A Constraint Programming approach to solve the clustering problem in open-pit mine planning

http://dx.doi.org/10.1590/0370-44672023770060

Jorge Luiz Valença Mariz[1,3]
https://orcid.org/0000-0002-4047-7642
Rodrigo de Lemos Peroni[1,4]
https://orcid.org/0000-0003-0131-5832
Ricardo Martins de Abreu Silva[2,5]
https://orcid.org/0000-0003-3117-6191

[1]Universidade Federal do Rio Grande do Sul – UFRGS, Departamento de Engenharia de Minas, Porto Alegre – Rio Grande do Sul - Brasil.

[2]Universidade Federal de Pernambuco - UFPE, Centro de Informática, Departameto de Informação e Sistemas, Recife – Pernambuco - Brasil.

E-mails: [3]jorge_valenca@hotmail.com, [4]peroni@ufrgs.br, [5]rmas@cin.ufpe.br

Abstract

Since the open-pit precedence-constrained production scheduling problem is an NP-hard problem, solving it is always a challenging task, especially from a long-term perspective because a mineral deposit containing millions of blocks would require several million precedence arcs as constraints, making the solution time grow exponentially and making a direct approach unfeasible. Therefore, different strategies have been employed since the 1960s to reduce the size of this problem, such as determining the ultimate pit limit, subdividing it into phases, segmenting the production scheduling problem into long-, mid-, and short-term plans, as well as aggregating blocks into clusters, thus significantly reducing the number of precedence arcs. Different modeling and clustering strategies have already been employed in an attempt to reduce the size of the mine sequencing problem, such as layer modeling, re-blocking, bench-phase clustering, or polygon (mining cut) clustering based on a similarity function. The mining cut clustering problem has been solved lately by machine learning and heuristics techniques, and this approach can also introduce operational constraints to the mine sequencing problem, such as equipment size, minimum pit width, and preferential mining direction. In this study, we propose a mining cut clustering model based on Mixed Integer Linear Programming (MILP). Then we solve it by an exact approach and by Constraint Programming (CP), analyzing the strengths and weaknesses of the Constraint Optimization Problem (COP) and Constraint Satisfaction Problem (CSP) techniques. Numerical experiments were carried out on the Bench 15 of the Newman1 dataset, demonstrating the superiority of the COP approach.

Keywords: mine planning, mining cut clustering, mixed integer linear programming, constraint programming.

1. Introduction

The representation of mineral deposits in block models proposed by Axelson (1964) revolutionized the way of approaching and studying mining operations in these deposits, mainly regarding the determination of mine sequencing. However, even if mathematical models capable of representing this problem were already available, such as the Linear Programming (LP) models of Manula (1965) and Johnson (1969), the limited computational capacity at that time was prohibitive even for solving small instances. The open-pit production scheduling problem can be seen as a precedence constrained knapsack problem, which is an NP-hard problem, signifying that there are no known algorithms capable of solving it exactly in polynomial time. Thus, depending on the number of constraints imposed, even solving small scale problems is still a challenge today, even though there are world-class deposits containing millions of blocks, which result in several millions of precedence arcs, making the monolithic approach to the problem impractical. Therefore, great effort was made to formulate subproblems of the main problem of mine sequencing, resulting in the subdivision of the monolithic problem into three distinct time horizons: long-, medium-, and short-term production scheduling problems.

Lerchs & Grossmann (1965) proposed a Dynamic Programming (DP) approach to solve exactly the ultimate pit limit (UPL) problem for 2D instances, which would make it possible to eliminate the blocks whose extraction would not be economically feasible from the mine sequencing problem. They also formulated the UPL problem as a maximum closure of a graph problem to solve exactly 3D instances, a solution that would wait for two decades until its full implementation (Whittle, 1988). Lerchs & Grossmann (1965) also proposed a parametric algorithm to solve the mine sequencing problem, an approach that is still widely used today in order to generate nested pits and phases, these being considered as a reference for the transition from strategic planning to tactical and operational plans. Nevertheless, even after employing these problem reduction strategies, the production scheduling can be very difficult to be solved, and an additional issue related to the convergence between the plans in the different time horizons should also be considered, since the operational plan requires several constraints that were ignored during the strategic plan determination, such as minimum pit width, positioning of ramps, equipment size, and preferred mining direction.

A strategy that was widely used in horizontally mineral deposits like coal, phosphate, and bauxite mines was to model these deposits as 2D seams, layers, or strata instead of using a regular block model, thus significantly reducing the size of the mine sequencing problem. Albach (1967) proposed a Chance-Constrained Programming (CCP) to assess the uncertainty in coal seam boundaries, while Metz and Jain (1978) proposed a DP-based model to solve the sequencing of phosphate layers, in addition to proposing a Mixed Integer Non-Linear Programming (MINLP) model and a heuristic to solve it. Klingman & Phillips (1988) proposed an Integer Programming (IP) model based on panels that considered the thickness of the phosphate layers and the size of a dragline, using a branch-and-bound algorithm for solving it, while Samanta *et al.* (2005) employed a genetic algorithm (GA) to determine the operational plan of a layered bauxite deposit. Although this type of model is efficient in simplifying the mine sequencing problem, it cannot be used in deposits that do not present this characteristic of great continuity in sub-horizontal directions. Another widely used technique, available in commercial mine planning software, is to group the blocks belonging to the same bench and the same phase into bench-phase units, determining the geometrical precedence and then sequencing these larger units, instead of operating on the individual blocks, from which we can quote the contributions of Menabde *et al.* (2004), Whittle (2004), Epstein *et al.* (2012), and Rezakhah *et al.* (2020).

