COMPARING HEURISTICS FOR THE OPEN PIT BLOCK SCHEDULING PROBLEM

Enrique Jélvez (Corresponding author), ejelvez@ug.uchile.cl.
Diego Mancilla, dmancillac@gmail.com.
Nelson Morales, nelson.morales@amtc.cl.
Pierre Nancel-Penard, pnancel@gmail.com.
Delphos Mine Planning Lab., AMTC, Universidad de Chile, Av. Tupper 2069, Santiago, Chile.

ABSTRACT

The open pit block scheduling problem is very important in mine planning. It consists of getting an extraction sequence that maximizes the net present value of the extracted blocks of an open pit, subject to various constraints, such as slope angles and upper and lower limits resource consumption, for instance, transportation and processing. Since 1960s it has been known that this problem can be modelled using mixed integer programming. Unfortunately, a deposit may consist of several thousand and up to millions of blocks, therefore, the mathematical optimization approach requires a very large number of variables, making it quite difficult to solve and, even more, requiring a lot of computation time only to find feasible solutions.

This paper deals with this last problem: how to find good feasible solutions, *i.e.*, block scheduling with a high net present value. With this aim, we have proposed and implemented some heuristic methods based upon the reduction of variables by means of an incremental approach and block aggregation. All the algorithms and instances are shared among the heuristics, so comparison in terms of performance is provided.

The heuristics that we implement in this paper to solve the open pit block scheduling problem were applied to different instances, from 62,000 blocks to 1,680,000 blocks, considering 12 time-periods. The experimental results obtained show that the algorithms are very promising for applying these techniques to large scale models, which is part of our future work.

INTRODUCTION

Mine planning is the discipline of mine engineering that transforms the geological resource into the best productive offer, subject to the strategic goals delineated by the shareholders and the owners of the business (Rubio, 2006). Having an orebody, a mining project must define the exploitation method (open pit or underground), the time when it will be exploited and how to use the available resources in order to obtain the major benefit for the project. The mine planning must connect appropriately all the variables that affect the mineral exploitation with the design of the extraction, such as resources, teams and people, answering what means are needed to know how, how much and when to extract the mineral, defining this way the mine plan. The productive promise generated in this plan is expressed by means of a production plan and supported by a production scheduling, which determines what portions of the mine will be extracted and which of them will be processed, specifying the time period in which every task will be carried out. In order to generate a production plan, the ore body is discretized in three-dimensional blocks, each of them has attributes as spatial coordinates, size, density and laws of different minerals, forming a block model developed by geologists and geostatisticians based on drillings. The importance of the stored data in this model is huge, since from them the costs and the expected profits are estimated, both being fundamental elements in the elaboration of the mine plan. The planning horizon, that is, the total time during which the mine will be exploited, is also discretized in time periods, which can vary from days up to weeks in short term planning; months or quarters in medium term planning and years in long term planning.

Considering the above mentioned, we can say in a general way that the open pit block scheduling problem consists of determining which blocks should be extracted, when they should be extracted and where they should be sent later, whether it is a processing plant or a waste dump. This problem can be formulated using integer linear programming techniques and its solution is key for the profitable operation of the mine, since it represents a discrete production plan in the time that it optimizes certain criterion of interest, either maximizing economic benefits or mineral production, either minimizing the exploitation costs, etc., subject to spatial precedence and resources consumption constraints. This problem can be extremely difficult to solve, due to the fact that a mine can have thousands or even millions of blocks and the planning horizon can reach ten periods or more, which implies that the resultant model can reach several millions of variables and constraints. Then, as the long-term models are so enormous, the cut-off grade or the destination where the blocks are sent is predetermined, with which an important number of variables is eliminated in the optimization model, allowing the problem to be more treatable. Also, we assume that the blocks are processed in the same time period in which are extracted from de mine (that is, we do not allow to stock material for future processing): this case will be address in this article.

Due to the previous complications, numerous heuristic methods have been developed to generate solutions, in general suboptimal for the open pit block scheduling problem. In particular, this work develop heuristic based on aggregation in order to obtain good feasible solutions for the problem, that is, a discrete production plan in time that seeks to maximize the net present value (NPV) of the extracted blocks.

In order to generate the different instances, we use the software BOS2M (Blending Optimization Sequencing and Scheduling Multi-destination), which implements a mathematical model using mixed integer programming as part of Delphos Mine Planning Laboratory research activities. The model maximizes the ore production value by computing an extraction sequence that considers stock, accessibility, blending and capacity constraints.

