the amount of operational resource r used to extract and, if applicable, process, block b (tons)
$\underline{R}_{rt}$: minimum availability of operational resource r in

Resumen ■ A pesar de que parte del predominio de software de programación de producción a cielo abierto se puede explicar por la preponderancia de la minería a cielo abierto en todo el mundo, hay otros factores que han dado lugar a un retraso en el software subterráneo correspondiente. Explicamos por qué matemáticamente la programación de la producción subterránea es más difícil que su contraparte a cielo abierto y daremos pautas para la investigación en el campo de la programación subterránea.

time period t (tons)

$\overline{R}_{rt}$: maximum availability of operational resource r in time period t (tons)

Variables:

$\hat{x}_b$: 1 if block b is in the final pit design; 0 otherwise

x_{bt} : 1 if we extract block b in time period t; 0 otherwise

$$(UPIT) \quad \max \sum_{b \in \mathcal{B}} p_b \hat{x}_b$$

$$\text{subject to} \quad \hat{x}_b \leq \hat{x}_{b'} \qquad \forall\, b \in \mathcal{B}, b' \in \widehat{\mathcal{B}}_b \qquad (1)$$

$$\hat{x}_b \in \{0,1\} \qquad \forall\, b \in \mathcal{B} \qquad (2)$$

The ($UPIT$) problem is the openpit design problem that the Lerchs-Grossmann algorithm solves. It maximizes profit by determining the pit size that contains the most economic selection of blocks. Each block b has a value, p_b, and is associated with the binary variable, $\hat{x}_b$, that assumes a value of 1 if $\hat{x}_b$ is chosen for extraction and 0 otherwise. Precedence constraints [the inequality shown as Eq. (1)] ensure that any block, $b \in \mathcal{B}$, can only be extracted once all of its predecessors, $b' \in \widehat{\mathcal{B}}_b$, have been extracted. The number of variables corresponds to the number of blocks in the problem, while the number of constraints depends on the precedence relationships between blocks.

The ($CPIT$) problem introduces the time dimension that ($UPIT$) lacks:

$$(CPIT) \quad \max \sum_{b \in \mathcal{B}} \sum_{t \in \mathcal{T}} \hat{p}_{bt} x_{bt}$$

$$\text{subject to} \quad \sum_{\tau \leq t} x_{b\tau} \leq \sum_{\tau \leq t} x_{b'\tau} \qquad \forall\, b \in \mathcal{B}, b' \in \widehat{\mathcal{B}}_b, t \in \mathcal{T} \quad (3)$$

$$\sum_{t \in \mathcal{T}} x_{bt} \leq 1 \qquad \forall\, b \in \mathcal{B} \qquad (4)$$

$$\underline{R}_{rt} \leq \sum_{b \in \mathcal{B}} q_{br}\, x_{bt} \leq \overline{R}_{rt} \qquad \forall\, r \in \mathcal{R}, t \in \mathcal{T} \qquad (5)$$

$$x_{bt} \in \{0,1\} \qquad \forall\, b \in \mathcal{B}, t \in \mathcal{T} \qquad (6)$$

The objective now maximizes discounted profits, while the constraints in Eq. (3) enforce precedence rules. The additional constraints in Eq. (4) restrict a block to be extracted at most once, and the constraints in Eq. (5) ensure that total resource usage does not exceed its availability in any given time period. The number of variables in the problem is given by $|\mathcal{B}| \times |\mathcal{T}|$, considerably more than the $|\mathcal{B}|$ contained in a ($UPIT$) problem of commensurate size. In addition to the precedence constraints, which also depend on time in ($CPIT$), the formulation has $|\mathcal{R}| \times |\mathcal{T}|$ resource constraints that may limit, for example, extraction and processing capacity. Although ($CPIT$) problems contain fewer resource constraints than precedence constraints, even a single such constraint destroys the network structure present in ($UPIT$).

Relative tractability of (UPIT) and (CPIT) problems

The tractability of models such as ($UPIT$) and ($CPIT$) depends on: (i) the size of the problem, in terms of the number of variables and constraints, and (ii) the structure of the constraint sets, including the resulting density of those constraints. Practical scheduling problems exhibiting simple and repeatable patterns are solved with specialized techniques that take advantage of their structure, that is, the underlying network in the ($UPIT$) problem, for which network algorithms can expedite solutions. Ahuja et al. (1993) show that by considering each block as a node and representing the precedence relationships between blocks with directed arcs, ($UPIT$) can be modeled as a maximum weight closure problem and solved with a polynomial-time algorithm quickly relative to solving the original problem with branch and bound. This structure allows for integer solutions even when the integrality requirements are ignored in the solution procedure.

