Advances on Block Aggregation using Hierarchical Clustering Techniques

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

Abstract

There are multiple stages in a mining operation in which a mining engineer must draw polygons to be used as operation guidelines. These polygons are drawn by hand and based on the engineer's experience and knowledge of the deposit. However, automatic procedures for forming the shapes can increase the quality and decrease the efforts required. Long-term planning requires large polygons that can be used as mining cuts. On the other hand, short-term planning requires mineable shapes to be used as mining units. These shapes need to be homogenous in grades and rock types so that the quality and dilution of material sent to the plant can be estimated with good approximation. In addition, the direction of mining can affect the desired shapes of the polygons. To address these problems, a clustering algorithm with shape control is introduced, which can provide reasonable guidelines for all the aforementioned shapes by calibrating its parameters. The implementations of the algorithm on two small datasets with 874 and 2794 blocks are illustrated. Performance of the algorithm on a real gold deposit with different mining strategies is also presented and evaluated based on homogeneity of grade, rock types, determined destinations, and run times.

1. Introduction

Clustering is a powerful tool for grouping similar objects together. However, similarity is not always the sole factor in determining the groups. One can name situations in which generated clusters have to satisfy some constraints such as mutually exclusive and inclusive objects, minimum and maximum cluster sizes, and constraints on the cluster shapes. Although it is possible to form a mathematical model for finding the optimum clustering scheme and add all sorts of constraints to the formulations, the clustering problem is proven to be an NP-Hard problem (Gonzalez, 1982; Mahajan et al., 2012). Therefore, heuristic techniques have been introduced in order to get a set of good clusters in reasonable time Hierarchical, k-means, and fuzzy c-means clustering are three of the most well-known techniques that are commonly used in solving engineering problems. These techniques can be categorised into two major groups: hierarchical clustering (agglomerative and divisive clustering) and partitional clustering (k-means, c-means, etc.). As its name implies, hierarchical clustering is performed by creating a hierarchy of clusters. On the other hand, partitional clustering is performed by partitioning data objects into a number of groups. Hierarchical clustering is known to result in better clusters rather than partitional algorithms but by taking more time (Feng et al., 2010).

Askari-Nasab, Hooman (2012), Mining Optimization Laboratory (MOL) – Report Four, © MOL, University of Alberta, Edmonton, Canada, Pages 340, ISBN: 978-1-55195-301-4, pp. 1-15.

2. Literature review

Various mine production planning mathematical models have been introduced in the literature, which rely on block grouping and aggregation. Busnach et al. (1985) try to group all the blocks on the same bench into horizontal layers to solve the long-term production planning in a phosphate mine using a zero-one mathematical model. Another early attempt can be found in Klingman and Phillips (1988) in which the authors use zero-one variables to determine whether a layer is economical to be mined and whether it should be processed or sent to the waste dump. Gershon (1983) uses the same approach but tries to label the layers as ore and waste before forming the mathematical formulation. Samanta et al. (2005) also group blocks into larger units and solve the problem using a genetic algorithm. A significant grouping approach for long-term production planning is by Ramazan (2001) who forms trees of objects using linear programs. This leads to an optimal formation of the trees and optimal solution of the planning problem. However, forming the trees for a real size block model is not practical due to the time and resources required. Zhang (2006) also groups the blocks into mining units and solves the problem using genetic algorithm. Although the author mentions that the groups are formed based on block properties, no explanation of the grouping procedure and the quality of results is presented. Askari-Nasab and Awuah-Offei (2010) use a fuzzy c-means clustering algorithm to form mining units and reduce the problem size. However, their approach does not account for similarities between the blocks for forming the mining units. Finally, Tabesh and Askari-Nasab (2011) use hierarchical clustering and Tabu search to form mining units homogenous in grade and rock type.

In addition to long-term production planning, block grouping is used to determine the dig limits and to distinguish between ore and waste in mining operations. Norrena and Deutsch (2002) develop an automatic procedure for determining the ore and waste zones while having digable shapes. Their procedure replaces the hand drawn polygons around ore and waste regions with consideration of the expected profit and mining equipment selectivity. Norrena and Deutsch (2002) first form a mathematical formulation for finding the limits and solve the problem using simulated annealing. Their approach is to use a block model with simulated grade value and separate ore and waste blocks based on cut-off grades. Afterwards, the simulated annealing procedure tries to smooth the block-wise polygons using a penalty function that represents the selectivity of the mining equipment.

3. Hierarchical clustering

Hierarchical clustering refers to a group of clustering algorithms that use similarity (or dissimilarity) measures to create a hierarchy of clusters. Two main divisions of the hierarchical clustering can be found in the literature: agglomerative and divisive clustering. The first set of algorithms start from very small clusters, i.e. consider each object a cluster and merge the most similar clusters together. The process continues until all the objects are grouped together or a stopping criterion is met. In contrast, divisive algorithms start from having all the objects in one group and dividing them into two smaller groups. Consequently, divisive algorithms stop when the number of clusters is equal to the number of objects. The common concepts between the two groups are the definition of similarity measures and the ways to update the similarity values when new clusters are defined.

Various similarity indices are used in this project based on the specific purpose defined. However, there are certain rules that are obeyed in defining the similarity index. First, all similarity indices are defined as a ratio in which the similarities measures (e.g. rock type similarity) go into the dominator and dissimilarity measures (e.g. distance) go into the denominator. The second rule is that all the similarity/dissimilarity factors have to be normalized to make sure that the effects of various parameters are comparable and weights can be applied to parameters of the same scale. The next rule is to use a penalty value between 0 and 1 for categorical measures (e.g. rock type,

destination and extraction period) and to use Minkowski distance (Pham et al., 2006) for numerical measures (e.g. distance and grade) and divide by the maximum value to scale values between 0 and 1. A more detailed definition of similarity measures can be found in Tabesh and Askari-Nasab (2011).

Similarity indices are calculated for blocks and need to be updated every time two blocks/clusters are merged to form a bigger cluster. Three main approaches for calculating the similarities between clusters exist: single link, complete link, and average link (Jain, 2010). Having all the approaches tested, the complete link approach is chosen for this project in which the similarity between two clusters is determined as the minimum similarity between all the pairs of blocks from the two clusters. The clustering pseudo code is shown in Fig 1.

```
FOR EACH bench IN dataset
    SimilarityMatrix = Calculate_Similarity_Matrix()
    AdjacencyMatrix = Create_Adjacency_Matrix()
    NumberofClusters = NumberofBlocks
    WHILE NumberofClusters > MaximumNumberofClusters
        (i,j) = Find_Most_Similar_Adjacent_Clusters()
        IF length(Clusters(i)) + length(Clusters(j)) <= MaximumClusterLength THEN
            SimilarityMatrix(i,:) = Min(SimilarityMatrix(i,:), SimilarityMatrix(j,:))
            SimilarityMatrix(j,:) = 0
            AdjacencyMatrix (i,:) = Max(AdjacencyMatrix (i,:), AdjacencyMatrix (j,:))
            AdjacencyMatrix (j,:) = 0
            Clusters(i) = Clusters(i) + Clusters(j)
            NumberofClusters = NumberofClusters – 1
        ELSE
            AdjacencyMatrix (i,j) = 0
        ENDIF
    ENDWHILE
    SmallClusters = Get_Small_Clusters(Clusters,MinimumClusterSize)
    Clusters  = Explode_Clusters(SmallClusters)
    FOR NumberofIterations
        CornerBlocks = Find_Corner_Blocks()
        Modify_Cluster_IDs(CornerBlocks)
        Clusters = Remove_Empty_Clusters(Clusters)
    NEXT
NEXT
```

Fig 1. Clustering pseudo code.

