Clustering Mining Blocks in Presence of Geological Uncertainty

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

ABSTRACT

A major trend in mine production planning research is incorporating geological uncertainty in the processes of planning. Many mathematical models and heuristic approaches are proposed to deal with the uncertainty. However, incorporating geological uncertainty in planning, usually through Geostatistical realizations, significantly increases the problem sizes and forces researchers to use heuristics or aggregation techniques to be able to obtain solutions in reasonable times. In this paper, we present four variations of agglomerative hierarchical clustering algorithm, one based on deterministic estimates of properties and three based on possible worlds approach which use Geostatistical realizations to form aggregates with regard to the geological properties and the existing uncertainties. We show, through case studies, that uncertainty based algorithms can result in aggregates that are less susceptible to uncertainties, and at the same time, the proposed algorithm can produce aggregates that are within a controlled size and have minable shapes.

1. Introduction

Open pit mine production scheduling problem (OPMPSP) is a complicated procedure which has attracted many researchers in the past decades. Many of the research is centred around formulating the problem using mixed integer linear programming and solving it through exact and heuristics methods. Bienstock and Zuckerberg (2010) proposed a technique (BZ) to solve the LP-relaxation of large instances of the OPMPSP in a few minutes. Similarly, Chicoisne, Espinoza, Goycoolea, Moreno, and Rubio (2012) proposed a topological sort algorithm to solve the LP-relaxation in a reasonable time. This encouraged many other researchers to use the algorithms to obtain the LP-relaxation solution and use heuristics to obtain integer solutions. Samavati, Essam, Nehring, and Sarker (2017), Samavati, Essam, Nehring, and Sarker (2017), Liu and Kozan (2016), Lamghari, Dimitrakopoulos, and Ferland (2015) can be named among those who took advantage of these advancements and developed heuristic solution techniques. However, most of the work in the literature is limited to the production planning problem in the absence of grade blending and stockpiling constraints as they change the structure of the model. Kumral (2011) propose a two stage modelling approach to incorporate geological uncertainty in production. Their approach aims at maximizing the minimum NPV using multiple geological and price uncertainty scenarios. However, the authors implement the approach on a very small block model with only 3,115 blocks in the final pit and do not provide processing times. Solving such model on a real-size block containing hundreds of thousands of blocks will not be possible.

On the other hand, when it comes to incorporating uncertainty in production planning, most of the researchers have to use heuristics to obtain good solutions. (e.g. Lamghari and Dimitrakopoulos (2016), Montiel and Dimitrakopoulos (2015)). Another approach that can help solve the production planning problem with presence of grade blending, stockpiling and uncertainty is to use aggregation techniques to reduce the size of the problem and increase the practicality of the generated solutions.

Boland, Dumitrescu, Froyland, and Gleixner (2009) propose a solution procedure based on aggregation and disaggregation. In their work, excavation decisions are made based on some given aggregates through an MIP. Afterwards, disaggregation techniques are proposed to enable the model make processing decisions in block level by introducing the concept of bins. Jélvez, Morales, Nancel-Penard, Peypouquet, and Reyes (2016) propose an aggregation-disaggregation technique to solve the OPMPSP. Their technique starts by aggregating (reblocking) the model into larger units. Then, they use a sliding time window heuristic to solve the OPMPSP for the aggregated block model and use the solution to categorize the aggregates into two groups. The first group consists of the blocks which all their direct predecessor and successor blocks are extracted in the same period. The rest are called border blocks. Afterwards, they fix the variables corresponding to non-border blocks, disaggregate the border blocks and solve the problem in disaggregated level. The authors implement their technique on MineLib (Espinoza et al., 2013) datasets and show that their technique can solve all the instances to near-optimality in a reasonable time period.

Goodfellow and Dimitrakopoulos (2016) propose a stochastic formulation for OPMPSP with uncertainty. Their model aims at optimizing the production schedule and downstream processes simultaneously. The authors incorporate uncertainty by minimizing deviations from target while maximizing the NPV. First, they use k-means clustering to group blocks into clusters with similar grades. Next, they formulate a non-linear mathematical formulation to maximize the NPV of the operation with respect to various constraints on capacities, grade control and stockpiling. Since the model is non-linear they use three different meta-heuristic algorithms to solve it. They implement their technique on a gold-copper deposit with one stockpile and six processing streams. They first solve the deterministic case with the three algorithms and then apply the same strategy to the stochastic model. They conclude that incorporating uncertainty in decision making results in higher NPV and less risk.

Although it is possible to cluster blocks using the expected attributes and use the clusters in production planning in presence of uncertainty, as done by Goodfellow and Dimitrakopoulos (2016), it is a major step forward to consider the uncertainty in attributes at the clustering stage. The literature in computing science shows good potential in treating object (block) attributes as uncertain variables and perform clustering with respect to this uncertainty. It has been shown that the outcome of such clustering technique is more reliable compared to the ones based solely on expected values or averages (Aggarwal, 2007). There are various methods for clustering uncertain objects proposed in the computing science literature. However, applying such techniques to blocks in a mining operation requires significant modifications and innovations. A clustering algorithm for mining blocks requires control over size, shape and number of created clusters, unlike general purpose clustering algorithms where the ultimate goal is to find the underlying similarities and dissimilarities between the data points. Moreover, techniques such as information-theoretic approach (Gullo et al., 2008), mixture model clustering (Hamdan and Govaert, 2005), FOPTICS (Kriegel and Pfeifle, 2005), FDBSCAN (Donghua and Lilei, 2011) and UK-means (Aggarwal et al., 2016) can be applied to blocks ignoring the strong dependency between attributes of one block and the adjacent one. However, the geographical dependency of block attribute distribution prohibits us from using these algorithms as they share the independency assumption. Therefore, another approach called possible worlds (Volk et al., 2009) has been chosen in this paper. Combining this approach with the deterministic approach in Tabesh, Mieth, and Askari–Nasab (2014) resulted in a clustering algorithm that, using sequential Gaussian simulation realizations, provides aggregates that incorporate uncertainty of block attributes and can be used in future research on open pit and underground mining simulation and production planning in presence of uncertainty.