Although there are re-blocking propositions (Chanda & Ricciardone, 2002; Jélvez *et al*., 2016), techniques that incorporate slope angle to clusters (Ramazan, 2007; Mai *et al*., 2018), and even temporal aggregation propositions (Newman & Kuchta, 2007) instead of aggregating blocks, the most applied methodology today to reduce the size of the open-pit mine sequencing problem is the mining cut clustering. Through this technique, the blocks belonging to a same bench are grouped into polygons based on a predetermined similarity function, and by seeing this process as an optimization problem apart from the production scheduling problem, it is possible to incorporate several constraints to this clustering problem, such as minimum and maximum size of the clusters, shape control, minimum width, mining direction, among others. Another challenge, however, is that the clustering problem is also NP-hard, such that the solution of larger instances is challenging for the algorithms, which is why so far, the majority of published articles have used heuristics and techniques based on machine learning in the solution of the mining cut clustering problem. Askari-Nasab *et al.* (2010a) proposed an approach based on Fuzzy Logic Clustering (FLC) to generate mining cuts from an iron ore deposit and input them into a production scheduling model based on Mixed Integer Linear Programming (MILP), an approach that was also implemented by Askari-Nasab *et al.* (2011), Ben-Awuah & Askari-Nasab (2011), and Ben-Awuah *et al.* (2012). Eivazy & Askari-Nasab (2012a), in turn, proposed a Fuzzy C-means (FCM) approach to generate mining cuts and input them into MILP-based mine planning models, technique also used by Koushavand *et al.* (2014). Askari-Nasab *et al.* (2010b) developed a two-stage mining cut clustering algorithm based on a hierarchical clustering (HC) step, which reduces the precedence arcs, and a refining step based on Tabu Search (TS), a technique that was enhanced by Tabesh & Askari-Nasab (2011, 2013) and has been widely used since then, i.e., Eivazy & Askari-Nasab (2012b), Badiozamani & Askari-Nasab (2014), and Upadhyay & Askari-Nasab (2016).

Askari-Nasab *et al.* (2013) and Tabesh *et al.* (2013, 2014, 2015) combined the bench-phase stage with the HC-TS approach to generate mining cuts with each bench-phase and input them into a MILP-based production scheduling model. Tabesh & Askari-Nasab (2019) introduced a K-means stage to the HC approach and developed a stochastic mining cut clustering workflow, which independently generates clusters for each equally probable realization of a deposit and then applies an aggregation technique to extract a final clustering model from all realizations. Salman *et al.* (2021) developed a multi-stage heuristic to generate mining cuts that uses K-means, fuzzy membership functions, post-processing and a geometry correction step. Nelis & Morales (2022) developed a MILP-based model that simultaneously solves the mining cut and production scheduling problems based on blasthole sampling and selective mining units (SMUs), so that the model elects a list of SMUs to be the representative of the clusters, attach the other SMUs to those clusters and then refine the result before the mine sequencing stage. Nelis *et al.* (2022) proposed a MILP-based mining cut clustering model and employed a column generation algorithm and Linear Programming relaxation to solve the problem, followed by an additional step that reaches an integer solution from the linear solution found in the previous steps. Figure 1 shows a diagram based on Nelis & Morales (2022) that explains the transition from strategic to operational mine planning, in which SMUs are generated from blasthole sampling and aggregated in mining cuts before the mine sequencing step, where the different colors represent the destinations for the mining cuts.

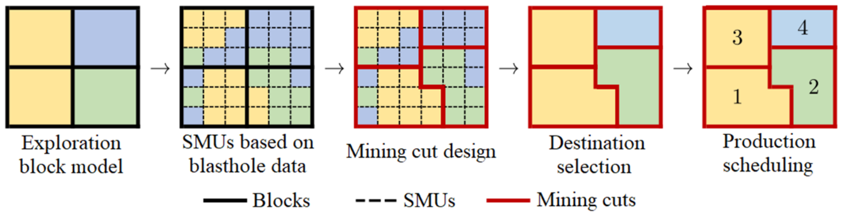

Figure 1 - Scheme showing the transition from strategic horizon, based on exploratory blocks, to operational mine planning (Modified from Nelis & Morales, 2022).

This study presents a MILP-based model for the mining cut clustering problem, in addition to a Constraint Programming (CP) approach to solve the proposed model, where the mining cut clustering problem, the similarity index employed, and the MILP-based mathematical model are defined. The proposed methodology is applied to the Newman1's Bench 15 (Espinoza *et al*., 2013), comparing techniques and discussing the results. However, CP is not exactly an algorithm, but a paradigm that integrates techniques from artificial intelligence, computer science, operations research and programming languages, including constraint propagation (Kleer & Sussman, 1978), branch-and-bound, and chronological backtracking (Golomb & Baumert, 1965). In this context, considering that a constraint represents a Cartesian product of the domains of the variables of a problem, a Constraint Satisfaction Problem (CSP) approach (Mackworth, 1977) can determine the finite sets of variables that satisfies all constraints simultaneously. Thus, a filtering step reduces the domain of each variable so that it still contains all the values that can be assumed to satisfy the constraints. The Constraint Optimization Problem (COP) approach, in turn, introduces the minimization (or maximization) of an objective function to this CSP workflow, resulting in an optimal solution for the variables without violating the constraints. CP has already been used by other authors in optimization problems, as in Khiari *et al*. (2010), Guns *et al*. (2013, 2016), and Dao *et al*. (2013, 2016, 2017), but never in solving the mining cut clustering problem.

2. Material and method

As aforementioned, the mining cut clustering problem is a block aggregation technique capable of reducing precedence arcs and adding operational constraints to the production scheduling problem model, requiring a similarity index to determine which features are considered and how the blocks will interact to generate these clusters. Bagirov *et al*. (2020) presented some variants of Minkowski norms that are often used in clustering problems, such as those based on the L1-norm (Manhattan of city block norm), on the L2-norm (this considers the Euclidean distance), or on the L∞-norm (Chebyshev norm). However, Tabesh & Askari-Nasab (2011) claim that the requirement of coding the variables of the mining cuts clustering problem as binary variables would prevent Minkowski norms from considering the similarities between the categories, as well as using a simple matching coefficient approach would fail, so they proposed a hierarchy of distances idea to calculate the similarities between variables.

