

Creating mining-cuts using hierarchical clustering and tabu search algorithms

Hooman Askari-Nasab, Mohammad Tabesh, and
Mohammad M. Badiozamani
Mining Optimization Laboratory (MOL)
University of Alberta, Edmonton, Canada

Abstract

Open pit mine plans define the complex strategy of displacement of ore and waste over the mine life. Various mixed integer linear programming (MILP) formulations have been used for production scheduling of open pit mines. The main problem with the MILP models is the inability to solve real size mining problems. The main objective of this study is to develop a clustering algorithm, which reduces the number of binary integer variables in the MILP formulation of open pit production scheduling. To achieve this goal, the blocks are aggregated into larger units referred to as mining-cuts. A meta-heuristic approach is proposed, based on Tabu Search (TS), to aggregate the individual blocks to larger clusters in each mining bench, representing a selective mining unit. The definition of similarity among blocks is based on rock types, grade ranges, spatial location, and the dependency of extraction of each mining-cut to the mining-cuts located above it. The proposed algorithm is tested on synthetic data.

1. Introduction

There are different approaches in solving optimization problems which are classified into three main categories: exact, heuristics, and meta-heuristics. In all three approaches the size of the problem and its complication is a key feature. Exact algorithms are based on mathematical programming which will find the optimal solution using different methods introduced in literature. Since most of the real world problems are large scale and complicated, non-exact algorithms have come into existence. Instead of looking for optimal solution, these algorithms just try to find a good solution within a reasonable time and resource usage. This resource usage has been of great concern when using both exact and non-exact algorithms. Different methods have been proposed for reducing the amount of resources required to solve optimization problems.

One known large-scale problem which cannot be solved using the exact optimization procedures is the open pit production scheduling. The number of variables and constraints in this problem is related to the size of the deposit and the number of blocks in the model. Open pit mine block models usually include millions of blocks, which makes the exact optimization method intractable for the open pit production scheduling problem. For the planning step the open pit mine production scheduling block models are used as the input. There will be variables, parameters and constraints regarding extraction and processing of these blocks in production scheduling formulations. The

typical mathematical model used to optimize the open pit mine production scheduling problem is Mixed Integer Programming (MIP).

If blocks are considered to be large in size in order to reduce the number of blocks, the precision required to model the overall pit slopes is lost. On the other hand, smaller blocks may result in more variables and constraints which make the problem unsolvable in a reasonable time. Therefore, methods have been proposed in the literature trying to introduce new modeling and solution procedures being able of creating the production plan and overcoming the curse of dimensionality in aforementioned MIPs.

The objective of this study is to develop an algorithm based on hierarchical clustering and Tabu Search to aggregate blocks into larger formations referred to as mining-cuts. Blocks within the same level or mining bench are grouped into clusters based on their attributes, spatial location, rock type, and grade distribution. Similar to blocks, each mining-cut has coordinates representing the center of the cut and its spatial location. Introduction of mining-cuts into mixed integer linear programming production scheduling models will reduce the number of variables and constraints and as a result, the production scheduling problem could be solved using current optimizers.

In the next section, the literature of production scheduling is reviewed. In the second section, the development of the clustering methodology based on hierarchical clustering and Tabu Search is presented, followed by an illustrative example from implementing the proposed algorithm. Then, the proposed algorithm is applied to a case study and the results are reported, followed by the conclusion.

2. Literature review

Martinez (2006) has worked on improving the performance of a mixed integer production scheduling model and has implemented his findings on a case study in Sweden. He has studied on an underground Iron ore mine and presents a combined (short and long-term) resolution model, using mixed integer programming. In order to decrease the solution time, he develops a heuristic consisting of two steps: (1) solving five sub problems and (2) solving a modified version of the original model based upon information gained from the sub problem solutions. His presented heuristic method, results in a better solution in less time comparing to solving the original MILP problem.

Newman (2007) performed another study on the same case study with Martinez (2006) on an underground Iron ore mine. They have designed a heuristic algorithm based on solving a smaller and more tractable model than the original model, by aggregating time periods and reducing the number of variables. Then, they solve the original model using information gained from the aggregated model. For the worst case performance of this heuristic, they compute a bound and show that their presented procedure produces good quality results while reducing the computation time.

Ramezan (2007) uses an algorithm entitled 'Fundamental Tree Algorithm' which is developed based on linear programming. This algorithm aggregates blocks of material to clusters and as a result, decreases the number of integer variables as well as the number of constraints in MIP formulation. The author proposes a new Fundamental Tree algorithm in optimizing production scheduling of open pit mine. The economic benefit of the proposed algorithm compared to existing methods is demonstrated through a case study.

Gaupp (2008) presents three approaches to make the MILP more tractable: (1) reducing the number of deterministic variables to eliminate blocks from consideration in the model; (2) strengthening the model's formulation by producing cuts and (3) using Lagrangian techniques to relax some constraints and simplify the model. By using these three techniques, he determines an optimal (or near-optimal) solution more quickly than solving the original problem.

Askari-Nasab (2008) have proposed two MILP formulations for long-term large-scale open pit mine production scheduling problem. They have used the fuzzy C-means clustering to aggregate the blocks in each elevation and reduce the number of blocks to smaller number of aggregated blocks. As a result, they have reduced the number of variables in the proposed MILP. They have implemented the proposed MILP theoretical frameworks for large-scale open-pit production scheduling.

