An improved meta-heuristic approach to extraction sequencing and block routing

by Y.A. Sari* and M. Kumral*

Synopsis

Mine production scheduling can be solved through many different techniques that have the drawbacks of either producing sub-optimal solutions or taking a long time. In this paper, a new approach based on a meta-heuristic is proposed. Meta-heuristic approaches use processing, inference, and memory at the same time in order to learn how to improve the solution. Different meta-heuristic techniques and their applications to mine production scheduling are discussed. A meta-heuristic approach, a combination of heuristic memory and simulated annealing, as demonstrated by means of a case study, takes a sub-optimal solution and improves it over time; thus it provides the best solution that it finds in the given time.

Keywords

Simulated annealing, mine production scheduling, open pit mining, mine planning, heuristic memory.

Introduction

Mineral deposits are most commonly represented by a block model that divides the orebody into a three-dimensional array of blocks. Each block consists of a cluster of similar characteristics such as rock type and ore grade, and has attributes such as tonnage of ore contained within the block and an expected economic value (Bley *et al.*, 2010). For each block, the mine production scheduling problem consists of the decisions of (1) whether to mine a block, (2) when to mine that block, and (3) how to process the mined block. The overall objective is to maximize the net present value (NPV) while meeting feasibility constraints such as production, blending, sequencing, and pit slope (Dagdelen, 2001).

Three main sub-problems of scheduling are the determination of production rates, discrimination between ore and waste, and block sequencing (Kumral, 2013a). These problems are interdependent; one sub-problem cannot be solved if the others have not been solved previously. However, in common applications, production rates are usually assumed and the other sub-problems are solved under this assumption (Menabde *et al.*, 2004; Nehring *et al.*, 2010; Asad and Topal, 2011). This leads to sub-optimal results. Our approach introduces a concept of cut-off range,

which regards the cut-off grade as guidance and optimizes it within the range provided. This is a step toward simultaneously optimizing production rates along with process destination discrimination and extraction sequencing.

Exact methods such as mixed integer programming (MIP) have been used for the block sequencing problem to obtain an optimal result for various cases (Kumral, 2013b; Little *et al.*, 2013; Nehring *et al.*, 2012; de Carvalho Jr. *et al.*, 2012) and yields a deterministic plan. However, MIP suffers from certain drawbacks. The size of the problem increases exponentially as the level of complexity (such as multiple metals, process destinations, rock types) increases (Rothlauf, 2011). To overcome the data size problem in MIP, block aggregation is suggested (Tabesh and Askari-Nasab, 2011; Topal, 2011) but naturally, this results in loss of optimality. Also, given that the block model is based on drill-hole data but is usually generated by geostatistical simulation, it is impossible in practice for the generated schedule to be optimal. Considering the amount of time MIP takes with large data-sets and that MIP is unnecessarily precise in our case, a faster, approximately-optimal algorithm is much more suited to the practical need.

Another widely used exact method is the Lerchs-Grossman algorithm (Lerchs and Grossman, 1964), which yields the ultimate pit. This is an algorithm based on graph theory that converts each block to nodes. Although faster than MIP, in addition to the problems in MIP, when using Lerchs-Grossman algorithm it is difficult to assign varying pit slopes at different points and determine mining and processing capacities for each period. Dagdelen and Johnson (1986) attempted to handle the

---

* *McGill University, Department of Mining and Materials Engineering, Montreal Canada.*

capacity constraints problem by incorporating the Lagrangean multiplier. The selection of the Lagrengean multiplier is a significant problem and the viability of the sequence generated depends on this selection. There is no clear way to determine the multiplier such that the NPV of the project is maximized.

Meta-heuristic approach to the mine production scheduling problem

In this research, simulated annealing (SA) meta-heuristic with addition of heuristic memory is utilized to solve mine production scheduling. The addition of heuristic memory helps to reduce the randomness of SA and improves computational efficiency. Heuristic memory learns the path of search in SA in such a way as to accelerate escape from local optima. As such, this addition can be seen as the incorporation of machine learning into the optimization process. Machine learning takes existing data a step further by automatically learning and improving the performance based on the data (Witten and Frank, 2005). Machine learning consists of many different techniques based on mathematical and empirical methods. These methods can be used to enhance the optimization and are especially easy to integrate with meta-heuristic approaches.

The application of SA to the mine production scheduling problem was developed by Kumral and Dowd (2005). This approach gradually improves an initial non-optimal solution by making several changes at each step and observing the effects of the changes. Although reaching the near-optimal solution takes time, the advantage of this technique is that it can be stopped at any time to obtain the most profitable solution so far.

The application of genetic algorithms was first introduced by Clement and Vagenas (1994). Based on the principles of natural selection, multiple feasible solutions are mixed by involving randomization. Similar to SA, solutions are improved gradually and the process can be stopped to obtain the best solution so far.