2. Theoretical framework

Clustering is the process of grouping similar objects together using measures of similarity and dissimilarity. The similarity of mining blocks is usually defined using the grade and rock type attributes of the blocks. However, these attributes are estimated based on the drillhole data collected and include uncertainty.

The possible worlds approach as proposed by Volk, et al. (2009) starts by using a value generator function (VG-function) to generate the possible worlds. In a mining block model, due to the geographical dependencies between the rock type and grades of the of the blocks, the best generator function to use is the Sequential Gaussian Simulation (SGS) (Rossi and Deutsch, 2014). The next step is to apply any clustering algorithm on every possible world and obtain a clustering model for every world. The clustering processes are independent, and therefore, they can run on parallel processes. Finally, an aggregation technique is required to derive the final clustering model from the generated clustering models for every possible world.

In this paper, we propose four variations of the agglomerative hierarchical clustering algorithm with shape and size control (AHCA) from Tabesh, et al. (2014). The first algorithm uses the expected values of parameters and runs the AHCA once to obtain the clustering model. The second algorithm runs the AHCA on every realization obtained from SGS and aggregates blocks into clusters if they happen to be grouped together in more than 50% of the single realization cluster models. The third algorithm runs the AHCA on every realization and uses the frequency of two blocks being grouped together as the similarity between the blocks and runs AHCA again to form the final clusters. The forth variation uses k-means to cluster every realization and aggregates the results using AHCA.

First three variations run the AHCA using the similarity index calculated based on equation (1). The equation is a simplified version of the original similarity index defined in M. Tabesh and Askari-Nasab (2013) and is divided into two parts for computation economics. The first part is the distance measure which is independent of the realizations and does not vary by sampled values of grade and rock type. The second part, however, is calculated for every realization in the realization based variations of the algorithm. $\tilde{D}_{ij}$ represents the normalized Euclidean distance between blocks i and j (equation (1)) and $\tilde{G}_{ij}^{k}$ represents the normalized difference between the grade values of blocks i and j in realization k. Although, a block model may have multiple elements, the formulation here, without losing generality, is presented for one major element grade only. Moreover, we can assign weights to these parameters and include more parameters in the similarity index definition. The effects of weights and other parameters are studied in Tabesh, et al. (2014) and Tabesh and Askari-Nasab (2013) and they are removed from this paper for brevity. R_{ij}^{k} is calculated based on equation (5), which considers a penalty value of $r \in [0,1]$ if the two blocks have different rock types and 1, if the blocks are of the same rock type in realization k. Finally, we need to determine the adjacency between the block using an adjacency threshold as in equation (6) in order to avoid forming fragmented clusters.

$$\tilde{D}_{ij} = \frac{D_{ij}}{\max(D_{ij})} \tag{1}$$

$$S_{ij} = \frac{1}{\tilde{D}_{ij}} \tag{2}$$

$$S_{ij}^{k} = \frac{R_{ij}^{k}}{\tilde{G}_{ij}^{k}} \tag{3}$$

$$S_{ij}^{k} = \frac{1}{\tilde{D}_{ij}} \times \frac{R_{ij}^{k}}{\tilde{G}_{ij}^{k}} \qquad (4)$$

$$R_{ij}^{k} = \begin{cases} 1 \; if \; blocks \, i \, and \; j \, are \; from \, the \, same \, rock \, type \\ r \qquad\qquad\qquad\qquad\qquad\qquad otherwise \end{cases} \qquad (5)$$

$$A_{ij} = (D_{ij} < AdjacencyTreshold) \qquad (6)$$

2.1. Algorithm 1 - Expected value algorithm (EVA)

This algorithm is very similar to the AHCA proposed in Tabesh, et al. (2014). Since the only uncertain parameters are rock type and grade we will determine the expected values of these attributes first and use them in forming the clustering model. The grade attributes are numerical, and therefore, we can use arithmetic average of all the realizations to determine the expected values. However, rock types are categorical attributes and arithmetic averages cannot be applied. Hence, we set the most frequent rock type from all the realizations as the expected rock type of each block. Since the realizations are preprocessed and we have assigned single values to grade and rock type, we only need to calculate the similarity index in equation (1) once and use it to run the AHCA. The resulted clustering model will be the final clustering scheme. Note that this algorithm is not based on possible worlds and is presented as a basis for comparison. The pseudo code of the algorithm is presented in Table 1.

Table 1.Expected Value Clustering Pseudo Code

```
FOR EACH i in Blocks
    FOR EACH j in Blocks
        $S_{ij} = \frac{1}{\tilde{D}_{ij}} \times \frac{R_{ij}}{\tilde{G}_{ij}}$
        $A_{ij} = (D_{ij} < AdjacencyTreshold)$
    NEXT
NEXT
FUNCTION Clusters = Clustering(S,A)
    NumClusters = NumBlocks
    WHILE NumClusters > DesiredNumofClusters
        $(i, j) = Max(S)$
        IF length(Clusters(i)) + length(Clusters(j)) <= MaxClusterSize THEN
            $S_{i:} = Min(S_{i:}, S_{j:})$
            $S_{j:} = 0$
            $A_{i:} = Max(A_{i:}, A_{j:})$
            $A_{j:} = 0$
            Clusters(i) = Clusters(i) + Clusters(j)
            NumClusters = NumClusters – 1
        ELSE
            $A_{ij} = 0$
        ENDIF
    ENDWHILE
    Clusters = ShapeRefinementProcedure(Clusters)
END FUNCTION
```

2.2. Algorithm 2 - Simple realization aggregation (SRA)

As explained earlier, possible world clustering works by sampling possible worlds from the uncertain data using a generator function, clustering every sampled set and aggregating the generated clustering models. Both following algorithms start by running the AHCA on every realization and obtain one clustering model per realization. The difference between the two algorithms is in the aggregation step. The aggregation step in the SRA algorithm is summarized in equations (7) and (8). First, we calculate the common cluster matrix (C_{ij}^{k}) for every realization cluster model. Next, the common cluster values are summed up over all the realizations and we calculate the frequency of two blocks being in the same cluster (F_{ij}). Finally, every pair of blocks with $F_{ij} \geq 0.5$ are grouped together in the final clustering model. The pseudo code of the algorithm is presented in Table 2.