Considering I as the domain of blocks, j as a partition of I for $j = 1,\ldots,n, \forall j \in J$, they proposed a similarity index between two blocks $b,c \in j$ as $S_{b,c} = R_{b,c} \times C_{b,c} / \tilde{D}_{b,c}^{\,W_D} \times \tilde{G}_{b,c}^{\,W_G}$. The $R_{b,c}$ and $C_{b,c}$ parameters are the penalties attributed to blocks with different lithologies and not located above the same cluster, respectively, considering an optimization in multiple benches. Therefore, if two blocks belong to the same lithology, their $R_{b,c}$ value will be greater than two blocks from different lithologies. Likewise, the value of $C_{b,c}$ will be higher if both blocks $b,c \in j$ are above blocks belonging to the same cluster, aiming to increase the similarity function $S_{b,c}$ in these specific circumstances. Conversely, $\tilde{D}_{b,c}$ and $\tilde{G}_{b,c}$ are the normalized distance and the normalized difference of grades between blocks b and c, respectively. Consequently, the greater these distances between two blocks $b,c \in j$, the more the similarity function $S_{b,c}$ will be penalized. W_D and W_G are calibration weights which, like penalties, can be assigned values between 0 and 1. Tabesh & Askari-Nasab (2013) formulated variants to this similarity function, including $S_{b,c} = T_{b,c} / \tilde{D}_{b,c}^{\,W_D} \times \tilde{G}_{b,c}^{\,W_G}$ where $T_{b,c}$ is the penalty applied to blocks that are not sent to the same destination. In this way, blocks that have different destinations receive lower $T_{b,c}$ values, thus penalizing the similarity function $S_{b,c}$. In this study, a similarity index $S_{b,c} = R_{b,c} \times T_{b,c} / \tilde{D}_{b,c} \times \tilde{G}_{b,c}$ is as a combination of the indices employed in Tabesh & Askari-Nasab (2011, 2013). As we approach block aggregation in mining cuts as an optimization problem, we also develop a MILP-based model to address the mining cut clustering problem, independently of the production scheduling problem, a model described below.

Indices and sets:

I: Set of blocks in a bench (the problem domain);
J: Set of partitions that, when added together, equal the domain;
i: Index of a block ($i = 1,\ldots,m, \forall i \in I$);
j: Index of a cluster ($j = 1,\ldots,n, \forall j \in J$);
b,c: Indices representing two blocks in the domain I ($b,c = 1,\ldots,m, \forall b, c \in I$);
$adj4_b$: Set of blocks adjacent to b in the N-S and E-W directions ($b = 1,\ldots,m, \forall b \in I$);
$adj8_b$: Set of blocks adjacent to b in the N-S, E-W, NE-SW and NW-SE directions ($b = 1,\ldots,m, \forall b \in I$);
$S_{b,c}$: Set that computes the similarity function between each block b and c ($b,c = 1,\ldots,m, \forall b, c \in I$);
$Dist_{b,c}$: Set that computes the Euclidian distance between each block b and c ($b,c = 1,\ldots,m, \forall b, c \in I$).

Decision Variables:

$M_{i,j} \in \{0,1\}$: Binary variable which assigns a cluster j to each block i ($i = 1,\ldots,m$, $\forall i \in I$, $j = 1,\ldots,n$, $\forall j \in J$);
$x_i \in \{1,J\}$: Integer variable which assigns to each block i the value corresponding to the index of its cluster j ($i = 1,\ldots,m$, $\forall i \in I$, $j = 1,\ldots,n$, $\forall j \in J$);
$y_j \in \{0,I\}$: Integer variable which counts the number of blocks assigned to each cluster j ($j = 1,\ldots,n$, $\forall j \in J$);
$p \in \{0,1\}$: Binary variable used to linearize the objective function.

Parameters:

min_size_cluster: Minimum number of blocks assigned to each cluster;
max_size_cluster: Maximum number of blocks assigned to each cluster;
min_n_cluster: Minimum number of clusters allowed for a bench;
max_n_cluster: Maximum number of clusters allowed for a bench;
γ: Maximum Euclidian distance (maximum diameter) between two blocks b,c assigned to a cluster j ($b,c = 1,\ldots,m$, $\forall b, c \in I$, $j = 1,\ldots,n$, $\forall j \in J$).

Constraints:

The constraints implemented in our MILP-based model are determined according to equations (1) to (9):

$$y_j = \sum_{i \in [1,m]} M_{i,j}\ ,\ \forall j \in J \quad (1)$$

$$x_i = \sum_{j \in [1,n]} M_{i,j} \times j,\ \forall i \in I \quad (2)$$

$$\sum_{j \in [1,n]} M_{i,j} = 1, \forall i \in I \quad (3)$$

$$j \neq \emptyset \Rightarrow y_j \geq min\_size\_cluster, \forall j \in J \quad (4)$$

$$j = \emptyset \Rightarrow y_j = 0, \forall j \in J \quad (5)$$

$$y_j \leq max\_size\_cluster,\ \forall j \in J \quad (6)$$

$$M_{b,j} = 1 \Rightarrow \sum_{j \in [1,n]} M_{c,j} \geq 1, \forall b \in j, c \in adj4_b \quad (7)$$

$$M_{b,j} = 1 \Rightarrow \sum_{j \in [1,n]} M_{c,j} \geq 2, \forall b \in j, c \in adj8_b \quad (8)$$

$$M_{b,j} = 1, M_{c,j} = 1 \Rightarrow Dist_{b,c} \leq \gamma, \forall b,c \in j, \forall j \in J \quad (9)$$