4. Implementations

Three case studies are used in this project to check the abilities of the clustering algorithm in different situations. The first data set is a gold mine with around 90,000 blocks of 50*50*15 feet. This data set is used to comment on the processing time and resources required to run the algorithm on a real-size block model. A small region of the same mine with 874 blocks is chosen for illustration purposes of different features of the clustering algorithm. Four elements are traced in the two block models: gold, silver, copper and sulphur. Rocks with high grade of gold or silver are processed in the mill and the ones with low grade of gold and high grade of copper are sent to a leach pad. The mine consists of 7 different rock types that are presented by numbers for confidentiality: 1, 6 and 7 are waste material, 2 and 3 are leach material, and 4 and 5 are mill material. Fig 2 represents the rock type and gold grade distribution over the chosen subset. The block model belongs to an ongoing mining operation and has to be extracted in 7 years based on the current long-term plan (Fig 3 (a)). Block destinations are determined based on the expected revenue of processing the block at each destination and presented in Fig 3 (b). The third dataset is a small part of an oil sands operation used to illustrate another implementation of the clustering technique. This dataset consists of 2,792 blocks which are categorized in 5 different material types numbered from 1 to 5 and shown in Fig 9.

The quality of the created clusters has to be measured and compared in different scenarios. Therefore, 4 quality measures are defined based on the objectives of forming the clusters. The first desired quality measure of clustering schemes is having a bounded size. Very large or very small clusters are not favorable if they are going to be used as planning or operational units. Thus, the size of the created clusters is considered a measure of evaluating clustering results. Formed clusters have to be homogenous in rock type, block destination, and sometimes the element's grade. Consequently, 3 quality measures are used to evaluate the homogeneity of the created clusters: rock unity, destination dilution factor, and grade variation in clusters. Rock unity is defined as the largest portion of each cluster that is from one rock type. Destination dilution factor (DDF) is the tonnage of blocks that share a destination divided by the total cluster tonnage. Coefficient of variation (CV) is a known parameter that is calculated by dividing the standard deviation of values by their mean. The average coefficient of variation of the major element of the mine is another performance measure which represents the homogeneity of grade inside clusters.

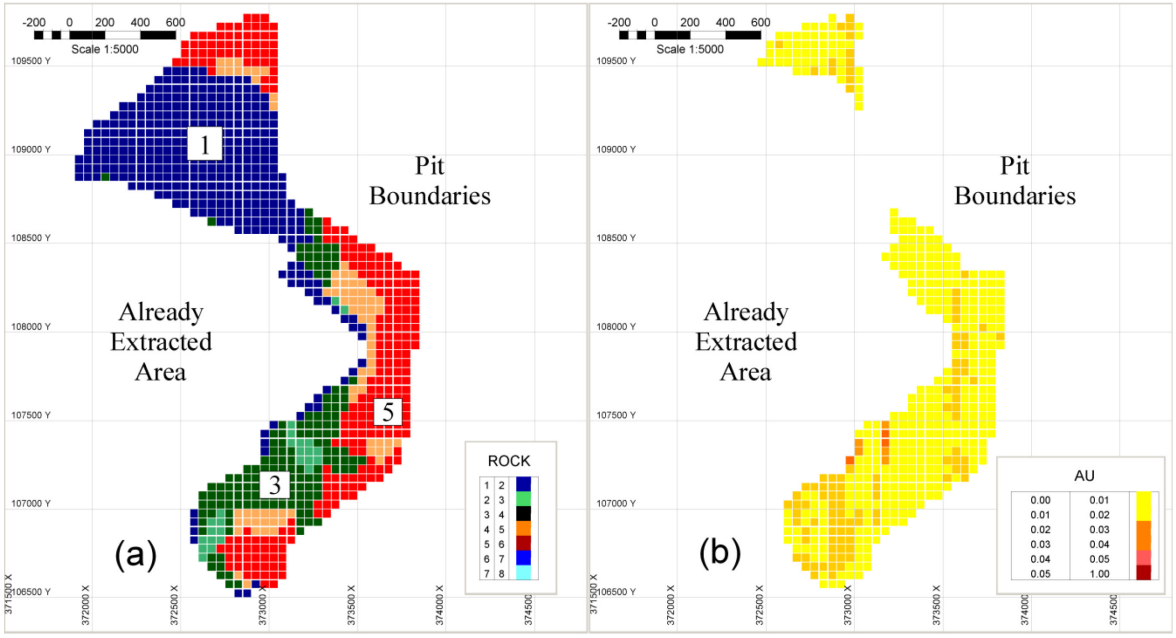

Fig 2. Rock type (a) and grade (b) distribution – units in feet.

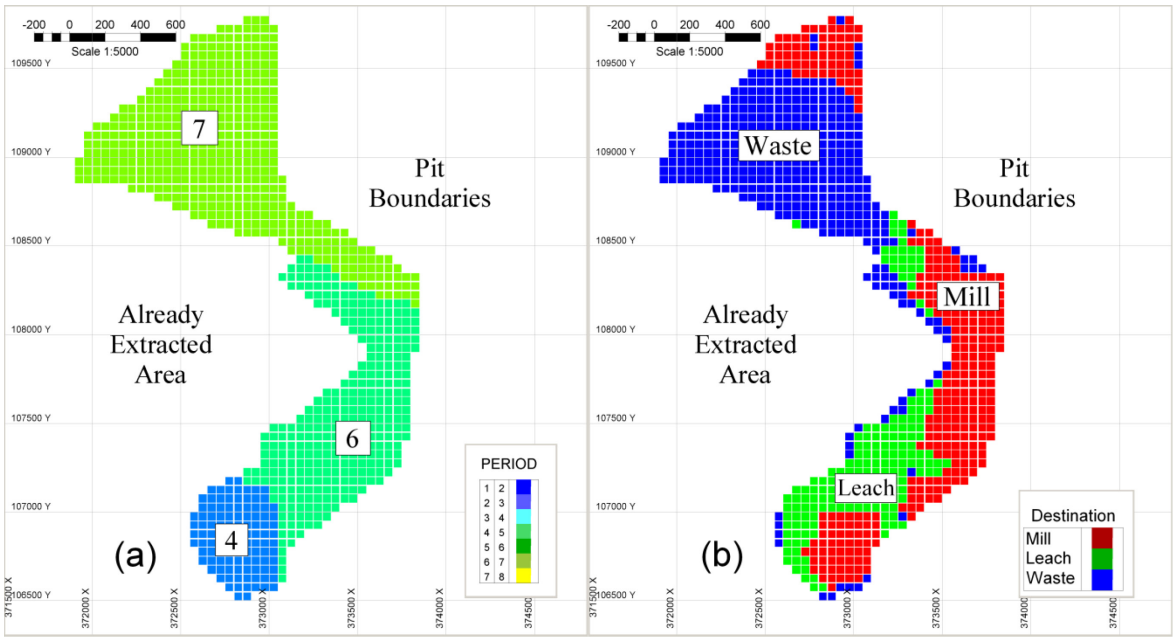

Fig 3. Extraction period (a) and destination (b) - units in feet.