$$C_{ij}^{k} = \begin{cases} 1 & \textit{if blocks } i \textit{ and } j \textit{ are from the same cluster} \\ 0 & \textit{otherwise} \end{cases} \tag{7}$$

$$F_{ij} = \frac{\sum_{r=1}^{R} C_{ij}^{k}}{R} \tag{8}$$

Table 2.Simple Realization Aggregation Pseudo Code

```
FOR EACH k in Realizations
    FOR EACH i in Blocks
        FOR EACH j in Blocks
            $S_{ij}^{k} = \frac{1}{\tilde{D}_{ij}} \times \frac{R_{ij}^{k}}{\tilde{G}_{ij}^{k}}$
            $A_{ij} = (D_{ij} < AdjacencyTreshold)$
        NEXT
    NEXT
    $Clusters^{k} = Clustering(S^{k}, A)$
    FOR EACH i in Blocks
        FOR EACH j in Blocks
            $C_{ij}^{k} = \begin{cases} 1 & \textit{if blocks } i \textit{ and } j \textit{ are from the same cluster} \\ 0 & \textit{otherwise} \end{cases}$
        NEXT
    NEXT
NEXT
FOR EACH i in Blocks
    FOR EACH j in Blocks
```

```
        $F_{ij} = \frac{\sum_{r=1}^{R} C_{ij}^{k}}{R}$
    NEXT
NEXT
FOR EACH i in Blocks
    $Clusters(i) = F_{i:} \geq 0.5$
NEXT
```

2.3. Algorithm 3 - Hierarchical realization aggregation (HRA)

In this algorithm, we run the AHCA on every realization and calculate C_{ij}^{k} and F_{ij} as explained earlier in (7) and (8). Afterwards, the F_{ij} values are used as similarity indices in the original AHCA instead of S_{ij} and the algorithm is run using the same settings as the original AHCA used on every realization. The pseudo code of the algorithm is presented in Table 3.

Table 3.Hierachical Realization Aggregation Pseudo Code

```
FOR EACH k in Realizations
    FOR EACH i in Blocks
        FOR EACH j in Blocks
            $S_{ij}^{k} = \frac{1}{\tilde{D}_{ij}} \times \frac{R_{ij}^{k}}{\tilde{G}_{ij}^{k}}$
            $A_{ij} = (D_{ij} < AdjacencyTreshold)$
        NEXT
    NEXT
    $Clusters^{k} = Clustering(S^{k}, A)$
    FOR EACH i in Blocks
        FOR EACH j in Blocks
            $C_{ij}^{k} = \begin{cases} 1 & if\ blocks\ i\ and\ j\ are\ from\ the\ same\ cluster \\ 0 & otherwise \end{cases}$
        NEXT
    NEXT
NEXT
FOR EACH i in Blocks
    FOR EACH j in Blocks
        $F_{ij} = \frac{\sum_{r=1}^{R} C_{ij}^{k}}{R}$
    NEXT
NEXT
$Clusters = Clustering(F, A)$
```

2.4. Algorithm 4 - K-Means with Hierarchical realization aggregation (KHRA)

K-means is a famous clustering algorithm which is known to be faster but less accurate than hierarchical clustering (Feng et al., 2010). Moreover, k-means, in its original form, is not designed to deal with categorical variables such as rock type (Mastrogiannis et al., 2009) and it does not provide control over shape and size of the generated clusters. Since block models have a mixture of numerical and categorical attributes we have modified the k-means algorithm to be able to apply it on block models.

In order to apply k-means clustering we first need to create the feature matrix. We use the same set of attributes, weights and penalties as AHCA to be able to compare the results. The first two columns of the feature matrix are X and Y coordinates of the blocks on each bench. The coordinates are powered to the distance weight and normalized by dividing all the values by the maximum values of each column. Similarly, the major element grade values are powered to the grade weight, normalized and added as the third column. Next, in order to add rock types to the feature matrix, we add as many columns as the number of categories i.e. number of rock types. If a block is of rock type *i,* the *i*th column will get a value of *1-r* (one minus rock type penalty). After preparing the feature matrix for every realization, k-means clustering is performed and C_{ij}^{k} and F_{ij} are calculated as explained earlier in (7) and (8). Finally, the F_{ij} values are used as similarity indices in AHCA to aggregate the clustering results and form the final clustering scheme. The pseudo code of the algorithm is presented in Table 4.

Table 4. K-Means with Hierachical Realization Aggregation Pseudo Code

```
FOR EACH k in Realizations
    $S^{k} = BuildFeatureMatrix()$
    $Clusters^{k} = KMeans(S^{k})$
    FOR EACH i in Blocks
        FOR EACH j in Blocks
            $C_{ij}^{k} = \begin{cases} 1 & if\ blocks\ i\ and\ j\ are\ from\ the\ same\ cluster \\ 0 & otherwise \end{cases}$
        NEXT
    NEXT
NEXT
FOR EACH i in Blocks
    FOR EACH j in Blocks
        $F_{ij} = \frac{\sum_{r=1}^{R} C_{ij}^{k}}{R}$
    NEXT
NEXT
$Clusters = Clustering(F, A)$
```

2.5. Result evaluation

Since the goal of block clustering is to create mining polygons which are homogenous in grade, rock type and destination, Tabesh and Askari-Nasab (2013) define three performance measures for a block clustering algorithm: rock unity (RU), destination dilution factor (DDF) and grade coefficient of variation (CV). Moreover, the generated clusters will be used as planning units, and thus, need to be of controlled size and shape. In addition, since all the created Geostatistical realizations reproduce the spatial continuity of the variables and have equal probability of occurrence (Rossi and Deutsch, 2014), a good clustering scheme is the one that produces better results over more realizations. Therefore, we evaluate every clustering result by looking at the average and standard deviations of the three performance measures, introduced by Tabesh and Askari-Nasab (2013), over all the realizations. In other words, after obtaining a clustering scheme from every algorithm we apply the scheme to every realization and calculate the performance measures. Moreover, we look at the cluster size variations and cluster shapes resulted from every algorithm.