Review of literature

In 1965 Lerchs and Grossman presented the final pit problem (to define what portion of blocks contains the maximum economic benefit) and developed an algorithm to solve it. An important achievement in the block scheduling problem formulation is due Thys B. Johnson (1968), who presented the problem under precedence, capacities and blending constraints (the last ones given by ranges of the processed ore grade), by means of a multidestination and multiperiod approach, this is, the optimization model decides what blocks to extract, what process to apply and when to do it. Our work considers a simplified version of this model, in which there is only two destinations (plant and waste dump) and there are not blending constraints.

In this literature survey, we will focus on block aggregation techniques, which reduce the size of the problem through the block aggregation with similar properties in larger units called *macroblocks.* Zhang (2006) uses genetic algorithms combined with a block aggregation technique based upon topological sort to reduce a number of variables in the model. Ramazan (2007) works on a model with fixed cut-off grades, upper and lower bound for processing and blending, but only upper bound in capacity. They aggregate blocks into *fundamental-trees* that have certain mathematical properties (like having a positive value, respecting slope constraints and being minimal in some sense) and present a case study of about 12,000 blocks that are aggregated into 1,600 fundamental trees that are scheduled over 4 time periods, with an overall solution time of 30 minutes. Boland et al (2009) use a different technique to approach the problem by considering that blocks belong to a certain and unique *bin*, where the number of bins is smaller than the number of blocks. The extraction of individual blocks is controlled with continuous variables, but binary variables are used at the bin level to impose slope constraints. Using this approach, they are able to *solve* instances up to 96,000 blocks on 25 time-periods in a few hundred seconds, using as a stopping criterion the difference between optimal values for the linear relaxations of their formulation and the original problem. Tabesh and Askari-Nasab (2011) present an algorithm that aggregates blocks into mining units and use Tabu Search to calibrate the number of final units. The resulting problem is then solved using standard mixed integer programming algorithm; the aggregation technique is interesting, because it is based upon a *similarity index* that considers attributes like rock type, ore grades and the distance between the blocks; the Tabu Search procedure is then used to further aggregate the blocks, while trying to balance the loss of selectivity due to this. The procedure is applied on 5 different instances, which show a variety of results whether improvements on the objective (NPV) or computation time are not consistent, thus the authors indicate that further research is required.

Formulation

Currently, the process of long term planning is performed using the nested pits methodology of Lerchs and Grossman (1965), indeed, the most specialized commercial software are based on this method; however, they are not completely automatics and do not consider time in its formulation neither capacity aspects, consuming hand calculation from mining engineer, based upon trial and error. Next, we will see how to add this information to the model.

Let B be the set of all blocks and $N = |B|$; the elements of B will be denoted as i, j. Similarly, we consider $T \in \mathbb{N}$ time-periods and denote individual time-periods as $s, t \in \boldsymbol{T} :=$

$\{1,2,\ldots,T\}$, where T is called the time horizon or planning horizon. The profit perceived if block $i \in B$ is processed at time-period $t = 1,2,..,T$ is $\rho^t \cdot v_i$, where ρ represents the discount factor ($\rho = 1/(1+dr)$, where dr is usually 10% in mining projects), and v_i is the net value obtained from processing the block i. Slope constraints are modeled as precedence constraints and encoded as a set of arcs $A \subset B \times B$, so $(i,j) \in A$ means that block j has to be extracted before block i. We say, in this case, that block j is a predecessor of block i, which in turn is a successor of j. Also, we define a set of resources R (for instance, transporting and processing, given by trucks and crushers capacities, respectively), and for each block $i \in B$ and resource $r \in R$, the quantity $a_{ir} \in \mathbb{R}$ of resource r that is used when block i is extracted or processed. Finally, for each time-period t, lower and upper bounds on the consumption of resource r are given by the quantities $C_{rt}^- \in \{-\infty\} \cup \mathbb{R}$ and $C_{rt}^+ \in \{+\infty\} \cup \mathbb{R}$, respectively. We define the following variables, for each $i \in B, t \in \boldsymbol{T}$:

$$x_{it} = \begin{cases} 1 & \text{, if block i is extracted during time period t or before} \\ 0 & \text{, otherwise} \end{cases}$$

According to previous definition, we introduce the following auxiliary variables for any $i \in B$.