4.1. Size control

Agglomerative hierarchical clustering is performed in a bottom-up fashion which provides an easy way of controlling the maximum cluster size. The size of the new cluster (tonnage or number of blocks) is checked against the specified maximum size every time two clusters are chosen to be merged together. The merging operation is terminated if merging the two clusters violates the maximum acceptable size and the next two most similar clusters are chosen. Another control mechanism is to calculate the total number of clusters by dividing the total number of blocks by the average number of blocks in each cluster and terminate the clustering procedure when the number of clusters is equal to the desired number of clusters. Minimum cluster size is also applicable in the post process step by removing clusters smaller than a specified threshold. The idea is borrowed from Barca and Rumantir (2007) where the authors remove small clusters in a k-means implementation. However, enforcing minimum cluster size may breach the maximum size constraint since it is implemented at a later stage.

4.2. Post process

A post process stage is considered to refine the shape of the created clusters. This stage is responsible for two changes: removing sharp corners and removing small clusters. The process is described using a sample clustering scheme as in Fig 4. First, clusters smaller than the specified minimum size, are disaggregated into single-block clusters and the shape refinement begins. This is an iterative procedure in which blocks with less than one adjacent block from the same cluster and more than one adjacent block from another cluster are detached from the original cluster and attached to the one with more neighbors. Assume that Fig 4 (a) is the result of the clustering and a minimum of 5 blocks per cluster is required.

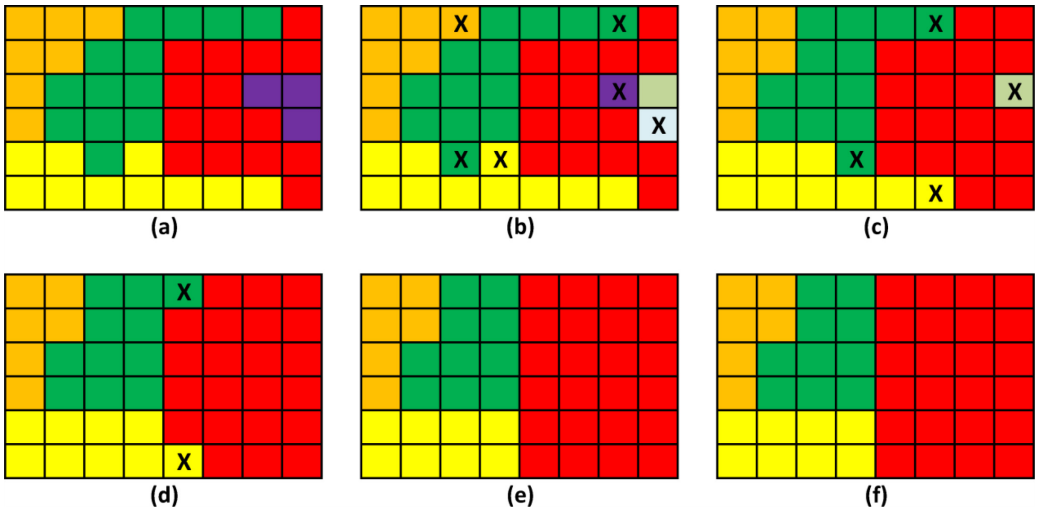

Fig 4. Shape refinement stage.

The purple cluster in Fig 4 (a) has only 3 blocks and has to disaggregate into 3 clusters in the first iteration. Afterwards, blocks marked with "X" in Fig 4 (b) are qualified as corner blocks and have to be detached from their original clusters. Three iterations will modify the clustering scheme from Fig 4 (a) to Fig 4 (e) where no more corner blocks can be identified.

Running the algorithm on the single bench block model, with only distance as the similarity index, results in a clustering scheme as shown in Fig 5 (a). The cluster shapes can be smoothed by 3 iterations of shape refinement as in Fig 5 (b). Average cluster size, maximum number of blocks in a cluster and minimum number of blocks in a cluster are set to 30, 35 and 10 respectively. The similarity index for the distance only clustering is calculated as in equation (1) where $\tilde{D}_{ij}$ is the normalized distance between blocks i and j.

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

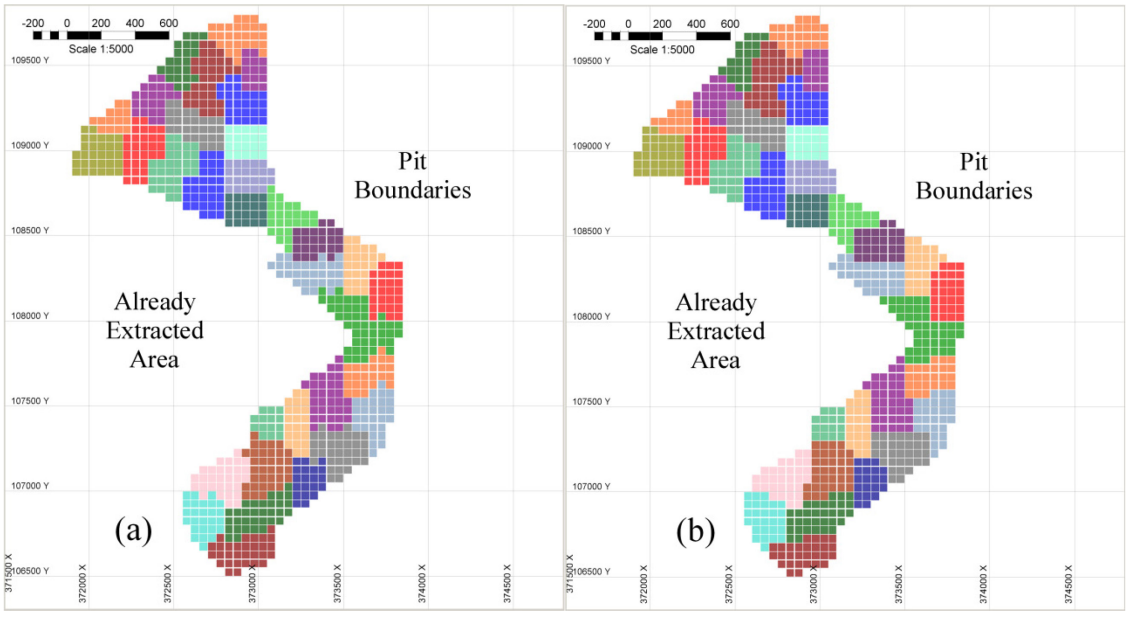

Fig 5. Clustering with distance as similarity measure - units in feet.