Amaya (2009) uses the concept of local-search based algorithm in order to obtain near optimal solutions to large problems in a reasonable time. They describe a heuristic methodology to solve very large scheduling problems with millions of blocks. They start from a known feasible solution, which they call “incumbent”. Their proposed algorithm seeks to find a similar solution to the current solution with an improved objective function value. To do that, the algorithm, by means of a random search, looks for solutions that partially coincide with the incumbent. Whenever any improvement in objective function is found, the incumbent is updated and the process is repeated to reach its stopping criteria.

Boland (2009) propose a method in which the mining and processing decisions are made based on different decision variables. They use aggregates of blocks in scheduling the mining process. Then, an iterative disaggregation method is used which refines the aggregates up to the point where the refined aggregates result in the same optimal solution to the relaxed LP, considering individual blocks. Then, their proposed algorithm uses individual blocks for making decision on processing. They have proposed several strategies to create refined aggregates. These refined aggregates provide high quality solutions in terms of net present value (NPV) in reasonable time.

Through a literature review, it can be inferred that most of researchers have focused on heuristic and meta-heuristic approaches to solve the mine production planning problem which do not guarantee the optimality of the solution. Also, there are some publications on penalty function and Lagrangian relaxation methods to scale down the problem and make it solvable with mathematical programming approaches. Only few studies are done to aggregate the blocks and reduce the number of variables to make the problem tractable. Among the latest aggregation approaches employed to solve the problem is what Askari-Nasab (2008) proposed using Fuzzy C-means clustering to create mining cuts. In the next section the theoretical framework, used in developing the algorithm, is reviewed.

3. Theoretical framework and models

3.1. Clustering

According to the handbook of applied algorithms “Clustering is the process of grouping together objects that are similar. The groups formed by clustering are referred to as clusters” (Stojmenović, 2008). This process is usually performed by calculating a measure of similarity (or dissimilarity) between different pairs of data. In addition to this measure, our purpose of clustering may also affect the result of the clustering process. In other words, the same set of data can be clustered in different ways when they are to be used for different purposes. Different characteristics of data which are taken into account in calculating similarity measure can be the cause of having different results as well as the clustering method used.

There have been many clustering algorithms proposed in the literature which are based on grouping data according to their different characteristics. Some of these groups are mentioned in this section.

Exclusive vs. Non-exclusive: an exclusive clustering technique ensures the resulting clusters to be disjoint in contrast with non-exclusive algorithms which result in clusters having overlaps. Most of the current famous algorithms belong to the first group.

Intrinsic vs. Extrinsic: when external effective parameters matter, clustering techniques are divided into another two categories. Intrinsic algorithms are the one which are unsupervised activities and

create the clusters based on the data itself. On the other hand, where clustering is done based on external sources of data such as predetermined objects which should or should not be clustered together the process is said to be extrinsic. Intrinsic clustering algorithms are more common.

Hierarchical vs. Partitional: hierarchical algorithms are the ones which create a sequence of partitions on the data and are divided into two groups. *Hierarchical agglomerative algorithms* are the one which start creating clusters with a single object and combine them together in order to find the final clusters. The opposite direction is used in *hierarchical divisive algorithm* where the big cluster containing all the objects is created first and it is segmented to smaller clusters in each step. Partitional clustering methods create partitions based on the similarity and dissimilarities being defined between objects such that more similar objects, according to the pre determined properties, are grouped together. A common example of these algorithms is the k-means algorithm and its many extensions.

Since in this paper, the focus is on a hierarchical clustering algorithm, a review of this algorithm is presented here. According to Johnson (1967), hierarchical clustering has the following steps:

1. Start by assigning each item to a cluster, so that if you have N items, you now have N clusters, each containing just one item. Let the distances (similarities) between the clusters be the same as the distances (similarities) between the items they contain.
2. Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one cluster less.
3. Compute distances (similarities) between the new cluster and each of the old clusters.
4. Repeat steps 2 and 3 until all items are clustered into a single cluster of size N.

3.2. Penalty Function

Despite clustering which reduces the number of decision variables, the penalty function commonly eliminates some or all of the constraints from a problem. Penalty functions are defined as follows: “Penalty functions, a technique used in solving constrained optimization problems, are often used to restrict the solution search to designs that meet all criteria. As the name implies, a penalty is assigned to the figure of merit or merit function if a constraint is violated during optimization” (Organization, 2009) If you consider a normal objective function to be a cost minimization function, a penalty value is the amount which is determined as the cost being applied because of violation of each constraint.

Two different kinds of constraints can be defined. The first group is called the “hard” constraints which cannot be violated during optimization procedure. The second group, which can be violated in order to have better results, are called “soft” constraints. In contrary with soft constraints, the amount of penalty for hard constraints is set so large that the solution procedure never violates the constraint.

The simplest way of representing the penalty function is a normalized weighted sum of deviations from design values of each constraint. These weights can be determined for both hard and soft constraints. This can be shown as a definition of Figure of Merit represented by Eq. (1).

$$FOM = \frac{\sum_{i=1}^N w_i (d_i - c_i)^2}{\sum_{i=1}^N w_i} \quad (1)$$

Where:

d_i is the design objective of constraint

c_i is the current value of constraint

w_i is the weight

N is the number of constraints

Although it seems to be easy and generic to implement penalty functions, it is practically difficult to use because different penalty functions for different problems should be determined.

4. Methodology

The mixed integer formulation of the open pit mine planning problem consists of different integer and continuous variables and different sets of constraints. One of these sets which are making the problem large in size is the order of extraction constraints. In order to reduce the number of constraints in this set, a clustering procedure based on fuzzy c-means algorithm is proposed in Askari-Nasab (2008). Developing these clusters is done based on similarities between blocks regarding physical location of the block and its grade which is taken into account as a (zero-one) ore-waste categorization.