$$\Delta x_{it} = \begin{cases} x_{it} & \text{, if } t = 1 \\ x_{it} - x_{it-1} & \text{, if } t > 1 \end{cases}$$

The idea is to know exactly the time-period in that block i is extracted. Considering all the above mentioned, we can formulate the open pit block scheduling problem as follow:

$$(OPBSP) \qquad \max \sum_{t=1}^{T} \sum_{i=1}^{N} \rho^t v_i \Delta x_{it} \tag{1}$$

$$x_{it} \le x_{jt} \qquad (\forall (i,j) \in A)(\forall t \in \boldsymbol{T}) \tag{2}$$

$$\Delta x_{i.t} \ge 0 \qquad (\forall i \in B)(\forall t \in \boldsymbol{T}) \tag{3}$$

$$\sum_i a_{ir} \Delta x_{it} \le C_{rt}^+ \qquad (\forall\, r \in R)(\forall t \in \boldsymbol{T}) \tag{4}$$

$$\sum_i a_{ir} \Delta x_{it} \ge C_{rt}^- \qquad (\forall\, r \in R)(\forall t \in \boldsymbol{T}) \tag{5}$$

$$x_{it} \in \{0\,,1\} \qquad (\forall\, i \in B)(\forall t \in \boldsymbol{T}) \tag{6}$$

Expression (1) presents the goal function, which is the discounted value of extracted blocks over the time horizon T. It turns, (2) corresponds to the precedence constraints given by the slope angle and (3) estates that blocks can be extracted only once. Moreover, (4) and (5) limit the maximum and minimum resource consumption in each time period, respectively. Finally, (6) establishes that all variables assume binary values.

The relation among the final pit problem, the scheduling problem and the Johnson model that considers the multidestination option is presented in Figure 1. It is possible to see, on one hand, that the Johnson model is more general than the one we consider in this paper, since the optimization model decides what to do with the mined blocks, besides considering the blending constraints; instead our model considers that the destination of each block is known in advance and it does not consider blending constraints. On the other hand, our model is more general than Lerchs and Grossman model, since the latter does not consider time in its formulation neither the capacity constraints for each resource.

As mentioned above, to solve the open pit block scheduling problem in real instances is very difficult, due to the enormous quantity of variables and constraints. The strategy for solving the integer programming formulation using standard implementation available in the optimization solver is based upon the *Branch & Bound* algorithm, which has an enumerative approach. Therefore, numerous heuristics have been developed to generate feasible solutions to the OPBSP, as we detail the next section.

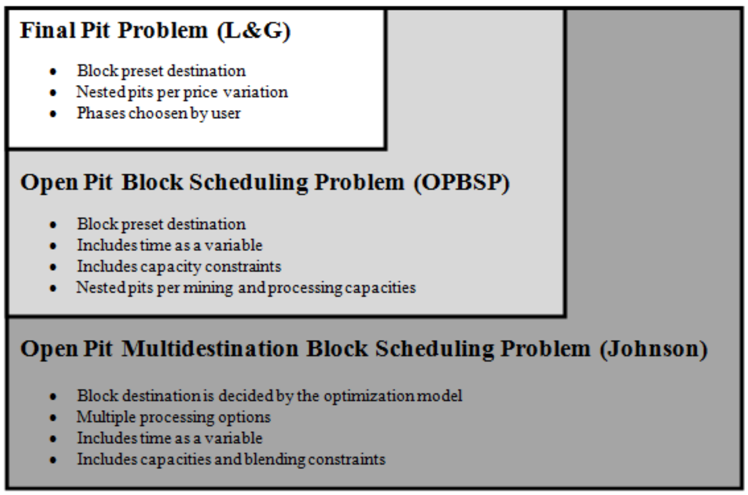

Figure 1 - Relation among final pit problem, OPBSP and Johnson´s model.

METHODOLOGY

The original block model BM1 has blocks of 30x30x30 cubic meters, but we use the same blocks to produce additional models with larger number of blocks by dividing the original blocks into blocks of 15x15x15 (BM2) and 10x10x10 (BM3). Each piece has economic values and tonnages that are proportional to their size.

Table 1 – Block model, precedence and integer problem size for different instances.