Ant colony optimization is a population-based metaheuristic method first developed by Dorigo and Birattari (2010) to imitate the foraging mechanism of ants. Ant colony optimization was proposed to solve the mine production scheduling problem by Sattarvand and Niemann-Delius (2013), Sattarvand (2009), and Shishvan and Sattarvand (2015). Using the Lerchs-Grossman method to produce an initial solution, the schedule was improved through iterations based on pheromone trails.

Reinforcement learning is similar to SA and genetic algorithms in terms of being an algorithm for searching the parameter space using the concept of reward; which in our case will be the improvement in the NPV. However, it yields better immediate results by applying a trial-and-error search having a memory-like system by incorporating historical error into its search mechanism (Sutton and Barto, 1998). Combined with dynamic programming, this type of learning can be used to adapt incoming updated information, for example during mine exploration.

Bayesian inference assumes the quantities of interest and parameters have an underlying probability distribution. By combining these probability distributions and observed data, optimal decisions can be made (Mitchell, 1997). Bayesian

learning can be used to estimate the parameters, their relations to other parameters, and update their values with the incoming new drilling data.

Method

SA was developed initially by Kirkpatrick *et al.* (1983) and Cerny (1985). The method was applied to open pit mine production scheduling by Kumral and Dowd (2005) and Kumral (2013a) by the following steps:

Step 1: Start with a non-optimal feasible solution
Step 2: Select a portion of the blocks
Step 3: Possibility 1: modify ore-waste discrimination. Change ore blocks to waste, and waste to ore by some probability
Possibility 2: modify the period of the given block to a previous or following period by some probability
Step 4: Recalculate NPV for the newly found solution
Step 5: Apply the Metropolis criterion as the acceptance criterion; accept the new solution with the probability yielded by Metropolis
Step 6: If the NPV has not increased for last n steps, terminate. Otherwise, go to Step 2.

The Metropolis criterion (Metropolis *et al.*, 2004), shown in Equation [1], is a criterion that takes two solutions and a temperature T as inputs and outputs a probability of acceptance between 0 and 1, where E_0 is the current solution's NPV and E is the newly found solution's NPV.

$$\min\left(1, exp\left\{-\frac{(E-E_0)}{T}\right\}\right) \tag{1}$$

T should be chosen high at first and then decreased slowly. If T decreases slowly enough, theoretically a global minimum will be reached (Lundy and Mees, 1986). According to how the Metropolis criterion is set up, at higher temperatures the criterion tends to accept solutions that are not improving as well as those are improving. This stage is called 'exploration of the parameter space', as shown in Figure 1a. As T is lowered, there is less chance of accepting solutions that are not improving. If T is not decreased slowly enough, there is a chance of becoming stuck at a local minimum as shown in Figure 1b.

a) Searching for solutions in parameter space (Red arrow: current position in the search, Grey arrow: search direction)

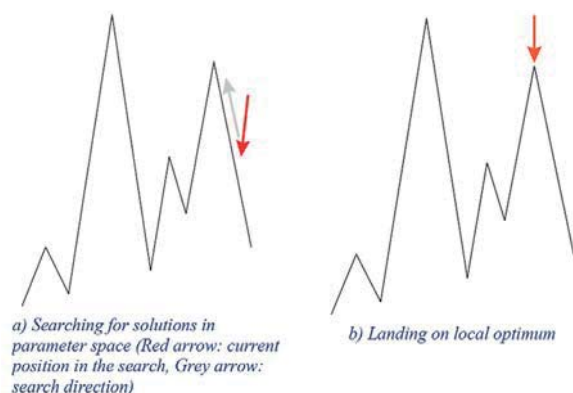b) Landing on local optimum

Figure 1—Parameter space

An improved meta-heuristic approach to extraction sequencing and block routing

As SA takes time, ideally the initial solution should provide very fast, although sub-optimal, results. The ranked positional weight (RPW) algorithm is a heuristic algorithm that draws a downward cone from each block and the block gains a score according to the economic values of the blocks in the downward cone (Gershon, 1987b). This approach follows the logic that if a block is underlain by a valuable block, it should gain more score as the removal of this block leads the way to the underlying valuable block. After the scoring has been completed, a schedule is generated such that starting from the first level, the highest, scored blocks will be mined. The RPW fits our purpose well because it produces a feasible solution rapidly.

A computer program was written to perform RPW and SA as demonstrated in Algorithm 1 to perform mine production scheduling. First, RPW is run to generate an initial feasible, sub-optimal solution. Then this solution is transferred to SA, which needs an initial input. SA gradually improves this solution at each iteration and outputs the result. A feasible solution respects the slope constraints, mining capacity constraints, and processing capacity constraints.


