TOPAL, E., KUCHTA, M., and NEWMAN, A. Extensions to an efficient optimization model for long-term production planning at LKAB's Kiruna Mine. *Application of Computers and Operations Research in the Minerals Industries,* South African Institute of Mining and Metallurgy, 2003.

Extensions to an efficient optimization model for long-term production planning at LKAB's Kiruna Mine

E. TOPAL, M. KUCHTA, and A. NEWMAN
Colorado School of Mines, Golden, Co, USA

LKAB's Kiruna Mine, located above the Arctic Circle in northern Sweden, is the second-largest underground mine in the world today. The orebody is a world-class high-grade magnetite deposit, from which three raw ore products are extracted. We use mixed integer programming to determine a production schedule, i.e., which production blocks to mine, and when to start mining them, so as to minimize deviation from planned production quantities while adhering to the operational constraints of the mine. The resulting model contains thousands of binary variables and a commensurate number of constraints, precluding us from obtaining a schedule in a reasonable amount of time. We describe a method that builds on prior, related work, to expedite solution time.

Keywords: underground mining, production planning, optimization, integer programming

Introduction

LKAB's Kiruna Mine is located above the Arctic Circle in northern Sweden, and produces about 24 million tons of iron ore per year, making it is the second largest underground mine in the world today. The orebody, a world-class high-grade magnetite deposit, is approximately 4 km long and 80 m wide on average, and lies roughly in the north-south direction, with a dip of some 60 degrees. Surface mining commenced around the turn of the last century, and half a century later, operations began underground. Since 1962, mining has been done exclusively via large-scale sublevel caving. The mine is divided into ten main production areas, about 400 m to 500 m in length, each with its own group of ore passes, also known as a *shaft group*, located at the centre of each production area and extending down to the current main 1045 m level (Figure 1). One to two 25-ton-capacity electric Load Haul Dump

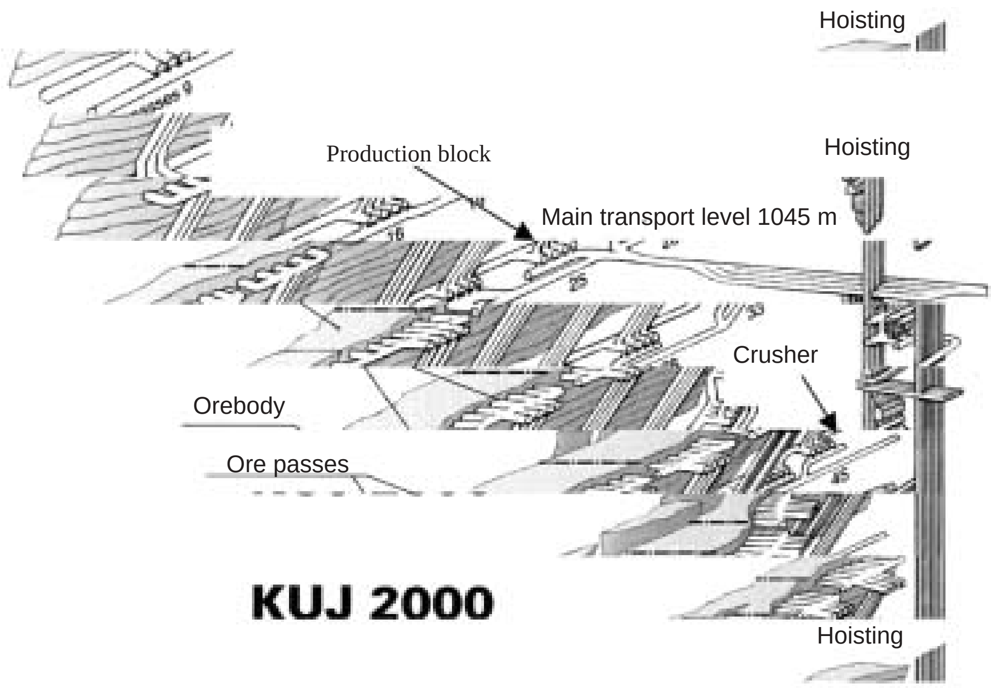

Figure 1. Schematic of the orebody being mined at its current level (LKAB, Internal promotional material, 2001)

Units (LHDs) operating on a sublevel within each production area transport the ore from the crosscuts to the ore passes. The site on which each LHD operates is also referred to as a *machine placement*. Each machine placement is usually 200 to 500 m in length and contains from one to three million tons of ore, equating to between one and five 100 m *production blocks*; the machine placements possess the same height as the mining sublevel and extend from the hangingwall to the footwall. Once started, mining restrictions require continuous production of the blocks within a machine placement until all available ore has been removed.