In the proposed method in this study the main idea is to create clusters in a way that the number of constraints regarding order of extraction is reduced in the MILP formulation. The number of constraints for each mining-cut is equal to the number of clusters in the upper level which are supposed to be extracted prior to extraction of the current mining-cut. Therefore, the algorithm has a look on the lower level and the dependencies between clusters while creating clusters in each bench.

The clustering procedure is a two step algorithm. In the first step an initial clustering is provided based on a similarity factor representing the spatial distance between blocks, the similarity of rock types, grades, and also the clusters in the lower bench. In this step, the hierarchical clustering approach is employed. In the next step a Tabu Search procedure is applied to modify clusters in a way that the number of dependencies between the two levels and accordingly the number of constraints is reduced.

The idea of using penalty values has been borrowed from Marcos Dósea (2008) where a clustering algorithm called “Adaptive Mean-Linkage with Penalty” is introduced. In proposing this algorithm object oriented characteristics have been considered. In this concept each object is supposed to have different properties. These properties can vary from simple spatial locations to complicated computational problem specific values. These values can be divided into two main groups: categorical and numerical values. The only important point in comparing categorical properties is to see whether the properties are equal or not, opposed to numerical properties where the amount of difference is of importance. While calculating dissimilarity measure in Adaptive Mean-Linkage with Penalty algorithm, both of these types are considered and a penalty value regarding difference is assigned to each comparing property.

In the proposed algorithm, two numerical and two categorical properties are defined for each block. Spatial location and grade values are the properties where the amount of difference matters in contrast to rock type and the lower bench cluster.

The final clusters created using this procedure have to have four main characteristics:

1. Each cluster has to be spatially united which means if two blocks are the members of a cluster there should be a set of blocks in the same cluster connecting these together.
2. Ore and waste blocks should be grouped together for further planning and decision making.
3. Clusters consisting of one rock type are easier to deal with in scheduling phases.

4. One of the most important goals of the proposed clustering algorithm is to create clusters having a look at the lower bench in order to reduce the number of technical extraction dependencies. This is satisfied by defining a property called “beneath cluster” for each block. In addition, in the Tabu Search, the concept of “beneath cluster” is taken into account in measuring the goodness of each state.

At the first step of the procedure one of the parameters considered in defining the similarity measure is the similarity between the clusters beneath each block. This will make the algorithm to group the blocks at the top of one cluster together. This will reduce the number of technical extraction constraints between clusters. In the next step the Tabu Search also tries to modify boundaries of the clusters in a way that the overlap between clusters is reduced.

In order to use Tabu Search (TS) algorithm to find a good solution for clustering in each bench, an initial state solution is required to feed into the algorithm. The initial solution can be found via several means. In this paper, Hierarchical clustering algorithm is used to generate the initial state solution. In order to use the clustering algorithm, the similarity between each pair of blocks should be defined first. To do so, the following steps are considered.

4.1. Similarity definition

1. Calculate the distance matrix between all blocks in the active bench. Distance between blocks is calculated by Eq.(2):

$$D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (2)$$

2. Calculate the rock type similarity matrix for all pairs of blocks in the active bench. Rock type similarity is calculated by Eq. (3) as assigning zero as the similarity between two blocks with different rock type and one, as the similarity between two blocks with same rock types.

$$R_{ij} = \begin{cases} 1 & \text{if blocks } i \text{ and } j \text{ are from the same rock type} \\ r & \text{otherwise} \end{cases} \quad (3)$$

3. Calculate the grade similarity matrix for all pairs of blocks in the active bench by Eq. (4).

$$G_{ij} = (G_i - G_j)^2 \quad (4)$$

Where G_i & G_j are grades of blocks i and j and G_{ij} is the grade similarity between these two blocks.

4. Calculate the beneath cluster similarity matrix for all pairs of blocks in the active bench. Beneath cluster similarity is calculated by assigning a number (less than one) as the similarity between two blocks which are above different clusters (in beneath bench) and one, as the similarity between two blocks which are above the same cluster. This is represented by Eq. (5).

$$C_{ij} = \begin{cases} 1 & \text{if blocks } i \text{ and } j \text{ are above the same cluster} \\ c & \text{otherwise} \end{cases} \quad (5)$$

5. Calculate the similarity matrix between all pairs of blocks in the active bench. Similarity between blocks i and j is defined by Eq. (6).

$$S_{ij} = \frac{R_{ij}^{W_R} \times C_{ij}^{W_C}}{G_{ij}^{W_G} \times D_{ij}^{W_D}} \quad (6)$$

As a result, the farther the blocks are from each other, the less similarity they have. In addition, the similarity between rock types, grades and beneath clusters of two blocks results in higher similarity between those blocks. By having the similarity matrix, it is possible to move forward to find out the initial state solution.

4.2. Hierarchical clustering algorithm:

Using the similarity matrix, the hierarchical algorithm can be described as:

1. Start by assigning each block to a cluster, so that if there are N blocks in the bench, there are N clusters, each containing just one block. Let the similarities between the clusters be the same as the similarities between the blocks they contain.
2. Find the most similar pair of clusters.
3. If clusters resulted from step 2 are neighbor (have common borders), then merge them into a single cluster, so that the number of clusters becomes one less and go to step 4. If not, ignore these clusters and return to step 2.
4. Compute similarities between the new cluster and each of the old clusters by considering the least similarity of new cluster members.
5. Repeat steps 2 and 3 until all blocks are clustered into a pre defined number of clusters.