To see the special structure of the ($UPIT$) problem, let us rearrange the constraints in Eq. (1) so that all variables are on the left-hand side:

$$\hat{x}_b - \hat{x}_{b'} \leq 0 \qquad \forall\, b \in \mathcal{B}, b' \in \widehat{\mathcal{B}}_b \qquad (7)$$

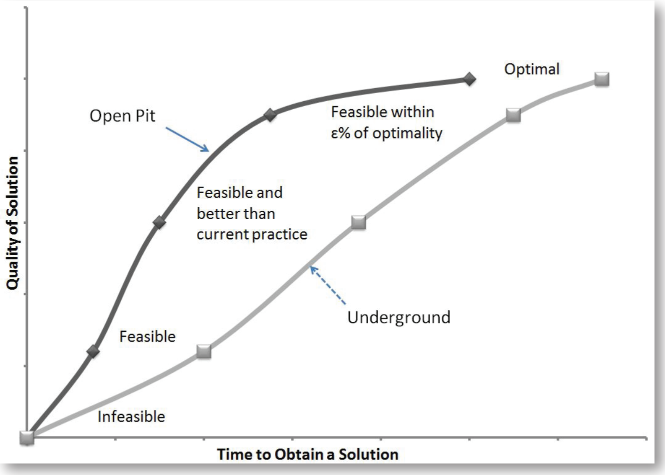
The matrix of left-hand-side constraint coefficients for this problem populate the so-called A-matrix with values of ±1 or 0, with at most one +1 and one −1 in each column:

$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & -1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & -1 \end{bmatrix}$$

On the other hand, the ($CPIT$) problem's constraint set does not possess as "easy" a structure. The precedence constraints contain the dimension of time, and while they still constitute a network, one can see that, without refor-

Figure 1

The total time to develop and solve an integer program for mine scheduling depends on the quality of the solution required. For openpit mines, solutions are often close to optimality or at least better than current practice. The more complicated and heterogeneous characteristics of underground mines result in more difficult problems that take longer to solve. In general, underground mine planners must be satisfied with a solution that is better than current practice and that can be produced in a reasonable amount of time.



mulation, there are many more +1's and −1's on the left-hand side of the constraint set (owing to the summation on t). A reformulation (Bienstock and Zuckerberg, 2010) retains the maximum closure network structure that the constraint set in (*UPIT*) possesses. However, the resource constraints [Eq. (5)] not only add rows with strings of coefficients corresponding to the amount of resource consumed in a time period (that is, the coefficients q_{br}), but also destroy the structure that admits integer solutions without those requirements (Lambert et al., 2014). Generally, there are few resource constraints relative to the precedence constraints, which is noteworthy for tractability purposes since the performance of commercial optimizers is significantly affected by the density of a problem's A matrix, that is, the ratio of nonzero to zero values. These solvers reduce problem size by storing only nonzero matrix coefficients in memory. However, when the A matrix averages more than 10 nonzero elements per column, it is considered dense. Consequently, with a large number of nonzero values with which the solver must compute, solution time slows considerably (Klotz and Newman, 2013a).

We observe that for the most "difficult" openpit scheduling models, the ease with which they can be solved is far greater than that with which an underground model instance of commensurate size can be solved (Fig. 1). We examine two principal reasons that, at the time of this writing, underground production scheduling models are more difficult to formulate and solve than their corresponding openpit counterparts: (i) the difference in structure between the two types of mines and (ii) the characteristics of the entities requiring "action."

Openpit versus underground mine structure

The characteristics of a mining operation define the mathematical structure of its corresponding production scheduling optimization problem; this structure, in turn, determines the tractability of the problem.

Precedence constraint structure. Perhaps the most significant difference between openpit and underground mine scheduling problems is in the structure that underlies the precedence rules governing the sequence of extraction between blocks of ore. For openpit mines that employ a repeatable precedence rule, such as the "plus sign convention" (Lambert et al., 2014), this underlying structure forms a network that can be exploited by the Lerchs-Grossmann algorithm when (i) solving the (*UPIT*) problem or (ii) solving the (*CPIT*) problem with a heuristic or exact method, for example, a Lagrangian relaxation procedure (Dagdelen and Johnson, 1986; Lambert and Newman, 2013). The mathematical structure of these types of constraints for (*UPIT*) and (*CPIT*) is given in Eq. (1) and Eq. (3), respectively.