Equations 1 and 2, respectively, aim to count the blocks assigned to each cluster y_j and ensure the assignment of a partition to each block x_i, while Equation 3 assures that only one cluster will be assigned to each block in the domain I. The *min_n_cluster* and *max_n_cluster* parameters are created to avoid determining the number of clusters in advance, allowing each partition to be empty or not at the end of the optimization, as long as all blocks have been assigned to some cluster. Equations 4 to 6 determine that each valid cluster must respect these boundaries, while any empty cluster must not contain blocks. In order to guarantee a smooth geometry for the clusters and avoid large recesses and corners, Equations 7 and 8 ensure that each block $b \in j$ has at least one neighbor in the set $adj4_b$ and two neighbors in the set $adj8_b$ belonging to the same partition j, resulting in clusters with adequate edges. This is an arbitrary parameter that can be managed depending on minimum pit width and size of the mining equipment. Equation 9, finally, assures that all blocks b and c only belong to a same cluster j if they have a maximum Euclidian distance less than or equal to the arbitrary parameter γ, then guaranteeing the continuity of the clusters. It is worth noting that the decisions about each j being empty or valid clusters in Equations 4 and 5 are indicator constraints, as well as $M_{b,j}$ being equal to 1 in Equations 7 to 9, such that these special constraints must be implemented according to the language or library being used, being linearized or not.

Objective Function:

The objective function aims to maximize the sum of the similarity functions calculated between each block b,c assigned to each cluster j, then the result is divided by two to avoid symmetry issues (computing $S_{b,c}$ and $S_{c,b}$). Thus, we would like to compute $S_{b,c}$ only when blocks b and c belong to the same cluster j, and the way to guarantee this would be to multiply the $M_{b,j}$ and $M_{c,j}$ variables by $S_{b,c}$, resulting in a non-linear model. Therefore, a binary variable $p \in \{0,1\}$ was introduced as an indicator constraint, assuming the value 1 only when $M_{b,j}$ and $M_{c,j}$ were equal to 1, resulting in the linearized objective function as written in equation (10):

$$Max \sum_{j \in [1,n]} \sum_{b,c \in j} (S_{b,c} \times p)/2 \quad (10)$$

However, to solve the MILP model using the CSP approach, just omit the objective function, or write it as ***Max 0***, as the algorithm extracts any feasible solution, not necessarily an optimal solution.

Comments:

The case study shown in the next section was run on a computer from the Laboratory of Mineral Research and Mining Planning (LPM) of the Federal University of Rio Grande do Sul (UFRGS), whose operating system has the Windows Server 2022, equipped with 2 Xeon 4214R 2.40 GHz 12-core processors, 256 GB of RAM, 4 SSDs of 1 TB and 1 external HD of 4 TB. The experiments were executed on bench 15 of the Newman1 dataset (Espinoza *et al.*, 2013), using the OR-Tools packages in Python, and those that rely on CP techniques used the CP-SAT solver, while those that rely on a MILP approach used the SCIP solver. The methodologies were compared in terms of processing time and the best solution achieved considering the objective function, as well as mine planning parameters, such as the mass of ore and waste sent to different destinations, the grades distribution, geometallurgical, and lithological classification dominant in each cluster. In addition, the silhouette coefficient, which has values between -1 and 1 and where high scores indicate better clusters, the Calinski-Harabasz index, where higher scores indicate better defined clusters, and the Davies-Bouldin index, which has a minimum score of 0 and where the lower the index, the better the separation between the clusters were also presented.

3. Results and discussion

The Newman1 dataset has 1,060 blocks distributed in 24 benches, containing 4 lithologies and 2 a priori destinations, and although the bench 15 has 83 blocks, 3 of them were excluded from the analysis because they did not have enough neighbor blocks to meet the parameters of the mathematical model, remaining 80 blocks. The *min_size_cluster* and *max_size_cluster* parameters were determined as 5 and 16 blocks, respectively, and the *min_n_cluster* was achieved by dividing the total number of blocks by *max_size_cluster*, while the *max_n_cluster* was achieved by dividing it by *min_size_cluster*. The true coordinates of the block model were not available, so the spacing between a block and its neighbors in the E-W or N-S directions equals 1. The parameter γ, related to the cluster's continuity, was set to 5. The adjustment of this parameter must be done meticulously for the instance studied. Low values may make it impossible to obtain feasible solutions, while high values would allow mixtures between elements from different clusters without violating the problem's constraints. Therefore, the γ value was obtained from trial-and-error experiments. Figure 2 shows the distribution of destinations (a), lithologies (b), and grades (c) of the Newman1's Bench 15.