Algorithm 1 Mine Production scheduling Using Ranked Positional Weight and Simulated Annealing

Input: Grade of each block
    Determine *cutOff* grade using Osanloo and Ataei's method(Osanloo & Ataei, 2003)
    function RANKED POSITIONAL WEIGHT(*blockGrades, cutOff*)
        Call Ranked positional weight algorithm to produce initial output
        return initialOutput
    end function
    function SIMULATED ANNEALING(*blockGrades, initialOutput, cutOff*)
        Call Simulated annealing algorithm to optimize the initial output
        return finalOutput
    end function
Output: finalOutput

In this paper, in addition to SA, the developed SA variant method with memory is used to solve a mine production scheduling problem. In SA, only improvement is tracked and the decisions are made based on improving the objective function. Thus, SA is memoryless (Glover and Kochenberger, 2003). Tabu search attempted to improve SA by creating a dynamic list of forbidden solutions, thus introducing a concept of memory (Glover, 1989, 1990). However, this is very specific and limited. Tabu search only attempts to decrease re-visitation of the same solutions; it does not attempt to utilize the information in the solutions in some way.

Our proposed SA variant method, improved SA with heuristic memory, deducts information from the 'big data' produced by SA through inputting a heuristic (a quantifiable component of the solution that is thought to influence the objective function) and recording the heuristics value and the corresponding objective functions value. This method adds a memory on top of SA, with the intent of making it faster to find the global optimum. The representation of heuristic memory-added SA can be followed through the pseudocode demonstrated in Algorithm 2. The algorithm contemplates whether the provided heuristics indeed have an effect on the objective function by collecting data and looking at the relationship between the heuristic and objective function. If the heuristic has an effect, the value of the heuristic, along

with a balancing parameter k_1, is also added into the objective function to increase its effect. With this approach, if there is a known or reasoned important component of the problem, it can be put forward rather than performing a wholly random search. A comparison of the SA flow chart and SA with the heuristic memory-added flow chart is given in Figure 2.

Algorithm 2 Improved Simulated Annealing by Adding Heuristic Memory

Input: Define n measurable heuristics suspected to influence the objective function, stored in array $h[n]$
    while SA produces a solution **do**
        for $i = 0$ to n **do**
            Save the value of $h[i]$ and the objective function for the current solution
            Perform function fitting using all recorded values of $h[i]$ vs objective function for all hitherto obtained solutions
        end for
        Objective function is modified to be submitted to Metropolis criterion such that $objective function \leftarrow objective function + k_1 \times fitted function \times heuristic$
    end while
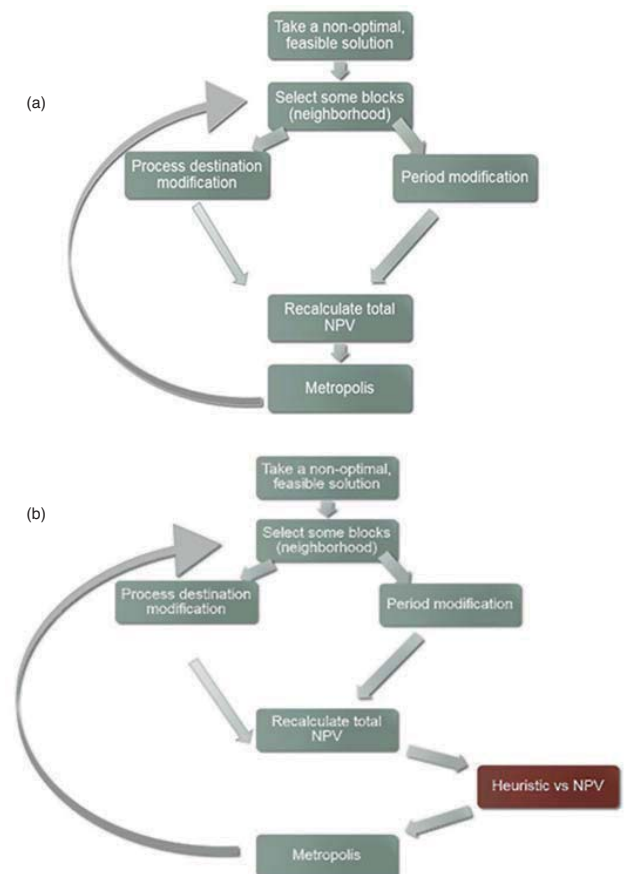
Figure 2—(a) Flow chart of SA from Kumral and Dowd (2005). Randomly selected blocks have either their process destination or period modified. If feasible, the new model's NPV is calculated and the objective function is evaluated by the Metropolis criterion. (b) Flow chart of modified SA with heuristic memory added. The modification is that after a new model is obtained and its NPV is calculated, the heuristic *vs* NPV data is saved. An updated objective function is submitted to the Metropolis criterion that maximizes NPV, pushes the number of blocks to capacity, and at the same time maximizes or minimizes the heuristic, depending on how it influences the NPV