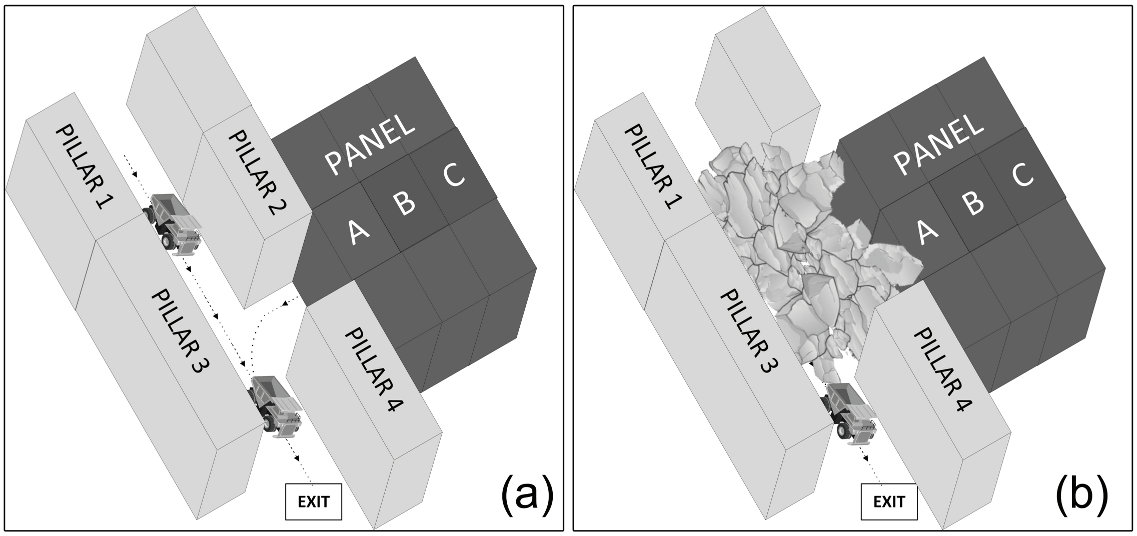
Underground mine precedence structure can differ greatly from one mine to the next. For the most part, the method of extraction used in an area, for example, a stope panel, of an underground mine dictates the order of mining in that area; underground mines often use a combination of mining methods, and precedence rules can relate extraction activities to non-extraction activities, such as ventilation requirements (Brickey, 2013), structural support or safety protocols. Consequently, even when underground mines possess a single, uniform mining method with a repeatable precedence pattern, such as sublevel caving at the Kiruna Mine in Sweden (Newman and Kuchta, 2007), other precedence rules may preclude the underlying network structure that commercial solvers could exploit to produce timely solutions. In addition, complex precedence logic (for example, Martinez and Newman, 2011) results in constraints with more variables, and this produces a dense A matrix, slowing computation.

As another example, consider Fig. 2, in which the Lisheen mine in Ireland possesses precedence rules between panel activities and haulage pillars, that is, designated pillars that are left in place to support the haulage routes (O'Sullivan and Newman, 2014). While the majority of the precedence constraints in the model are similar to the constraints in Eq. (3), which possess an underlying network structure, others are more mathematically troublesome. Specifically, while extracting a haulage pillar is optional, it may prevent extraction in other areas of the mine. This type of precedence constraint incorporates an "or" decision and cannot be incorporated into a network model. Because precedence rules governing haulage pillars do not have an underlying network structure, the tractability of the Lisheen scheduling problem is largely determined by the number of haulage pillars included in the model (Fig. 3).

Operations and activities. The openpit scheduling problem can be segmented into a series of smaller, more tractable subproblems; blocks are assigned a phase number that can be included in the precedence constraints

Figure 2

We show a conceptual illustration of precedence rules at the Lisheen underground mine. In (a), we render a panel of ore extraction activities adjacent to a haulage route comprising haulage pillars containing ore. Within the panel, strict precedence rules, similar to the constraints in Eq. (1), dictate that Activity A must be taken before Activity B and Activity B before Activity C, that is, it is impossible to take Activity C without extracting both Activity A and Activity B in advance. Precedence rules of this type have an underlying network structure – that is, by representing the activities as nodes, arcs between pairs of nodes define the precedence rules; (b) illustrates that once a haulage pillar (Pillar 2) is extracted, the roof caves in, blocking access to pillars (Pillar 1) and activities (A, B, and C) upstream of the extracted pillar. Consequently, this precedence rule, that is, of the form of a packing constraint that if you extract Pillar 2, you cannot extract Pillar 1 nor perform activities (A, B, and C), cannot be incorporated into a network model.



when determining a schedule in linear- or mixed-integer-programming-based scheduling software. Because the (*UPIT*) problem is specific to openpit mines, there is no equivalent decomposition technique that can be applied to underground mines.