From the two main *in situ* ore types, the mine delivers three raw ore products used to supply four ore post-processing plants, or mills. Phosphorus is the main ore contaminant. The B1 product contains the least amount of phosphorus and is used to produce high-quality fines. The other ore types, B2 ore and D3 ore, are processed into pellets and possess medium- and high-phosphorous content, respectively.

Scheduling model

Optimization models are commonly used, not only in the mining sector, to make decisions, represented with *decision variables*, so as to attain a goal, specified as an *objective function*, while meeting operational limits, called *constraints*. We employ such an optimization model, specifically, a multi-period mixed integer program, to provide mine planners with a production schedule that details when to start mining each production block within each machine placement in order to satisfy demand for each ore type at the mills as closely as possible while not violating any mine sequencing constraints. An *optimal* solution would give the *minimum* deviation from planned production quantities and the means by which to attain this minimum, i.e., for our application, the time at which a given production block should be mined; a *sub-optimal* solution would give a value for the deviation that is higher than what could theoretically be realized. In either case, the solution would satisfy all operational constraints. A sub-optimal solution may still be usable, if its deviation from optimality is relatively low, e.g., 1–5%.

For ease of presentation, we assume that each production block requires exactly one time period to mine. The model follows:

Indices

- a machine placement
- b,b' production block
- k ore type, i.e., B1, B2, D3
- t time period (month)
- v shaft group, i.e., 1…10

Sets

- Ω_b = set of eligible time periods in which production block b can be mined (restricted by production block location and the start time of other relevant production blocks)
- A_a = set of production blocks within machine placement a
- BlockV_b = set of production blocks whose access is restricted vertically by production block b
- BlockR_b = set of production blocks whose access is forced by right adjacency to production block b
- BlockL_b = set of production blocks whose access is forced by left adjacency to production block b
- S_v = set of machine placements contained in shaft group v

Parameters

- r_{bk} = amount of ore type k in block b (tons)
- d_{kt} = demand for ore type k in time period t (tons)
- $Early_b$ = earliest start time for production block b
- $Late_b$ = latest start time for production block b
- T = length of the planning horizon
- LHD_v = the maximum number of simultaneously-operational LHDs in each shaft group v

$$P_{abv} = \begin{cases} 1, \text{ If block } b \text{ of machine placement } a \text{ is in shaft group } v \\ 0, \text{otherwise} \end{cases}$$

Decision variables

$$y_{bt} = \begin{Bmatrix} 1, \text{ If start mining production block } b \text{ in time period } t \\ 0, \text{otherwise} \end{Bmatrix}$$

- du_{kt} = amount mined above the desired demand of ore type k in time period t (tons)
- dd_{kt} = amount mined below the desired demand of ore type k in time period t (tons)

Objective function

$$Min \sum_{k,t} \left(du_{kt} + dd_{kt} \right)$$

Constraints

$$\sum_b r_{bk} * y_{bt} - du_{tk} + dd_{tk} = d_{kt} \quad \forall k, t \in \Omega_b \qquad [1]$$

$$\sum_{t \in \Omega_b} y_{bk} \geq y_{bt} \quad \forall b, b' \in BlockV_b, t' \in \Omega_{b'} \qquad [2]$$

$$\sum_{t \in \Omega_b} y_{bt} \leq y_{b't'} \quad \forall b, b' \in BlockR_b, t' \in \Omega_{b'} \qquad [3]$$

$$\sum_{t \in \Omega_b} y_{bt} \leq y_{b't'} \quad \forall b, b' \in BlockL_b, t' \in \Omega_{b'} \qquad [4]$$