An improved meta-heuristic approach to extraction sequencing and block routing

The program is able to work with multiple metals, process destinations, and rock types. Other details of the program are as follows:

➤ *Cooling schedule*—If initial temperature is too high, all new solutions are accepted. This will lead to undirected search. If initial temperature is too low, only improved solutions will be accepted and the annealing process will be reduced to a local search. Therefore, the initial temperature is set by taking the first two solutions and finding T in Equation [1] such that the equation will be equal to 0.5. This sets T such that, in the beginning, a solution will have a 50% chance of being accepted even if it is not improving. Decrementing T is accomplished by $T \leftarrow T \times 0.9999$ to ensure it decreases slowly enough to accept more solutions. Each time a fixed number of solutions are found (40 solutions), T is updated as described

➤ *Stopping criteria*—There are two conditions that can stop the annealing loop:
  • When the Metropolis Criterion does not accept the solution for a pre-set, empirically selected amount of iterations (in our case, four iterations)
  • When the program loops for a user-set amount of value. The second condition exists to produce solution under limited time. However, the longer the program is allowed to run, the better the results

➤ *Maximum number of solutions at each temperature*—This is a parameter that sets the number of generated solutions before decreasing the temperature. This should depend on the size of the data, so in our program we set it to 200 solutions

➤ *Cut-off range*—SA uses the guidance of the cut-off grades. However, it does not adhere to them strictly. During the generation of transition destinations process, the cut-off range is used to decide to which extent the blocks out of the limits of the cut-off grade could be accepted. This parameter may have a major effect on the results. If set high enough, it can remove cut-off grade boundaries altogether

➤ *Number of iterations*—SA terminates either when there is no improvement or when the given number of iterations is reached. Mining problem sizes are very large and thus the number of iterations is usually reached sooner than settling on the ideal solution. This parameter should be selected as large as possible as time permits

➤ *Short-coming process blocks effect*—This parameter is used as the balance between maximizing NPV and satisfying capacity constraints in the objective function. The parameter specifies how important it is to fulfill the process capacities. This value ranges between zero where it is not considered and unity where this criterion is all that matters

➤ *Mining cost adjustment factor*—The modifiable mining cost adjustment factor (MCAF) is used to reflect the increased cost of transport in deeper levels of the deposit. MCAF is entered by the user and the MCAF is added to the mining cost using the Equation [2].

$$\text{miningCost} \leftarrow \text{miningCost} + \text{level} \times \text{MCAF} \qquad [2]$$

➤ *Heuristic memory*—For each heuristic, the heuristic value and the objective function value are stored. When enough data is produced, a function-fitting method (Equation [3]) is performed to deduct information of how this heuristic influences the objective function. In our case, the number of blocks was the heuristic used and the fitting function was linear regression. This influence, along with a balancing parameter (k), is included in the objective function. The balancing parameter depends on the coefficient of determination, R^2, of the fitting function. Possible heuristics include the number of blocks, number of ore blocks, block grade *versus* process destination, coordinates of the main ore clusters, mine depth, and mine life.

$$\text{Objective Function} \leftarrow \\ \text{Objective Function} + heuristic \times R^2 \times k \qquad [3]$$

Case study

To demonstrate an application of meta-heuristic optimization on mine production scheduling, a program has been written using SA with the heuristic method approach.

The case study considers a copper and molybdenum deposit generated from a public drill-hole data-set in http://www.kriging.com/datasets/ Using sequential Gaussian simulation, a 3D block model of 595 046 blocks was created, where each block is 10×10×10 m in size. The mining company has one waste dump and three process destinations (low-, middle-, and high-grade processing), where the ore is processed by different procedures and thus their costs and recoveries are different. The slopes are 45 degrees in four directions (north, south, east, and west). Parameters for the case study are given in Table I. With 595 046 blocks, four periods, and four total destinations there are 595 046 × 4 × (4 + 1) = 11 900 920 decision variables. In the calculation, destinations are incremented by one because the the decision can also be taken not to extract the block.

Cut-off grades were calculated using the method of Osanloo and Ataei (2003) for finding the equivalent cut-off grade for multiple metal deposits, yielding 0.4859%, 0.6006%, and 0.7257 % respectively for each process. First, the RPW algorithm (Gershon, 1987a,b) was run to output a sub-optimal initial result. This result was input to the SA and SA with heuristic memory as an initial solution. All solutions respect the slope, mining, and process capacity constraints.

