

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE MINAS,
METALÚRGICA E DE MATERIAIS**

**PROGRAMAÇÃO POR RESTRIÇÕES APLICADA AO
AGRUPAMENTO DE BLOCOS NO PLANEJAMENTO DE LAVRA**

JORGE LUIZ VALENÇA MARIZ

PORTO ALEGRE
2024

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE MINAS,
METALÚRGICA E DE MATERIAIS**

**PROGRAMAÇÃO POR RESTRIÇÕES APLICADA AO
AGRUPAMENTO DE BLOCOS NO PLANEJAMENTO DE LAVRA**

JORGE LUIZ VALENÇA MARIZ

Defesa de Tese apresentada ao Programa de Pós Graduação em Engenharia de Minas, Metalúrgica e de Materiais da Universidade Federal do Rio Grande do Sul como parte dos requisitos para obtenção do título de Doutor em Engenharia de Minas.

PORTO ALEGRE
2024

JORGE LUIZ VALENÇA MARIZ

**PROGRAMAÇÃO POR RESTRIÇÕES APLICADA AO
AGRUPAMENTO DE BLOCOS NO PLANEJAMENTO DE LAVRA**

Esta defesa de Tese foi julgada adequada e aprovada pelo Orientador e pela Banca Examinadora.

Aprovada em: _____

Orientador: Prof. Dr. Rodrigo de Lemos Peroni – UFRGS

BANCA EXAMINADORA:

Prof. Dr. Marcel Antonio Arcari Bassani – UFRGS

Prof. Dr. Ricardo Martins de Abreu Silva – UFPE

Dr. Luciano Nunes Capponi – Companhia Brasileira de Metalurgia e Mineração

PORTO ALEGRE
2024

AGRADECIMENTOS

Primeiramente, gostaria de agradecer a Deus e a meu guru, Paramahansa Yogananda, que diariamente me dão suporte, proteção, energia e discernimento para que eu empregue minha força de vontade em prol de um mundo mais justo e menos obscuro. Bárbara e Rany foram peças fundamentais nesta jornada, pois foi no cotidiano com elas, pautado no amor e no respeito, dia após dia, hora após hora, que eu pude acrescentar todas as pequenas pedras que, juntas, compõem este longo pavimento que representa a consolidação do meu entendimento científico. Agradeço a meus pais, Luciano e Valéria, por terem fornecido suporte e guiado meus passos enquanto eu não podia fazê-lo sozinho, permitindo assim que um potencial, uma promessa, se concretizasse em realidade, mas nunca tolhendo minhas escolhas. Sou grato ao meu orientador Rodrigo Peroni e ao Ricardo Martins, pois graças aos seus ensinamentos e às oportunidades a mim concedidas que esta pesquisa tomou forma. Também a contribuição do Prof. Hooman Askari-Nasab e do Mohammad Mahdi Badiozamani foram fundamentais para a qualidade e o êxito das publicações relacionadas a este trabalho. Por fim, agradeço a todos os professores que tive em vida, especialmente aos da UFRGS, pois ao lançar luz sobre as sombras da ignorância, nos fazem questionar nossas convicções e perceber que a realidade está repleta de dúvidas, e não de certezas, nos dando assim um vislumbre da nossa pequenez perante o universo.

RESUMO

O problema mais importante do planejamento de lavra é a definição do sequenciamento de lavra (*production scheduling problem*), quando deve-se decidir quando cada porção do depósito mineral será extraída e qual sua destinação. Entretanto, este é um problema em que os algoritmos conhecidos não são capazes de resolvê-lo de forma exata em tempo polinomial, sendo classificado como NP-difícil. Conseqüentemente, algumas estratégias são empregadas para simplificá-lo, como a definição prévia de uma cava final (*ultimate pit problem*) e a subdivisão do problema do sequenciamento sob as ópticas do longo, médio e curto prazo, que apresentam suas próprias particularidades, objetivos e restrições. Outra maneira de simplificar este problema é o agrupamento (*clustering*) de blocos em polígonos de lavra (*mining cuts*) com base em algum critério de similaridade, reduzindo assim o tamanho do problema do sequenciamento e adicionando restrições operacionais à solução, como largura mínima da cava e dimensão do equipamento de lavra. O problema do agrupamento também é NP-difícil, sendo frequentemente abordado por heurísticas, metaheurísticas e técnicas baseadas em aprendizado de máquina, geralmente prescindindo da definição formal de um modelo matemático. Neste estudo, é apresentada uma formulação matemática para o problema do agrupamento de polígonos de lavra em minas a céu aberto. Essa proposta baseia-se em Programação Inteira Mista (*Mixed-Integer Linear Programming, MILP*) e é projetada para ser abordada de forma independente do problema do sequenciamento de produção. Além disso, o problema é resolvido por programação por restrições (*Constraint Programming, CP*), que possibilita a obtenção de um conjunto de soluções viáveis ao amostrar o espaço de soluções a partir do modelo de otimização *Constraint Satisfaction Problem (CSP)*. Por outro lado, o modelo *Constraint Optimization Problem (COP)* obtém soluções ótimas para o problema, podendo até encontrar soluções que são ótimos globais. Como a solução direta do modelo proposto pode ainda gerar uma solução inviável do ponto de vista operacional, este estudo propõe ainda uma otimização multiestágio baseada em programação por restrições. Foi proposta ainda a heurística de propagação geométrica (*geometric propagation heuristic, GPH*), capaz de efetuar o refinamento dos *clusters* gerados e garantir que a solução respeite a dimensão do equipamento de lavra. Para ilustrar a metodologia proposta, são apresentados dois estudos de caso em bancos de dados reais, um consistindo na aplicação direta das abordagens *CSP* e *COP*, ao passo que o outro aplica a otimização multiestágio e a *GPH* em três cenários com diferentes parâmetros. Por fim, uma metodologia multiestágio reduzida composta de uma etapa empregando *k-means* e outra etapa empregando a *GPH* foi proposta,

cujos resultados foram comparados aos obtidos pela metodologia multiestágio baseada em programação por restrições. Os resultados comprovam que as técnicas empregadas neste estudo foram capazes de gerar polígonos de lavra que são ótimos locais em um tempo aceitável, além de ressaltarem o potencial da heurística de propagação geométrica, algoritmo capaz de rapidamente refinar a geometria de *clusters*.

Palavras-chave: Mineração a céu aberto; Planejamento de lavra; Planejamento operacional; Agrupamento de polígonos de lavra; Programação por restrições; Heurística de propagação geométrica.

ABSTRACT

The most important problem in mining planning is the definition of the mining sequencing, when it must be decided when each portion of the mineral deposit will be extracted and what its destination will be. However, this is a problem in which known algorithms are not able to solve it exactly in polynomial time, being classified as NP-hard. Consequently, some strategies are employed to simplify it, such as the prior definition of an ultimate pit and the subdivision of the sequencing problem from the perspective of long-, medium- and short-term, which present their own particularities, objectives and constraints. Another way to simplify this problem is to aggregate blocks into mining cuts based on a similarity criterion, reducing the size of the sequencing problem and adding operational criteria to the solution, such as the minimum pit width and size of mining equipment. The clustering problem is also NP-hard, and is often addressed by heuristics, metaheuristics and techniques based on machine learning, generally dispensing with the formal definition of a mathematical model. In this study, a mathematical formulation is presented for the mining cut clustering problem in open-pit mines. This proposal is based on Mixed-Integer Linear Programming (MILP) and is designed to be approached independently of the production scheduling problem. Furthermore, the problem is solved by Constraint Programming (CP), which makes it possible to obtain a set of feasible solutions by sampling the solution space using the Constraint Satisfaction Problem (CSP) optimization model. On the other hand, the Constraint Optimization Problem (COP) model obtains optimal solutions to the problem, and can even find solutions that are global optima. As the direct solution of the proposed model can still generate an unfeasible solution from an operational point of view, this study also proposes a multi-stage optimization based on Constraint Programming. The geometric propagation heuristic (GPH) was also proposed, capable of refining the generated clusters and ensuring that the solution respects the size of the mining equipment. To illustrate the proposed methodology, two case studies on real databases are presented, one consisting of the direct application of the CSP and COP approaches, while the other applies multistage optimization and GPH in three scenarios with different parameters. Finally, a reduced multi-stage methodology composed of a k-means step and a GPH step was proposed, the results of which were compared to those obtained by the multi-stage optimization based on Constraint Programming. The results prove that the techniques used in this study were capable of generating mining cuts that are local optima in an acceptable time, in addition to highlighting the potential of the geometric propagation heuristic, an algorithm capable of quickly refining the geometry of clusters.

Keywords: Open-pit mining; Mine planning; Operational planning; Mining cut clustering; Constraint Programming; Geometric propagation heuristic.