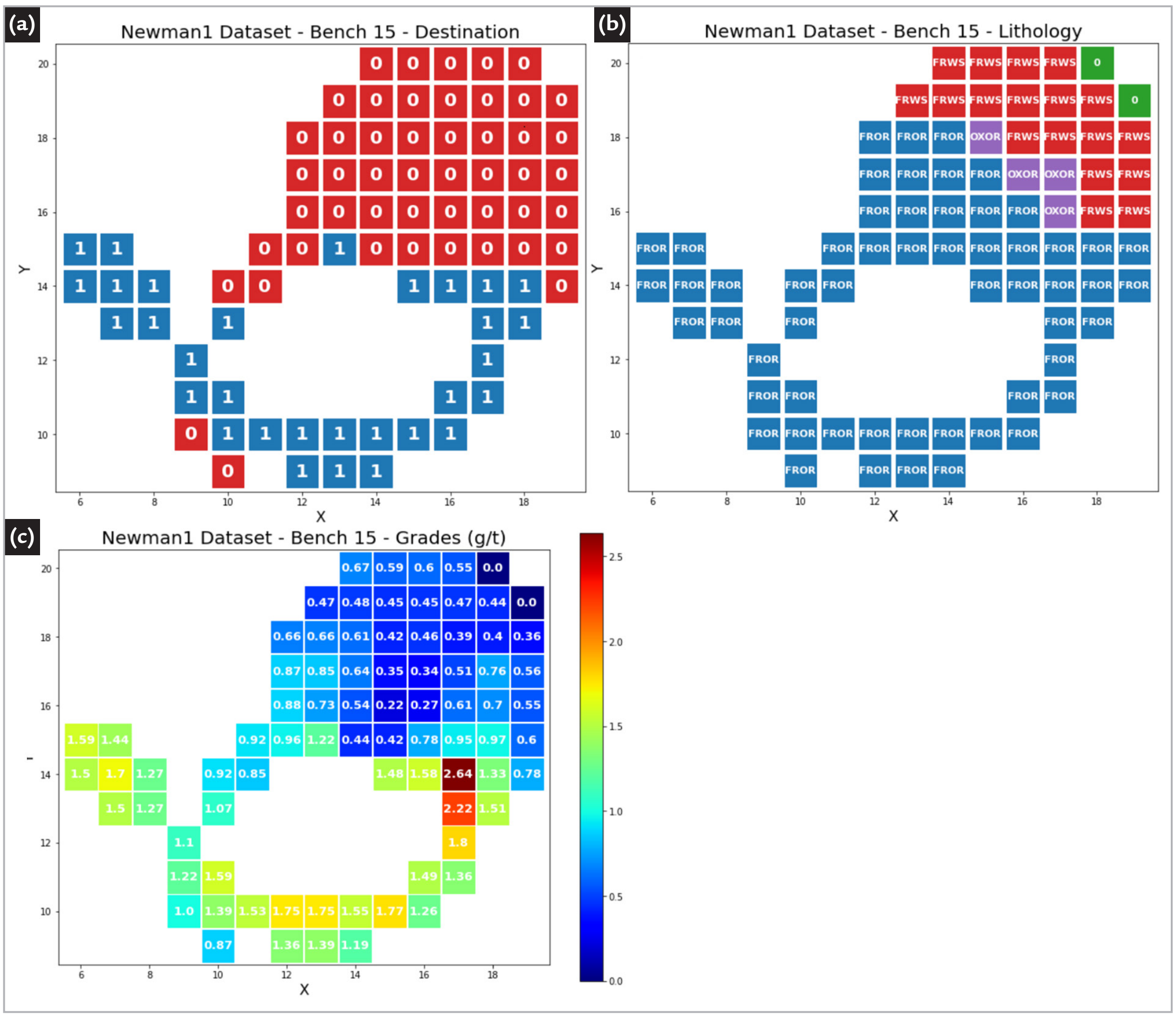

Figure 2 - Distribution of destinations (a), lithologies (b), and grades (c) of the Newman1's Bench 15.

Initially, an attempt was made to obtain an exact MILP solution so that it could be compared to the solutions obtained by CP, but after running for 1,180,110.13 seconds (13.66 days) considering a gap of 10% as a stopping criterion, no optimum solution was reached. To illustrate how it is difficult to solve NP-hard problems by exact algorithms as their size increases, it was possible to obtain an exact solution for the Newman1's Bench 3, containing 15 blocks, in 11.19 seconds, while a solution with a gap of 1% was obtained for Bench 9, with 49 blocks, in 2,472.97 seconds, thus taking 221 times longer for a problem approximately 3 times larger. Therefore, given the impossibility of obtaining an exact solution to serve as a parameter in an acceptable time, we chose to explore the feasible solution space of the Bench 15 through the CSP approach, and considering a stopping criterium of 7 days (604,839.38 seconds), the CSP algorithm identified 74,581 feasible solutions, although there is no guarantee that the entire solution space has been explored or that an optimal solution has been identified. It was possible to identify a discontinuity in the solution space, since 27,308 feasible solutions (36.62%) presented values between 246,607,154.18 and 404,699,200.11, while 47,273 solutions (63.38%) reached values between 4,727,646,352.89 and 4,891,735,908.29, the latter being the best solution found. Furthermore, we solved the proposed model by the COP approach considering a stopping criterion of 60 seconds and, including the time required to generate the mathematical model in the solver, in 117.14 seconds the algorithm reached a solution with an objective function of 4,893,625,450.99, slightly better than the best solution obtained by the CSP approach, but 5,163 times faster than investigating exhaustively the solution space to find it. Figure 3 presents the clusters generated by the best solutions of the Newman1's Bench 15 achieved by CSP (a) and COP (b) approaches, respectively.