3. Case studies

In order to evaluate the proposed algorithms we implement them on two case studies provided by Malaki, Khodayari, Pourrahimian, and Liu (2017) and Cabral Pinto (2016). The first dataset is a copper deposit with three different rock types. The dataset block dimensions are $10\times10\times10$ meters and there are a total of 189,000 blocks in the model. The dataset contains 20 realizations of rock types and element grade estimates created using SGS (Malaki et al., 2017). The second dataset is a multi element dataset where the element names and coordinates are distorted for confidentiality. The block model dimensions are $300\times300\times20$ feet and there are a total of 570,770 blocks in 176 benches. The element grades and rock types are estimated from the drill hole data and 100 realizations of geological uncertainty are created through SGS. The case studies are tested on machine with Intel® Xeon® CPU with eight cores with 2.8 GHz speed and 24 GB of ram.

3.1. Case study 1

The first case study is a Copper deposit borrowed from Malaki, et al. (2017). Although the dataset is a block-caving block model, the results can be generalized to other open-pit and underground methods that follow the same procedure of creating block model realizations and grouping the blocks prior to planning. A 3D presentation of the ore body is shown in Fig. 1.

Every bench of the block model contains 7,200 blocks which are clustered using the three algorithms. The first algorithm uses the expected values for rock type and grade while the other algorithms use every realization in the block model. Sample plan views of expected rock type (Fig. 2), expected copper grade (Fig. 3), realization 1 rock type (Fig. 4) and realization 1 copper grade (Fig. 5) are presented here to provide a sense of attribute variations within the block model and between the realizations.

In order to have simple and comparable results, the four algorithms are implemented using the same set of weights and control parameters as summarized in Table 5. The first group of runs are set to aggregate 30 blocks into one cluster and the second group are set to aggregate 60 blocks into one cluster with maximums set to 35 and 70 respectively. Minimum cluster size is not applied and the number of shape refinement iterations is set to 3 for all the runs. Afterwards, the three algorithms are run and the resulted clustering scheme is applied to every realization of the block model. Since all the realizations are equiprobable, we look at the average values of grade CV and rock unity among all the realizations. Destination dilution factor is omitted as blocks do not have a predetermined destination assigned to them.