ÍNDICE DE TABELAS

TABELA 1 - RESUMO DOS ESTUDOS APRESENTADOS NESTE TÓPICO, CONTEMPLANDO OS PRINCIPAIS ASPECTOS RELACIONADOS AOS PROBLEMAS DO AGRUPAMENTO E DO SEQUENCIAMENTO DE LAVRA.	68
TABELA 2 - DESCRIÇÃO DOS MODELOS DE BLOCOS E DAS RESPECTIVAS BANCADAS UTILIZADAS NESTE ESTUDO.	94
TABELA 3 - PARÂMETROS DAS MELHORES SOLUÇÕES ALCANÇADAS PELAS TÉCNICAS <i>CSP</i> E <i>COP</i>, RESPECTIVAMENTE, AO RESOLVER O PROBLEMA DE AGRUPAMENTO DE CORTE DE MINERAÇÃO PARA A BANCADA 15 DO <i>NEWMAN1 DATASET</i>.	98
TABELA 4 - EVOLUÇÃO DOS PARÂMETROS DURANTE A OTIMIZAÇÃO NO EXPERIMENTO NUMÉRICO 1.	103
TABELA 5 - EVOLUÇÃO DOS PARÂMETROS DURANTE A OTIMIZAÇÃO NO EXPERIMENTO NUMÉRICO 2.	105
TABELA 6 - EVOLUÇÃO DOS PARÂMETROS DURANTE A OTIMIZAÇÃO NO EXPERIMENTO NUMÉRICO 3.	107
TABELA 7 - RESULTADOS OBTIDOS DURANTE OS EXPERIMENTOS NUMÉRICOS RELATIVOS À QUALIDADE DOS <i>CLUSTERS</i> E INDICADORES-CHAVE DE DESEMPENHO NO PLANEJAMENTO DE LAVRA.	108
TABELA 8 – COMPARAÇÃO ENTRE OS RESULTADOS OBTIDOS DURANTE O EXPERIMENTO NUMÉRICO 1 E UMA VERSÃO REDUZIDA DA METODOLOGIA MULTISTÁGIO.	111
TABELA 9 – COMPARAÇÃO ENTRE OS RESULTADOS OBTIDOS DURANTE O EXPERIMENTO NUMÉRICO 2 E UMA VERSÃO REDUZIDA DA METODOLOGIA MULTISTÁGIO.	114
TABELA 10 – COMPARAÇÃO ENTRE OS RESULTADOS OBTIDOS DURANTE O EXPERIMENTO NUMÉRICO 3 E UMA VERSÃO REDUZIDA DA METODOLOGIA MULTISTÁGIO.	117

ÍNDICE DE ILUSTRAÇÕES

FIGURA 1 – EXEMPLO DE APLICAÇÃO DA METODOLOGIA SIMPLIFICADA, CONSISTINDO NA RESOLUÇÃO DIRETA DO MODELO MATEMÁTICO ATRAVÉS DE OTIMIZAÇÃO, ONDE A) BLOCOS COM DESTINAÇÃO PREDETERMINADA SÃO AGRUPADOS EM <i>CLUSTERS</i> B) QUE COMPARTILHAM UMA MESMA DESTINAÇÃO.	15
FIGURA 2 - EXEMPLO DE APLICAÇÃO DA METODOLOGIA MULTISTÁGIO COMPLETA. INICIALMENTE, A) BLOCOS COM DESTINAÇÃO PREDETERMINADA SÃO AGRUPADOS POR <i>K-MEANS</i> (B). EM SEGUIDA, ESTA SOLUÇÃO PARCIAL É SUBMETIDA À OTIMIZAÇÃO (C), QUANDO A HEURÍSTICA <i>GPH</i> CORRIGE A GEOMETRIA DOS <i>CLUSTERS</i> (D).	16
FIGURA 3 - EXEMPLO DE APLICAÇÃO DA METODOLOGIA MULTISTÁGIO REDUZIDA, ONDE A) BLOCOS COM DESTINAÇÃO PREDETERMINADA SÃO AGRUPADOS POR <i>K-MEANS</i> (B) E OS <i>CLUSTERS</i> TÊM SUAS GEOMETRIAS REFINADAS PELA HEURÍSTICA <i>GPH</i> (C).	17
FIGURA 4 – ETAPAS PREVISTAS DURANTE A ELABORAÇÃO DESTE ESTUDO.	18
FIGURA 5 - REPRESENTAÇÃO DE QUATRO CAVAS ANINHADAS COM FATORES DE RETORNO CRESCENTES, TAL QUE $B_1 < B_2 < B_3 < B_4$	22
FIGURA 6 - REPRESENTAÇÃO DE UMA CAVA A CÉU ABERTO CONTENDO TRÊS <i>PUSHBACKS</i> COM TAMANHOS MUITO DIFERENTES, OCORRENDO O <i>GAP PROBLEM</i>	23
FIGURA 7 – REPRESENTAÇÃO DE UMA ENVOLTÓRIA CONVEXA SUPERIOR BEM DEFINIDA (DIREITA) E UMA DEFICIENTE E COM UMA GRANDE LACUNA (ESQUERDA) PLOTADOS EM UM GRÁFICO DE VALOR DA CAVA VS. TAMANHO DA CAVA.	24
FIGURA 8 - METODOLOGIA TRADICIONAL PARA OS PLANEJAMENTOS ESTRATÉGICO E TÁTICO EM CAVAS A CÉU ABERTO, INCLUINDO O USO DE UMA FUNÇÃO DE TRANSFERÊNCIA PARA CONVERTER O MODELO DE TEORES EM UM MODELO ECONÔMICO, A GERAÇÃO DE CAVAS ANINHADAS, A ESCOLHA DE <i>PUSHBACKS</i> , SEGUIDA DA OPERACIONALIZAÇÃO E O SEQUENCIAMENTO DOS BLOCOS DENTRO DE CADA <i>PUSHBACK</i>	26
FIGURA 9 – ESQUEMA REPRESENTANDO A CONVERSÃO DE BLOCOS DA EXPLORAÇÃO MINERAL EM BLOCOS DE CURTO PRAZO ATRAVÉS DA ADIÇÃO DE INFORMAÇÃO ORIUNDA DAS AMOSTRAS DE PERFURAÇÃO, SEGUIDO DO AGRUPAMENTO DESTES BLOCOS EM POLÍGONOS DE LAVRA, CUJAS AS CORES REPRESENTAM DIFERENTES DESTINOS E OS NÚMEROS REPRESENTAM PERÍODOS DE SEQUENCIAMENTO.	27
FIGURA 10 - AGRUPAMENTO DE DADOS EM A) DOIS <i>CLUSTERS</i> CIRCULARES E B) EM TRÊS HIPERPLANOS.	29
FIGURA 11 – RESULTADO DO AGRUPAMENTO DE UM CONJUNTO DE DADOS POR DIFERENTES CRITÉRIOS, ONDE FOI UTILIZADO EM A) UM CRITÉRIO SUBJETIVO, EM B) UM <i>COMPLETE-LINK CLUSTERING</i> , EM C) UM <i>SINGLE-LINK CLUSTERING</i> E EM D) A MÉTRICA <i>WCSS</i>	34
FIGURA 12 - EXEMPLO DE UMA APLICAÇÃO DO <i>K-MEANS</i> AO DESMONTE DE ROCHAS, ONDE UM CONJUNTO DE DADOS É SUBDIVIDIDO EM DOIS <i>CLUSTERS</i> A PARTIR DA DISTÂNCIA AO LOCAL DE DETONAÇÃO E DA CARGA MÁXIMA POR ESPERA.	38
FIGURA 13 – EXEMPLOS DE DENDROGRAMAS OBTIDOS POR MEIO DOS MÉTODOS (B) <i>SINGLE-LINK CLUSTERING</i> E (C) <i>COMPLETE-LINK CLUSTERING</i> A PARTIR DE UMA (A) MATRIZ DE DISSIMILARIDADE CALCULADA PARA QUATRO PONTOS DE DADOS ARBITRÁRIOS.	39