Table 1. Clustering based on distance.

Description	Refinement Iterations	Rock Unity	DDF	Au Grade CV
Distance Only	0	78%	87%	35%
Distance Only (Refined Shape)	3	77%	87%	35%

Clustering with only distance as the similarity measure does not seem to be useful. Clusters that are uniform in their grade, rock type or block destination can be more reasonable to form. However, if these clusters are to be used later as extraction units they have to have minable shapes. Therefore, clusters with grade and destination similarities are created and compared in Fig 6 and Fig 7. Fig 6 (a) shows the result of clustering with grade as the sole similarity measure. Shapes of the created clusters are irregular and with sharp corners but follow the changes in grade values as compared against Fig 6 (b). Simultaneous consideration of distance and grade as similarity measures can solve the shape problem and create minable clusters with a decrease in grade homogeneity (Fig 7 (a)). However, the element grade is not always the best way of grouping the blocks since they are going to be processed based on cut-offs; i.e. two blocks with grades right above and below the cut-off grade have similar grades but one is going to be processed and the other is considered as waste. Therefore, the predetermined destination of the block is used in Fig 7 (b) as the similarity measure to form clusters. The similarity index is calculated using equation (2) where T_{ij} is the penalty applied if blocks i and j are not sent to the same destination and $\tilde{G}_{ij}$ is the normalized difference between element grades of blocks i and j. The quality measures of the created clusters and parameters used are summarized in Table 2.

As shown in Table 2, the lowest grade variation can be achieved by considering the grade as the sole similarity measure. However, refining the cluster shapes causes a significant increase in the grade variation. On the other hand, clustering based on destination and distance, results in smoother cluster shapes and reasonable grade, destination and rock type variation.

$$S_{ij} = \frac{T_{ij}}{\tilde{D}_{ij}^{W_D} \times \tilde{G}_{ij}^{W_G}} \qquad (2)$$

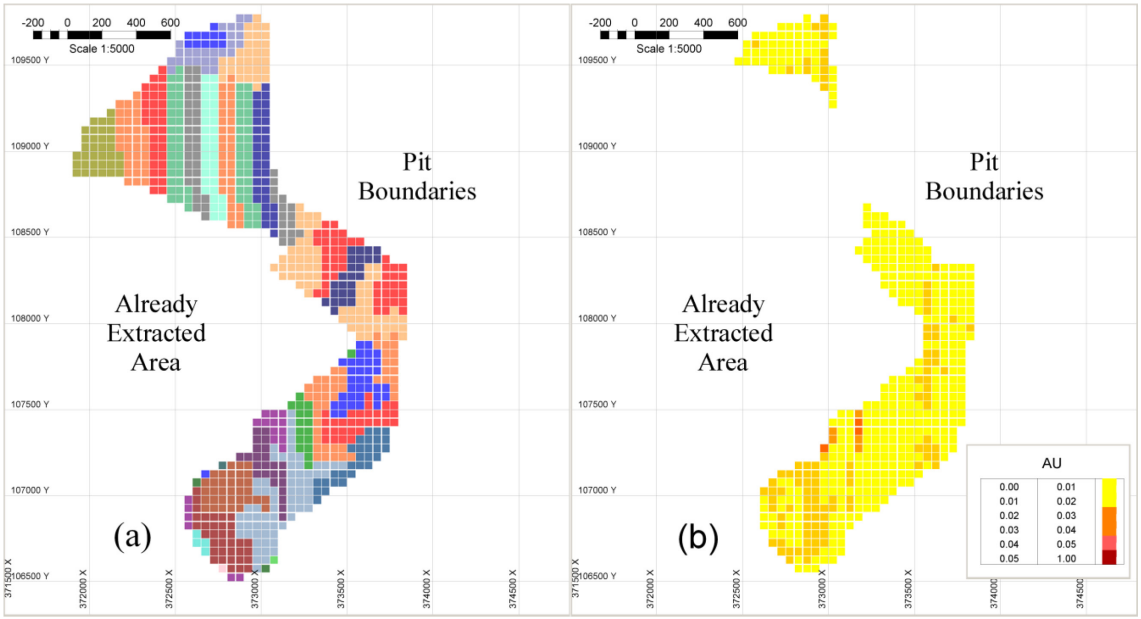

Fig 6. Clustering with grade as similarity measure - units in feet.

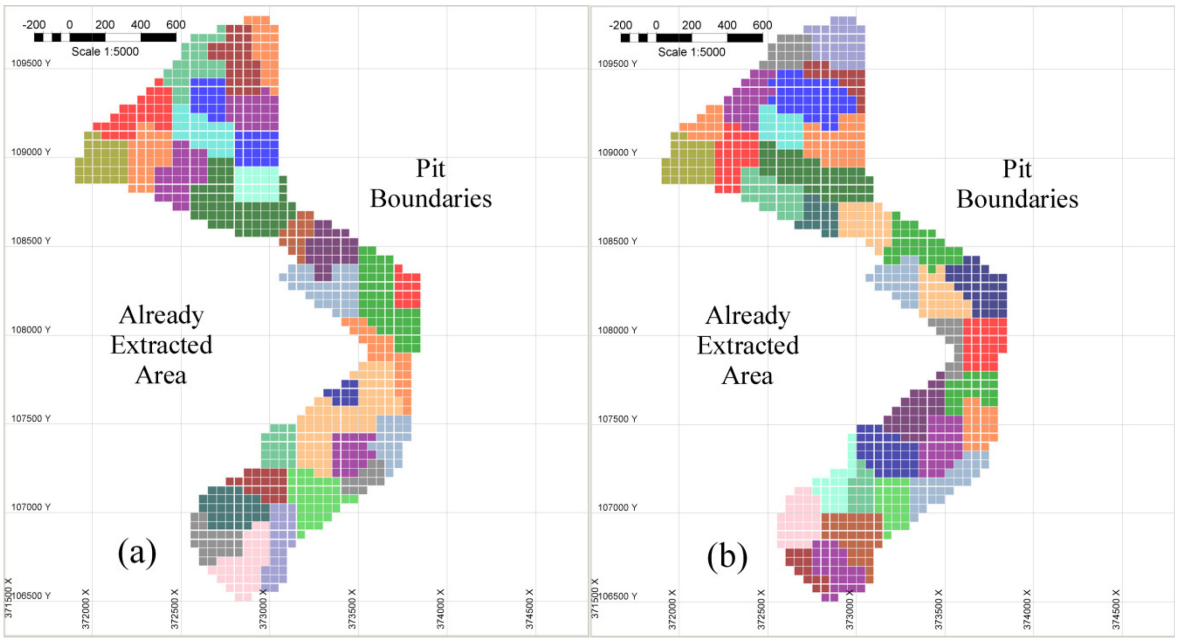

Fig 7. Clustering with grade and shape (a) and destination (b) as similarity measure

Table 2. Clustering based on distance, grade and destination – units in feet.