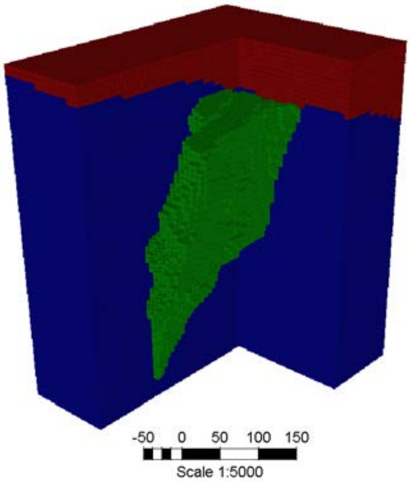

Fig. 1 3D presentation of the ore body

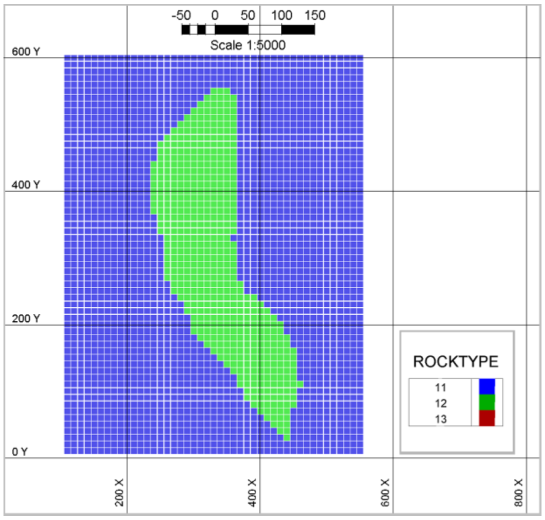

Fig. 2 Expected rock type at level 40

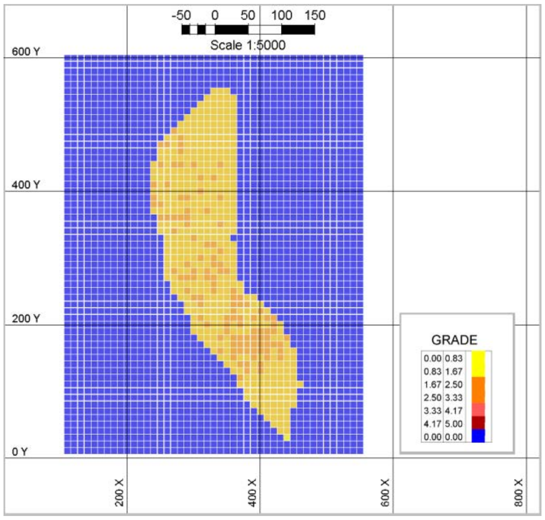

Fig. 3 Expected grade at level 40

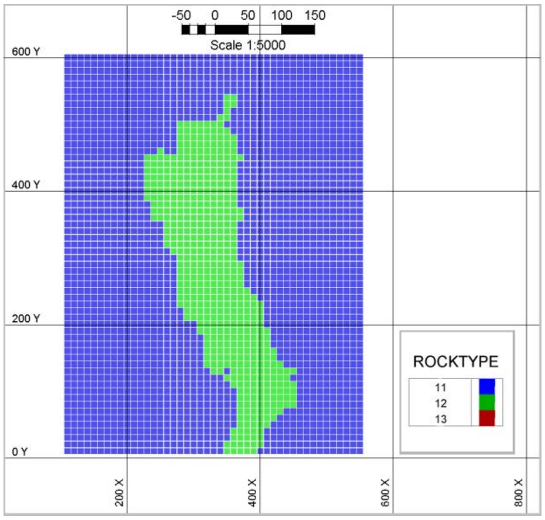

Fig. 4 Realization 1 rock type at level 40

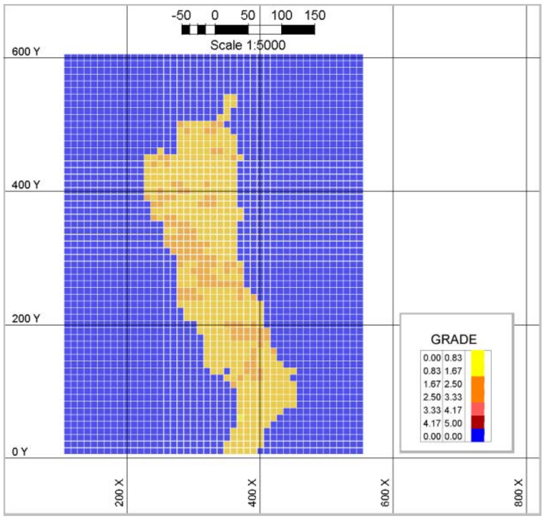

Fig. 5 Realization 1 grade at level 40

Table 5.Case study 1 clustering runs settings

ID	Algorithm	Avg. Block per cluster	Max. Block per cluster	Shape Refin. Iter.
1-EVA-30-35	Expected	30	35	3
1-SRA-30-35	SRA	30	35	3
1-HRA-30-35	HRA	30	35	3
1-KHRA-30-35	KHRA	30	35	3
1-EVA-60-70	Expected	60	70	3
1-SRA-60-70	SRA	60	70	3
1-HRA-60-70	HRA	60	70	3
1-KHRA-60-70	KHRA	60	70	3

The clustering performance measures are presented in Table 6. The first column presents the average copper grade CV over the realizations. A lower grade CV means a better clustering scheme since it illustrates the variation of block grades within a cluster. In contrast, rock unity measures the homogeneity of block rock types within a cluster and a larger value means a better clustering scheme. We are testing the algorithms on two different size settings. The question of the proper size of clusters is thoroughly discussed in Tabesh and Askari-Nasab (2013) and the goal of this case study is to compare the proposed algorithms. According to Table 6, we can conclude that the SRA is performing best while EVA is performing worst. However, a major requirement in clustering blocks in a mining operation is to have clusters of the same size to able to use them as planning units. In this sense, SRA results are significantly poorer than the other algorithms when comparing the size variations. Moreover, comparing the shapes of the generated clusters show that clusters created using SRA are not minable and cannot be used as units of production planning. The sample plan views in Fig. 6 and Fig. 7 show the difference between the resulted clusters from SRA and HRA respectively. EVA and KHRA results have cluster shapes similar to HRA and are presented in Fig. 8 and Fig. 9 respectively. In addition to unacceptable size and shape properties of SRA results, looking at the realization by realization statistics shows that EVA, HRA and KHRA results are more consistent over different realizations while the SRA is varying significantly. Fig. 10 shows the variations in average cluster grade CVs over different realizations and Fig. 11 shows the variations in average cluster rock unities. The two Fig.s show that EVA, HRA and KHRA follow the same patterns with KHRA performing better over all the realizations. In contrast, SRA performance, especially regarding grade variations, is highly sensitive to variations in realizations. In addition to better results, KHRA is significantly faster which can be an important algorithm selection criterion even on a small scale case study as presented here.

Table 6.Case study 1 clustering Results

ID	Average Grade CV	Average Rock Unity	Cluster Size CV	Time (min)
1-EVA-30-35	41.3%	95.7%	20.8%	14
1-SRA-30-35	24.1%	98.6%	357.7%	171
1-HRA-30-35	34.9%	96.4%	32.3%	181
1-KHRA-30-35	28.6%	97.0%	35.6%	13
1-EVA-60-70	75.3%	95.1%	21.4%	13
1-SRA-60-70	50.7%	97.9%	245.4%	167
1-HRA-60-70	63.3%	95.9%	35.3%	176
1-KHRA-60-70	54.2%	96.3%	38.5%	13