The resultant NPV of each algorithm is given in Table II. Using SA improved the RPW results by $75 743 914, which is 4.70%. SA with heuristic memory, on the other hand, improved the NPV by $77 386 239, which is a 4.80% improvement, when run for the same amount of time.

Figures 3 and 4 show various cross-sections of the orebody. These figure also compare the RPW and SA outputs, with each colour corresponding to an extraction period (1: light blue, 2: green, 3: orange, 4: red, dark blue: not extracted). It can be seen from these figures that compared to the RPW algorithm, SA is inclined to mine the blocks in the earlier periods to increase the NPV. However, the SA results
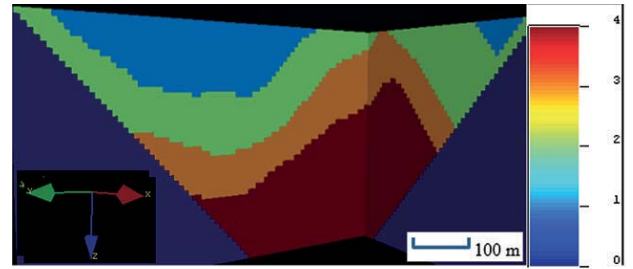
Table I

Parameters of the case study

Values	Explanation
10 10 10	Block dimensions (m)
98 132 46	No. of blocks in x, y, z directions
4	No. of total destinations
3	No. of ore destinations
1	No. of waste destinations
4	No. of periods
2	No. of rock types
3 3 3 3	Mining cost of each destination rock type 1 ($ per ton
4 4 4 4	Mining cost of each destination rock type 2 ($ per ton)
0 15 30 50	Mineral processing cost of each destination rock type 1 ($ per ton)
0 18 38 53	Mineral processing cost of each destination rock type 2 ($ per ton)
0 60 70 80	Sales cost of each destination ($/ton concentrate)
5 5 5 5	Specific gravity of each destination (t/m^3)
60	Mining capacity (ore and waste) (in thousands of number of blocks)
25 20 15	Processing capacity of each destination (in thousands of number of blocks)
0.01	Cut-off range
2	Number of metals (Cu and Mo)
10000 30000	Ore price ($ per ton) (Cu and Mo)
0 40 70 95	Recovery for each destination metal 1 rock type 1 (%)
0 30 40 75	Recovery for each destination metal 2 rock type 1 (%)
0 40 80 95	Recovery for each destination metal 1 rock type 2 (%)
0 30 40 75	Recovery for each destination metal 2 rock type 2 (%)
0.1	Discount rate
300	Number of iterations
0.9	Shortcoming process block effect
0.1	MCAF
0.600 0.680 0.780	Copper grade requirement (%)
0.043 0.063 0.010	Molybdenum grade requirement (%)

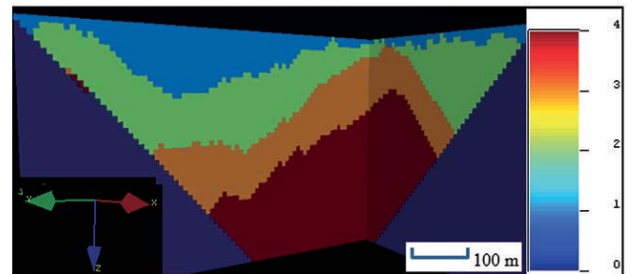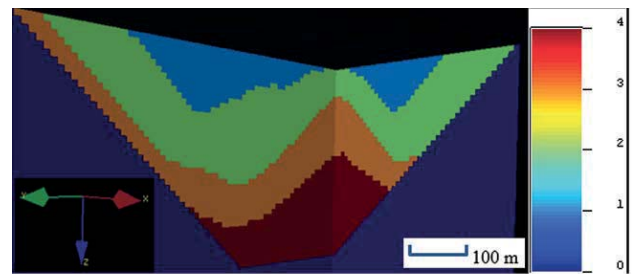(a) Output from Ranked Positional Weight

Figure 3—Cross-sections (sections: 48, 52, 0)
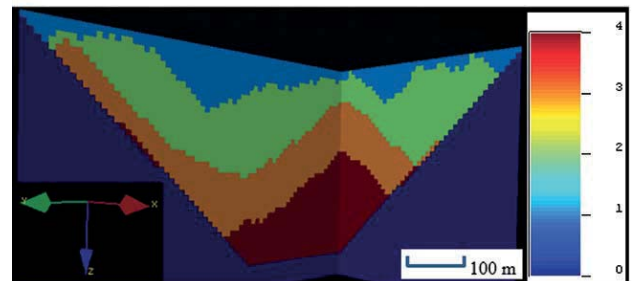

(a) Output from Ranked Positional Weight

Figure 4—Cross-sections (sections: 56, 65, 0)

Table II

Resultant NPVs for each applied technique

Algorithm	Resultant NPV
Ranked positional weight	$1 613 205 645
Ranked positional weight + simulated annealing	$1 688 949 559
Ranked positional weight + simulated annealing with heuristic memory	$1 690 591 884

look less smooth than the ranked positional algorithm's result. This is mainly because of the structure of annealing, where blocks are switched between the periods one by one, causing the sections to look rugged.