Description	Distance Weight	Grade Weight	Destination Penalty	Rock Unity	DDF	Au Grade CV
Grade Control	0	1	0	85%	94%	24%
Grade Control (Refined Shape)	0	1	0	81%	92%	31%
Grade & Shape Control	1	1	0	79%	89%	33%
Destination & Shape Control	1	0	0.9	87%	94%	28%

4.3. Directional mining units

Clustering can also be used to create mining units for short term planning, ore control and blasting. However, the shapes of the clusters have to be consistent with the direction of mining or their usage will be limited and the number of drop cuts will increase. A similarity measure based on the mining direction is defined in this project to account for mining direction. Two coordinates have to be determined by the user as the start and end points of the mining direction vector. The distance of each block from these points is calculated and this value is then used as a similarity measure. This can result in clusters perpendicular to the mining direction. Equation (3) is used to calculate the mining direction similarity index. Assume that M_i^1 refers to the Euclidean distance from block i to the starting point of the direction vector and M_i^2 refers to its distance to the end point of the direction vector. Note that the sign has to be preserved in order to distinguish between the start and end point of the direction vector. The mining direction index (M_i) can now be used to determine the similarity between blocks i and j and can be used in combination with other similarity measures such as distance, grade, or destination with their specified weights and penalties. Note that considering too many parameters decreases the quality of clustering since their effects can neutralize each other, e.g. it is recommended to use either grade or destination factors since they are usually correlated. Two examples of the directional mining based on mining direction, distance and destination are presented in Fig 8. The clusters are set to have 20, 50 and 55 blocks in minimum, average and maximum respectively.

$$M_i = sign\left(\left(M_i^1\right)^2 - \left(M_i^2\right)^2\right) \times \sqrt{\left|\left(M_i^1\right)^2 - \left(M_i^2\right)^2\right|} \tag{3}$$

$$S_{ij} = \frac{T_{ij}}{\tilde{D}_{ij}^{W_D} \times \tilde{G}_{ij}^{W_G} \times \tilde{M}_{ij}^{W_M}} \tag{4}$$

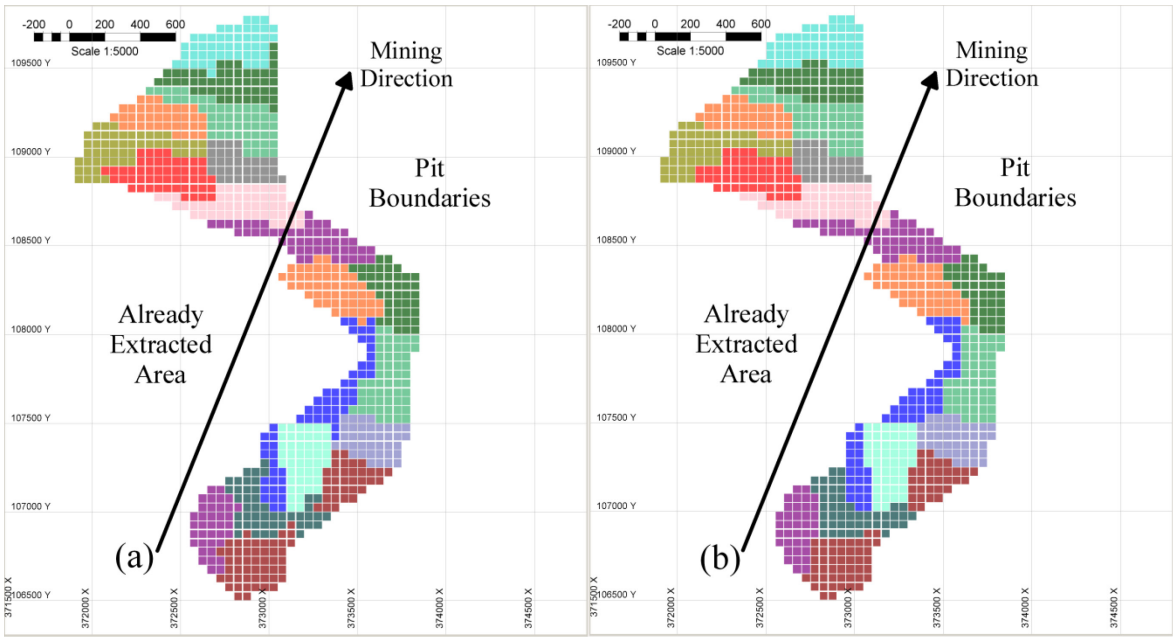

Fig 8. Clustering with direction before (a) and after (b) Shape refinement – units in feet.

Table 3. Clustering with direction.

Description	Rock Unity	DDF	Au Grade CV	Cluster Size	
				Min	Max
Directional Mining Units	81%	90%	38%	21	53
Directional Mining Units (Refined Shape)	80%	89%	39%	23	52

Another modification to the clustering algorithm makes it useful for stratified deposits such as oil sands and coal. The mining units used in this group of operations have their own shape specifications that are different from metal mining. Strips of single typed material with appropriate width and smooth corners are required to do short-term and operational planning. The only change required to get the strips is to replace the definition of the mining direction in equation (3) by equation (5) and continuing with the clustering procedure as before. Instead of specifying a direction vector the two user-specified points represent the starting point of the mining operation.

$$M_i = \sqrt{\left|\left(M_i^1\right)^2 + \left(M_i^2\right)^2\right|} \tag{5}$$

The performance of the clustering algorithm for stratified deposits units is tested on a small oil sands block model with 2794 blocks in one bench. The clustering is performed based on the reference points shown in Fig 10 and with material type and distance considered as the similarity indices. Cluster size average, minimum and maximum are set to be 100, 120 and 50 respectively. The quality measures of the clustering algorithm can be found in Table 4.