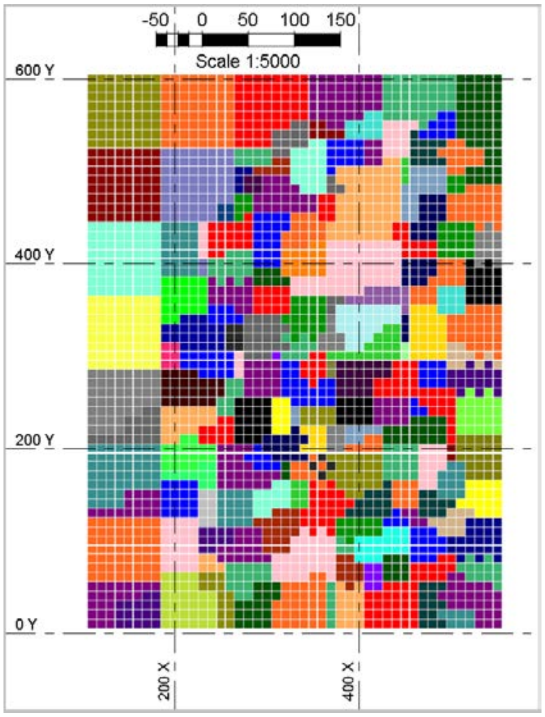

Fig. 6 1-SRA-60-70 results at level 40

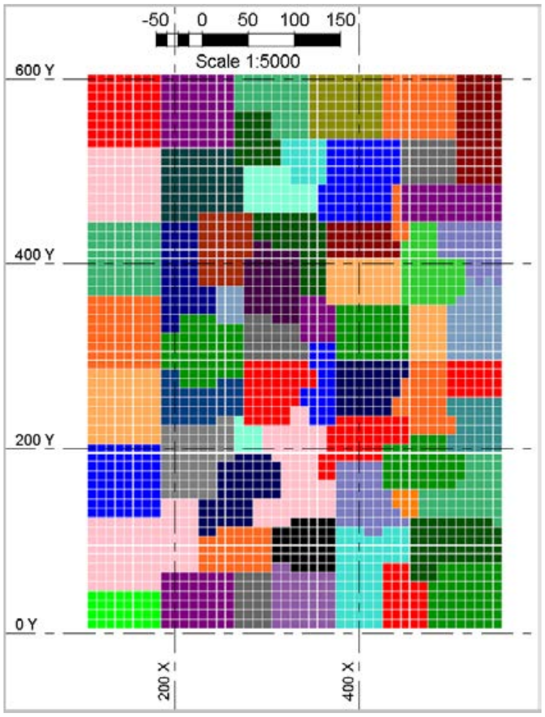

Fig. 7 1-HRA-60-70 results at level 40

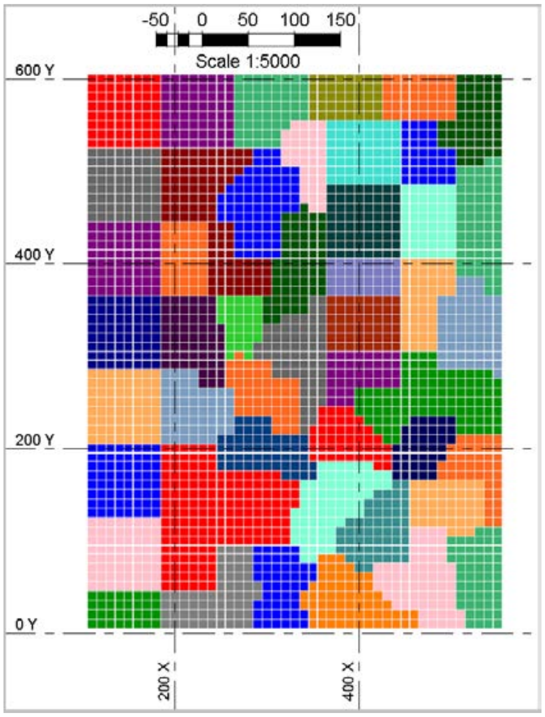

Fig. 8 1-EVA-60-70 results at level 40

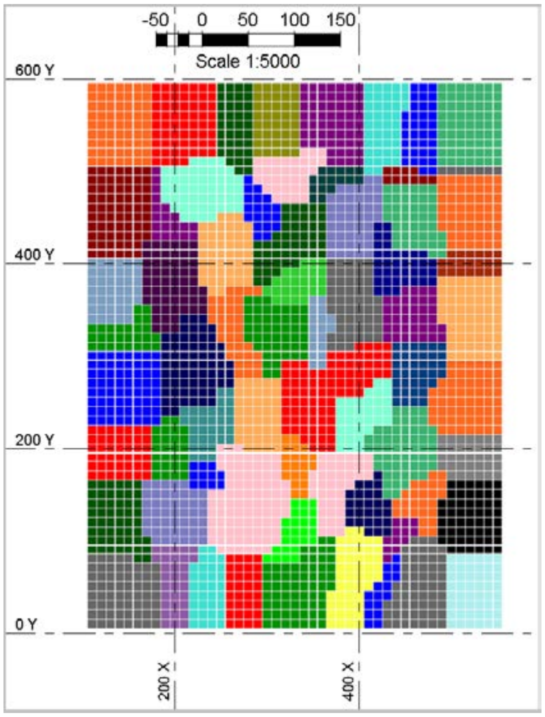

Fig. 9 1-KHRA-60-70 results at level 40

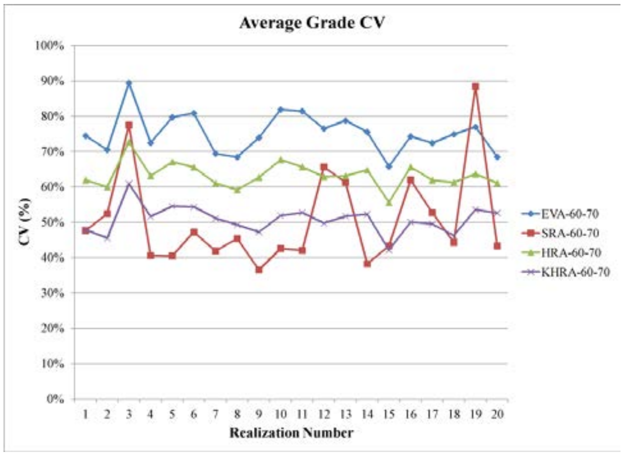

Fig. 10 Average grade CV over realizations

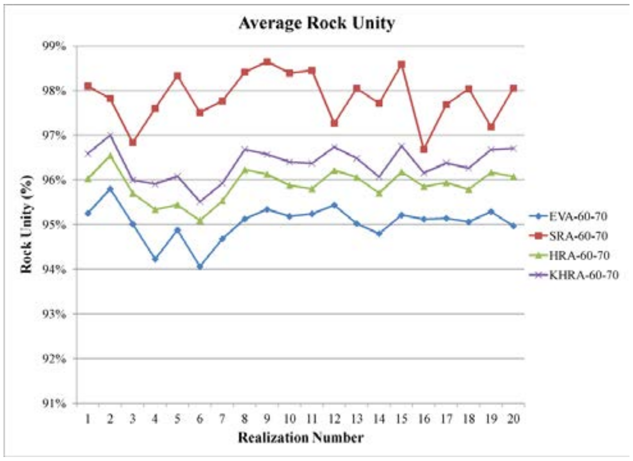

Fig. 11 Average rock unity over realizations

3.2. Case study 2

The second case study is borrowed from Cabral Pinto (2016) with three elements and nine rock types. The element and rock type names are masked for confidentiality as well as the block coordinates and dimensions. However, the large number of blocks and realizations in the block model make it a good measure for evaluating the performance of the proposed algorithms. We applied the same clustering settings as the first case study except we did not perform the SRA settings as it was shown that the results would not be acceptable and they would take a long time to process. The clustering settings for the second case study are summarized in Table 7.