Figure 5 shows the average grade of Cu and Mo at each period for each process, as well as the number of blocks extracted at each period. These results belong to the SA

(a) Low grade process
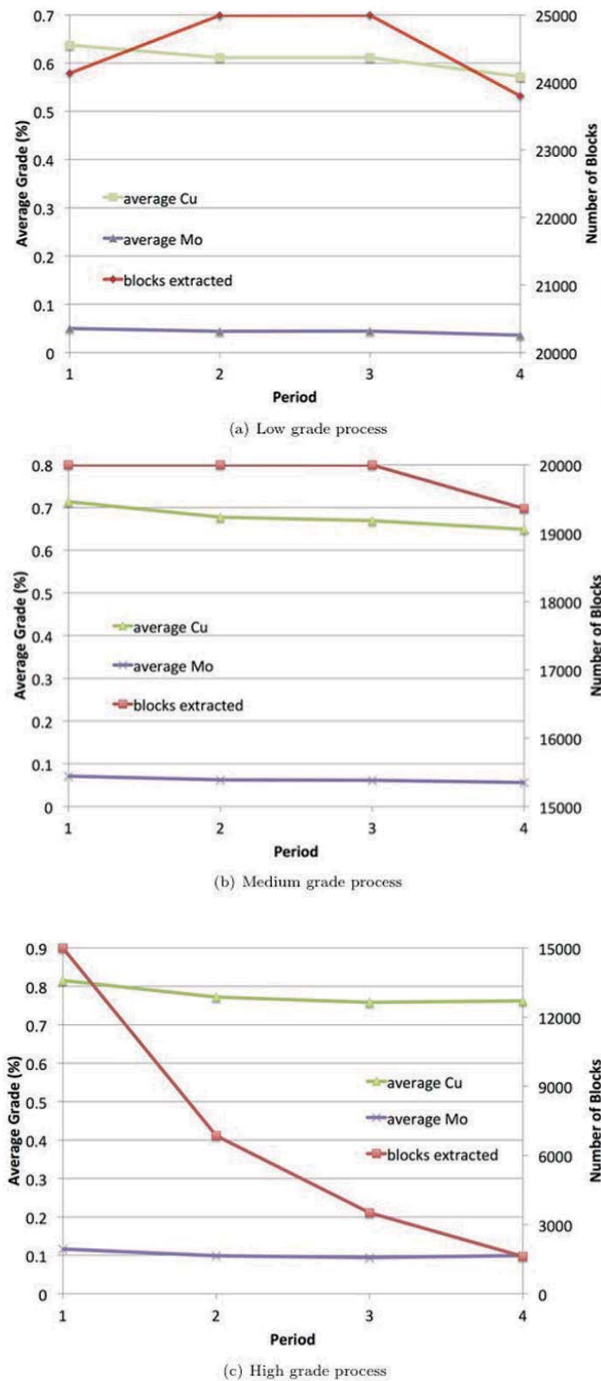

(b) Medium grade process

(c) High grade process

Figure 5—Average grades and number of blocks in each destination and period

method, but the SA with heuristic memory results are very similar and not distinctive and thus yield the same figures because of the process capacity push mechanism. On the other hand, the intelligent search mechanism through heuristic memory also reduces the running time by about 25%. To maximize NPV, the approach forces to reach the capacities. Therefore, the results are similar but the NPVs are different. This can be observed in the NPV increases in Table II. While the number of blocks extracted and the average grades are similar, the block configurations are different with these two methods. It should also be noted

that average grade is persistent within a range throughout the periods at each destination. Therefore, there are no distinctive changes in average grades. This is important for the processes to choose and maintain a recovery, where fluctuations in the block grades affect the recovery process negatively. Another point to note is that process capacity push is working well, except for the high-grade process. The reason for this is the grade is not homogeneous and there are not enough high-grade blocks for the later periods. Most of the high-grade material is near the surface, thus the capacity during the first period is completely filled. It can also be

An improved meta-heuristic approach to extraction sequencing and block routing

observed from Figure 5 that number of blocks sent to each process at each period is below the corresponding capacity, satisfying the process capacity constraints. The processing capacities are 25 000, 20 000, and 15 000 blocks for the low-, medium-, and high-grade processes, respectively (Table I). As can be seen from Figure 5, for the low- and medium-grade processes, capacity satisfaction is quite good. Since the number of high-grade blocks is low, there is a decreasing order of number of blocks. In this case, there may be a few solutions: establishing stockpiles, changing the high-grade process design to meet the grade requirement, or re-installed high-grade process capacities. This is a common problem in mining operations because the capacity installation ignores ore material heterogeneity. As can be also seen from Figure 5, the grades at each destination are consistent in terms of periods. Mining capacity was 60 000 and was also satisfied, as the numbers of blocks extracted in each period are 60 000, 60 000, 59 703, and 46 609 respectively.