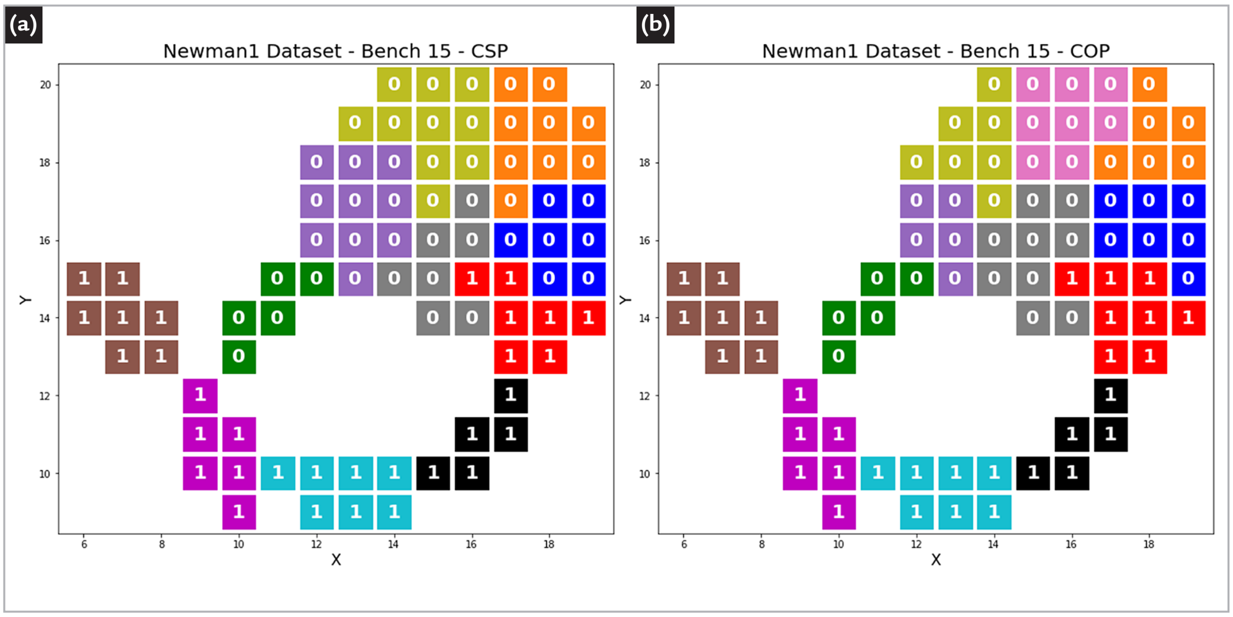

Figure 3 - Clusters generated in Newman1's Bench 15 by the CSP (a) and COP (b) approaches, respectively.

The CP approaches resulted in solutions with different numbers of clusters, and the optimization indicators were favorable to the COP solution, proving the speed and convergence capacity of the technique. However, the right side of the solution obtained by COP draws attention, including one unnecessary waste block in the red cluster sent to the processing plant, which would increase the dilution and reduce the economic return of the solution. Considering that the entire Bench 15 was excavated, the present value of each block was calculated, all of which presented mining costs, but only the blocks sent to the processing plant presented processing costs and benefits.

Mining costs are equivalent to the tonnage multiplied by the mining cost of each block, information that is already available in the dataset. The processing cost calculation is similar, although the cost of processing an OXOR lithology block differs from the cost of processing a block containing other lithologies. Finally, the processing benefit can be calculated by multiplying the selling price by the grade, mass, and recovery of the processed block. Furthermore, if this revenue is subtracted by mining and processing costs, we have the present value of a processed block. As it was also not indicated in the dataset what the monetary unit of the economic calculations would be, we chose to just call them units. Therefore, the present value of the CSP solution was equivalent to 217,379.77 units, while the solution achieved by COP resulted in a present value of 202,523.99 units. This difference must be attributed to the chosen similarity function, as the COP approach was more efficient in optimizing the model than the CSP approach, but as there were several factors composing this function, the trade-off among the factors affected the economic return of the best overall solution. Table 1 presents a summary of the best results obtained by the two techniques.

Table 1 - Parameters of the best solutions achieved by the CSP and
COP techniques, respectively, when solving the mining cut clustering problem for the Newman1's Bench 15.

Parameter	Best CSP	COP	Parameter	Best CSP	COP
Objective Function	4,891,735,908.29	4,893,625,450.99	Average grade (g/t)	1.38	1.37
Computational time (s)	604,839.38	117.14	Processed ore (t)	2,109.18	2,164.20
Number of clusters	11	12	Disposed ore (t)	1,533.60	1,478.58
Blocks sent to the processing plant	32	33	Processed waste (t)	150,313.82	155,922.80
Calinski-Harabasz	11.39	10.92	Disposed waste (t)	255,982.72	250,373.74
Davies-Bouldin	1.96	1.93	Dilution (t)	27,504.90	33,168.90
Silhouette	0.05	0.00	Ore loss (t)	13,145.73	13,145.73

4. Conclusions

In this study, we propose a MILP-based mathematical model to address the mining cut clustering problem and introduce a modification to the similarity indices developed by Tabesh & Askari-Nasab (2011, 2013), employing Constraint Programming techniques for the first time to solve this problem. When analyzing the performance of the proposed methods for a case study applied to Newman1's Bench 15, it was verified that the exact MILP approach could not converge in feasible time, so the CSP approach was employed to explore the solution space during 7 days, thus identifying 74,581 feasible solutions. However, in just 117.14 seconds, the COP approach was able to converge to an even better solution than the best solution obtained by the CSP approach, 5,163 times faster than exhaustively exploring the solution space. Nevertheless, it was identified that the solution obtained by CSP presented a better economic result than the solution obtained by COP, which can be attributed to a need to recalibrate the similarity index used, being a proposal for future studies. A multi-objective optimization approach could also be considered in the future, such that there is no interference between conflicting objectives grouped in a single equation, but that elite solutions can be identified that offer a trade-off between the objectives and allow more accurate decision-making. Applying the proposed model to a more complex case study, with well-defined parameters regarding equipment size and minimum pit width, would also be important. Finally, different models and algorithms can be explored, such as metaheuristic techniques, especially when considering the value of the CSP approach in generating viable solutions, and it is also possible to verify the production scheduling stage after solving the clustering problem.

Acknowledgments

This study was (partially) supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) through a PROEX (Programa de Excelência Acadêmica) Doctoral Scholarship [grant number 88887.487786/2020-00].

References

ALBACH, H. Long range planning in open-pit mining. ***Management Science,*** v. 13, n. 10, B549-B568, 1967.

ASKARI-NASAB, H.; AWUAH-OFFEI, K.; EIVAZY, H. Large-scale open pit production scheduling using Mixed Integer Goal Programming. ***International Journal of Mining and Mineral Engineering,*** v. 2, n. 3, p. 185-214, 2010a.

ASKARI-NASAB, H.; TABESH, M.; BADIOZAMANI, M. Creating mining cuts using hierarchical clustering and tabu search algorithms. *In*: CASTRO, R.; EMERY, X.; KUYVENHOVEN, R. (ed.) *In*: INTERNATIONAL CONFERENCE ON MINING INNOVATION, 4, 2010. [***Proceedings...***]. Santiago: University of Chile, 2010b. p. 159-171.

ASKARI-NASAB, H., POURRAHIMIAN, Y., BEN-AWUAH, E., KALANTARI, S. Mixed Integer Linear Programming formulations for open pit production scheduling. ***Journal of Mining Science,*** v. 47, n. 3, p. 338-359, 2011.