FIGURA 14 - REPRESENTAÇÃO DE (A) UM GRAFO NÃO DIRECIONADO E DE (B) UM GRAFO DIRECIONADO.....	47
FIGURA 15 – EXEMPLIFICAÇÃO DAS ESTRATÉGIAS (A) BUSCA EM LARGURA E (B) BUSCA EM PROFUNDIDADE.	49
FIGURA 16 - COMPARAÇÃO ENTRE OS ALGORITMOS (A) <i>BFS</i> , (B) <i>DIJKSTRA</i> E (C) <i>A* SEARCH</i> NA BUSCA PELO MENOR CAMINHO ENTRE UM NÓ DE ORIGEM (VERDE) E UM NÓ TERMINAL (VERMELHO) EM UM CENÁRIO COM OBSTÁCULO.	51
FIGURA 17 - MODELO SINTÉTICO EM QUE A) EQUIVALE AOS <i>CLUSTERS</i> APÓS A CLUSTERIZAÇÃO HIERÁRQUICA, B) CONSISTE NA DISTRIBUIÇÃO DE TEORES, C) EQUIVALE ÀS LITOLOGIAS, D) REPRESENTA OS <i>CLUSTERS</i> APÓS UMA ITERAÇÃO <i>TS</i> E E) EQUIVALE AOS <i>CLUSTERS</i> APÓS OITO ITERAÇÕES <i>TS</i>	55
FIGURA 18 - VISTA EM PLANTA DE UMA BANCADA MOSTRANDO A) A DIFERENCIAÇÃO ENTRE AS ZONAS DE ESTÉRIL E MINERALIZADA, B) O SEQUENCIAMENTO ÓTIMO NO CENÁRIO BASE, C) O SEQUENCIAMENTO ÓTIMO NO CENÁRIO NW-SE E D) O SEQUENCIAMENTO ÓTIMO NO CENÁRIO SE-NW.	59
FIGURA 19 - FLUXOGRAMA DA METODOLOGIA DE SEQUENCIAMENTO DE LAVRA SUGERIDA POR TABESH, MIETH E ASKARI-NASAB (2013, 2014).	61
FIGURA 20 - EXEMPLO DE CUSTOS DA INCERTEZA VERSUS DIFERENTES CAPACIDADES DE LAVRA E BENEFICIAMENTO EM UM CASO SINTÉTICO.	62
FIGURA 21 - DEFINIÇÃO DE POLÍGONOS DE LAVRA COM BASE NA METODOLOGIA PROPOSTA POR NELIS E MORALES (2022), ONDE UMA LISTA DE BLOCOS CANDIDATOS A REPRESENTANTES É DETERMINADA E ALGUNS DESTES SÃO ELEITOS COMO REPRESENTANTES DOS <i>MINING CUTS</i> , EXPRESSOS EM DIFERENTES CORES. EM SEGUIDA, TODOS OS BLOCOS SÃO ASSOCIADOS A ALGUM DESTES REPRESENTANTES, COMPARTILHANDO SUA DESTINAÇÃO.	66
FIGURA 22 - FLUXOGRAMA SINTETIZANDO OS PROCEDIMENTOS EMPREGADOS NESTE ESTUDO NO ÂMBITO DA OTIMIZAÇÃO DO PROBLEMA DO AGRUPAMENTO PARA GERAÇÃO DE POLÍGONOS DE LAVRA.	71
FIGURA 23 - ILUSTRAÇÃO REPRESENTANDO OS CONJUNTOS ADJ_{4B} E ADJ_{8B} PARA UM DADO BLOCO $B \in I$	73
FIGURA 24 - PSEUDOCÓDIGO DA FASE DE INICIALIZAÇÃO DA OTIMIZAÇÃO MULTISTÁGIO.	82
FIGURA 25 - PSEUDOCÓDIGO DA FASE DE AJUSTE DE PARÂMETROS DA OTIMIZAÇÃO MULTISTÁGIO.....	83
FIGURA 26 - PSEUDOCÓDIGO DO PROCEDIMENTO DE JANELA MÍNIMA EMPREGADO NA OTIMIZAÇÃO MULTISTÁGIO.	84
FIGURA 27 - PSEUDOCÓDIGO DA FASE DE EXPLOTAÇÃO DA OTIMIZAÇÃO MULTISTÁGIO... ..	84
FIGURA 28 - PSEUDOCÓDIGO DAS ETAPAS DE INICIALIZAÇÃO E RECONSTRUÇÃO DA SOLUÇÃO POR <i>GPH</i> DURANTE A OTIMIZAÇÃO MULTISTÁGIO.....	87
FIGURA 29 - PSEUDOCÓDIGO DO PROCEDIMENTO SIMPLES DE REFINAMENTO DE FORMAS EMPREGADO PELA <i>GPH</i>	89
FIGURA 30 - PSEUDOCÓDIGO DO PROCEDIMENTO COMPLETE DE REFINAMENTO DE FORMAS EMPREGADO PELA <i>GPH</i>	90
FIGURA 31 - PSEUDOCÓDIGO DA HEURÍSTICA <i>GPH</i>	91

FIGURA 32 – DISTRIBUIÇÃO ESPACIAL A) DAS LITOLOGIAS, B) DAS DESTINAÇÕES A PRIORI E C) DOS TEORES DOS BLOCOS DA BANCADA 15 DO <i>NEWMAN1 DATASET</i>	95
FIGURA 33 - <i>CLUSTERS</i> GERADOS NA BANCADA 15 DO <i>NEWMAN1 DATASET</i> A PARTIR DAS ABORDAGENS A) <i>CSP</i> E B) <i>COP</i> , RESPECTIVAMENTE, REPRESENTADOS EM DIFERENTES CORES.....	97
FIGURA 34 - DISTRIBUIÇÃO ESPACIAL A) DAS LITOLOGIAS, B) DESTINAÇÕES A PRIORI E C) TEORES DE AU NOS BLOCOS DA BANCADA 4 DO <i>W23 DATASET</i>	100
FIGURA 35 - CARACTERÍSTICAS OPERACIONAIS DE UMA ESCAVADEIRA KOMATSU PC-200.	101
FIGURA 36 - EXPERIMENTO NUMÉRICO 1: A) SOLUÇÃO DICA 1, OBTIDA POR <i>K-MEANS</i> ; B) SOLUÇÃO DICA 2, OBTIDA POR <i>COP</i> ; C) SOLUÇÃO EXPLOTATÓRIA, OBTIDA POR <i>COP</i> ; D) E SOLUÇÃO FINAL COM 6 <i>CLUSTERS</i> , OBTIDA POR <i>GPH</i>	103
FIGURA 37 - EXPERIMENTO NUMÉRICO 2: A) SOLUÇÃO DICA 1, OBTIDA POR <i>K-MEANS</i> ; B) SOLUÇÃO DICA 2, OBTIDA POR <i>COP</i> ; C) SOLUÇÃO EXPLOTATÓRIA, OBTIDA POR <i>COP</i> ; E D) SOLUÇÃO FINAL COM 7 <i>CLUSTERS</i> , OBTIDA POR <i>GPH</i>	105
FIGURA 38 - EXPERIMENTO NUMÉRICO 3: A) SOLUÇÃO DICA 1, OBTIDA POR <i>K-MEANS</i> ; B) SOLUÇÃO DICA 2, OBTIDA POR <i>COP</i> ; C) SOLUÇÃO EXPLORATÓRIA, OBTIDA POR <i>COP</i> ; E D) SOLUÇÃO FINAL COM 10 <i>CLUSTERS</i> , OBTIDA POR <i>GPH</i>	107
FIGURA 39 - CONSIDERANDO AS CONFIGURAÇÕES DO EXPERIMENTO NUMÉRICO 1, COMPARAÇÃO ENTRE OS AGRUPAMENTOS GERADOS PELA A) APLICAÇÃO DO <i>K-MEANS</i> COMBINADO COM A HEURÍSTICA <i>GPH</i> E PELA B) OTIMIZAÇÃO MULTIESTÁGIO BASEADA EM <i>COP</i>	110
FIGURA 40 - CONSIDERANDO AS CONFIGURAÇÕES DO EXPERIMENTO NUMÉRICO 2, COMPARAÇÃO ENTRE OS AGRUPAMENTOS GERADOS A) PELA APLICAÇÃO DO <i>K-MEANS</i> COMBINADO COM A HEURÍSTICA <i>GPH</i> E B) PELA OTIMIZAÇÃO MULTIESTÁGIO BASEADA EM <i>COP</i> . CONSIDERANDO AINDA C) O <i>CLUSTER</i> DESTINADO AO DEPÓSITO DE ESTÉRIL PELA METODOLOGIA MULTIESTÁGIO COMPLETA (EM PRETO), É POSSÍVEL IDENTIFICAR QUAIS DESTES BLOCOS FORAM MAL CLASSIFICADOS.	113
FIGURA 41 - CONSIDERANDO AS CONFIGURAÇÕES DO EXPERIMENTO NUMÉRICO 3, COMPARAÇÃO ENTRE OS AGRUPAMENTOS GERADOS A) PELA APLICAÇÃO DO <i>K-MEANS</i> COMBINADO COM A HEURÍSTICA <i>GPH</i> E B) PELA OTIMIZAÇÃO MULTIESTÁGIO BASEADA EM <i>COP</i> . CONSIDERANDO AINDA C) OS <i>CLUSTERS</i> DESTINADOS AO DEPÓSITO DE ESTÉRIL PELAS METODOLOGIAS MULTIESTÁGIO REDUZIDA (EM AMARELO) E COMPLETA (EM PRETO), É POSSÍVEL IDENTIFICAR QUAIS DESTES BLOCOS FORAM MAL CLASSIFICADOS.	116