The maximum number of blocks in each cluster is controlled in order not to have unequal sizes for clusters. The Hierarchical algorithm to find an initial state solution is presented as a flow chart in Fig. 1 (left).

4.3. Tabu search (TS)

Having the initial state solution, now the solution can be improved by applying the Tabu Search (TS). The following steps define our solution algorithm based on TS:

1. Determine the neighbors of each state through the following steps:
 - 1.1. Calculate the number of arcs each cluster has produced/caused (it is determined by looking at previous bench clusters (level 0) and counting the number of arcs each cluster at current bench (active level) has produced in respect to clusters of level 0). Sort the clusters in respect to the mentioned number of arcs.
Choose \underline{m} clusters which have produced more arcs (m first clusters from the sorted list of step 1.1).
 - 1.2. For each block in those \underline{m} clusters selected in step 1.2, calculate the number of arcs that are caused by the block and sort the blocks with respect to this number.
 - 1.3. From top of the sorted list of step 1.3, choose \underline{n} blocks which are on borders of the cluster.
 - 1.4. Consider all situations in which these \underline{n} blocks are disconnected from previous clusters and connected to other neighbor clusters (there are at most 3 neighbor clusters for each of bordering blocks of a cluster). As the result, the maximum number of neighbors in each state is $m \times n \times 3$.
2. Update the goodness measure for all new clusters in new situation as defined by Eq. (7). This is the intra-cluster similarity measure:

$$ICM_i = \frac{1}{n_i^2} \sum_{i=1}^{n_i} \sum_{k=1}^{n_i} S_{jk} \quad (7)$$

Where

n_i : The number of block in cluster i

S_{ij} : The similarity between blocks j and k .

State measure is presented by Eq. (8).

$$SM_i = \frac{\overline{ICM}^{w_s}}{N^{w_N}} \tag{8}$$

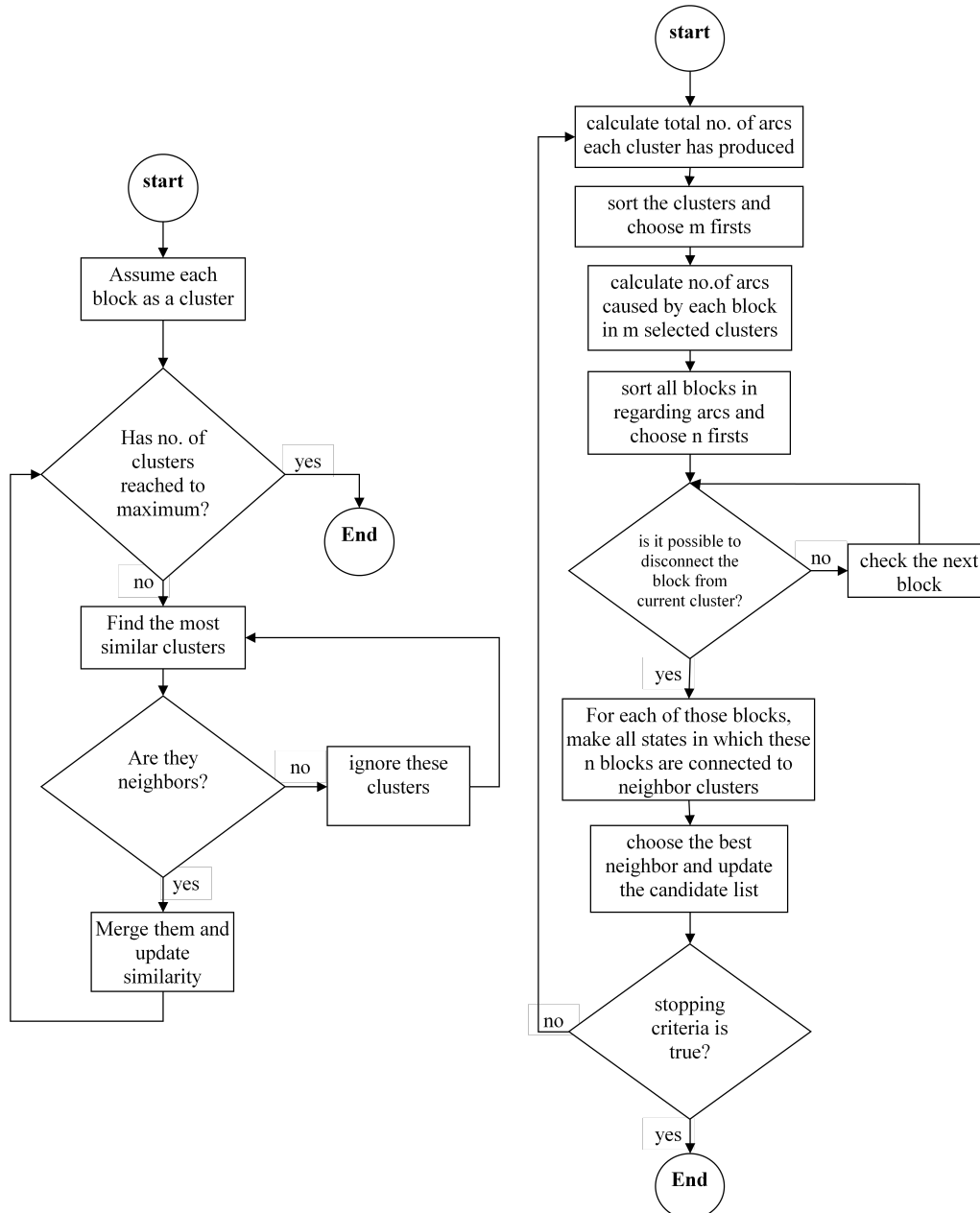


Fig. 1. Algorithm flow charts.