$$\sum_{a \in S_v} \sum_{b \in A_a} \sum_{u \in \Omega_b \setminus u \leq t} P_{abv} * y_{bu} \leq \mathrm{LHD_v} \ \forall v \qquad [5]$$

$$y_{bt} = 1 \ \forall b \mid Early_b = Late_b \qquad [6]$$

$$\sum_t y_{bt} \leq 1 \ \forall \ b \mid Late_b > T \qquad [7]$$

$$\sum_t y_{bt} = 1 \ \forall \ b \mid Late_b \leq T \qquad [8]$$

$$d\mathrm{u}_{tk}, \mathrm{dd}_{tk} \geq 0 \ \forall t, \mathrm{k} \qquad [9]$$

The objective function minimizes the deviation from the production targets for each ore type so that the mills can meet their respective production demands. Constraints [1] calculate the tons of each ore type mined per time period and the corresponding deviations from the specified production levels. Constraints [2], the vertical sequencing constraints between mining sublevels, preclude mining a production block under a given production block until operationally feasible. Constraints [3] and [4] enforce horizontal sequencing constraints between adjacent production blocks. Constraints [5] limit the number of LHDs active within a shaft group to the maximum allowable number. Constraints [6] place active production blocks into the production schedule. Constraints [7] preclude a production block from starting to be mined more than once during the time horizon if its late start date occurs beyond the maximum time horizon. Constraints [8] require a production block to start being mined at some point during the time horizon if its late start date falls within the time horizon. Constraints [9] enforce non-negativity and integrality of the variables, as appropriate.

Literature review

Open pit mining problems[1,2] are early applications of optimization to problems in the mining sector. Some of these models possess special structures that allow the optimization problems to be solved despite the primitive state of hardware and software at the time. Subsequently, another special class of optimization models, linear programs, is applied in underground mines, e.g., copper or coal mines [3–7]. Unfortunately, *these* models lack the ability to capture discrete decisions, e.g., whether or not to mine a given production block. In an effort to capture these discrete decisions, some models combine linear programming with simulation or manual intervention[8–12]. For example, a stochastic dynamic program determines long-term optimal generation levels for a coal- and gas-fired power station under a set of scenarios; the authors then use simulation to determine corresponding mining and stockpiling strategies[13]. Researchers recognize the need to incorporate discrete decisions in their models[14–16], but none of these models provide detailed optimal multi-period mine schedules. One such multi-period model[17] uses discrete decision variables to determine the location of processing facilities and whether a mine produces or not, but does not provide a detailed production schedule. Several prior models for production planning at Kiruna Mine do not yield adequate multi-period schedules in an operationally acceptable amount of time[18–20]. These models therefore either provide sub-optimal solutions or production schedules over a shorter timeframe than required. An integer-programming model provides a production schedule for a sublevel stoping operation at Stillwater Mining Company[21]. This model is the closest to ours, and yields near-optimal solutions to maximize revenue from mining platinum and palladium; however, the authors do not describe any special techniques to expedite solution time.

Increasing the efficiency of the optimization model

The size and structure of our model motivates our current work. MIP solution time increases exponentially with the number of integer variables. To determine a three-year production schedule requires 36 (time periods)*1173 (production blocks) = 42,228 variables. Given the mathematical structure of our model, especially the complicating sequencing constraints, the corresponding solution time can be hours, or even days. In a separate research endeavour[22], we reduce the number of integer variables by preprocessing the production data, carefully formulating the model, and implementing several algorithms to eliminate unnecessary variables. Relevant to our current discussion is the formulation of the model to make decisions for *machine placements*, rather than for *production blocks.* Because all production blocks within a machine placement must be mined continuously in a specific order, it suffices to define a binary variable indicating whether *machine placement* a starts to be mined in time period t, i.e., y_{at}, rather than defining a variable for whether or not each individual production block is mined. This change of variables requires nontrivial accounting to consider the amount of time required to mine each production block, and, given the number of production blocks in each machine placement, the amount of time to mine (some portion of) the machine placement. However, because there are far fewer machine placements than individual production blocks, using y_{at}, rather than y_{bt}, as the binary variable reduces the number of binary variables in our model by an order of magnitude. In the ensuing discussion, we use these new variables, y_{at}, rather than the variables that appear in the original formulation.

