A Framework for Stope Sequence Optimization

John Manchuk

Centre for Computational Geostatistics (CCG)
Department of Civil & Environmental Engineering
University of Alberta

Underground mining is becoming more economically marginal as deposits become deeper and more challenging to excavate. New technology will make marginal deposits more feasible. A recent addition to aid in the economics of marginal underground mines is automatic stope sequence optimization. Choosing the best sequence for assigning equipment, blending ore and ordering stopes can be a very complex problem to undertake. Advances in combinatorial optimization algorithms have provided a means to address complex problems such as the sequential ordering problem for a set of stopes. This paper will explore the sequential ordering problem for a two dimensional panel of stopes with simplified constraints. The objective will be to maximize net present value. A framework for solving more complex problems is developed.

Introduction

Previous CCG research has focused on optimizing the geometry of a single stope. This work builds on those previous developments. The focus here is to establish an optimal or near optimal sequence of stope extraction.

Many methods of extracting ore in an underground mining context exist. The stope sequencing methods developed here are applicable to steeply dipping ore bodies that can be segregated into panels of stopes. The mining method used, possibly sublevel stoping or vertical crater retreat, must involve fairly consistent development procedures and stope dimensions. Natural variability of orebody geometry and grades leads to stopes of differing economic value. These grade variations together with mining costs leads to significantly different values for each stope. We must incorporate this with predicted changes in operating costs over time and/or depth and the value of the commodity of interest.

Two techniques are explored for a simplified version of the stope sequencing problem (SSP): simulated annealing (SA) and a probabilistic decision making heuristic (PDM). Simulated annealing was explored to provide results from a classical technique and provide a benchmark to compare other techniques. PDM is a more complex logic-driven technique that introduces precedence parameters. Maximizing net present value (NPV) is the objective. The simplified problem involves mining a steeply dipping ore body with development levels above each stope and extraction levels below. There is only one extraction crew, one stope preparation crew, and one development crew. Drifting rates are assumed constant. Stope preparation time is assumed to be a function of stope volume and extraction equipment operating parameters are limited to number of units, loading and dumping time and haul rate loaded and empty.

Background

Stope sequencing as an optimization problem is combinatorial in nature. Generally speaking, combinatorial optimization is concerned with choosing a best configuration of items or parameters for a particular objective (Papadimitriou and Steiglitz, 1998). Given the well known traveling salesman problem for example, the objective is to minimize total traversed distance between a set of cities that the salesman is slated to visit. The class of combinatorial optimization problems which stope sequencing falls under is referred to as the sequential ordering problem (SOP). This problem has been tackled with various algorithms including branch and bound (Hernadvolgyi, 2004) and ant colony optimization (Gambardella et al, 2000).

Optimization of underground mining operations is a recent area of interest. Previous works done included: optimization of equipment placement and blending schedules to minimize operating costs using mixed integer programming at the Kiruna Mine in Sweden (Kutchta et al, 2003); scheduling of basic mining operations including developing drifts and ramps, sampling, stope preparation and ore extraction with the objective of increasing production and estimating operating costs using mixed integer programming as well (Carlyle and Eaves, 2001); and, the use of a genetic algorithm to schedule a mine's extraction sequence from various stopes to achieve some specified objective based on each stopes mineral properties (Lindon et al, 2005).

This paper considers two optimization methods: (1) simulated annealing because it is well known and easy to implement for generating benchmark solutions; and (2) a logical decision making algorithm similar to branch and bound that avoids the need to generate a lower bound for a given sequence state.

Simplified Problem Development

Algorithm development will be done for a simplified problem. A two dimension panel of stopes applicable to steeply dipping ore bodies and consistent operations will be considered. Variables that will be considered from each stope include economic value, average grade(s), tonnage, depth, and preparation, mining and backfilling cost. All of these properties can be utilized to find optimal or near optimal stope sequences for logically driven algorithms; however, in the case of a random algorithm like simulated annealing, none of these parameters are important. Once a random sequence is chosen, these variables are simply used to calculate the objective and either accept or reject the sequence.