LISTA DE ABREVIATURAS E SÍMBOLOS

ABC – *Artificial Bee Colony*

ACO – *Ant Colony Optimization*

BB – *Branch-and-Bound*

BFS – *Breadth-First Search*

BIRCH – *Balanced Iterative Reducing and Clustering using Hierarchies*

CCP – *Chance-Constrained Programming*

CGA – *Column Generation Algorithm*

CP – *Constraint Programming*

CLP – *Constraint Logic Programming*

COP – *Constraint Optimization Problem*

CSP – *Constraint Satisfaction Problem*

CURE – *Clustering Using REpresentatives*

DBSCAN – *Density-Based Spatial Clustering of Applications with Noise*

DFS – *Depth-First Search*

DES – *Discrete Event Simulation*

DP – *Dynamic Programming*

FIFO – *First-In, First-Out*

FFA – *Firefly Algorithm*

FLC – *Fuzzy Logic Clustering*

GA – *Genetic Algorithm*

GPH – *Geometric Propagation Heuristic*

GRASP – *Greedy Randomized Adaptive Search Procedures*

GWO – *Grey Wolf Optimization*

HC – *Hierarchical Clustering*

IP – *Integer Programming*

KNNCLUST – *k-NN Density-Based Clustering*

LB – *Lower Bound*

LGP – *Linear Goal Programming*

LIFO – *Last-In, First-Out*

LP – *Linear Programming*

LG – *Algoritmo de Lerchs - Grossmann*

LR – *Lagrangian relaxation*

MIGP – *Mixed Integer Goal Programming*
MIP – *Mixed Integer Programming*
MILGP – *Mixed Integer Linear Goal Programming*
MILP – *Mixed Integer Linear Programming*
MINLP – *Mixed Integer Nonlinear Programming*
ML – *Machine Learning*
PCKP – *Precedence Constrained Knapsack Problem*
PSO – *Particle Swarm Optimization*
QCQP – *Quadratically Constrained Quadratic Program*
QP – *Quadratic Programming*
SMU – *Selective Mining Unit*
SOCP – *Second-Order Cone Programming*
SOM – *Self-Organizing Maps*
TS – *Tabu Search*
UB – *Upper Bound*
VPL – *Valor Presente Líquido*
WCSD – *Within-Cluster Sum of Dissimilarities*
WCSS – *Within-Cluster Sum of Squares*

SUMÁRIO

1 INTRODUÇÃO	11
1.1 HIPÓTESE INVESTIGATIVA E META.....	13
1.2 OBJETIVOS	14
1.3 ETAPAS DA PESQUISA	15
2 REVISÃO DO ESTADO DA ARTE.....	19
2.1 OTIMIZAÇÃO MATEMÁTICA E PLANEJAMENTO DE LAVRA	19
2.1.1 Forma geral de um problema de otimização matemática	19
2.1.2 O problema do sequenciamento de lavra.....	20
2.1.3 Horizontes temporais no sequenciamento de lavra: planejamento estratégico, tático e operacional.....	24
2.2 O PROBLEMA DO AGRUPAMENTO	27
2.2.1 Medidas de similaridade entre agrupamentos.....	28
2.2.2 Medidas de similaridade em polígonos de lavra por Tabesh e Askari-Nasab (2011, 2013).....	30
2.2.3 Modelos de otimização na análise de agrupamentos	32
2.2.4 Métricas de avaliação da qualidade de agrupamentos.....	35
2.3 PROGRAMAÇÃO POR RESTRIÇÕES E ALGORITMOS RELEVANTES	36
2.3.1 Heurísticas no problema do agrupamento.....	36
2.3.2 Programação por restrições e seu emprego no problema do agrupamento	40
2.3.2.1 Fundamentos da programação por restrições	40
2.3.2.2 Aplicação da programação por restrições ao problema do agrupamento	42
2.3.3 Busca pelo menor caminho em um grafo.....	47
2.4 O PROBLEMA DO AGRUPAMENTO DE POLÍGONOS DE LAVRA	51
3 METODOLOGIA.....	70
3.1 MODELO MATEMÁTICO DO PROBLEMA DO AGRUPAMENTO DE POLÍGONOS DE LAVRA	72
3.1.1 Índices e conjuntos	72
3.1.2 Parâmetros.....	73
3.1.3 Variáveis de decisão	74
3.1.4 Critério de similaridade.....	74
3.1.5 Função objetivo	75
3.1.6 Restrições do problema.....	76
3.2 METODOLOGIA SIMPLIFICADA: PARTICULARIDADES DA MODELAGEM MATEMÁTICA CONSIDERANDO DIFERENTES ABORDAGENS POR OTIMIZAÇÃO	77
3.3 METODOLOGIA MULTISTÁGIO BASEADA EM PROGRAMAÇÃO POR RESTRIÇÕES	80
3.3.1 Fase de inicialização.....	80
3.3.2 Fase de ajuste de parâmetros	82
3.3.3 Fase de exploração.....	84

3.3.4 Fase de pós-processamento	85
3.3.4.1 Etapa de inicialização heurística.....	85
3.3.4.2 Etapa de reconstrução da solução.....	85
3.3.4.3 Etapa de refinamento de formas.....	88
3.4 METODOLOGIA MULTISTÁGIO REDUZIDA.....	91
3.5 VALIDAÇÃO E COMPARAÇÃO ENTRE METODOLOGIAS DE AGRUPAMENTO DE POLÍGONOS DE LAVRA.....	92
4 ANÁLISE DE RESULTADOS E DISCUSSÃO	94
4.1 METODOLOGIA SIMPLIFICADA: BANCADA 15 DO <i>NEWMAN 1 DATASET</i>	94
4.2 METODOLOGIA MULTISTÁGIO COMPLETA: BANCADA 4 DO <i>W23 DATASET</i>	99
4.2.1 Experimento numérico 1	102
4.2.2 Experimento numérico 2	104
4.2.3 Experimento numérico 3	106
4.2.4 Comparação entre os experimentos numéricos.....	108
4.3 UMA METODOLOGIA MULTISTÁGIO REDUZIDA SERIA SUFICIENTE?	109
5 CONCLUSÕES	120
5.1 CONSIDERAÇÕES FINAIS EM RELAÇÃO AOS EXPERIMENTOS	120
5.2 TRABALHOS FUTUROS.....	122
REFERÊNCIAS	125
APÊNDICE I – CÓDIGO EM <i>PYTHON</i> DO MODELO DO AGRUPAMENTO DE POLÍGONOS DE LAVRA SOLUCIONADO POR PROGRAMAÇÃO POR RESTRIÇÕES NO <i>OR-TOOLS</i>	142
APÊNDICE II – CÓDIGO EM <i>PYTHON</i> DO MODELO DO AGRUPAMENTO DE POLÍGONOS DE LAVRA SOLUCIONADO POR PROGRAMAÇÃO INTEIRA MISTA NO <i>OR-TOOLS</i>	145

1 INTRODUÇÃO

Pequenas variações nas condições de contorno de um projeto podem ter impacto significativo sobre a configuração final de uma mina, seu sequenciamento, seu fluxo de caixa e, conseqüentemente, no retorno final deste projeto. Devido ao grande número de parâmetros considerados, o planejamento de mina está associado a inúmeras fontes de incerteza, que são relevantes para o desenvolvimento de projetos de mineração, principalmente devido ao grande impacto financeiro inerente à tomada de decisão nesta etapa. Um modelo não é capaz de mapear todas as fontes de incerteza, sendo estas dependentes dos parâmetros de entrada do modelo ou do cenário simulado. Destarte, faz-se necessário entender e identificar essas fontes para medir e controlar riscos, como incertezas geológicas, econômicas e quaisquer outras que impactem na conversão de recursos para reservas (DIMITRAKOPOULOS; FARRELLY; GODOY, 2002; PERONI, 2002; DIMITRAKOPOULOS; MARTINEZ; RAMAZAN, 2007; CHERCHENEVSKI; COSTA; RUBIO, 2019).

Pensando ainda no planejamento estratégico, voltado principalmente ao longo prazo, a avaliação de projetos de mineração se inicia na sondagem e amostragem, passa pela criação de um modelo geológico e de estimativa representativo, decisão dos métodos de lavra e beneficiamento, avaliação dos custos de capital e operacionais e, finalmente, elaboração de um plano técnico e financeiro que contemple toda a vida útil da mina, incluindo seu fechamento. A determinação de cenários e indicadores econômicos (como o Valor Presente Líquido, ou VPL) é fundamental para que uma avaliação econômica robusta seja efetuada, possibilitando assim avaliar a rentabilidade e exequibilidade do projeto, sendo fortemente desejado o mapeamento e quantificação dos riscos envolvidos na operação. Uma fonte crítica de risco técnico está no teor e na tonelagem esperados de minério; a capacidade de modelar e integrar esse risco à otimização e ao planejamento é de suma importância, e permite que uma abordagem mais consciente seja adotada na avaliação de um ativo ou gerenciamento de um projeto (DIMITRAKOPOULOS; FARRELLY; GODOY, 2002).

Entretanto, espera-se do planejamento estratégico não somente a maximização de uma função matemática (*e.g.* VPL) mediante as restrições impostas (precedência, ângulo de talude, tores, balanço metalúrgico, definição de capacidade de equipamentos de lavra e beneficiamento, dentre outros); o plano estratégico deve ser exequível, flexível e capaz de contemplar os riscos, caso contrário suas previsões jamais irão se concretizar. O planejamento tático tem como objetivos detalhar melhor o planejamento estratégico em relação aos períodos mais imediatos, mantendo a ele uma aderência satisfatória enquanto visa tornas as cavas

matemáticas operacionais. O planejamento operacional, por sua vez, deve buscar meios de atender aos requerimentos da usina de beneficiamento enquanto satisfaz restrições físicas, de equipamentos e recursos necessários para operacionalizar as atividades, tomando sempre como referência a aderência ao plano tático, que por sua vez é balizado pelo plano estratégico. Embora os riscos associados à operação sejam altos e diversos, a ausência de harmonia entre estes diferentes horizontes temporais pode igualmente levar o projeto ao insucesso ou frustração de resultados esperados, mesmo que uma ou outra etapa tenha sido executada individualmente de maneira supostamente “ótima”.