A fundamental difference between openpit and underground mining is the treatment of scheduling waste material. From an analytical perspective, the openpit mining process from ore to product can be separated into three decisions: (1) which blocks to extract at time *t*; (2) whether, upon extraction, to send a block to the mill, leaching heap, stockpile or dump and (3) what ore blocks to select from the stockpiles at time *t* to satisfy grade control. While there is a degree of interdependency between these decisions, there may be sufficient separation to model and optimize each one independently and then employ a heuristic to provide a solution for the monolith. Consequently, for openpit mining operations, each model need only consider the subset of operational constraints that is relevant to the particular decision, and each constraint has fewer relevant decisions associated with it. Fewer variables in each constraint reduces the density of the *A* matrix; this, combined with fewer constraints overall, improves tractability.

By contrast, underground miners seek to extract only the stopes, referred to as activities, that they decide

Figure 3

We illustrate the effect of complex precedence rules on tractability. For the Lisheen mine, if we exclude haulage pillars from the scheduling problem, we can solve a 52-week instance of the model in a matter of seconds. As we introduce haulage pillars, complexity and solution time increase (approximately exponentially) until problem instances are no longer tractable.

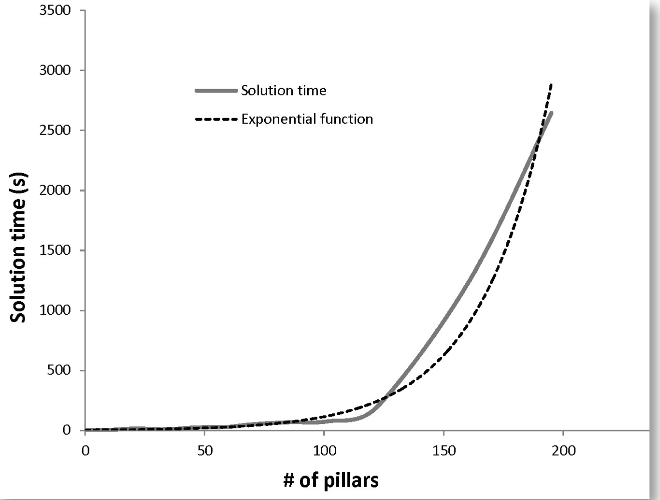
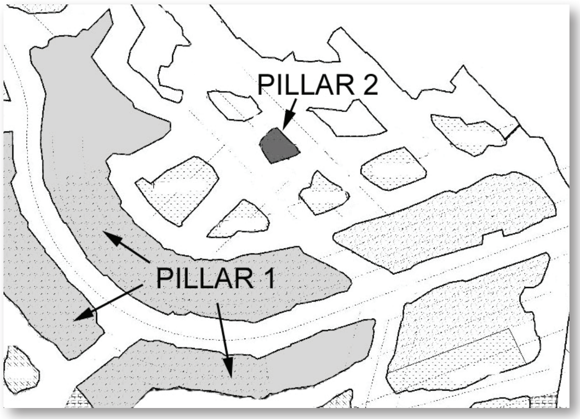


Figure 4

In the section of the Lisheen mine shown here, Pillar 1 contains 34,000 tonnes of ore and requires 100 days for extraction. By contrast, Pillar 2 containing 634 tonnes of ore can be extracted in two days. Such heterogeneity complicates precedence rules and makes the choice of a time fidelity difficult.



a priori will be processed. Many underground mines do not stockpile ore; consequently, underground mines that blend must coordinate the extraction of blocks so that the flow of ore directly satisfies the blending requirements at the mill. It is difficult to decompose the problem into sequences of phases. Therefore, decomposition heuristics cannot be applied as frequently, and the monolith remains difficult to solve.

Openpit versus underground development activity characteristics

In addition to the factors mentioned above, the number of ore blocks (openpit) or activities (underground), the shapes and sizes (of underground activities), and the number and types of blocks or activities being scheduled, can significantly influence the tractability of mine scheduling problems, particularly with respect to the application of heuristic techniques that rely on spatial and/or temporal aggregation to produce solutions.