Operational aspects of the SSP involve development crew(s) for making stopes accessible, preparation and mining crew(s) for supporting, drilling and blasting ore and extracting stopes and crew(s) for backfilling completed stopes. For the simplified case, only one development crew and one mining crew were considered and the mining crew performs extraction and haulage of ore to dump sites, perhaps through the use of LHD's. In a more complex case, there may be multiple crews operating simultaneously and some form of secondary ore haulage to extraction points (LHD's for extraction and trucks for haulage). Order of operations for the case considered here are as follows, see Figure 1:

1. Make a stope accessible in time $t_a \leq t_e\text{-}t_p$. The constraint ensures constant flow of ore.
2. Prepare the stope in time t_p. Preparation includes:
    a. Adding required support.
    b. Drilling, loading, and blasting.

    c. Developing the extraction system.

3. Completely extract ore from the stope in time t_e.
4. Return to step 1.

Other operational parameters are crew specific and include such things as development rate along ramps, drifts and crosscuts, haulage rates and tonnage per load, equipment availability and utilization, operating costs, time and costs associated with relocating equipment to other drifts or stopes, and stope preparation time. All of these parameters are utilized in calculating costs and times associated with mining a panel of stopes for discounted value calculations. Mining constraints depend on the mining method, geological properties of the deposit and other operational parameters. Additional constraints make finding a solution to the SSP more difficult; however, some of these constraints decrease the number of feasible sequence combinations dramatically. Possible constraints include:

1. Each stope must be mined prior to mining the next. A more flexible system would be one in which a stope is only mined partially prior to equipment moving to a different stope.
2. Development can only take place towards one stope at a time per crew. Depending on the time available, development crews could develop to the next stope to mine as well as develop partially to a different area of the panel.
3. Mining and development crews cannot experience more than a specified amount of idle time.
4. Ore must be constantly fed to the processing plant. This is equivalent to having no gaps in time between mining stopes.
5. From the time mining begins on a stope until it is backfilled and has settled for some time, its neighboring stopes must stand as pillars.
6. Developing and/or mining immediately above or below a stope being mined is restricted for stability purposes.
7. Advancing development on the same level as a stope being extracted is restricted to minimize equipment interaction.

For the simplified case with only one mining and one development crew, constraints 1, 2 and 4 will be considered; a stope that is selected to be mined is completely mined prior to starting the next; the development crew only advances to a stope to be mined an nowhere else; and there can be no gaps in ore extraction.

The geometry of the problem is quite extensive if we consider it a continuous process, but it can be simplified by moving to a segmented approach. A complex panel may involve numerous stopes along strike, depth and possibly along dip. Required drifts and crosscuts to access, prepare and mine these stopes could be quite extensive as well. Various scenarios for ore drop locations also exist:

1. Ore is hauled to one or multiple common ore passes. In this case, LHD's would acquire a load, haul it to the ore pass location and return to the stope.
2. Ore is hauled along crosscuts only and loaded into secondary haulage equipment. This case would involve more parameters for the secondary equipment and its interaction with LHD's (haul rate, tonnage per load, queue time, LHD wait time, etc).

For the simplified case only panels with stopes along strike and depth were considered. Drifts were segmented up based on the stope configuration. If a segment must be mined to make a stope accessible, the entire segment must be completed before preparation and mining operations can commence. Ore is hauled to one common ore pass located at one end of the panel. A summary of parameters, constraints and assumptions for the simplified case to be explored in this paper is as follows, aided by Figure 2:

- Multiple stopes along strike and depth comprise a panel.
- One mining crew and one development crew are considered.
- Stopes must be mined sequentially (cannot mine stopes partially) and development only advances towards stopes to be mined.
- No time gaps in mining are permitted.
- One common ore pass exists at one end of the panel. Development starts from this location and progresses into the panel.
- Development drifts are segmented according to the stope configuration.
- There are no restrictions on mining or development while a stope is being mined (neighboring stopes can be mined and development can advance anywhere in the panel).
- Time to transport crews to different locations is negligible.
- Economic and operations related parameters remain constant over time and through space.