Em um cenário utópico, os investidores gostariam de conhecer com precisão todos os aspectos relacionados à operação de uma mina da abertura ao fechamento, onde seriam antecipados todos os parâmetros-chaves ao sucesso do empreendimento, como distribuição de teores e mineralogias, preço de venda do minério, balanço metalúrgico, dentre outras características, de preferência em um horizonte temporal de dias ou horas, de forma que as dificuldades pudessem ser antecipadas adequadamente. Entretanto, não somente é impossível prever todas as incertezas do projeto durante toda sua vida útil, como ainda é computacionalmente impossível criar um modelo tão complexo como o do sequenciamento de lavra em minas a céu aberto considerando tantos arcos de precedência entre blocos, principalmente em grandes depósitos, que podem conter milhões de blocos e diversas restrições, bem como em períodos curtos durante intervalos de tempo tão longos quanto a vida útil de uma mina. Isso se dá porque o problema do sequenciamento de lavra pode ser formulado como o problema da mochila com restrição de precedência (*Precedence Constrained Knapsack Problem, PCKP*), um notório problema NP-difícil (*Non deterministic polynomial-time hardness* ou *NP-hard*), ou seja, não existem algoritmos conhecidos capazes de os resolver em tempo polinomial (COWLING; GREENBERG; DE JONG, 2013).

Consequentemente, os problemas relativos aos horizontes temporais do planejamento de lavra (longo, médio e curto prazos) foram formulados de diversas maneiras desde a década de 1960, sendo vasta a gama de algoritmos empregados na resolução destas formulações. É ainda comum a formulação do problema em etapas distintas, onde primeiro se aborda o problema da cava final (*Ultimate Pit Problem*), depois são geradas fases (*pushbacks*) contidas nesta cava final e, somente então, o sequenciamento propriamente dito (*Production Scheduling Problem*) é otimizado em cada *pushback*, tendo sido empregadas heurísticas, metaheurísticas, algoritmos exatos e híbridos em cada uma destas etapas, conforme revisões elaboradas por Caccetta e Hill (2003), Osanloo, Gholamnejad e Karimi (2008), Newman *et al.*

(2010), Espinoza *et al.* (2013), Meagher, Dimitrakopoulos e Avis (2014), Blom, Pearce e Stuckey (2019), Flores-Fonseca, Linfati e Escobar (2021) e Mariz e Soofastaei (2022).

Destarte, dada a complexidade do *Production Scheduling Problem*, além da subdivisão do problema e da abordagem em diferentes horizontes temporais, uma maneira efetiva de reduzir a complexidade do problema é criar agrupamentos de blocos (*block aggregation*), visto que assim os arcos de precedência seriam reduzidos significativamente, já que o sequenciamento considera agora os *clusters*, e não mais os blocos originais. Além de reduzir o tamanho do problema do sequenciamento de lavra, o agrupamento de blocos pode adicionar critérios operacionais que de outro modo seriam ignorados durante a resolução do problema. Embora existam proposições baseadas em reblockagem (CHANDA; RICCIARDONE, 2002; JÉLVEZ *ET AL.*, 2016), técnicas que incorporam ângulo do talude às partições (RAMAZAN, 2007; MAI; TOPAL; ERTEN, 2018) e até propostas de agregação temporal (NEWMAN; KUCHTA, 2007) ao invés da agregação de blocos, a metodologia mais aplicada atualmente é o agrupamento de polígonos de lavra (*mining cuts*).

Através desta técnica, os blocos pertencentes a uma mesma bancada em uma cava a céu aberto são agrupados em polígonos de lavra com base em uma função de similaridade pré-determinada, e ao isolar este procedimento como um problema de otimização à parte do problema de sequenciamento de lavra é possível incorporar várias restrições ao problema, como tamanho mínimo e máximo dos *clusters*, controle geométrico, largura mínima de operação, dimensão dos equipamentos de escavação, direção preferencial de lavra, dentre outros (TABESH; ASKARI-NASAB, 2011, 2013). No entanto, o problema do agrupamento (*clustering problem*), tal qual o sequenciamento de lavra, também é NP-difícil, sendo frequente abordado por heurísticas, principalmente técnicas relacionadas ao aprendizado de máquina (*Machine Learning, ML*), como o *K-means* (MACQUEEN, 1967) e clusterização hierárquica (*Hierarchical Clustering, HC*), mas também sendo frequentemente solucionado por meio de metaheurística evolucionárias como o algoritmo genético (*Genetic Algorithm, GA*) (GOLDBERG, 1989; AGGARWAL; REDDY, 2014; BAGIROV; KARMITSA; TAHERI, 2020).

1.1 HIPÓTESE INVESTIGATIVA E META

Devido à dificuldade de resolução dos problemas NP-difíceis, o problema do agrupamento de polígonos de lavra vem sendo resolvido por heurísticas e técnicas baseadas em *ML*, prescindindo de um modelo matemático formal. Destarte, a hipótese investigativa

considera que desenvolver um modelo matemático para este problema e solucioná-lo por meio de uma abordagem que seja eficiente dos pontos de vista da qualidade da função objetivo e do tempo de processamento gerará *inputs* para o problema do sequenciamento de lavra operacional melhores do que os blocos individualizados. Ainda, se considera que implementar uma etapa de pós-processamento após a etapa de otimização gerará soluções ainda mais robustas, principalmente nos casos em que a solução otimizada não seja adequada quanto às fronteiras dos *clusters* e à dimensão do equipamento de lavra.

A meta deste estudo, portanto, é propor uma nova solução baseada em otimização matemática capaz de resolver o problema do agrupamento de blocos em polígonos de lavra para minas a céu aberto, visando melhorar o desempenho e a operacionalidade do sequenciamento de lavra determinístico no nível do planejamento operacional.

1.2 OBJETIVOS

Contextualizado o problema do planejamento de lavra determinístico, seus diferentes objetivos e horizontes temporais, este estudo volta-se principalmente à proposição de uma nova abordagem para solução do problema do agrupamento de polígonos de lavra ou *mining cuts*, visando melhorar o desempenho do sequenciamento de lavra no âmbito do médio e do curto prazo. Para atingir esta meta, foram definidos os seguintes objetivos:

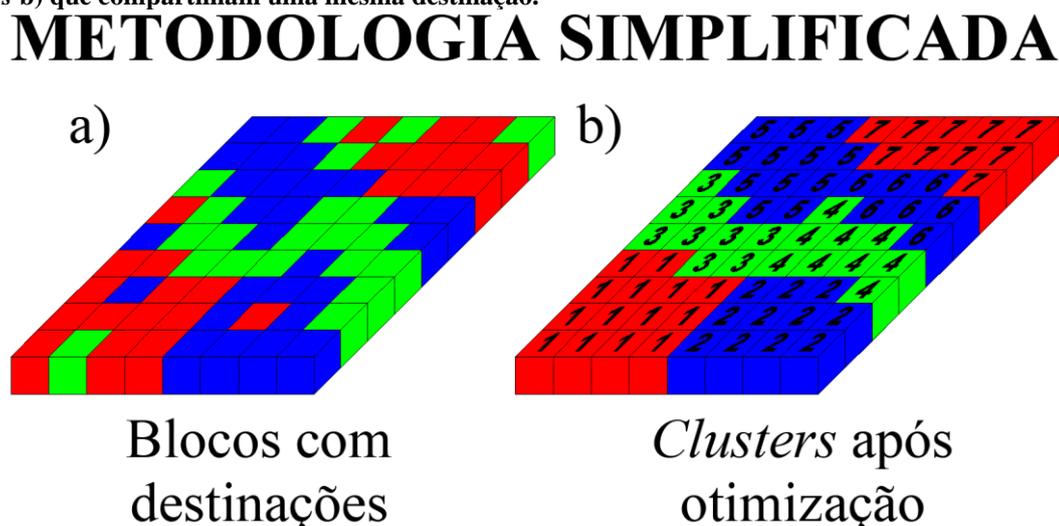
1. Mapear as principais formulações e variantes do problema do agrupamento de blocos em *mining cuts* para o sequenciamento de lavra em minas a céu aberto;
2. Analisar os diferentes algoritmos empregados na otimização do problema do agrupamento de blocos em *mining cuts* para o sequenciamento de lavra de minas a céu aberto;
3. Desenvolver um modelo matemático para abordar o problema do agrupamento de polígonos de lavra;
4. Avaliar o desempenho dos modelos matemáticos com dados reais e de *benchmarks* para aplicação do problema do agrupamento de polígonos de lavra voltado ao planejamento de lavra operacional;
5. Desenvolver uma metodologia robusta para solução computacionalmente eficiente do modelo matemático voltado ao problema do agrupamento de polígonos de lavra associado ao planejamento de lavra operacional;
6. Desenvolver uma solução de pós-processamento capaz de corrigir possíveis erros ou inconsistências apresentados pelos *clusters* gerados durante a otimização;

7. Empregar uma metodologia de testes estruturada, que contemple a validação dos modelos gerados através de métricas de otimização (*e.g.*: valor de função objetivo e tempo de execução), bem como a validação por meio de métricas relacionadas à mineração (*e.g.*: diluição e perda de minério);
8. Publicar um ou mais artigos científicos em periódicos de reconhecimento internacional, mantendo a aderência dos artigos ao tema sugerido, culminando na defesa de uma tese perante uma banca avaliadora.