Where

\overline{ICM} : Average of all intra-cluster similarities

N : Total number of arcs

W_s : Weight of similarity

W_N : Weight of number of arcs

3. Update the candidate list by choosing the state with the maximum measure as the new member.
4. Repeat steps 1 to 3 until the stopping criteria happen.

Stopping criteria: number of candidate list members reaches to a pre-determined number or in s consecutive states, the best solution remains unchanged. The proposed TS algorithm is presented as a flow chart in Fig. 1 (right).

5. Numerical example

In order to show how the proposed algorithm works, the hierarchical clustering algorithm is applied to find the initial state solution to a simple example. The block model data is as presented in Table 1.

Table 1: Sample of block model data base

Block ID	X coordinate	Y coordinate	Rock type	Average grade	Beneath block
A	1	1	1	5.00%	1
B	2	1	1	4.00%	1
C	3	1	1	1.00%	2
D	4	1	2	0.50%	2
E	5	1	2	0.00%	2
F	1	2	1	4.50%	1
G	2	2	2	3.00%	1
H	3	2	2	2.00%	1
I	4	2	2	1.00%	2
J	5	2	3	0.90%	2
K	1	3	2	4.00%	1
L	2	3	3	5.20%	1
M	3	3	3	3.00%	1
N	4	3	2	2.00%	4
O	5	3	3	0.90%	2
P	1	4	2	3.00%	3
Q	2	4	1	4.00%	3
R	3	4	3	5.50%	4
S	4	4	2	5.10%	4
T	5	4	3	4.50%	4
U	1	5	2	1.00%	3
V	2	5	1	1.50%	3
W	3	5	1	2.00%	3
X	4	5	1	2.70%	4
Y	5	5	3	3.00%	4

For the sake of this problem, the beneath bench (level 0) is assumed to be clustered as presented in Figure 2(a). It is assumed that this level has 25 blocks on a regular grid of size 5×5. According to the data presented in Table 1, there are 25 blocks in the active bench, labeled 1 to 25. Furthermore, like the lower bench, they are assumed to be aligned in a regular 5×5 grid on the active bench. The sketches of grade distribution and rock types are presented in Fig. 2(b) and 2(c) respectively.

As it was mentioned in previous sections, some parameters are required in hierarchical clustering algorithm. These parameters are r , c , W_D , W_G , W_R and W_C . The parameters used in this example are presented in Table 2.

Table 2: Parameters

W_D	W_G	W_R	W_C	r	c
1	1	1	1	0.8	0.2

The similarity matrix used in hierarchical clustering algorithm is presented in appendix A. The following sketch presented in Fig. 3(a) shows the initial state solution from hierarchical clustering algorithm. As the result of applying the hierarchical algorithm, five clusters are made in the last iteration. Applying the Tabu Search to improve the initial state solution has resulted in Fig. 3(b). After 8 Tabu Search iterations, the total number of arcs is reduced from 20 (as result of hierarchical) to 17.

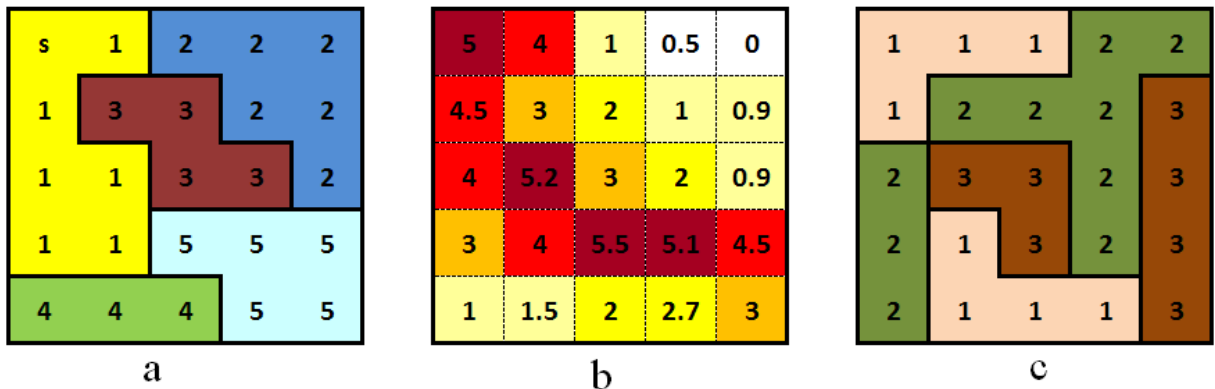


Fig. 2. (a): beneath clusters, (b): grade distribution, (c): rock types.

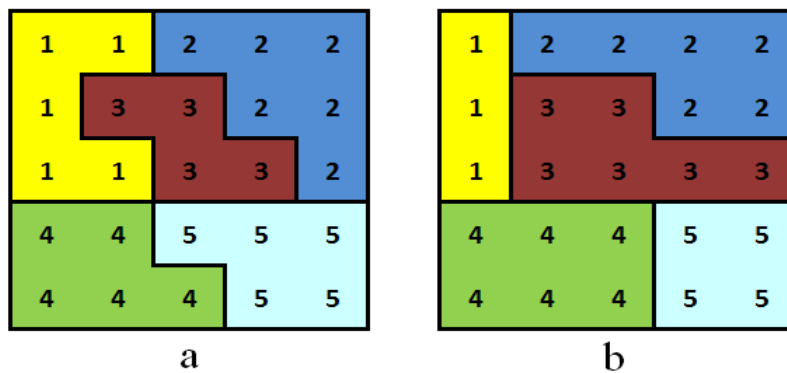


Fig. 3. (a): before Tabu Search, (b): after Tabu Search.