Algorithms

Simulated Annealing

The simulated annealing algorithm works by first randomly selecting a feasible stope sequence, then perturbing that sequence randomly in an attempt to increase NPV. Perturbations are accepted according to either an increase in NPV or a probability of the new sequence leading to a more optimal one with future perturbations. Initial system temperature is set as the NPV from the first random sequence and for each iteration is decreased: For accepted perturbations a temperature reduction factor of 0.99 was used and for rejected ones, a factor of 0.999 was used. Fast or even moderate reduction rates resulted in early program completion and suboptimal NPV as compared to those for slow temperature reduction rates. General flow of the algorithm is as follows:

1. Randomly choose an initial feasible stope sequence.
    1.1. Calculate its NPV and set as the initial temperature.
2. While the temperature is greater than a user specified tolerance:
    2.1. Randomly pick a stope from the panel (not the stope to be mined last from the previous sequence).
    2.2. Mine up to but not including the chosen stope following the previous sequence.
    2.3. Create a list of feasible stopes given the current state of the panel.

2.4. Randomly pick another stope from the feasible list.

2.5. Swap the positions of the stope picked in 2.1 with that of 2.4.

2.6. Mine the remaining stopes given the updated sequence. If the new order is not feasible, the sequence will be augmented so that it is feasible with as little change as possible.

2.7. Calculate the updated NPV and the probability of success:

$$Prob = \exp\left(-\frac{\left|NPV_k - NPV_{k-1}\right|}{T_{k-1}}\right)$$

If $NPV_k > NPV_{k-1}$ or the probability is greater than 0.9, the temperature is reduced by 0.99 and NPV is updated. Otherwise, the temperature is reduced by 0.999 and the previous best sequence is restored.

3. Done.

Probabilistic Decision Making (PDM)

PDM is a more involved algorithm for logically determining a stope sequence rather than randomly. It utilizes properties of each stope at each state of the mining sequence to calculate a probability that a stope would be the next best to mine. Stope properties are anything that can be quantified numerically and are deemed important to determining a mining sequence. Depending on the mining method and constraints imposed, properties could be any of the following:

1. Stope profit.
2. Time to completely extract a stope.
3. Time to make a stope accessible and ready for mining.
4. Cost to make a stope accessible and ready for mining.
    a. 3 and 4 involve mining drifts and crosscuts and possibly ramps to access a stope as well as time and cost for preparation.
5. Loss in potential profit from stopes made inaccessible by mining a particular stope. Determining inaccessible stopes depends on the constraints.
6. Gain in potential profit from stopes made accessible by mining a particular stope. These would include stopes passed by development to reach the stope of interest.
7. Time to move equipment to required areas.
8. Cost to move equipment to required areas.
    a. 7 and 8 involve moving development equipment from their current location to a new heading to make the next stope accessible. Time and cost to move mining equipment can also be considered.
9. Others…

For the simplified case, properties 1 to 4 were utilized for decision making. Each stope is also associated with a weight vector, $\boldsymbol{w}$, with the same length as its property vector, $\boldsymbol{F}$. These weights are used for calculation of probabilities (see Equation 1). For decision k, the feasible stope having the highest probability is chosen to be next in the sequence. It should also be noted that the properties, which can take on very large values, are normalized to sum to one.

$$P_{ij} = \frac{\exp\left(w_{ij}^T F_{ij}\right)}{\sum_{i=1}^{m}\sum_{j=1}^{n}\exp\left(w_{ij}^T F_{ij}\right)} \tag{1}$$

$$\begin{aligned} \textit{where}\quad & P_{ij} \text{ is the probability of being a good decision} \\ & w_{ij}^T \text{ are the weights for stope } ij \text{ in the panel} \\ & F_{ij} \text{ are the properties for stope } ij \text{ in the panel} \end{aligned}$$

Rationale behind the properties selected is based on the nature of calculating net present value. Revenue from the stope was selected because we would like to mine more valuable stopes earlier in the sequence. The remaining properties were selected to offset the revenue. Time to mine the stope along with time to access it will increase the degree of discounting incurred, which should make the stope less desirable. The premise of PDM is to weight these properties in such a way that decisions are made leading to an optimal or near optimal sequence.

Choosing initial weights is done based on their effect being positive or negative towards NPV. Property 1, stope profit, is considered positive and was given an initial weight of 0.5. Properties 2, 3 and 4 are considered negative and received weights of -0.5. Since these weights will not lead to the optimal sequence, PDM is a learning process. Weights are updated according to a decision's effect on the final NPV of the panel. Calculating the NPV for PDM to attempt to learn and improve upon was done using a greedy algorithm. An initial order and NPV is determined by choosing a sequence based purely on profit. For a particular decision *k*, the stope providing maximum return is chosen for position *k* in the sequence.