1.3 ETAPAS DA PESQUISA

Para atender aos objetivos propostos nessa pesquisa, o problema do agrupamento de blocos em polígonos de lavra foi modelado por Programação Linear Inteira Mista (*Mixed Integer Linear Programming, MILP*) e resolvido de forma exata e por programação por restrições (*Constraint Programming, CP*). A resolução direta do modelo matemático através de otimização foi chamada neste estudo de metodologia simplificada, representada na Figura 1 em um exemplo que considera somente a destinação dos blocos como critério de similaridade.

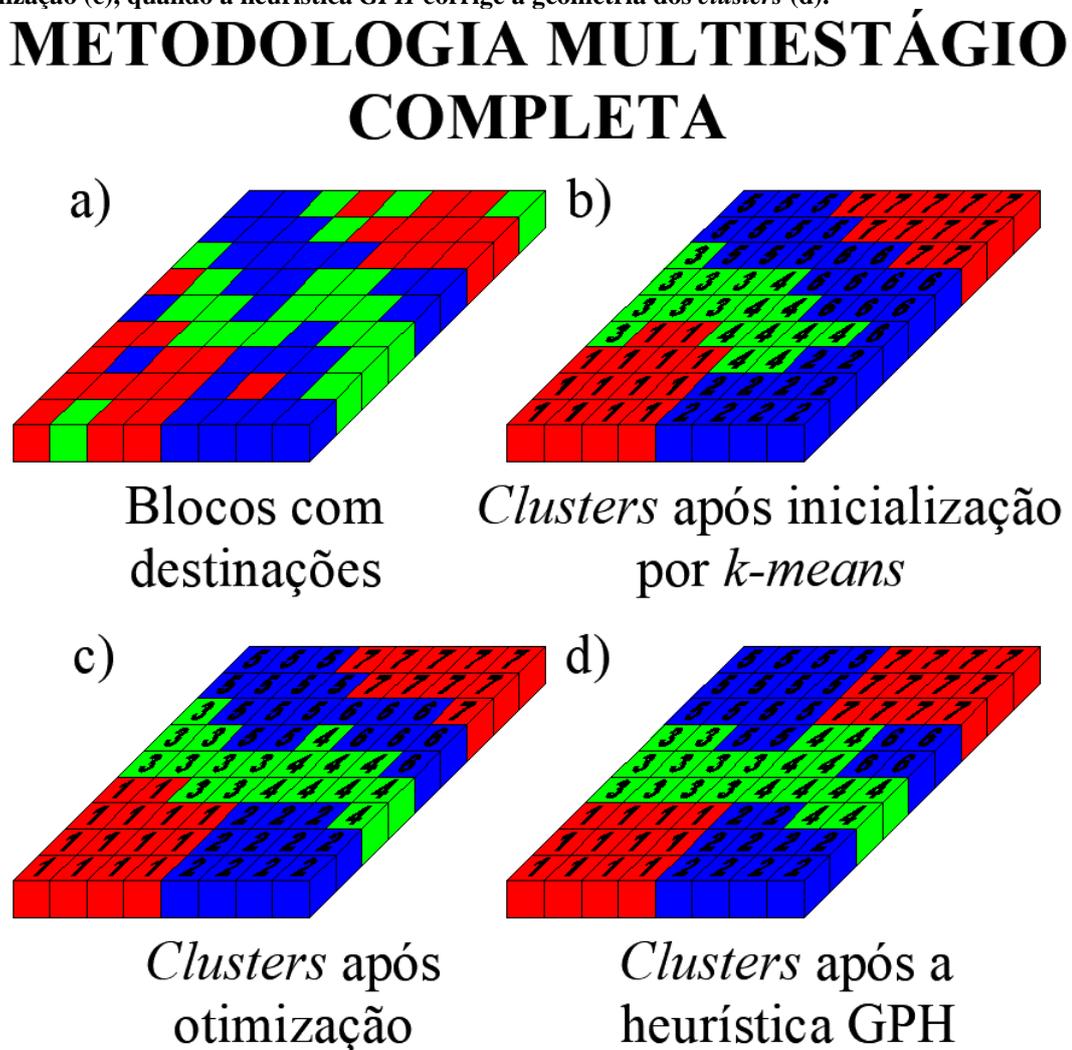
Figura 1 – Exemplo de aplicação da metodologia simplificada, consistindo na resolução direta do modelo matemático através de otimização, onde a) blocos com destinação predeterminada são agrupados em clusters b) que compartilham uma mesma destinação.



Para possibilitar que modelos maiores fossem resolvidos por otimização, foi proposta uma metodologia multiestágio baseada em *Constraint Optimization Problem (COP)*, uma das abordagens da programação por restrições. Esta metodologia tem início com a obtenção de

uma solução parcial por *k-means*. Considerando esta solução parcial como solução de dica (*hint solution*), são empregadas duas etapas de otimização, onde a primeira ajusta os parâmetros do modelo e obtém uma solução parcial melhor, enquanto a segunda explora o espaço de soluções vizinho à *hint solution* em busca de uma solução que seja um ótimo local. Finalmente, foi proposta a heurística de propagação geométrica (*Geometric Propagation Heuristic, GPH*) para refinar e corrigir falhas na geometria dos *mining cuts* gerados durante a otimização. A Figura 2 apresenta um exemplo da metodologia multiestágio completa considerando somente a destinação dos blocos como critério de similaridade.

Figura 2 - Exemplo de aplicação da metodologia multiestágio completa. Inicialmente, a) blocos com destinação predeterminada são agrupados por *k-means* (b). Em seguida, esta solução parcial é submetida à otimização (c), quando a heurística *GPH* corrige a geometria dos *clusters* (d).

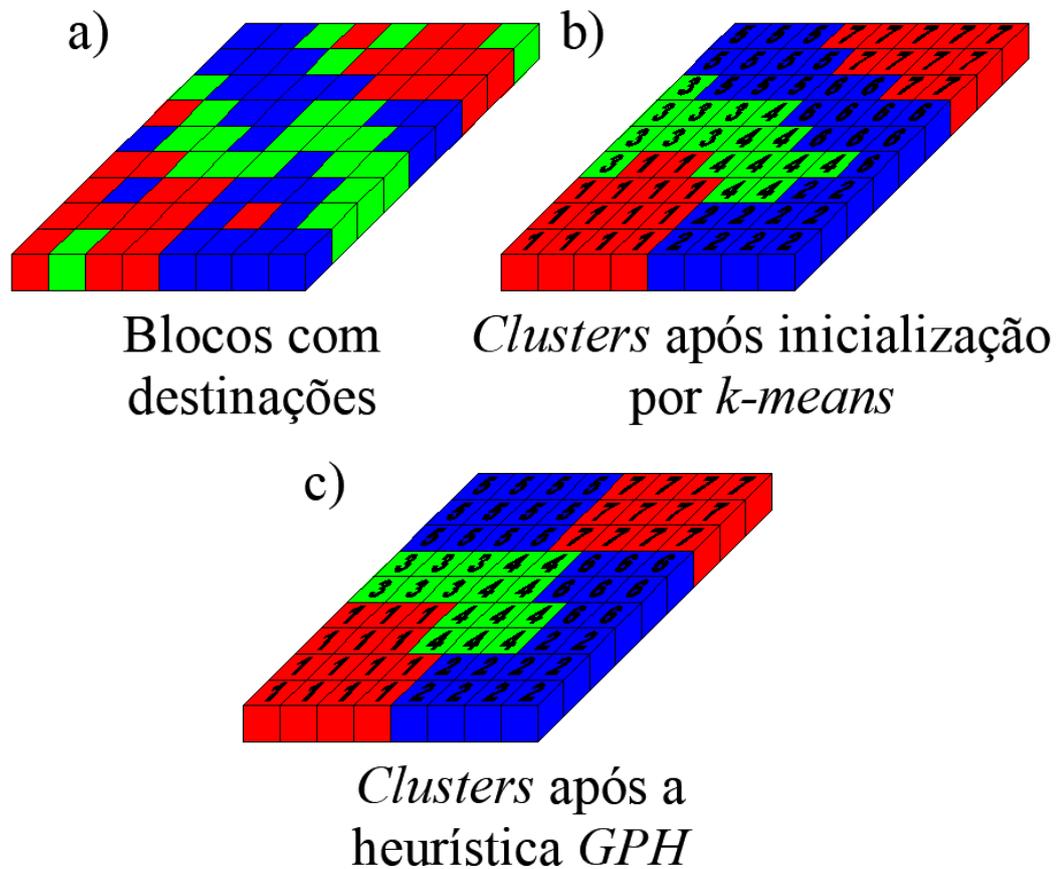


Foi proposta ainda uma metodologia multiestágio reduzida para verificar se é possível obter soluções com *mining cuts* satisfatórios mesmo sem o emprego das etapas de otimização.

Conforme apresentado na Figura 3, esta metodologia reduzida emprega somente o algoritmo *k-means* para obter uma solução inicial e corrige suas fronteiras por meio da heurística *GPH*.