Table 7.Case study 2 clustering runs settings

ID	Algorithm	Avg. Block per cluster	Max. Block per cluster	Shape Refin. Iter.
2-EVA-30-35	Expected	30	35	3
2-HRA-30-35	HRA	30	35	3
2-KHRA-30-35	KHRA	30	35	3
2-EVA-60-70	Expected	60	70	3
2-HRA-60-70	HRA	60	70	3
2-KHRA-60-70	KHRA	60	70	3

As can be seen in Fig. 12 this deposit is more variable in rock types. Fig. 13 and Fig. 14 are two sample plan views of expected grade and rock type distribution in the deposit. The clustering results of HRA and KHRA for the same plan view are presented in Fig. 15 and Fig. 16 for comparison. The summary of running the clustering algorithms on the second case study is presented in Table 8. Similar to the previous case study, KHRA is performing better regarding grade variation and it performs slightly worse regarding rock unity compared to HRA. However, considering the processing time required for running HRA, KHRA is the more reasonable choice of algorithm. Finally, it can be seen that both uncertainty based algorithms outperform the deterministic algorithm over multiple realizations.

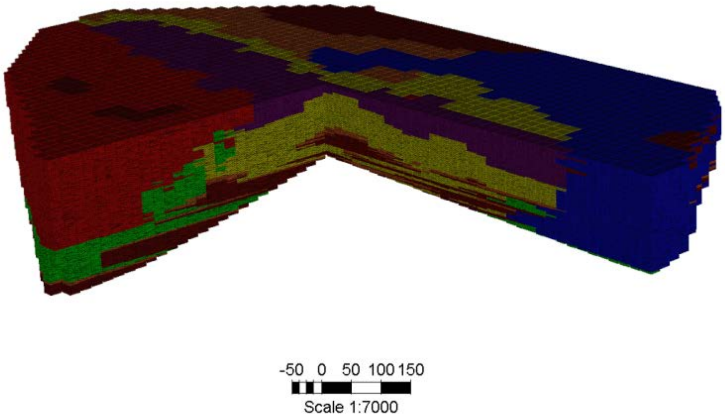

Fig. 12 3D presentation of the ore body

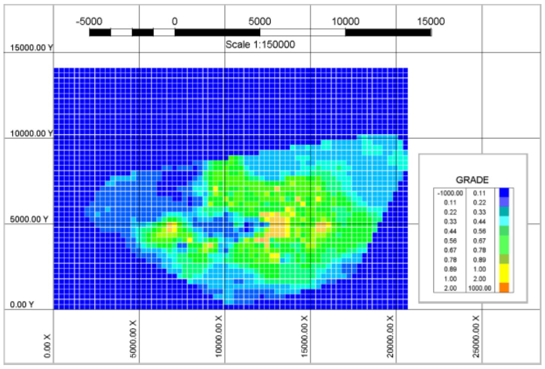

Fig. 13 Expected rock type at level 40

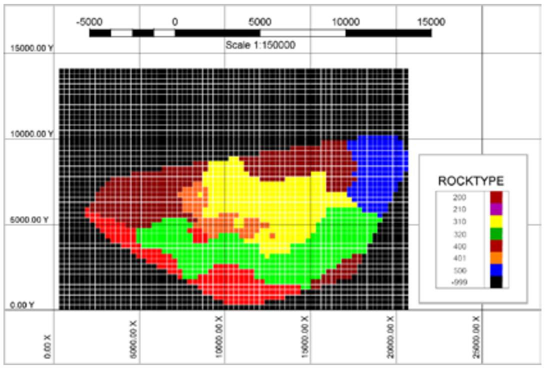

Fig. 14 Expected grade at level 40

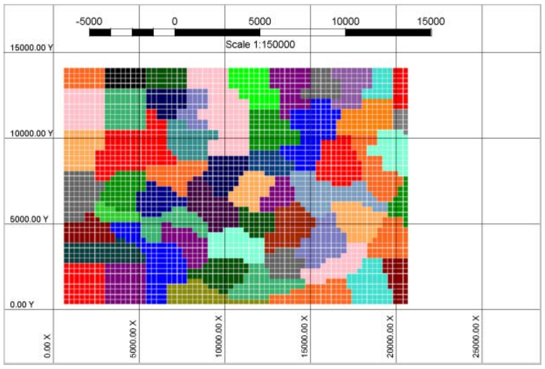

Fig. 15 2-HRA-60-70 results at level 40

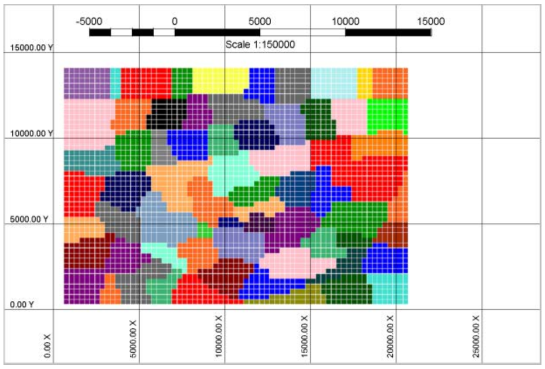

Fig. 16 2-KHRA-60-70 results at level 40

Table 8.Case study 2 clustering Results

ID	Average Grade CV	Average Rock Unity	Cluster Size CV	Time (min)
2-EVA-30-35	34.8%	93.5%	24.0%	54
2-HRA-30-35	35.5%	93.7%	23.6%	3,452
2-KHRA-30-35	27.8%	95.8%	30.1%	70
2-EVA-60-70	38.7%	92.8%	23.8%	53
2-HRA-60-70	39.5%	93.1%	23.7%	3,363
2-KHRA-60-70	29.6%	95.5%	33.4%	69

4. Conclusion

In this paper we presented a hierarchical agglomerative clustering technique based on possible worlds uncertainty approach to aggregate mining blocks into larger units in a mining operation. The clustering technique deals with geological uncertainties in rock type and grade estimates. The first step is to create the possible worlds using SGS realizations as the generator function. Next, we proposed four different algorithms that use these realizations and form one clustering scheme to apply to the block model. Afterwards, we defined quality measures to evaluate the result of each algorithm and choose the one that provides the best results. Finally, we implemented the algorithms on two case studies and showed how considering uncertainties in clustering can improve the results and provide clustering schemes that have higher quality measures over more possible realizations. Moreover, we showed that the hybrid algorithm using k-means and hierarchical aggregation can be implemented on real-size block models in a reasonable processing time to create reliable clustering schemes in presence of geological uncertainty.