There are two stages to the learning process: (1) decisions are made and weights are updated with the greedy NPV as the goal, and (2) weights are used to calculate probabilities which are used to choose pairs of feasible stopes to exchange order. Both processes rely on the same stope properties. For the second process, weights are updated if the change leads to no improvement in NPV.

Updating weights is accomplished using a steepest descent method. Classic steepest descent is designed to find a local minimum of a function, but in this case, we only need to update the weights until the probability of choosing a particular stope is made smaller than the next best decision. The gradient for Equation 1 is defined as follows:

$$\frac{\partial P_{ij}}{\partial w_{ijl}} = \frac{\left(\sum P - P_{ij}\right)\cdot F_{ijl}\cdot \exp\left(w_{ijl}F_{ijl}\right)}{\left(\sum P\right)^2} \tag{2}$$

$$\begin{aligned} \textit{where}\quad & \sum P \text{ is the sum of all probabilities} \\ & F_{ijl} \text{ is factor } l \text{ for stope } ij \\ & w_{ijl} \text{ is weight } l \text{ for stope } ij \end{aligned}$$

The probabilistic decision making algorithm is as follows:

1. Determine initial NPV using the greedy algorithm.

2. Initialize all weights.
3. Until all stopes receive an order:
    3.1. Create a list of feasible stopes for decision *k*.
    3.2. Calculate stope properties for the feasible stopes.
    3.3. Calculate probabilities via Equation 1.
    3.4. Choose the stope with the highest probability.
    3.5. Calculate the new NPV by attempting to replicate as close as possible the previous best order.
        3.5.1. If the new NPV is better than previous, update the best order and NPV variables and move on to decision $k + 1$.
        3.5.2. Otherwise perform gradient descent until this stope is not chosen and retry decision *k*.
4. While stopping criteria is larger than input tolerance:
    4.1. Build a matrix of factors for the current best order.
    4.2. Calculate probabilities using those factors.
    4.3. Choose the stope with highest probability; refer to it as stope A.
    4.4. Build a list of stopes whose order can be feasibly swapped with stope A's order.
    4.5. While New NPV < Current NPV or all possible swap combinations are attempted:
        4.5.1. Calculate probabilities of stopes in the swap list.
        4.5.2. Choose that with the highest probability; refer to it as stope B.
        4.5.3. Swap the order of stope A with B and calculate the NPV of the new order.
        4.5.4. If New NPV < Current NPV, update the weights applied to stopes in the swap list until stope B is not the best choice.
    4.6. If all possible swaps were attempted and no improvement in NPV was found, update the weights for stope A until it is not the best choice.
    4.7. If NPV is improved, update the best order and NPV.
    4.8. Update the stopping criteria
5. Done.

Test Runs

Preliminary testing and a comparison of the two methods described were done using various sized actual panels ranging from 19 stopes over three levels to 70 stopes over eight levels. Table 1 summarizes various aspects of the runs. These two methods provide similar final net present values. Initial NPV for simulated annealing is the first random order it selects, which is sub optimal when compared to the initial order selected by PDM. The initial order for PDM is selected using the greedy algorithm mentioned above.

Table 1: Sequence run results with profit values in millions and time in seconds.

Stopes	Simulated Annealing			PDM			ΔP_f	ΔT
	P_i	P_f	Time	P_i	P_f	Time		
19	9.528	9.594	1.98	9.588	9.595	1.88	0.000	0.10
29	7.744	7.829	3.64	7.826	7.829	5.75	0.000	-2.11
59	10.578	10.972	12.98	10.934	10.949	48.08	0.023	-35.10
70	11.694	12.064	16.03	11.952	12.004	88.61	0.061	-72.58

Conclusions

Providing a means to solve the combinatorial optimization problem of stope sequencing is not trivial. Approaching the problem from a simplified perspective has provided a framework for future work in advancing this area of research. It was shown that either a random algorithm such as simulated annealing or a logic-driven one can be implemented. It is difficult to say which approach would find a better solution upon increasing the complexity of the underground mining scenario; however, non-random algorithms will likely be more efficient. Future work consists of advancing this framework into a more functional application and attempting other algorithms along the way including branch and bound and local search heuristics.