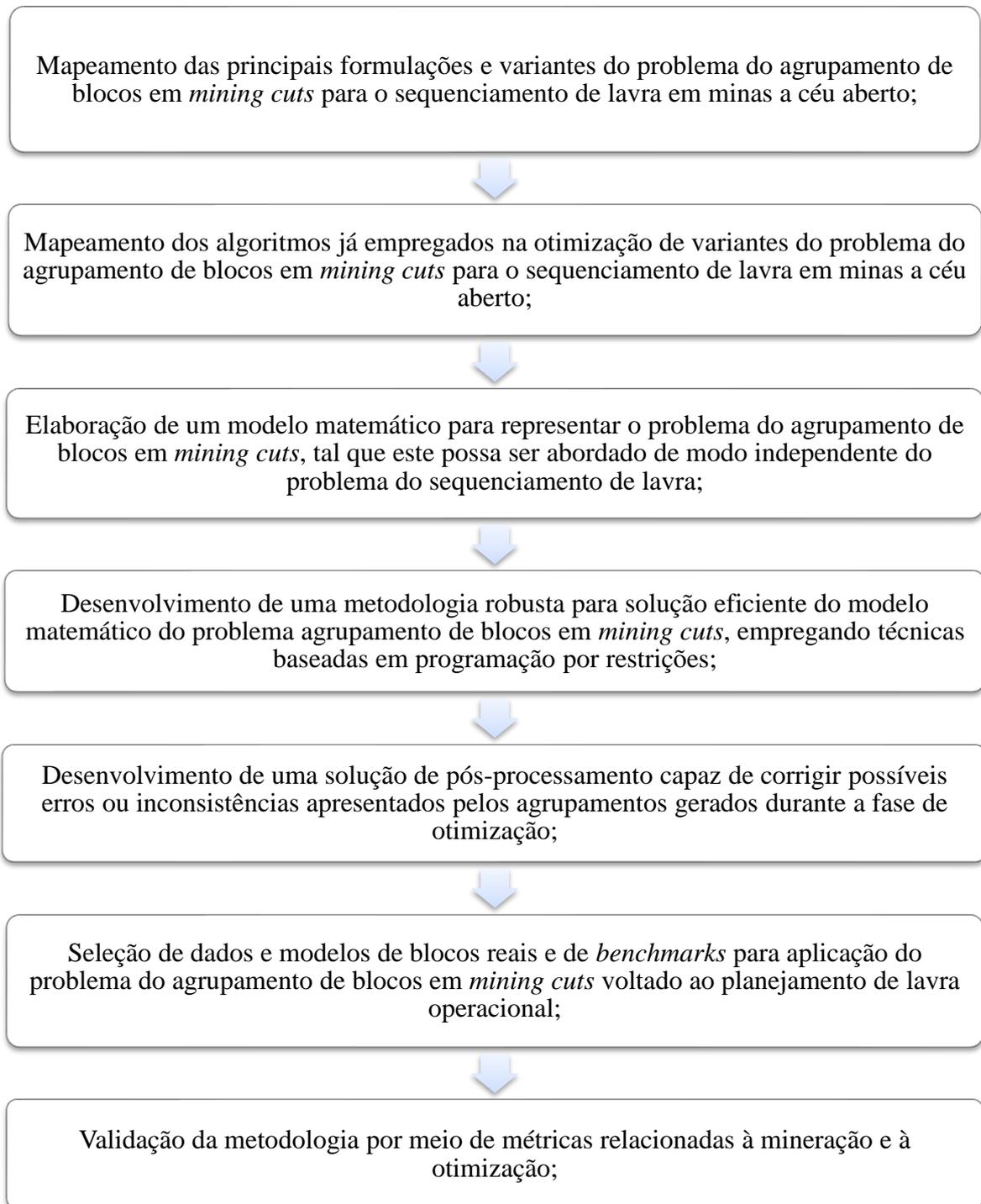
Figura 3 - Exemplo de aplicação da metodologia multiestágio reduzida, onde a) blocos com destinação predeterminada são agrupados por *k-means* (b) e os *clusters* têm suas geometrias refinadas pela heurística *GPH* (c).

METODOLOGIA MULTIESTÁGIO REDUZIDA



Portanto, para a consecução deste estudo, foram definidas as etapas apresentadas na Figura 4:

Figura 4 – Etapas previstas durante a elaboração deste estudo.



2 REVISÃO DO ESTADO DA ARTE

2.1 OTIMIZAÇÃO MATEMÁTICA E PLANEJAMENTO DE LAVRA

O planejamento de lavra pode ser subdividido em dois problemas matemáticos principais: o problema da cava final (*Ultimate Pit Problem*) e o problema do sequenciamento de lavra (*Production Scheduling Problem*), tendo sido elaboradas desde os anos 1960 diversas formulações e algoritmos para abordá-los, podendo a abordagem ser individualizada ou conjunta, como o sequenciamento direto de blocos (*Direct Block Sequencing, DBS*) (MORALES *et al.*, 2015; MARIZ; SOOFASTAEI, 2022). Antes de abordar os diferentes problemas do planejamento de lavra e suas formulações matemáticas, entretanto, deve-se aludir ao que seria um problema de otimização matemática.

2.1.1 Forma geral de um problema de otimização matemática

Um problema de otimização matemática tem como forma geral o seguinte Problema 1, contemplando as Equações (2.1) e (2.2):

$$\text{Minimizar } \mathbf{f}_0(\mathbf{x}) \tag{2.1}$$

$$\text{Sujeito a } \mathbf{f}_i(\mathbf{x}) \leq \mathbf{b}_i \text{ para } i = 1, \dots, m. \tag{2.2}$$

onde:

o vetor $\mathbf{x} = (x_1, \dots, x_n)$ é a variável de otimização do problema;

a função $\mathbf{f}_0 : \mathbf{R}^n \rightarrow \mathbf{R}$ é a função objetivo;

as funções $\mathbf{f}_i : \mathbf{R}^n \rightarrow \mathbf{R}$, $i = 1, \dots, m$, são as funções de restrição (desigualdade);

as constantes $\mathbf{b}_1, \dots, \mathbf{b}_m$ são os limites ou fronteiras para as restrições.

Um vetor \mathbf{x}^* é chamado ótimo, ou solução do Problema 1, caso apresente o menor valor objetivo em 2.1 entre todos os vetores que satisfaçam as restrições: para qualquer \mathbf{z} com $\mathbf{f}_1(\mathbf{z}) \leq \mathbf{b}_1, \dots, \mathbf{f}_m(\mathbf{z}) \leq \mathbf{b}_m$, tem-se $\mathbf{f}_0(\mathbf{z}) \geq \mathbf{f}_0(\mathbf{x}^*)$. O Problema 1 é uma abstração do problema de fazer a melhor escolha possível de um vetor em \mathbf{R}^n a partir de um conjunto de candidatos. A variável \mathbf{x} representa a escolha feita; as restrições $\mathbf{f}_i(\mathbf{x}) \leq \mathbf{b}_i$ em 2.2 representam requisitos ou especificações que limitam as opções possíveis, e o valor objetivo $\mathbf{f}_0(\mathbf{x})$ em 2.1 representa o custo de escolha de \mathbf{x} (assim como $-\mathbf{f}_0(\mathbf{x})$ seria o ganho de escolha de \mathbf{x}). A solução do Problema 1 corresponde à escolha que tem custo mínimo (ou

ganho máximo) dentre todas as opções que atendem às restrições (BOYD; VANDENBERGHE, 2009).

2.1.2 O problema do sequenciamento de lavra

O problema do sequenciamento de lavra em minas a céu aberto pode ser definido como a determinação da sequência de remoção de blocos que irá maximizar o fluxo de caixa descontado (VPL) do projeto, sujeito a uma variedade de restrições físicas e econômicas, geralmente relacionadas à precedência, ângulo de talude, capacidade dos equipamentos de lavra e beneficiamento, controle de teores, largura mínima operacional, dentre outras, podendo ser incorporadas ao problema tantas restrições quanto se julgue necessário. Vários autores, inspirados em problemas de pesquisa operacional, desenvolveram modelos *LP* e Programação Inteira (*Integer Programming, IP*) para resolver este problema, mas a combinação das restrições inerentes ao sequenciamento e lavra resultam um problema *NP-hard*, como supracitado, cuja solução ótima não pode ser obtida em um tempo de cálculo aceitável (polinomial), de modo que simplificações e o uso de metaheurísticas são comumente empregadas na busca por soluções aproximadas (CACCETTA; HILL, 2003; ALIPOUR *et al.*, 2020; MARIZ; SOOFASTAEI, 2022).