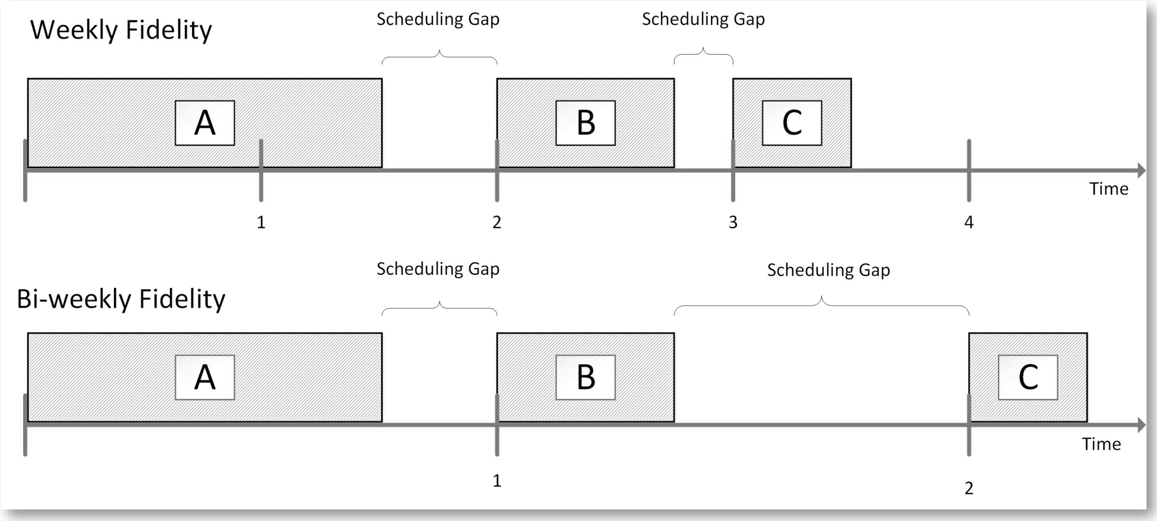
Block shape and size. Characterization of blocks in an openpit mine as identical cuboids allows researchers to formulate more tractable openpit scheduling problems (Johnson, 1969). The regularity of block size and shape facilitates: (i) the definition of repeatable precedence rules (ii) the selection of a suitable time fidelity for the problem and (iii) the aggregation of ore blocks to reduce problem size. An obvious way to cope with heterogeneously sized blocks is to aggregate them, for example, based on a measure of similarity (Tabesh and Askari-Nasab, 2011). An aggregated solution that must be disaggregated, even heuristically, is better than no solution at all.

In underground mining, the volume and dimensions of the ore extraction activities, that is, stopes, can vary greatly (Fig. 4). This heterogeneity is most often a consequence of the technical design, which is based on the mining method and the distribution of mineral concentration. Once the mine is operational, additional factors, such as fissures in the rock and/or unpredictable results of blasting, can impact heterogeneity. As a consequence, precedence rules cannot necessarily be easily articulated mathematically (see discussion on precedence constraint structures).

Additionally, it can be difficult to choose a time fidelity for the schedule; the fidelity must be small enough so that activities of shorter duration do not unnecessarily push the schedule forward (Fig. 5). However, because we must account for finer fidelity by defining variables for each activity and start-time combination, we must also be

Figure 5

At weekly fidelity, the total scheduling gap between Block A, Block B and Block C is much smaller than at bi-weekly fidelity. These gaps unnecessarily extend the start times for extraction of dependent blocks. While Block C can start during week 3 with weekly fidelity, at bi-weekly fidelity, the same block must wait until period 2 (week 4) before it can start.



mindful to select a time fidelity that is large enough to produce a tractable model.

For underground mines that blend ore, possess complex precedence rules relating extraction activities to non-extraction activities, and/or contain irregular ore activities, aggregation of these activities is not a viable approach to reduce problem size and improve tractability, even for long-term planning. The activities are simply too different, in too many ways, even if they lie in close proximity to each other. Aggregation would lead to faulty precedence rules, and, depending on the nature of the deposit, could grossly mis-estimate the ore grade recoverable from the aggregated activity.

Entity types. In openpit production scheduling, the principal decision is always whether or not to extract a given block (cuboid) of ore, and when. In underground production scheduling models, there are many activities that need to be scheduled, for example, access, development and extraction, before the "block" is extracted, and the subsequent backfill of a void after extraction. Because these different activities do not need to occur in an uninterrupted sequence, aggregating, say, a development-extraction-backfill chain into a single "super activity" could grossly compromise solution quality. Therefore, model size cannot be as easily reduced as in a corresponding openpit production scheduling model. Additionally, each of these activities requires different resource constraints of the form given in Eq. (5). The incorporation of these constraints in and of themselves makes the model less tractable [see discussion on the relative tractability of the *(UPIT)* and *(CPIT)* problems].