Despite prior work to expedite solution times, integer programs are notorious for their lack of robustness with respect to tractability, and even these advanced techniques fail to produce a five-year schedule in a timely fashion for all the mining scenarios we have encountered at Kiruna Mine. Motivated by new data sets that not only increase the size of the model, but also introduce symmetry (making various feasible solutions less distinguishable from one another), we explore additional techniques to reduce the search space, i.e., the set of feasible solutions from which we derive the optimal.

Specifically, we add redundant, but valid, constraints to our model. In other words, these constraints are not necessary to eliminate operationally infeasible solutions (because the constraints are redundant with those already in the model), but they are valid, i.e., they are satisfied by every feasible solution in the original model. Intuitively, this may appear to exacerbate problem tractability by (unnecessarily) increasing the size of the model. However, unlike the addition of variables, which provides more options (feasible solutions) the solution algorithm must explore, adding constraints reduces the number of feasible possibilities, thereby reducing the search space and expediting solution time. In fact, constraints [7] and [8] in the existing model already serve this purpose, but we can also add redundant vertical and horizontal sequencing constraints.

We add constraints between pairs of machine placements if both the early and the late start of each machine placement lie within the time horizon. Specifically, if we define a as a machine placement and a' as a machine placement whose mining start date is affected by a, $ES(a)$ and $LS(a)$ as the early and late start dates, respectively, for machine placement a, and assume that $ES(a) \geq ES(a')$, we can add a (redundant) vertical sequencing constraint of the form:

$$\sum_{t=ES(a)}^{LS(a)} t * y_{at} - \sum_{t'=ES(a')}^{LS(a')} t' * y_{a't} \geq ES(a) - ES(a') \qquad [10]$$

$$\forall a, a' \text{ affected by } a$$

In other words, given machine placement a starts to be mined in some time period t (where t lies between the earliest and latest possible times that a can start to be mined), machine placement a' can start to be mined no earlier than at some time period t' (where t' lies between the earliest and latest possible times that a' can start to be mined) less the difference between the early start times between the two production blocks, a and a', i.e., the required lag time after a starts being mined and before a' starts to be mined.

We can construct similar (redundant) horizontal sequencing constraints as follows, where the first set of constraints corresponds to left adjacency constraints and the second set to right adjacency constraints:

$$\sum_{t=ES(a)}^{LS(a)} t * y_{at} - \sum_{t'=ES(a')}^{LS(a')} t' * y_{a't} \leq lag_l(a) - 1 \qquad [11]$$

$\forall a, a'$ affected by a

$$\sum_{t=ES(a)}^{LS(a)} t * y_{at} - \sum_{t'=ES(a')}^{LS(a')} t' * y_{a't} \leq lag_r(a) - 1 \qquad [12]$$

$\forall a, a'$ affected by a

where $lag_l(a)$ and $lag_r(a)$ represent the maximum amount of time between when machine placement a starts to be mined and when machine placement a′ must correspondingly start to be mined for machine placements to the left and to the right of machine placement a, respectively. Analogous to the first set of constraints, these two constraints require machine placement a' to be mined within the required amount of time after machine placement a starts to be mined.

All three of these sets of constraints are met by using the original horizontal and vertical sequencing constraints, i.e., constraints [2]–[4] of the formulation shown earlier. However, the addition of constraints [10]–[12] to this original formulation helps expedite solution time.

Numerical results

We illustrate the effect on the solution time of imposing constraints [10]–[12] of our model by providing results for three separate scenarios. The first scenario uses the original model presented earlier. The second scenario replaces the sequencing constraints ([2]–[4]) in the original model with constraints [10]–[12]. The third scenario combines the original constraints with our redundant constraints, [10]–[12]. We use both actual data from LKAB's Kiruna Mine, and other data sets for which we modify demand data and the available number of LHDs to reflect realistic changes given the availability of each ore type in each machine placement and the availability of load haul dump units, respectively. The first four data sets possess a three-year time horizon and the fifth data set a five-year horizon. As is common with integer programming algorithms, our algorithm provides both the best objective function value it has found at any point in its search, and the best bound on the optimal objective function value, i.e., an objective function value at least as good as the optimal solution (and perhaps better). In our numerical study here, we attempt to solve all instances within a 4-hour time limit to within 5% of optimality, i.e., to within 5% of what is guaranteed to be the 'best possible' solution.