References

Carlyle, W.M. and Eaves, B.C., *Underground Planning at Stillwater Mining Company*, Interfaces, Vol. 31, No. 4, July/August 2001, pp. 50-60

Gambardella, L.M. and Dorigo, M., *An Ant Colony System Hybridized with a New Local Search for the Sequential Ordering Problem*, Informs Journal on Computing, Vol. 12, No. 3, 2000

Hernadvolgyi, I.T., *Solving the Sequential Ordering Problem with Automatically Generated Lower Bounds*, Department of Information and Communication Technology, University of Trento, Technical Report DIT-04-018, 2004

Kuchta, M., Newman, A. and Topal, E., *Production scheduling at LKAB's Kiruna Mine using mixed-integer programming*, Mining Engineering, Vol. 55, No. 4, April 2003, pp. 35-40

Lindon, L.F., Goforth, D., Wageningen, A., Dunn, P., Cameron, C., and Muldowney, D., *A Parallel Composite Genetic Algorithm for Mine Scheduling*, Artificial Intelligence and Soft Computing Proceedings (481), 2005

Papadimitriou C.H. and Steiglitz K., *Combinatorial Optimization Algorithms and Complexity*, Dover Publications Inc., 1998

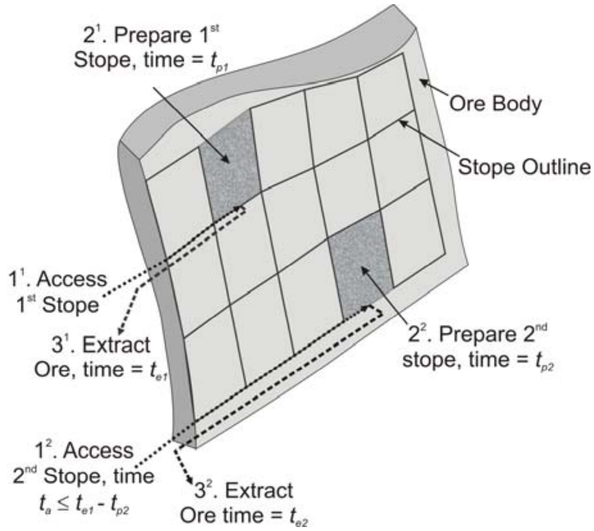

Figure 1: Order of operations for the simplified stope sequencing problem.

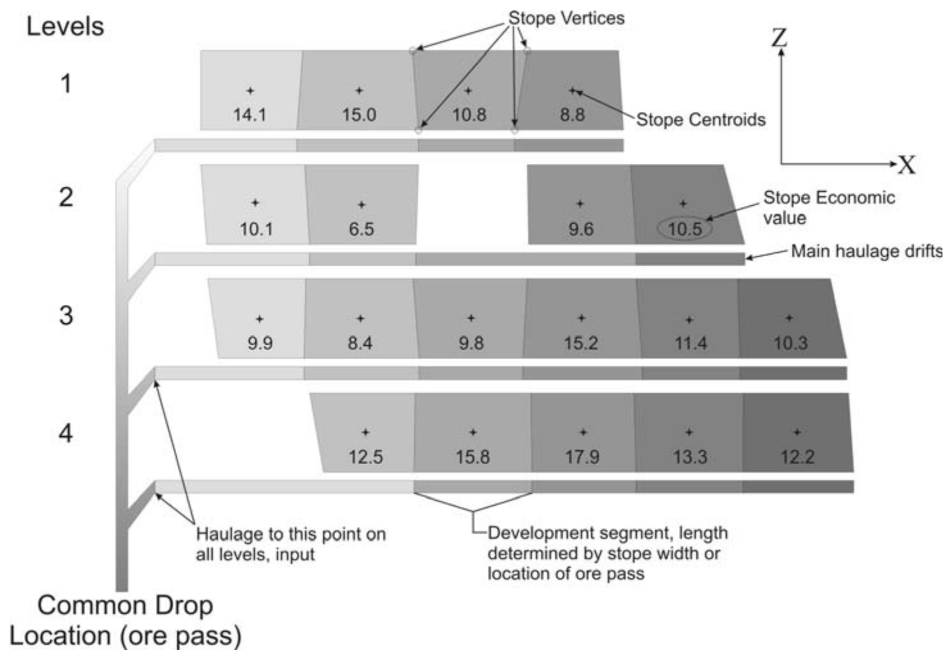

Figure 2: Simplified panel of stopes.