6. Illustrative examples

The proposed algorithm is applied to different data sets. Fig. 4 and Fig. 5 illustrate a gold, silver, and copper deposit. The blue section shows waste material and the red blocks illustrate ore. Another case is illustrated with an iron ore deposit, there are 2598 blocks in seven benches. The size of blocks is $50 \times 25 \times 25$ meters. The algorithm is executed by a MATLAB code. The plan views of the benches 13 and 14 showing the orebody and clusters created by hierarchical clustering and tabu search illustrated in Fig. 6 and Fig. 7.

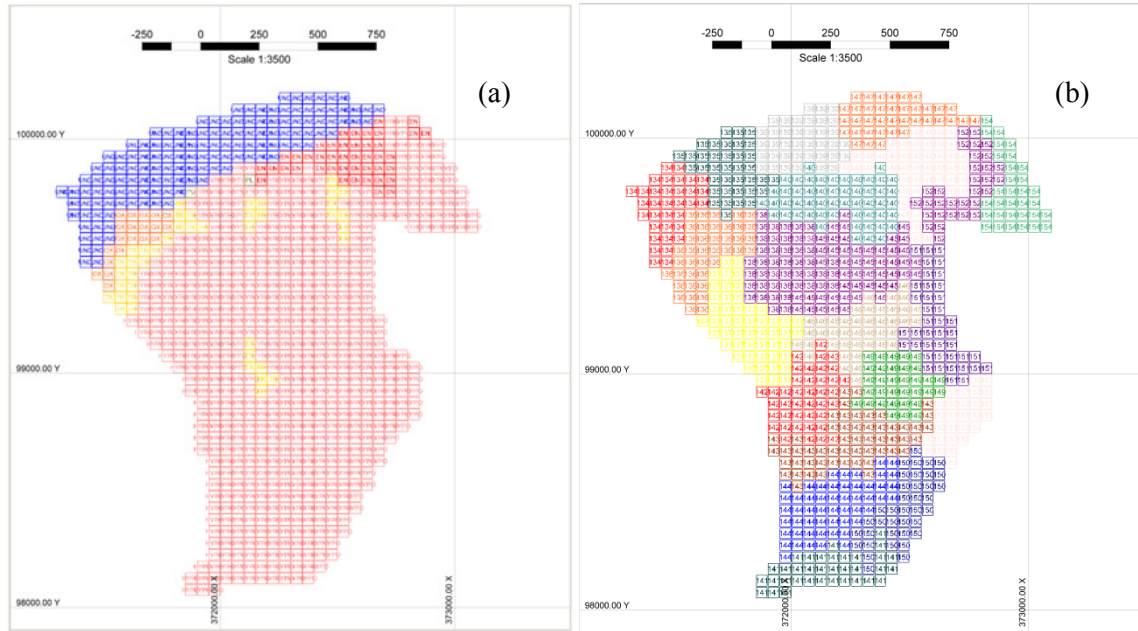


Fig. 4. Gold, Silver, Copper deposit, (a) rock type (b) mining-cuts.

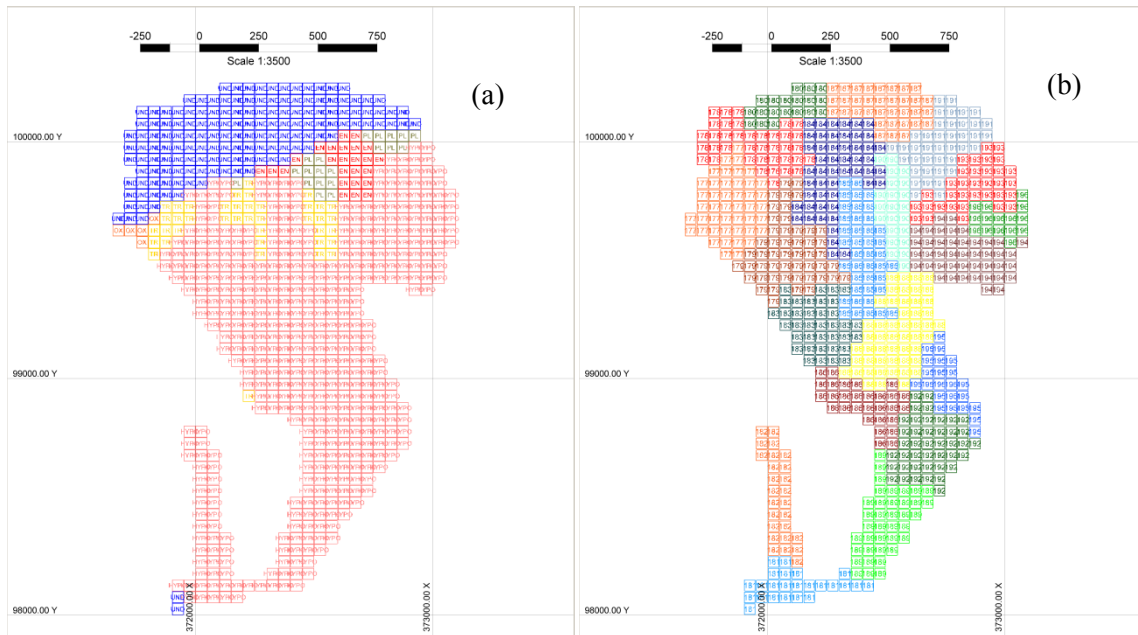


Fig. 5. Gold, Silver, Copper deposit, (a) rock type (b) mining-cuts.