We implement our mixed integer program using the AMPL programming language[23,24], and the CPLEX solver, Version 7.0[25]. We run our model instances on a Sun Ultra 10 machine with 256 MB RAM. We report results in Table I.

Results are inconclusive as to whether adding constraints [10]–[12] or replacing [2]–[4] in the original model with constraints [10]–[12] produces faster solution times. However, the run time of the original model is clearly dominated by the introduction of the new constraints. While the run times for the model with the first two data sets exhibit modest improvements with the addition of the redundant constraints, the solution times for the model with the third data set are significantly reduced by adding the redundant constraints. We were unable to obtain a solution provably within 5% of optimality for the scenario with the fourth data set. However, the optimality gap is much lower with, than without, the redundant constraints. The most significant improvement occurs with the fifth data set. Without the redundant constraints, the algorithm fails to find a solution provably within 5% of optimality within the 4-hour time limit we impose. By contrast, using the new constraints, the algorithm finds a solution within 5% of optimality in less than 30 minutes.

Conclusions and extensions

We use integer programming to generate production schedules at LKAB's Kiruna Mine; however, the size of our model requires that we develop techniques to expedite solution time. By reformulating the model to reduce the number of integer variables, and by employing tightening constraints, in all but one case we test, we generate near-optimal three- to five-year schedules in less than an hour. Extensions to our research include continuing to evaluate our current solution methodology, and generating new techniques to even further reduce solution times for other types of data sets. We note that Kiruna Mine has adopted schedules that we have produced with our optimization model, and is now using optimization to aid in its schedule generation. We present a complete comparison of current practice and production scheduling at Kiruna Mine prior to the introduction of this optimization model elsewhere[26].

Acknowledgements

The authors would like to thank LKAB for the opportunity to work on this challenging project and for permission to publish these results. The authors would also specifically like to thank LKAB employees Hans Engberg, Anders Lindholm, and Jan-Olov Nilsson for providing information, assistance, and support in this endeavour.

References

1. LERCHS, H. and GROSSMANN, I.F. Optimum

Table I
Solution times for five sets of data using the three models: (i) the original model, (ii) replacing constraints [2]–[4] with [10]–[12], and (iii) the original model with constraints [10]–[12]

Scenario	Original model	Model replacing [2]–[4] with [10]–[12]	Model with [2]–[4] and [10]–[12]
Data Set 1	1100 seconds	1200 seconds	950 seconds
Data Set 2	1200 seconds	1500 seconds	750 seconds
Data Set 3	820 seconds	350 seconds	200 seconds
Data Set 4	14400 seconds (7.5% gap)	14400 seconds (5.7% gap)	14400 seconds (5.0% gap)
Data Set 5	14400 seconds (5.2% gap)	730 seconds	1300 seconds

design of open pit mines. Canadian Institute of Mining, January, 1965. pp. 47–54.