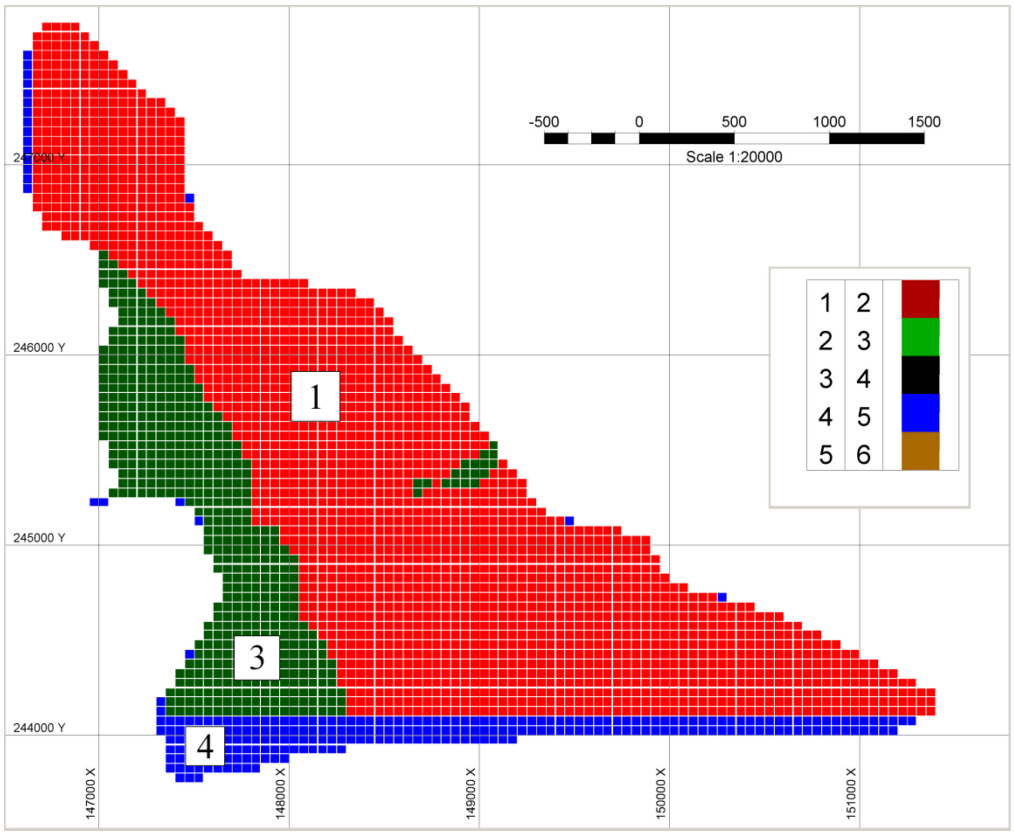

Fig 9. Material type distribution in oil sands dataset – units in meters.

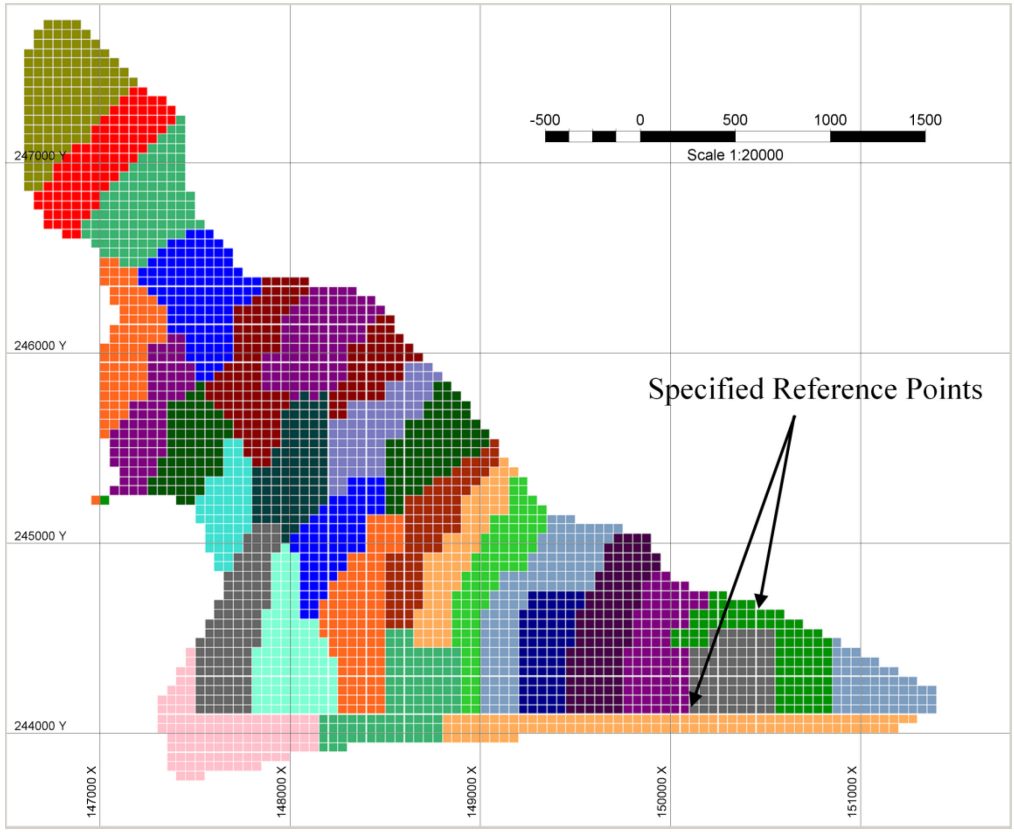

Fig 10. Stratified deposit clustering – units in meters.

Table 4. Stratified deposits clustering summary.

Rock Unity		Material Dilution		Cluster Size		Cluster Tonnage (ton)		CPU Time (s)
Avg	Std	Avg	Std	Avg	Std	Avg	Std	
92%	10%	97%	8%	85	28	6,352,450	2,436,927	52

4.4. Clustering within boundaries

It is sometimes required to create the clusters within some boundaries. A mine planner may need short-term mining units within the long-term schedule, mine push backs or mining zones. This can be achieved by manipulating the adjacency matrix. The zero-one matrix representing block adjacencies can be multiplied by another zero-one matrix indicating whether the two blocks are from the same region or not. Not being considered neighbors, the two blocks cannot merge to form a cluster in any further steps. This can be added to any of the clustering configurations described earlier. The same configuration used for creating directional mining units is used to create the clusters within the long-term production schedule and shown in Fig 11. Obviously, the quality of the clusters will decrease because of the added constraints. The shape refinement procedure is also bounded to the long-term schedule.

Another implementation of the bounded clustering would be to have clusters with absolute distinction between ore and waste. However, this can lead to very small and tortuous clusters, based on the nature of the deposit (Fig 12 (a)). On the other hand, the shape refinement procedure can be permitted to ignore the strict boundaries and manipulate clustering to get better shaped clusters (Fig 12 (b)). Note that the resulted clusters may have higher dilution and variation in rock type and grade than the original constrained clusters. The quality measures of the clustering schemes are compared in Table 5.