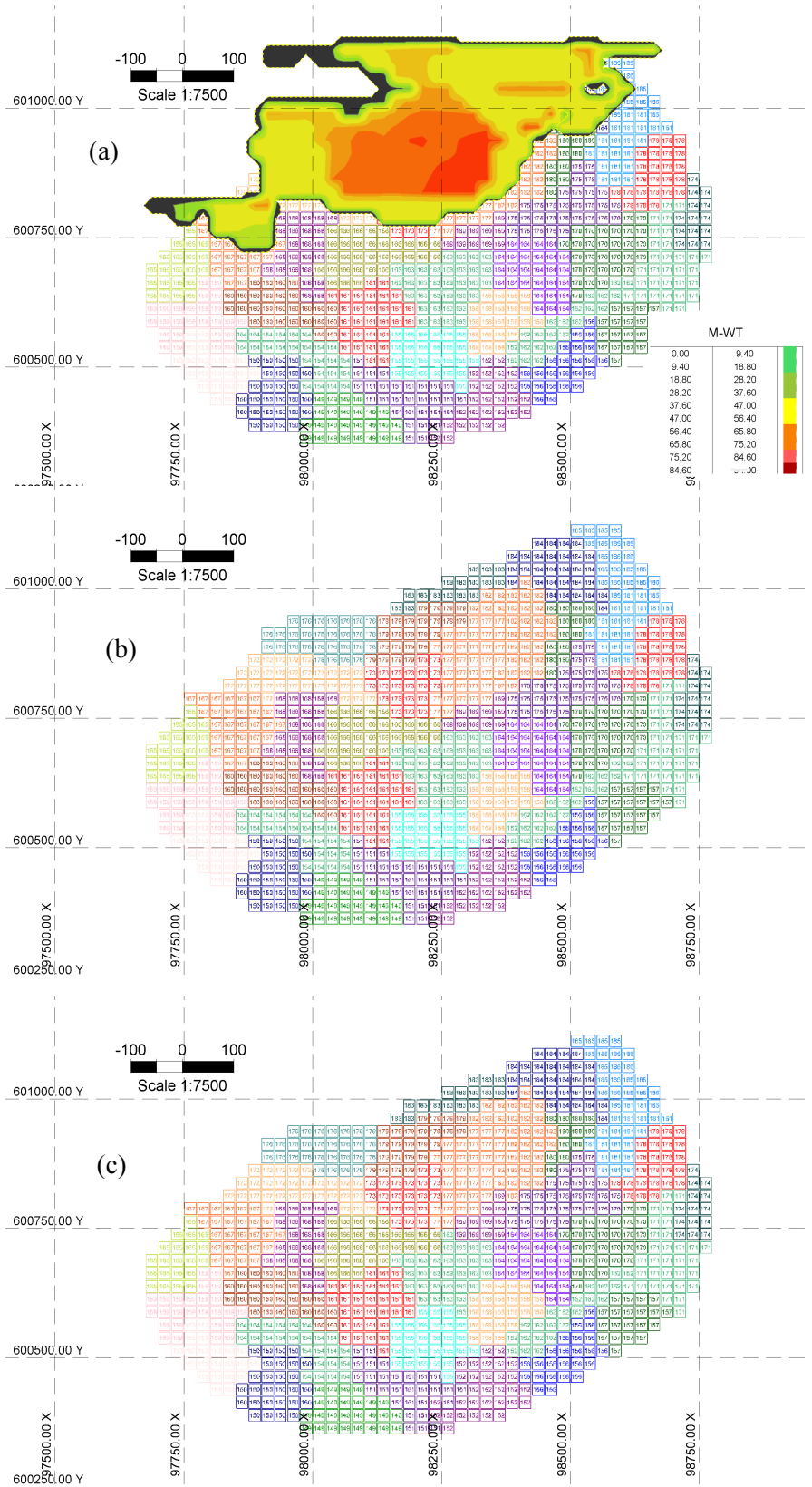


Fig. 6. Plan view of aggregated mining-cuts, bench 13, (a) orebody, (b) hierarchical clustering, and (c) tabu search.