ASKARI-NASAB, H.; MIETH, C.; TABESH, M. A hybrid local search/mixed integer programming approach to open pit controlled phase-design. *In*: COSTA, J. F.; KOPPE, J.; PERONI, R. (ed.). Application of computers and operations research in the mineral industry. *In*: APCOM SYMPOSIUM, 36, 2013, Porto Alegre. [***Proceedings...***]. Porto Alegre: Fundação Luiz Englert, 2013. p. 1-14.

AXELSON, A. H. A practical approach to computer utilization in mine planning. *In*: APCOM SYMPOSIUM, 4, 1964. [***Proceedings...***]. Golden: Colorado School of Mines, 1964. p. 593-622.

BADIOZAMANI, M. M.; ASKARI-NASAB, H. Integration of reclamation and tailings management in oil sands surface mine planning. ***Environmental Modelling & Software,*** v. 51, p. 45-58, 2014.

BAGIROV, A. M.; KARMITSA, N.; TAHERI, S. ***Partitional clustering via nonsmooth optimization***: unsupervised and semi-supervised learning. Cham: Springer, 2020.

BEN-AWUAH, E.; ASKARI-NASAB, H. Oil sands mine planning and waste management using Mixed Integer Goal Programming. ***International Journal of Mining, Reclamation and Environment,*** v. 25, n. 3, p. 226-247, 2011.

BEN-AWUAH, E.; ASKARI-NASAB, H.; AWUAH-OFFEI, K. Production scheduling and waste disposal planning for oil sands mining using Goal Programming. ***Journal of Environmental Informatics,*** v. 20, n. 1, p. 20-33, 2012.

CHANDA, E. K.; RICCIARDONE, J. Long term production scheduling optimisation for a surface mining operation: an application of MineMax™ scheduling software. ***International Journal of Surface Mining, Reclamation and Environment,*** v. 16 n. 2, p. 144-158, 2002.

DAO, T.-B.-H.; DUONG, K.-C.; VRAIN, C. A declarative framework for constrained clustering. *In*: ***Machine Learning and Knowledge Discovery in Databases.*** Cham: Springer, 2013. p. 419-434.

DAO, T.-B.-H.; VRAIN, C.; DUONG, K.-C.; DAVIDSON, I. A framework for actionable clustering using constraint programming. *In*: KAMINKA, G. A.; FOX, M.; BOUQUET, P.; HÜLLERMEIER, E.; DIGNUM, V.; DIGNUM, F.; VAN HARMELEN, F. (ed.). *In*: EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE, 22, 2016, [***Proceedings…***]. The Hague. Amsterdam: IOS Press, 2016. p. 453-461.

DAO, T.-B.-H.; DUONG, K.-C.; VRAIN, C. Constrained clustering by constraint programming. ***Artificial Intelligence,*** v. 244, p. 70-94, 2017.

EIVAZY, H.; ASKARI-NASAB, H. A mixed integer linear programming model for short-term open pit mine production scheduling. ***Mining Technology,*** v. 121, n. 2, p. 97-108, 2012a.

EIVAZY, H.; ASKARI-NASAB, H. A hierarchical open-pit mine production scheduling optimisation model. ***International Journal of Mining and Mineral Engineering,*** v. 4, n. 2, p. 89-115, 2012b.

EPSTEIN, R.; GOIC, M.; WEINTRAUB, A.; CATALÁN, J.; SANTIBÁÑEZ, P.; URRUTIA, R.; CANCINO, R.; GAETE, S.; AGUAYO, A.; CARO, F. Optimizing long-term production plans in underground and open-pit copper mines. ***Operations Research,*** v. 60, n. 1, p. 4-17, 2012.

ESPINOZA, D.; GOYCOOLEA, M.; MORENO, E.; NEWMAN, A. MineLib: a library of open pit mining problems. ***Annals of Operations Research,*** v. 206, n. 1, p. 93-114, 2013.

GOLOMB, S. W.; BAUMERT, L. D. Backtrack Programming. ***Journal Of The ACM,*** v. 12, n. 4, p. 516-524, 1965.

GUNS, T.; NIJSSEN, S.; RAEDT, L. de. K-pattern set mining under constraints. ***IEEE Transactions on Knowledge and Data Engineering,*** v. 25, n. 2, p. 402-418, 2013.

GUNS, T.; DAO, T.-B.-H.; VRAIN, C.; DUONG, K.-C. Repetitive branch-and-bound using constraint programming for constrained minimum sum-of-squares clustering. ***In***: KAMINKA, G. A.; FOX, M.; BOUQUET, P.; HÜLLERMEIER, E.; DIGNUM, V.; DIGNUM, F.; VAN HARMELEN, F. (ed.). EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE, 22, 2016. [***Proceedings…***]. The Hague.. Amsterdam: IOS Press, 2016. p. 462-470.

JÉLVEZ, E.; MORALES, N.; NANCEL-PENARD, P.; PEYPOUQUET, J.;REYES, P. Aggregation heuristic for the open-pit block scheduling problem. ***European Journal of Operational Research,*** v. 249, n. 3, p. 1169-1177, 2016.

JOHNSON, T. B. Optimum production scheduling. In: INTERNATIONAL SYMPOSIUM ON COMPUTERS AND OPERATIONS RESEARCH, 8, 1969, Salt Lake City. [***Proceedings…***]. New York: SME/AIME, 1969. p. 539-562.

KHIARI, M.; BOIZUMAULT, P.; CRÉMILLEUX, B. Constraint Programming for mining n-ary patterns. *In*: ***Principles and Practice of Constraint Programming - CP 2010. Berlin: Springer, 2010.*** p. 552-567.

KLEER, J. de; SUSSMAN, G. J. Propagation of constraints applied to circuit synthesis. ***International Journal of Circuit Theory and Applications,*** v. 8, n. 2, p. 127-144, 1978.