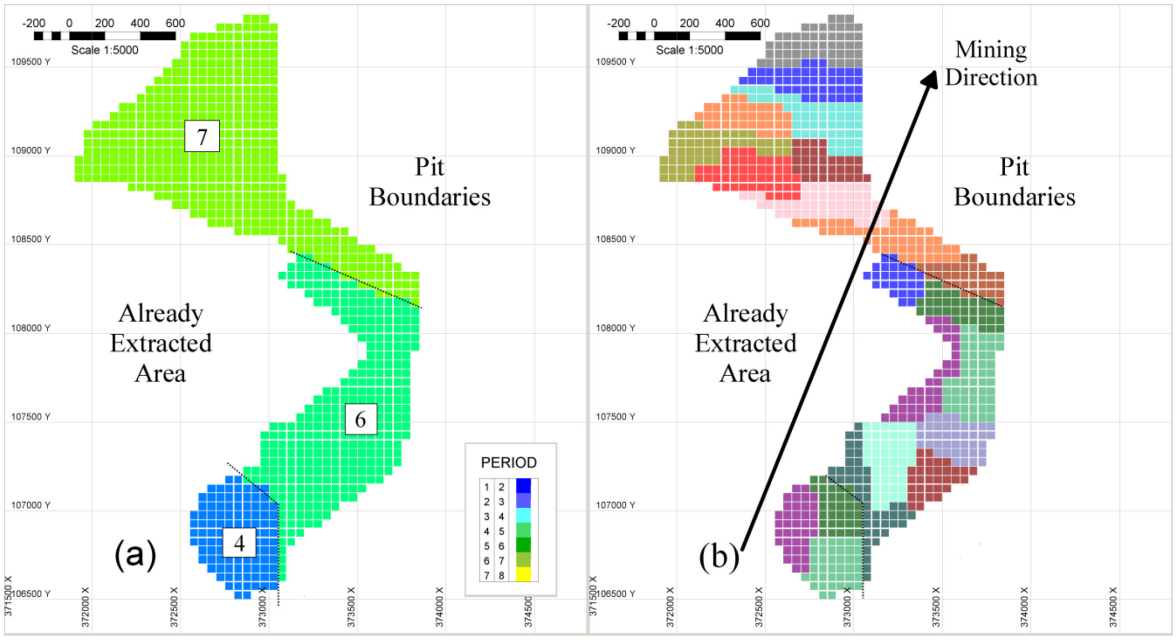

Fig 11. Directional clustering withing long-term schedule – units in feet.

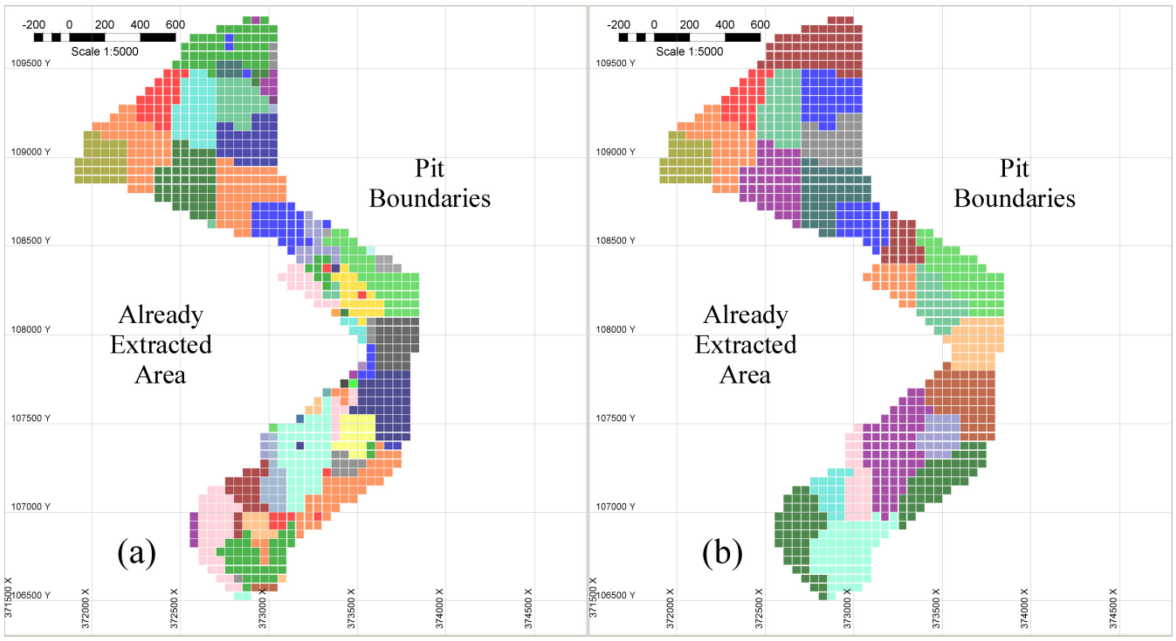

Fig 12. Clustering within destinations before (a) and after (b) shape refinement – units in feet.

Table 5.Clustering within boundaries

Description	Rock Unity	DDF	Au Grade CV	Number of Clusters	Cluster Size	
					Min	Max
Directional Mining Units	80%	89%	39%	20	23	52
Directional Mining Units (Within Long-Term Schedule)	76%	87%	38%	21	26	52
Within Destination	99%	100%	14%	76	1	50
Within Destination (Unconstrained Refined Shape)	85%	93%	33%	23	19	68

4.5. Processing times

The processing times for all the aforementioned implementations, are around 2 seconds on the small dataset with 874 blocks. However, a real-size block model consists of hundreds of thousands of blocks and the processing time is an important issue. Therefore, the algorithm is run on a real block model with 91,448 blocks and the processing times are summarized in Table 6 and Table 7. The machine used for the performance measures has 8 processing cores at 2.4 GHz speed and 24 GB of RAM. However, the code does not meet parallel processing requirements and uses one core only. Forming the similarity matrices is the stage with the most memory consumption and uses 8 GB of RAM at its peak. The processing times for all instances are around 6 minutes.

Table 6 .Cluster size and CPU time.

#	Description	Cluster Size		Cluster Tonnage (tonne)		CPU Time (s)
		Avg	Std	Avg	Std	
1	Distance Only	23	7	81,556	30,338	343
2	Distance Only (Refined Shape)	24	7	83,090	29,238	362
3	Grade Control	16	14	55,067	51,923	363
4	Grade Control (Refined Shape)	28	10	97,238	37,719	383
5	Grade & Shape Control	26	9	90,027	36,075	371
6	Destination & Shape Control	26	8	89,596	32,290	391
7	Within Destination (Refined Shape)	25	25	86,572	90,262	357

Table 7. Homogeneity measures.

#	Description	Rock Unity		DDF		Au Grade CV	
		Avg	Std	Avg	Std	Avg	Std
1	Distance Only	88%	17%	87%	16%	43%	33%
2	Distance Only (Refined Shape)	88%	17%	87%	16%	44%	32%
3	Grade Control	93%	15%	93%	13%	19%	27%
4	Grade Control (Refined Shape)	88%	17%	88%	15%	38%	32%
5	Grade & Shape Control	88%	17%	88%	16%	38%	61%
6	Destination & Shape Control	90%	15%	93%	12%	41%	34%
7	Within Destination (Refined Shape)	94%	13%	95%	9%	25%	34%

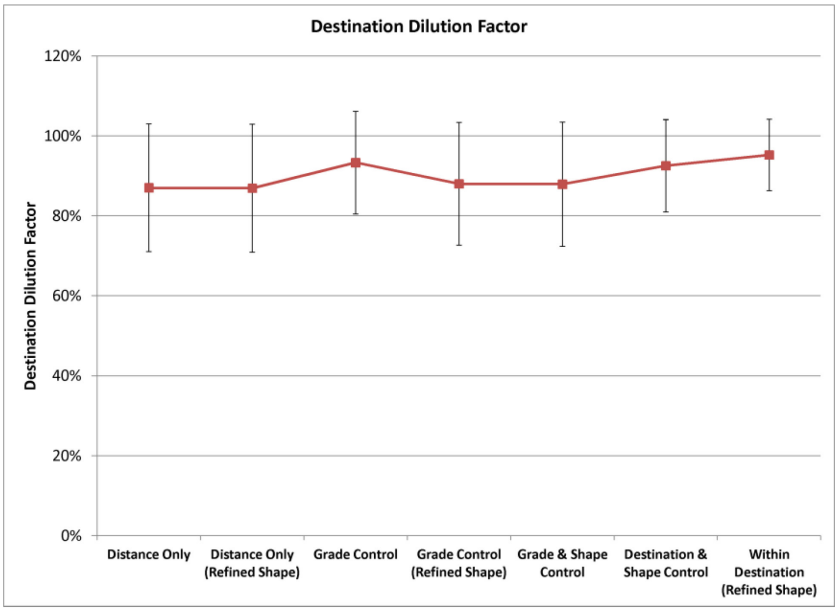

Fig 13. Destination dilution factor for various weight and penalty configurations.