Lastly, the overall average grades of low-, medium-, and high-grade processes are compared in Figure 6. Average Cu and Mo grades obtained at each process destination are given as 0.6081% and 0.043 275% for low-grade processing, 0.677 325% and 0.062 778 25% for medium-grade processing, and 0.776 868 25% and 0.101 760 25% for high-grade processing. The average grades are highly compatible with the grade requirements. As expected, the high-grade process has the highest average grade, followed by the medium and low grades.

Conclusions and future work

The use of SA after a heuristic-based method guarantees that it will either produce a better solution or return the initial solution. It is true that SA takes time to reach the optimal value. However, unlike exact methods, it can be stopped at any point and best solution found so far can be returned. Moreover, almost all parameters can be integrated into SA, such as process capacity, transportation cost, and multiple process destinations, which are impossible to integrate in some other techniques. In exact methods, as the number of parameters increases, the problem size increases exponentially, whereas with SA the problem size increases proportionally; only as much as the expansion of the search space. SA is also more convenient to apply to our problem than other meta-heuristic methods such as genetic algorithms, particle swarm optimization, and evolutionary search because these types of algorithms require a pool of initial solutions. In our case, we used RPW to generate the initial solution, which can provide only one solution. For such a large problem, generating more than one solution is hard and time-consuming.

It is observed from the case study that usage of SA can add large gains to the revenue compared to RPW. The average grade and number of blocks sent to destinations were overall stable. Moreover, the case study has shown that the revenue of the solution obtained in the same amount of time has been increased by SA with heuristic memory. As the running time increases, further improvement can be achieved.

In the case study of the heuristic-memory-based SA, a linear fitting function was used. Efficiency of the memory enhancement can be increased through improving this fitting function. Also, in our case most parameters related to heuristic-memory-based SA were chosen empirically, such as when to produce the first function, how often to update the function, and how to balance the optimal function with the heuristic. Research can be conducted on how to optimize these parameters.

The main issue in all meta-heuristic applications is the parameter selection. This is also true for all types of SA. Selection of SA-related parameters such as the temperature, number of iterations, and maximum number of solutions at each temperature can affect the running time of the program to a great extent. If the parameters are poorly set and the program is run for a short time, the results may not be optimal.

References

Asad, M. and Topal, E. 2011. Net present value maximization model for optimum cut-off grade policy of open pit mining operations. *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 111. pp. 741–750.

Bley, A., Boland, N., Fricke, C., and Froyland, G. 2010. A strengthened formulation and cutting planes for the open pit mine production scheduling problem. *Computers and Operations Research*, vol. 37. pp. 1641–1647.
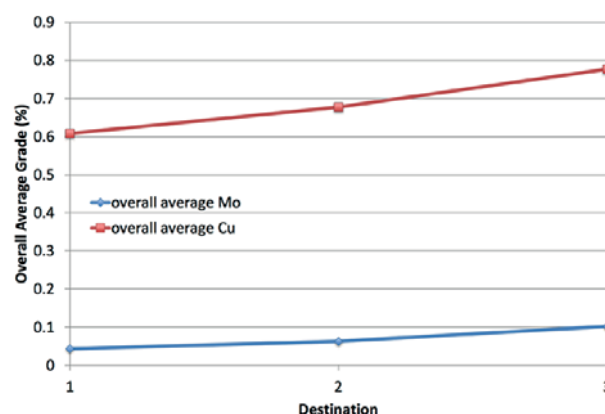
Figure 6—Overall average grades compared between low-, medium-, and high-grade process destinations

An improved meta-heuristic approach to extraction sequencing and block routing

De Carvalho Jr., J., Koppe, J., and Costa, J. 2012. A case study application of linear programming and simulation to mine planning. *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 112. pp. 477–484.

Cěrnÿ, V. 1985. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, vol. 45. pp. 41–51.

Clement, S. and Vagenas, N. 1994. Use of genetic algorithms in a mining problem. *International Journal of Surface Mining and Reclamation*, vol. 8. pp. 131–136.