Block model	Block size	# Blocks	# Prec. arcs	# Variables
BM1	30x30x30	62,220	274,300	746,640
BM2	15x15x15	497,760	2,353,494	5,973,120
BM3	10x10x10	1,679,940	7,856,462	20,159,280

It is possible to reduce the block model calculating the final pit as preprocessing (Caccetta and Hill, 2003) proved that it suffices to consider the blocks inside the final pit limits. This result allows us to reduce the block model that needs to be considered). There are 12 time periods (years) and a discount rate d$r = 10\%$ applies. We consider two types of resources: mining (transported), maximum 70,000 tons per day; and processing, maximum 30,000 tons per day.

In order to run the different heuristics we use the BOS2M tool, by assuming that the blocks are processed in the same time period in which they are extracted from the mine (that is, we do not allow to stock material for future processing). Our model incorporates a fixed cut-off grade (as typical long term models), which implies that if a block contains sufficiently high ore content, it is always processed; otherwise, it is never processed. Therefore, the decision of the block destination is done beforehand. Finally, we consider a slope angle of 45° at two levels.

Heuristics

In this section we introduce the heuristics proposed for solving the open-pit block scheduling problem. We assume that OPBSP cannot be solved optimally in any of these cases.

Incremental heuristic

This heuristic proposes to solve the problem incrementally, i.e., it takes fewer time periods (which we call window time), solves and removes the scheduled blocks in this window time; then it repeats this process to the remaining blocks and time periods, adjusting the constraints accordingly and moving the window time until to complete the horizon planning. The simplest example of this is taking just one time period as our window time and solving repeatedly. The Figure 2 shows the procedure to $\boldsymbol{T = 3}$ and a window time equal to one: in the first time period it selects the blocks to be extracted $\boldsymbol{P_1}$, then remove these blocks and repeat this procedure to the second period, obtaining $\boldsymbol{P_2}$. Finally, the heuristic solves for the ultimate time period.

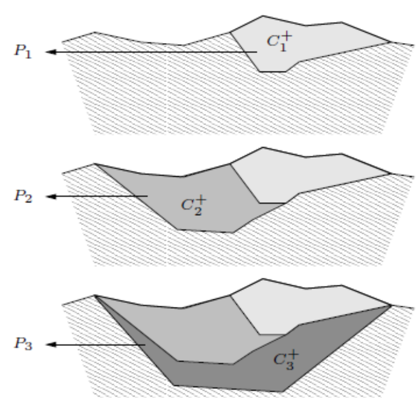

Figure 2 – Example of the incremental heuristic for $T = 3$ and window time equal to one. C_t^+ represents the maximum resource consumption in time period t.

Block aggregation heuristic

This heuristic reduces the number of blocks considered in the optimization process. For this, blocks are aggregated into larger units, several times if necessary, until the problem can be solved (either optimally or by using another heuristic). The solution at the aggregated level is then used as a guide to set variables and generate solutions for the original blocks. An illustration of this heuristic can be found in Figure 3 in which each individual block in B is reblocked into a big block containing exactly 9 of them. This means that, while the original block model has 315 blocks, the reblocked model contains only 35 blocks. Then (a) the problem is solved on these 35 blocks and (b) we select blocks that are at the interior part of each period (grey blocks). (c) The original blocks corresponding to these aggregated blocks are removed from the original model, the capacities are updated and the problem is solved for the remaining blocks. (d) Finally, all blocks are scheduled by mixing the solutions

Note that reblocking procedure may be used several times depending on whether it was possible (or not) to solve the corresponding instance. In this case we propose to use the heuristic recursively, which leads to two phases in the algorithm: The first phase (forward) in which we try to solve the original problem and reblock on failure until some instance is solved; and the second phase (backward) in which we use the solution for reblocked problem to fix variables and solve this instance with smaller blocks, repeating (if necessary) until it finds a feasible solution to the original problem.