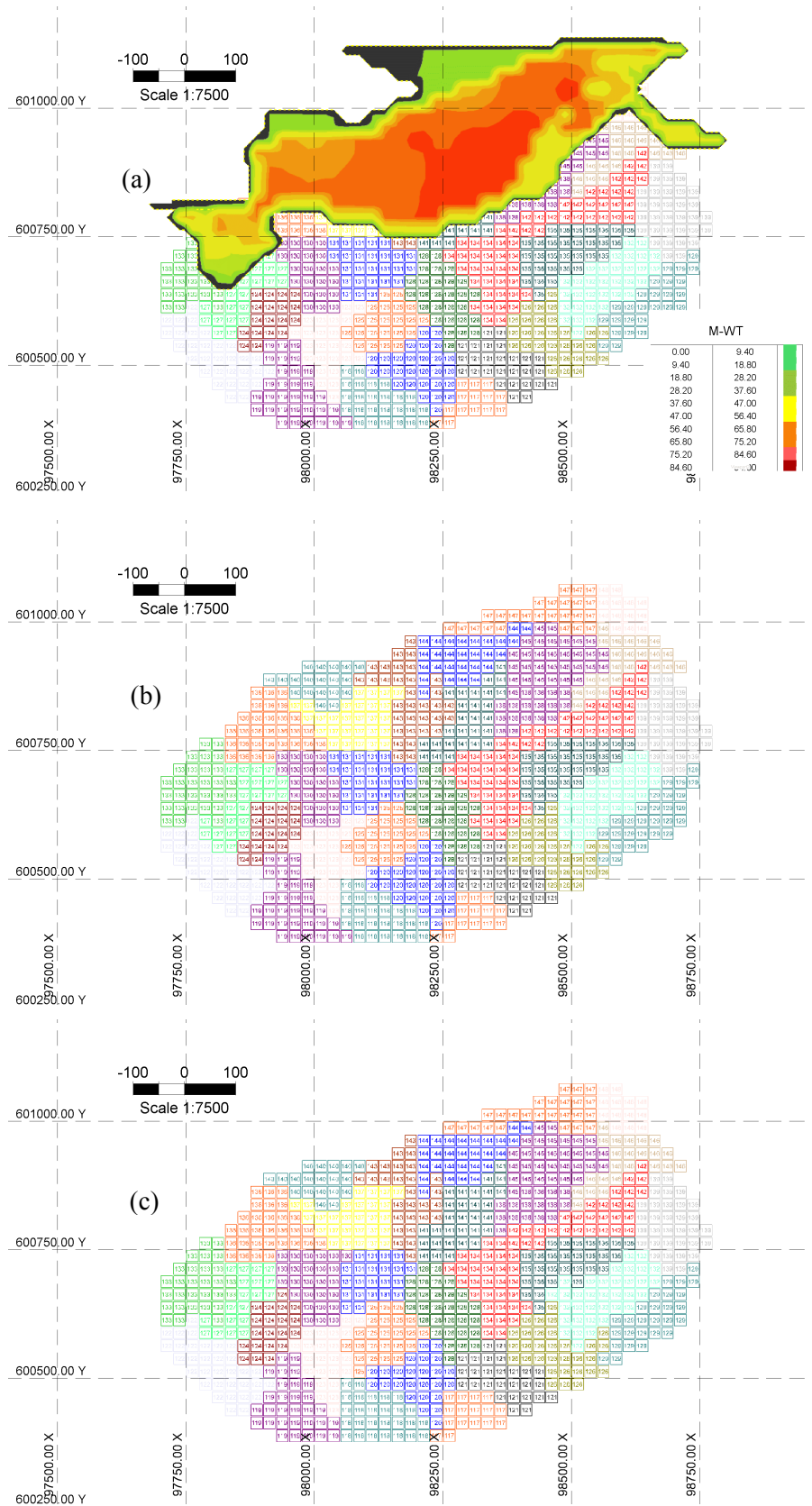


Fig. 7. Plan view of aggregated mining-cuts, bench 14, (a) orebody, (b) hierarchical clustering, and (c) tabu search.

7. Conclusions

A meta-heuristic approach is developed, based on Tabu Search (TS), which aggregates the blocks into larger clusters in each mining bench representing a selective mining unit. The main objective of the developed clustering algorithm is to reduce the number of binary integer variables in the mixed integer linear programming formulation of open pit production scheduling problem. To achieve this goal, the blocks are aggregated into larger units referred to as mining-cuts. The definition of similarity among blocks is based on rock types, grade ranges, spatial location, and the dependency of extraction of a mining-cut to the mining-cuts located above it. The algorithm is developed and tested on synthetic data. As future work it seems relevant to fully integrate this clustering algorithm into an MILP mine production scheduling framework. The efficiency of the new platform can also be evaluated through case studies on large-scale open pit mines.

8. References

- [1] Amaya, J. e. a. (2009). A scalable approach to optimal block scheduling.
- [2] Askari-Nasab, H., & Awuah-offei, K. (2008). Mixed integer linear programming formulations for open-pit production scheduling. *Mining optimization laboratory (MOL)*,
- [3] Boland, N., et al. (2009). LP-based disaggregation approaches to solving the open pit mining production scheduling problem with block processing selectivity. *Computers & Operations Research*, 36,(4), 1064 - 1089.
- [4] Gaupp, M. (2008). Methods for improving the tractability of the block sequencing problem for open-pit mining. PhD thesis Thesis, Colorado school of Mines, Golden,
- [5] Johnson, S. C. (1967). Hierarchical Clustering Schemes. *Psychometrika*, 32,(3), 241-254.
- [6] Marcos Dósea, L. S., Maria A. Silva, Sócrates C.H. Cavalcanti (2008). Adaptive Mean-Linkage with Penalty: A new algorithm for cluster analysis. *Chemometrics and Intelligent Laboratory Systems*, 94,(1), 1-8.
- [7] Martinez, M. A. (2006). Improving the Performance of a Mixed-Integer Production Scheduling Model for LKAB's Iron Ore Mine, Kiruna, Sweden. Thesis, Colorado school of Mines, Golden, Pages 70.
- [8] Newman, A. M., Kuchta, M. (2007). Using aggregation to optimize long-term production planning at an underground mine. *European Journal of Operational Research*, 176,(2), 1205-1218.
- [9] Organization, B. R. (2009). Optimization - Penalty Functions, Solving constrained optimization problems. ASAP TECHNICAL, Retrieved from: <http://www.breault.com/k-base.php?kbaseID=252&catID=44>
- [10] Ramezan, S. (2007). The new Fundamental Tree Algorithm for production scheduling of open pit mines. *European Journal of Operational Research*, 177,(2), 1153-1166.
- [11] Stojmenovi'c, A. N. I. (2008). *Handbook of Applied Algorithms: Solving Scientific, Engineering and Practical Problems*. John Wiley & Sons, Inc.

9. Appendix

[MATLAB code and documentation for hierarchical clustering and tabu search algorithms](#)