KLINGMAN, D.; PHILLIPS, N. Integer programming for optimal phosphate-mining strategies. ***Journal of the Operational Research Society,*** v. 39, n. 9, p. 805-810, 1988.

KOUSHAVAND, B.; ASKARI-NASAB, H.; DEUTSCH, C. V. A linear programming model for long-term mine planning in the presence of grade uncertainty and a stockpile. ***International Journal of Mining Science and Technology,*** v. 24, n. 4, p. 451-459, 2014.

LERCHS, H.; GROSSMANN, I. F. Optimum design of open pit mines. ***CIM Bulletin,*** v. 68, p. 17-24, 1965.

MACKWORTH, A. K. Consistency in networks of relations. ***Artificial Intelligence,*** v. 8, n. 1, p. 99-118, 1977.

MAI, N. L.; TOPAL, E.; ERTENT, O. A new open-pit mine planning optimization method using block aggregation and integer programming. ***Journal of the Southern African Institute of Mining and Metallurgy,*** v. 118, n. 7, p. 705-714, 2018.

MANULA, C. B. ***Application of linear programming methods to mine planning and scheduling.*** The Pennsylvania State University, 1965. (Special Research Report Number SR-49).

MENABDE, M.; FROYLAND, G.; STONE, P.; YEATES, G. Mining schedule optimization for conditionally simulated orebodies. *In*: OREBODY MODELING AND STRATEGIC MINE PLANNING SYMPOSIUM, 1, 2004, Perth. [***Proceedings…***]. Uncertainty and Risk Management. Perth: The Australasian Institute of Mining and Metallurgy, 2004. p. 353-357.

METZ, H. J.; JAIN, S. K. Optimal mining and processing decisions in stratiform phosphate deposits. ***Interfaces,*** v. 9, n. 1, p. 1-12, 1978.

NELIS, G.; MORALES, N. A mathematical model for the scheduling and definition of mining cuts in short-term mine planning. ***Optimization and Engineering,*** v. 23, p. 233-257, 2022.

NELIS, G.; MEUNIER, F.; MORALES, N. Column generation for mining cut definition with geometallurgical interactions. ***Natural Resource Research,*** v. 31, p. 131-148, 2022.

NEWMAN, A. M.; KUCHTA, M. Using aggregation to optimize long-term production planning at an underground mine. ***European Journal of Operational Research,*** v. 176, n. 2, p. 1205-1218, 2007.

RAMAZAN, S. The new fundamental tree algorithm for production scheduling of open pit mines. ***European Journal of Operational Research,*** v. 177, n. 2, p. 1153-1166, 2007.

REZAKHAH, M.; MORENO, E.; NEWMAN, A. Practical performance of an open pit mine scheduling model considering blending and stockpiling. ***Computers & Operations Research,*** v. 115, 104638, 2020.

SALMAN, S.; MUHAMMAD, K.; KHAN, A.; GLASS, H. J. A block aggregation method for short-term planning of open pit mining with multiple processing destinations. ***Minerals,*** v. 11, n. 3, 288, 2021.

SAMANTA, B.; BHATTACHERJEE, A.; GANGULI, R. A genetic algorithms approach for grade control planning in a bauxite deposit. *In*: DESSUREAULT, S.; GANGULI, R.; KECOJEVIC, V.; GIRARD-DWYER, J. (ed.). ***Application of computers and operations research in the mineral industry.*** London: Taylor & Francis, 2005. p. 337-342.

TABESH, M.; ASKARI-NASAB, H. Two-stage clustering algorithm for block aggregation in open pit mines. ***Mining Technology***, v. 120, n. 3, p. 158-169, 2011.

TABESH, M.; ASKARI-NASAB, H. Automatic creation of mining polygons using hierarchical clustering techniques. ***Journal of Mining Science,*** v. 49, n. 3, p. 426-440, 2013.

TABESH, M.; MIETH, C.; ASKARI-NASAB, H. Open pit production planning using controlled pushbacks and aggregates. *In*: WORLD MINING CONGRESS, 23, 2013, Montreal. [***Proceedings…***]. Montreal: Canadian Institute of Mining, Metallurgy and Petroleum, 2013. p. 273-298.

TABESH, M.; MIETH, C.; ASKARI-NASAB, H. A multi-step approach to long-term open-pit production planning. ***International Journal of Mining and Mineral Engineering,*** v. 5, n. 4, p. 273-298, 2014.

TABESH, M.; ASKARI-NASAB, H.; PERONI, R. A comprehensive approach to strategic open pit mine planning with stockpile consideration. *In*: BANDOPADHYAY, S. (ed.). Application of Computers and Operations Research in the Mineral Industry, 37, 2015, Fairbanks. *In*: APCOM SYMPOSIUM, 37, 2015. [***Proceedings…***]. Englewood: Society for Mining, Metallurgy and Exploration Inc., 2015. p. 326-332.

TABESH, M.; ASKARI-NASAB, H. Clustering mining blocks in presence of geological uncertainty. ***Mining Technology,*** v. 128, n. 3, p. 162-176, 2019.

UPADHYAY, S. P.; ASKARI-NASAB, H. Truck-shovel allocation optimization: a Goal Programming approach. ***Mining Technology,*** v. 125, n. 2, p. 82-92, 2016.

WHITTLE, G. Global asset optimization. In: OREBODY MODELING AND STRATEGIC MINE PLANNING SYMPOSIUM, 1, 2004, Perth. [***Proceedings…***]. Uncertainty and risk management. Perth: The Australasian Institute of Mining and Metallurgy, 2004. p. 361-366.

WHITTLE, J. Beyond optimization in open pit design. *In*: FYTAS, K.; COLLINS, J. L.; SINGHAL, R. K. (ed.). ***Computer applications in the mineral industry***: first canadian conference. Rotterdam: A. A. Balkema, 1988. p. 331-337.

Received: 28 May 2023 - Accepted: 16 November 2023.