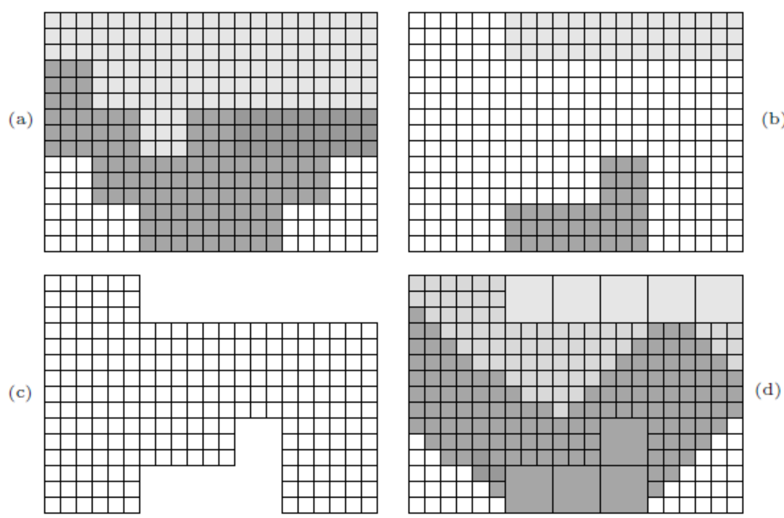

Figure 3 – Example of the block aggregation heuristic. (a) Aggregated block model is constructed and the scheduling problem is solved. (b) Inner blocks have their extraction time period fixed. (c) Border blocks are left for solving at the original scale. (d) Final solution is constructed.

RESULTS AND DISCUSSION

In this section we present and discuss the results obtained in the numerical experiments by using the heuristics described above. We present the results for different instances in the Table

2, which has one column per heuristic and for each we report: total execution time (in seconds) and solution value (in millions of US dollars). The notation for the heuristics is the following:

- **IP**: Corresponds to run the optimization solver on the monolithic problem (this is, not using a heuristic at all).
- **LP**: Is the result of solving the linear relaxation of the problem, which does not produce an integer feasible solution, but is useful for reference. A very efficient algorithm to solve it can be found in Bienstock and Zuckerberg (2010).
- **HInc**: Corresponds to the incremental heuristic.
- **HReb/HInc**: Represents the block aggregation heuristic when using HInc as the algorithm for solving the instances in the backward and forward phases.

In all heuristics, we run the algorithms with only one time period per iteration (length time window equal to one). The time rows consider the time required for reblocking (case HReb) and calculating precedences, which is shorter than nine seconds for the larger instance. The notation ∞ represents that it is not possible solve the monolithic problem directly. We do not report the execution time of LP because is not important for our results, but the solution value is given for comparison purpose

Table 2 – Numerical results per heuristic for different block sizes

Block model		IP	LP	HInc	HReb/HInc
BM1	Ex. time [s]	∞	-	115	115
	Sol.value [MMUS$]	-	1,246	1,126	1,126
BM2	Ex. time [s]	∞	-	12,554	649
	Sol.value [MMUS$]	-	1,246	1,129	1,181
BM3	Ex. time [s]	∞	-	$> 6\ hrs$	11,954
	Sol.value [MMUS$]	-	1,246	-	1,182

The LP relaxation value provides an estimate of the optimal value of the problem, and therefore, enables us to estimate the gap of the found solution using HReb/HInc (about 5.13 % in the larger case). However, there is less accuracy in the aggregated case (BM1) with all heuristic, because the optimality gap was up to 9%. This is explained because some decisions lose their freedom in the aggregated block model. Optimal values across heuristics are consistent, which was expected thus validating the results in terms of the implementation. Among the tested heuristics, HReb seems to be the more promising, because it is able to produce solutions for all instances. Finally, we present the pits obtained by using HReb/HInc (see Figure 4), because it had the best performance. In them, we can see the geometry of the pits, which are very similar, validating our approach, in the sense that block aggregation can be used to derive solutions to the larger models.