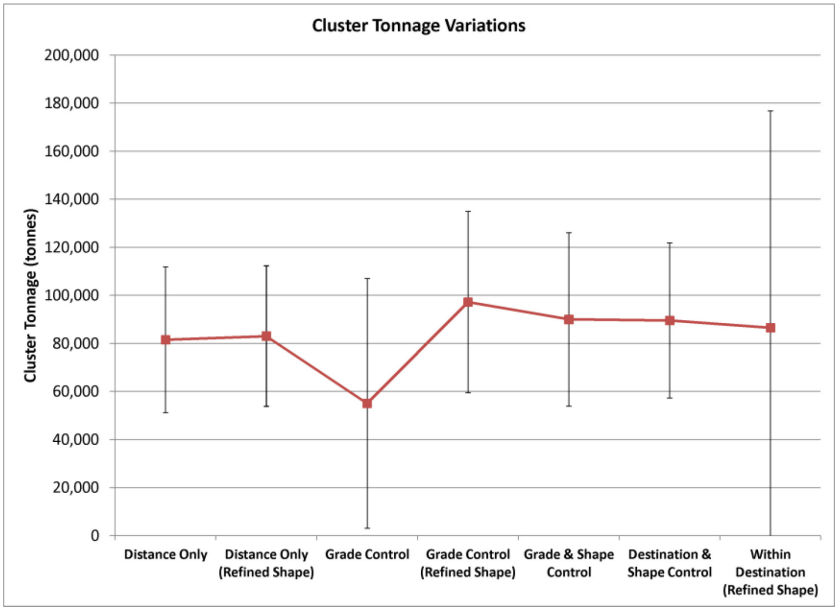

Fig 14. Cluster tonnage variation for various weight and penalty configurations.

Comparing all the implementations it seems that each scenario has its own advantages and shortcomings. A mine engineer has to select the weight and penalty combination which suits his goal of using a clustering algorithm. Scenarios 3 and 4 have the lowest grade variation and highest rock and destination homogeneity but significant variations in cluster size and tonnage. On the other hand, the first 2 scenarios have the lowest variation in cluster size and highest variations in grade, rock type and destination. The destination and shape control scenario seems to be able to balance required qualifications. However, a mine engineer may be interested in any configuration based on the objective of clustering.

5. Conclusions

Clustering has been widely used as a tool of grouping objects in various science and engineering areas. It has also been used in mining engineering to reduce the size of production scheduling mathematical formulations and to get more practical results. However, the shape of the generated clusters was not in control and could decrease the practicality of their usage. A clustering procedure with shape control is proposed in this paper which can create clusters of blocks with smoothed edges and based on the direction of mining. The minimum and maximum size of the clusters can also be controlled. The performance of the proposed algorithm with various configurations is tested on 3 datasets with 874, 2,794 and 91,448 blocks. Running various configurations of the algorithm on the large dataset takes around 6 minutes, and rock type and destination homogeneities of around 90% are achieved in different scenarios. More research is underway by the authors to define quantitative measures to evaluate the shape of the generated clusters.

6. References

[1] Askari-Nasab, H. and Awuah-Offei, K. (2010). A Clustering Algorithm for Open Pit Mine Production Scheduling, in Proceedings of Society for Mining, Metallurgy & Exploration (SME) Annual Meeting Preprint, Phoenix, Arizona, USA.

[2] Barca, J. C. and Rumantir, G. (2007). A Modified K-means Algorithm for Noise Reduction in Optical Motion Capture Data, in Proceedings of Computer and Information Science, 2007. ICIS 2007. 6th IEEE/ACIS International Conference on, Melbourne, pp. 118-122.

[3] Busnach, E., Mehrez, A., and Sinuany-Stern, Z. (1985). A Production Problem in Phosphate Mining. *Operational Research Society*, *36* (4), 285-288.

[4] 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.

[5] Gershon, M. E. (1983). Optimal mine production scheduling: evaluation of large scale mathematical programming approaches. *International Journal of Mining Engineering*, *1* 315-329.

[6] Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters, Award winning papers from the 19th International Conference on Pattern Recognition (ICPR), 19th International Conference in Pattern Recognition (ICPR)*, *31* (8), 651-666.

[7] Klingman, D. and Phillips, N. (1988). Integer Programming for Optimal Phosphate-Mining Strategies. *The Journal of the Operational Research Society*, *39* (9), 805-810.

[8] Norrena, K. P. and Deutsch, C. V. (2002). Optimal Determination Of Dig Limits For Improved Grade Control, in Proceedings of APCOM 2002 - Application Of Computers And Operations Research In The Minerals Industry, Littleton, Colorado, pp. 11.

[9] Pham, D. T., Prostov, Y. I., and Suarez-Alvarez, M. M. (2006). Statistical approach to numerical databases: clustering using normalised Minkowski metrics. in *Intelligent Production Machines and Systems*, D. T. Pham, E. E. E. E.E. Eldukhri and A.J. SorokaA2 - D.T. Pham, and A. J. Soroka, Eds., Oxford, Elsevier Science Ltd, pp. 356-361.

[10] Ramazan, S. (2001). Open pit mine scheduling based on fundamental tree algorithm.Thesis,

[11] Samanta, B., Bhattacherjee, A., and Ganguli, R. (2005). A genetic algorithms approach for grade control planning in a bauxite deposit. in *Application of Computers and Operations Research in the Mineral Industry*, null SV - null DO - doi:10.1201/9781439833407.ch44 ed, Taylor & Francis, pp. 337-342.

[12] Tabesh, M. and Askari-Nasab, H. (2011). A Two Stage Clustering Algorithm for Block Aggregation in Open Pit Mines. *Transactions of the Institution of Mining and Metallurgy. Section A, Mining industry*,

[13] Zhang, M. (2006). Combining genetic algorithms and topological sort to optimise open-pit mine plans, in Proceedings of 15th International Symposium on Mine Planning and Equipment Selection (MPES), FIORDO S.r.l., Torino, Italy, pp. 1234-1239.