Dagdelen, K. 2001. Open pit optimization–strategies for improving economics of mining projects through mine planning. *Proceedings of the 17th International Mining Congress and Exhibition of Turkey*, Ankara, 19-22 June 2001. Unal, E., Unver. B., Tercan, E., and Odası, M.M. (eds.). Chamber of Mining Engineers of Turkey, Ankara. pp. 117–121.

Dagdelen, K. and Johnson, T.B. 1986. Optimum open pit mine production scheduling by Lagrangian parameterization. *Proceedings of the 19th APCOM Symposium on Application of Computers and Operations Research in the Mineral Industry*, Pennsylvania University, 14-16 April. Ramani, R.V. (ed.). Society of Mining Engineers, Littleton, CO.

Dorigo, M. and Birattari, M. 2010. Ant colony optimization. *Encyclopedia of Machine Learning*. Springer. pp. 36–39.

Gershon, M. 1987a. Heuristic approaches for mine planning and production scheduling. *International Journal of Mining and Geological Engineering*, vol. 5. pp. 1–13.

Gershon, M. 1987b. An open-pit production scheduler: algorithm and implementation. *Mining Engineering*, vol. 39, no. 8. pp. 793-795.

Glover, F. 1989. Tabu search - part i. *ORSA Journal on Computing*, vol. 1. pp. 190–206.

Glover, F. 1990. Tabu search - part ii. *ORSA Journal on Computing*, vol. 2. pp. 4–32.

Glover, F. and Kochenberger, G.A. 2003. Handbook of Metaheuristics. Springer.

Kirkpatrick, S., Gelaatt, C.D. Jr., and Vecchi, M. 1983. Optimization by simulated annealing. *Science,* vol. 220. pp. 671–680.

Kumral, M. 2013a. Optimizing ore–waste discrimination and block sequencing through simulated annealing. *Applied Soft Computing*, vol. 13. pp. 3737–3744.

Kumral, M. 2013b. Multi-period mine planning with multi-process routes. *International Journal of Mining Science and Technology*, vol. 23. pp. 317–321.

Kumral, M. and Dowd, P. 2005. A simulated annealing approach to mine production scheduling. *Journal of the Operational Research Society*, vol. 56. pp. 922–930.

Lerchs, H. and Grossman, F. 1964. Optimum design of open-pit mines. *Operations Research*, vol. 12. p. B59.

Little, J., Knights, P., and Topal, E. 2013. Integrated optimization of underground mine design and scheduling. *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 113. pp. 775–785.

Lundy, M. and Mees, A. 1986. Convergence of an annealing algorithm. *Mathematical Programming*, vol. 34. pp. 111–124.

Menabde, M., Froyland, G., Stone, P., and Yeates, G. 2004. Mining schedule optimisation for conditionally simulated orebodies. *Proceedings of the International Symposium on Orebody Modelling and Strategic Mine Planning: Uncertainty and Risk Management*, Perth, 22-23 November 2004. Australasian Institute of Mining and Metallurgy. pp. 347–52.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., and Teller, E. 2004. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, vol. 21. pp. 1087–1092.

Mitchell, T.M. 1997. Machine Learning. McGraw Hill, Burr Ridge, IL. p. 45.

Nehring, M., Topal, E., Kizil, M., and Knights, P. 2012. Integrated short- and medium-term underground mine production scheduling. *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 112. pp. 365–378.

Nehring, M., Topal, E., and Knights, P. 2010. Dynamic short term production scheduling and machine allocation in underground mining using mathematical programming. *Mining Technology*, vol. 119. pp. 212–220.

Osanloo, M. and Ataei, M. 2003. Using equivalent grade factors to find the optimum cut-off grades of multiple metal deposits. *Minerals Engineering*, vol. 16. pp. 771–776.

Rothlauf, F. 2011. Design of Modern Heuristics: Principles and Application. Springer.

Sattarvand, J. 2009. Long-term open-pit planning by ant colony optimization. MSc thesis, RWTH Aachen University.

Sattarvand, J. and Niemann-Delius, C. 2013. A new metaheuristic algorithm for long-term open-pit production planning. *Archives of Mining Sciences*, vol. 58. pp. 107–118.

Shishvan, M.S. and Sattarvand, J. 2015. Long term production planning of open pit mines by ant colony optimization. *European Journal of Operational Research*, vol. 240. pp. 825–836.

Sutton, R.S. and Barto, A.G. 1998. Introduction to Reinforcement Learning. MIT Press.

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.

Topal, J.L.E. 2011. Strategies to assist in obtaining an optimal solution for an underground mineplanning problem using mixed integer programming. *International Journal of Mining and Mineral Engineering*, vol. 3. pp. 152–172.

Witten, I.H. and Frank, E. 2005. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann. ◆