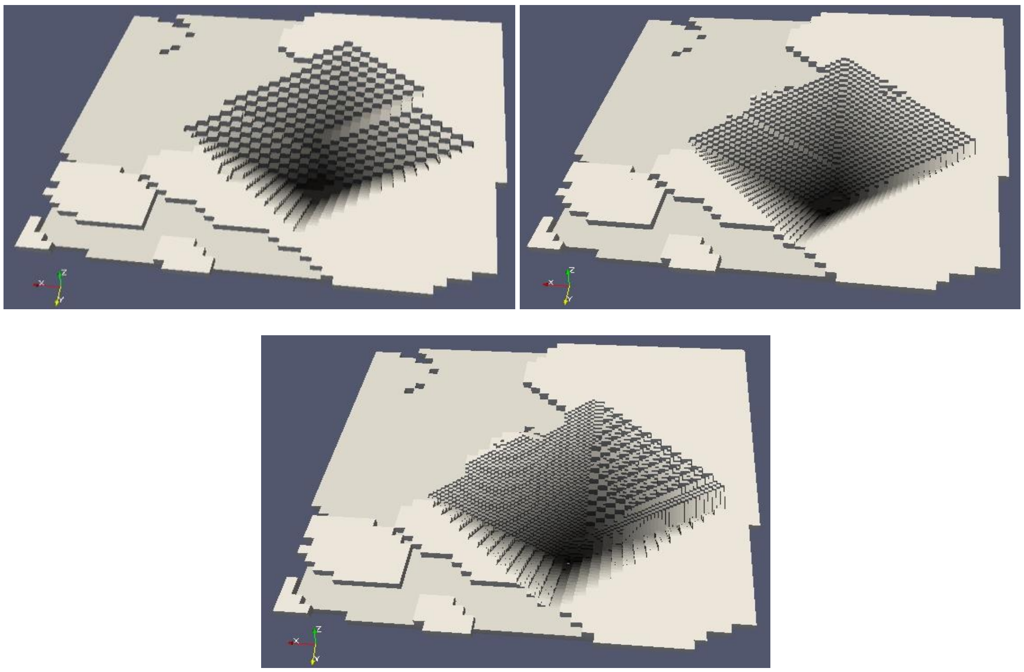

Figure 4 – Pits obtained through HReb/HInc for all block models.

BM1 (upper left), BM2 (upper right) and BM3 (lower)

CONCLUSION

We have presented a number of heuristics to tackle the open-pit block scheduling problem that approaches the problem by reducing the size of the binary linear formulation. The main result of the heuristics is that we moved from a scenario in which using the standard approach produced no feasible solutions due to the size of the problem, to one in which we are able to produce solutions for a number of different instances, within reasonable times and optimality gap.

An interesting property of the heuristics presented is that they can be easily combined with other techniques in order to further improve the efficiency of them, for instance, toposort heuristic developed by Chicoisne et a. (2012) or sliding time window heuristic developed by Cullenbine et al (2011), in the forward and/or backward phases of block aggregation heuristic. There are several possible extensions for the heuristics, like using more than one time-period at each iteration of the incremental heuristic and using others heuristics methods in forward and backward phases in HReb, as mentioned recently. We expect to improve this in the near future.

Extended comparison with other results in the bibliography is possible, but it is difficult because, even if the same block model is used, it is not always clear that the parameters of the model, like overall capacity or number of periods, coincide.

REFERENCES

Bienstock, D. and Zuckerberg, M. (2010). Solving LP relaxations of large-scale precedence constrained problems. In Proc. 14th Conf. Integer Programming Combin. Optim. (IPCO). Berlin. Lecture Notes in Computer Science. v. 6080. Springer-Verlag. pp. 1–14.

Boland, N., Dumitrescu, I., Froyland, G. and Gleixner, A. M. (2009). LP-based disaggregation approaches to solving the open pit mining production scheduling problem with block processing selectivity. Comput. Oper. Res. v. 36(4). pp. 1064–1089.

Caccetta, L. and Hill, S. (2003). An application of branch and cut to open pit mine scheduling. Journal of Global optimization. v. 27. pp. 349-365.

Chicoisne, R., Espinoza, D., Goycoolea, M., Moreno, E. and Rubio, E. (2012). A new algorithm for the open pit mine Scheduling problem. Operation Research. v. 60 (3). pp. 517-528.

Cullenbine, C., Wood, R. K. and Newman, A. (2011). A sliding time window heuristic for open pit mine block sequencing. Optimization Letters. v. 5 (3). pp. 365-377.

Johnson, T. (1968). Optimum open pit mine production scheduling. Ph.D thesis. Berkeley. Operations Research Department. University of California,

Lerchs, H. and Grossman, H. C. (1965). Optimal design of open-pit mines. Transactions C.I.M. v. 58. pp. 47-54.

Ramazan, S. (2007). The new fundamental tree algorithm for production scheduling for open pit mines. European Journal Operation Research. v. 177(2). pp. 1153–1166.

Rubio, E. (2006). Block cave mine infrastructure reliability applied to production planning. PhD Thesis. Vancouver. Faculty of Graduate Studies (Mining Engineering). University of British Columbia.

Tabesh, M. and Askari-Nasab, H. (2011). Two-stage clustering algorithm for block aggregation in open pit mines. Mining Technology. v. 120 (3). pp. 158-169.

Zhang, M. (2006). Combining genetic algorithms and topological sort to optimize open-pit mine plans. Proc. 15th Internat. Sympos. Mine Planning Equipment Selection (MPES), M. Cardu, R. Ciccu, E. Lovera, E. Michelotti, eds., FIORDO S.r.l. Torino, Italy. pp. 1234 -1239.