5. References

[1] Aggarwal, C. C. (2007). *On Density Based Transforms for Uncertain Data Mining*. in Proceedings of 2007 IEEE 23rd International Conference on Data Engineering, pp. 866-875.

[2] Aggarwal, S., Agarwal, N., and Jain, M. (2016). *Uncertain data mining: A review of optimization methods for UK-means*. in Proceedings of 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 3672-3677.

[3] Bienstock, D. and Zuckerberg, M. (2010). Solving LP Relaxations of Large-Scale Precedence Constrained Problems. in *Integer Programming and Combinatorial Optimization: 14th International Conference, IPCO 2010, Lausanne, Switzerland, June 9-11, 2010. Proceedings*, F. Eisenbrand and F. B. Shepherd, Eds., Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 1-14.

[4] 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. *Computers & Operations Research, 36* (4), 1064-1089.

[5] Cabral Pinto, F. A. (2016). Advances in data spacing and uncertainty.Thesis, University of Alberta, Edmonton, Pages 177.

[6] Chicoisne, R., Espinoza, D., Goycoolea, M., Moreno, E., and Rubio, E. (2012). A New Algorithm for the Open-Pit Mine Production Scheduling Problem. *Operations Research, 60* (3), 517-528.

[7] Donghua, P. and Lilei, Z. (2011). *Uncertain data cluster based on DBSCAN*. in Proceedings of 2011 International Conference on Multimedia Technology, pp. 3781-3784.

[8] Espinoza, D., Goycoolea, M., Moreno, E., and Newman, A. (2013). MineLib: a library of open pit mining problems. *Annals of Operations Research*, *206* (1), 93-114.

[9] Feng, L., Qiu, M.-H., Wang, Y.-X., Xiang, Q.-L., Yang, Y.-F., and Liu, K. (2010). A fast divisive clustering algorithm using an improved discrete particle swarm optimizer *Pattern Recogn. Lett.* , *31* (11), 1216-1225

[10] Goodfellow, R. C. and Dimitrakopoulos, R. (2016). Global optimization of open pit mining complexes with uncertainty. *Applied Soft Computing*, *40* 292-304.

[11] Gullo, F., Ponti, G., Tagarelli, A., and Greco, S. (2008). *A Hierarchical Algorithm for Clustering Uncertain Data via an Information-Theoretic Approach*. in Proceedings of 2008 Eighth IEEE International Conference on Data Mining, pp. 821-826.

[12] Hamdan, H. and Govaert, G. (2005). *Mixture Model Clustering of Uncertain Data*. in Proceedings of The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ '05., pp. 879-884.

[13] Jélvez, E., Morales, N., Nancel-Penard, P., Peypouquet, J., and Reyes, P. (2016). Aggregation heuristic for the open-pit block scheduling problem. *European Journal of Operational Research*, *249* (3), 1169-1177.

[14] Kriegel, H. P. and Pfeifle, M. (2005). *Hierarchical density-based clustering of uncertain data*. in Proceedings of Fifth IEEE International Conference on Data Mining (ICDM'05), pp. 4 pp.

[15] Kumral, M. (2011). Incorporating geo-metallurgical information into mine production scheduling. *Journal of the Operational Research Society*, *62* (1), 60-68.

[16] Lamghari, A. and Dimitrakopoulos, R. (2016). Network-flow based algorithms for scheduling production in multi-processor open-pit mines accounting for metal uncertainty. *European Journal of Operational Research*, *250* (1), 273-290.

[17] Lamghari, A., Dimitrakopoulos, R., and Ferland, J. A. (2015). A hybrid method based on linear programming and variable neighborhood descent for scheduling production in open-pit mines. *Journal of Global Optimization*, *63* (3), 555-582.

[18] Liu, S. Q. and Kozan, E. (2016). New graph-based algorithms to efficiently solve large scale open pit mining optimisation problems. *Expert Systems with Applications*, *43* 59-65.

[19] Malaki, S., Khodayari, F., Pourrahimian, Y., and Liu, W. V. (2017). An application of mathematical programming and sequential Gaussian simulation for block cave production scheduling. in *Proceedings of the First International Conference on Underground Mining Technology*, M. Hudyma and Y. Potvin, Eds. Sudbury: Australian Centre for Geomechanics, pp. 323-337.

[20] Mastrogiannis, N., Giannikos, I., Boutsinas, B., and Antzoulatos, G. (2009). CL.E.KMODES: a modified k-modes clustering algorithm. *Journal of the Operational Research Society*, *60* (8), 1085-1095.

[21] Montiel, L. and Dimitrakopoulos, R. (2015). Optimizing mining complexes with multiple processing and transportation alternatives: An uncertainty-based approach. *European Journal of Operational Research*, *247* (1), 166-178.

[22] Rossi, M. E. and Deutsch, C. V. (2014). Recoverable Resources: Simulation. in *Mineral Resource Estimation*, Dordrecht, Springer Netherlands, pp. 167-192.

[23] Samavati, M., Essam, D., Nehring, M., and Sarker, R. (2017). A local branching heuristic for the open pit mine production scheduling problem. *European Journal of Operational Research*, *257* (1), 261-271.

[24] Samavati, M., Essam, D., Nehring, M., and Sarker, R. (2017). A methodology for the large-scale multi-period precedence-constrained knapsack problem: an application in the mining industry. *International Journal of Production Economics*, *193* 12-20.

[25] Tabesh, M. and Askari-Nasab, H. (2013). Automatic Creation of Mining Polygons using Hierarchical Clustering Techniques. *Journal of Mining Science*, *49* (3), 426-439.

[26] Tabesh, M., Mieth, C., and Askari-Nasab, H. (2014). A multi–step approach to long–term open–pit production planning. *International Journal of Mining and Mineral Engineering*, *5* (4), 273-298.

[27] Volk, P. B., Rosenthal, F., Hahmann, M., Habich, D., and Lehner, W. (2009). *Clustering Uncertain Data with Possible Worlds*. in Proceedings of 2009 IEEE 25th International Conference on Data Engineering, pp. 1625-1632.