Future directions

Bienstock and Zuckerberg (2010) present a novel algorithm for production scheduling in openpit mines. Their method solves the linear programming relaxation of the integer programming problem, that is, the original problem with the integer variables relaxed to be continuous. Their linear programming algorithm, combined with rounding heuristics (Chicoisne et al., 2012), produces results for large problem instances significantly faster than standard techniques. Despite the reported results at the time of this writing on problem types as simple as (*CPIT*), their approach, like the Lerchs-Grossmann algorithm of the 1960s, highlights a new research direction. ■

References

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B., 1993, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs.

Bienstock, D., and Zuckerberg, M., 2010, "Solving LP relaxations of large-scale precedence constrained problems," *Integer Programming and Combinatorial Optimization, 14th International Conference, IPCO 2010*, F. Eisenbrand and F.B. Shepherd, eds., June 9-11, 2010, Lausanne, Switzerland, Springer, Vol. 6080, Lecture Notes in Computer Science.

Brickey, A., 2013, "Underground Mine Production Schedule Optimization with Ventilation," INFORMS Annual Meeting 2013, Minneapolis, MN.

Chicoisne, R., Espinoza, D., Goycoolea, M., Moreno, E., and Rubio, E., 2012, "A new algorithm for the open-pit mine production scheduling problem," *Operations Research*, Vol. 60, pp. 517-528.

Dagdelen, K., and Johnson, T. B., 1986, "Optimum open pit mine production scheduling by Lagrangian parameterization," *Proceedings of the 19th Symposium APCOM*, PA, Jostens Publications, pp. 127-141.

Espinoza, D., Goycoolea, M., Moreno, E., and Newman, A., 2013, "Minelib: A library of open pit mining problems," *Annals of Operations Research*, Vol. 206, pp. 93-114.

GEOVIA, 2012, Geovia, Vancouver, Canada.

Hartman, H. L., 2007, "Introductory mining engineering," retrieved from Wiley.com.

IBM, 2013, ILOG CPLEX, Incline Village, NV.

Johnson, T., 1968, "Optimum Open Pit Mine Production Scheduling," Ph.D. Thesis, University of California at Berkeley.

Johnson, T., 1969, "Optimum open pit mine production scheduling," *A Decade of Digital Computing in the Mining Industry*, A. Weiss, ed., American Institute of Mining Engineers, New York, Chap. 4, pp. 17-24.

Klotz, E., and Newman, A.M., 2013a, "Practical guidelines for solving difficult linear programs," *Surveys in Operations Research and Management Science*, Vol. 18, pp. 1-17.

Klotz, E., and Newman, A. M., 2013b, "Practical guidelines for solving difficult mixed integer programs," *Surveys in Operations Research and Management Science*, Vol. 18, pp. 18-32.

Lambert, B., Brickey, A., Newman, A., and Eurek, K., 2014, "Open pit block sequencing formulations: A tutorial," *Interfaces*, Vol. 44, pp. 127-142.

Lambert, W., and Newman, A., 2013, "Tailored Lagrangian relaxation for the open pit block sequencing problem," *Annals of Operations Research*, Vol. 222, No. 1, pp. 419-438.

Maptek, 2012, Maptek Software Pty Ltd, Vulcan Chronos, Sydney, Australia.

Martinez, M.A., and Newman, A.M., 2011, "A solution approach for optimizing long- and short-term production scheduling at LKAB's Kiruna mine," *European Journal of Operational Research*, Vol. 211, pp. 184-197.

MineMax (n.d.), MineMax Software Pty Ltd, Perth, Australia.

Newman, A., and Kuchta, M., 2007, "Using aggregation to optimize long-term production planning at an underground mine," *European Journal of Operational Research*, Vol. 176, pp. 1205-1218.

O'Sullivan, D., and Newman, A., 2014, "Extraction and backfill scheduling in a complex underground mine," *Interfaces*, Vol. 44, pp. 204-221.

Rardin, R.L., 1998, *Optimization in Operations Research*, Prentice Hall, Upper Saddle River.

Tabesh, M., and Askari-Nasab, H., 2011, "Two-stage clustering algorithm for block aggregation in open pit mines," *Mining Technology*, Vol. 120, pp. 158-169.