2. UNDERWOOD, R. and TOLWINSKI, B. A mathematical programming viewpoint for solving the ultimate pit problem. European Journal of Operational Research, vol. 107, no. 1. 1998. pp. 96–107.
3. WILLIAMS, J.K., SMITH, L., and WELLS, P.M. Planning of underground copper mining. 10th *International Symposium on the Application Computers and Operations Research in the Mineral Industry (APCOM) Proceedings*, Johannesburg, South Africa, 1992. pp. 251–254.
4. JAWED, M. Optimal production planning in underground coal mines through goal programming—A case study from an Indian mine. *24th International APCOM Symposium Proceedings*, Montreal, Quebec, Canada, 1993. pp. 43–50.
5. RAMIK, J. and HANELOVA, J. Optimization models of production of a coal-mining company, *Operations Research Proceedings*, Zimmerman, U., Derigs, U., Gaul, W., Morig, R.H. *et al*., (eds.). 1997. pp. 445–450.
6. PENDHARKAR, P.C. A fuzzy linear programming model for production planning in coal mines. *Computers & Operations Research*, vol. 24, no. 12. 1997. pp. 1141–1149.
7. WINKLER, B.M. and GRIFFIN, P. Mine production scheduling with linear programming—Development of a practical tool. *27th International APCOM Symposium Proceedings*, Royal School of Mines, London, United Kingdom, 1998. pp. 673–681.
8. WILKE, F.L. and REIMER, T.H. Optimizing the short term production schedule for an open pit iron ore mining operation. *15th International APCOM Symposium Proceedings*, Brisbane, Australia, 1977. pp. 425–433.
9. TANG X., XIONG, G., and LI, X. An integrated approach to underground gold mine planning and scheduling optimization. *24th International APCOM Symposium Proceedings*, Montreal, Quebec, Canada, 1993. pp. 148–154.
10. TOLWINSKI, B. and UNDERWOOD, R. A scheduling algorithm for open pit mines. *IMA Journal of Mathematics Applied in Business & Industry*. 1996. pp. 247–270.
11. WINKLER, B.M. Mine production scheduling using linear programming and virtual reality. *27th International APCOM Symposium Proceedings*, Royal School of Mines, London, United Kingdom, 1998. pp. 663–673.
12. KUCHTA, M. A database application for long term production scheduling at LKAB's Kiruna Mine. *30th International APCOM Symposium Proceedings*, Phoenix, Arizona, 2002. pp. 797–804.
13. BAKER, W.R. and DAELLENBACH, H.G. Two-phase optimization of coal strategies at a power station. *European Journal of Operational Research,* vol. 18, no. 3. 1984. pp. 304–314.
14. WINKLER, B.M. Using MILP to optimize period fix costs in complex mine sequencing and scheduling problems. *26th International APCOM Symposium Proceedings*, Pennsylvania State University, University Park, Pennsylvania, 1996. pp. 441–446.
15. SMITH, M.L. Optimizing short-term production schedules in surface mining: Integrating mine modeling software with AMPL/CPLEX. *International Journal of Surface Mining*. 1998. pp. 149–155.
16. TROUT, L.P. Underground mine production scheduling using mixed integer programming. *25th International APCOM Symposium Proceedings*, Brisbane, Australia, 1995. pp. 395–400.
17. BARBARO R.W. and RAMANI, R.V. Generalized multiperiod MIP model for production scheduling and processing facilities selection and location. *Mining Engineering*, vol. 38, no. 2. 1986. pp. 107–114.
18. ALMGREN, T. An approach to long range production and development planning with application to the Kiruna Mine, Sweden, Luleå University of Technology, Doctoral thesis number 1994:143D, 1994.
19. TOPAL, E. Long and short term production scheduling of the Kiruna iron ore mine, Kiruna, Sweden, Master of Science thesis, Colorado School of Mines, Golden, Colorado, 1998.
20. DAGDELEN, K., KUCHTA, M., and TOPAL, E. Linear programming model applied to scheduling of iron ore production at the Kiruna Mine, Kiruna, Sweden. *Society of Mining Engineers Annual Meeting*, Denver, Colorado, 1999, Preprint Number pp. 01–199.
21. CARLYLE, M. and EAVES, B.C. Underground planning at Stillwater Mining Company. *Interfaces*, vol. 31, no. 4. 2001. pp. 50–60.
22. NEWMAN, A., TOPAL, E., and KUCHTA, M. An efficient optimization model for long-term production planning at LKAB’s Kiruna mine, Working paper, Colorado School of Mines, 2002.
23. AMPL, Version 10.6.16, Bell Laboratories, 2001.
24. FOURER, R., GAY, D. and KERNIGHAN, B.W. *AMPL: A Modeling Language for Mathematical Programming*, Boyd and Fraser, Massachusetts, 1993.
25. CPLEX, Version 7.0, ILOG Corporation, 2001.
26. KUCHTA, M., NEWMAN, A., and TOPAL, E. Implementing a Production Schedule at LKAB’s Kiruna Mine, Working paper, Colorado School of Mines, 2002.