Osanloo, Gholamnejad e Karimi (2008) formularam o problema do sequenciamento de lavra como o Problema 2, contemplando as Equações (2.3) a (2.9), baseados na formulação *LP* de Johnson (1969):

$$\text{Maximizar} \quad Z = \sum_{t=1}^T \sum_{m=1}^M \sum_{n=1}^N C_i^{tm} \cdot TB_i \cdot x_i^{tm} \quad (2.3)$$

$$\text{Sujeito a} \quad G_{min}^{tm} \leq \left(\frac{\sum_{i=1}^N g_i \cdot TB_i \cdot x_i^{tm}}{\sum_{i=1}^N TB_i \cdot x_i^{tm}} \right) \leq G_{max}^{tm} \text{ para } t =$$

$$1, \dots, T \text{ e } m = 2, \dots, M.$$

$$PC_{min}^{tm} \leq \sum_{i=1}^N TB_i \cdot x_i^{tm} \leq PC_{max}^{tm} \text{ para } t = 1, \dots, T \text{ e } m =$$

$$2, \dots, M. \quad (2.5)$$

$$MC_{min}^t \leq \sum_{i=1}^N \sum_{m=1}^M TB_i \cdot x_i^{tm} \leq MC_{max}^t \text{ para } t = 1, \dots, T. \quad (2.6)$$

$$\sum_{t=1}^T \sum_{m=1}^M x_i^{tm} = 1 \text{ para } i = 1, \dots, N. \quad (2.7)$$

$$\sum_{m=1}^M x_b^{tm} - \sum_{r=1}^t \sum_{m=1}^M x_l^{rm} \leq 0 \text{ para } t = 1, \dots, T, b =$$

$$1, \dots, N \text{ e } \forall l \in \Gamma_b. \quad (2.8)$$

$$0 \leq x_i^{tm} \leq 1 \text{ para } t = 1, \dots, T, b = 1, \dots, N \text{ e } m = 1, \dots, M. \quad (2.9)$$

onde:

T equivale ao número máximo de períodos;

N equivale ao número máximo de blocos a serem sequenciados;

i representa o índice do bloco ($i = 1, \dots, N$);

C_i^{tm} equivale ao VPL resultante de minerar o peso unitário de rocha do bloco i durante o período t considerando o processo m ;

x_i^{tm} equivale à proporção do bloco i a ser lavrada no período t e enviada ao processo m ;

g_i equivale ao teor médio do bloco i ;

TB_i equivale à massa total no bloco i ;

G_{min}^{tm} e G_{max}^{tm} equivalem aos teores médios mínimo e máximo do material m enviado ao beneficiamento no período t ;

PC_{min}^{tm} e PC_{max}^{tm} equivalem às capacidades mínima e máxima de beneficiamento do material m enviado em qualquer período;

MC_{min}^t e MC_{max}^t representam as capacidades mínima e máxima de lavra em qualquer período;

b representa o índice do bloco disponível para lavra no período t ;

m equivale ao tipo de rocha ou rota de processo (*i.e.* estéril = 1, minério = 2, ..., M);

l equivale ao contador para $l \in \Gamma_b$;

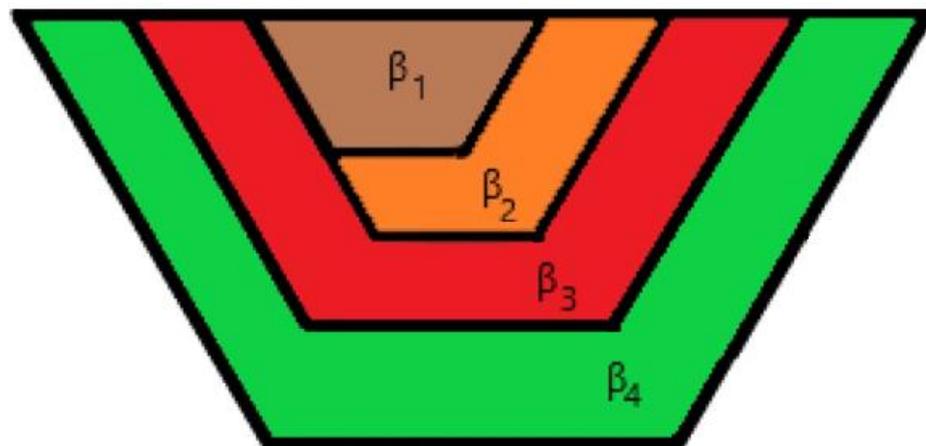
Γ_b equivale ao conjunto de índices de blocos definidos para um bloco b , ou seja, os índices dos blocos que devem ser removidos para permitir a extração do bloco b , respeitando o ângulo máximo de talude.

Devido à citada dificuldade em abordar o problema diretamente, principalmente quando trata-se de depósitos multivariados e de grandes dimensões, dotados de milhões de blocos e dezenas de períodos a serem sequenciados, ainda hoje a capacidade computacional disponível limita a formulação direta. Destarte, geralmente se decompõe o problema do sequenciamento de lavra multiperíodo em um *master problem* e define-se um conjunto de subproblemas, tal qual proposto na decomposição de Dantzig-Wolfe, onde cada subproblema é então resolvido como um problema de período único, possuindo as mesmas características do problema da definição da cava final. Após solucionar todos os subproblemas, resolver o *master problem* é relativamente simples, embora Osanloo, Gholamnejad e Karimi (2008) afirmem que o método de Johnson (1969) seja capaz de gerar resultados ótimos para cada período individualmente, mas não para abordar de forma exata o problema da cava final

(DANTZIG; WOLFE, 1960; JOHNSON, 1969; OSANLOO; GHOLAMNEJAD; KARIMI, 2008).

Anos antes da formulação de Johnson (1969), Lerchs e Grossmann (1965) já haviam sugerido ainda em sua publicação seminal um terceiro algoritmo, propondo um método para abordar o sequenciamento de lavra de longo prazo baseado no conceito de parametrização, introduzindo o conceito das cavas aninhadas. Ao se considerar o modelo não descontado e variar o valor econômico de cada bloco i de C_i para $(C_i - \lambda)$ onde $\lambda \geq 0$, uma sequência crescente de valores de λ dá origem a um conjunto de cavas aninhadas. Em outras palavras, as cavas aninhadas são cavas consecutiva geradas à medida que se reduz o retorno econômico dos blocos (fator de retorno, *revenue factor*) na cava de modo constante, sendo definida assim a sequência de extração de uma forma gulosa (*greedy*) desde a menor cava aninhada (aquela que só é viável quando o teor é muito elevado, visto que o preço de venda foi reduzido) até a maior (aquela que é viável mesmo quando os teores são muito baixos, visto que o preço de venda foi aumentado). (LERCHS; GROSSMANN, 1965; BOND, 1995; CACCETTA; HILL, 2003; OSANLOO; GHOLAMNEJAD; KARIMI, 2008). A Figura 5 contém uma representação de quatro cavas aninhadas com fatores de retorno crescentes, tal que $\beta_1 < \beta_2 < \beta_3 < \beta_4$.

Figura 5 - Representação de quatro cavas aninhadas com fatores de retorno crescentes, tal que $\beta_1 < \beta_2 < \beta_3 < \beta_4$.



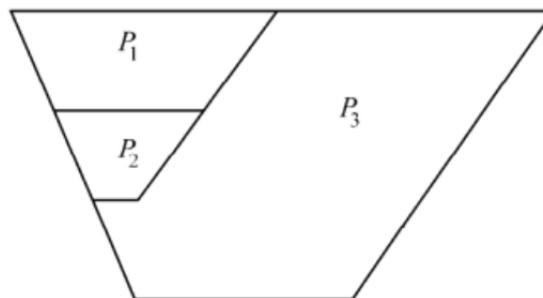
Fonte: CANESSA; MORENO; PAGNONCELLI (2020).

A abordagem tradicional do sequenciamento de lavra em minas a céu aberto consiste, ainda hoje, em definir o limite final da cava, empregar um algoritmo paramétrico para gerar cavas aninhadas, selecionar algumas destas cavas paramétricas como fases (*phases* ou *pushbacks*) para servir de referência e limitar o problema do sequenciamento de produção,

incorporar aspectos operacionais às fases matemáticas, como rampas e geometrias exequíveis e, finalmente empregar variações do Problema 2 para, enfim, sequenciar a cava.

Ampliando o conceito de *pushbacks*, são volumes gerenciáveis de material, ou cavas individuais menores, contendo suas respectivas frentes de trabalho e dinâmica de mineração, tendo como objetivo reduzir a demanda computacional dos algoritmos de sequenciamento (visto que não abordariam o modelo monolítico de blocos de uma única vez). Essa redução de tamanho permite o desenvolvimento dos planos de lavra em menores horizontes temporais (médio e curto prazo), sendo que estas etapas podem ser determinadas subjetivamente ou por alguma estratégia de otimização. Foram desenvolvidas algumas heurísticas ao longo dos anos, como as propostas por Dagdelen e François-Bongarçon (1982), Gershon (1987), Whittle (1988), Wang e Sevim (1995) e Ramazan e Dagdelen (1998). Meagher, Dimitrakopoulos e Avis (2014) elaboraram uma extensa revisão sobre a otimização de *pushbacks* e o *gap problem*, termo usado para descrever incrementos substanciais de tamanho entre *pushbacks* sucessivos, o que acarreta em cenários de cava sub ótimos em termos da maximização do VPL (GERSHON, 1983; MEAGHER; DIMITRAKOPOULOS; AVIS, 2014). A Figura 6 representa uma cava a céu aberto cujos três *pushbacks* possuem tamanhos muito distintos, ocorrendo o *gap problem*.

Figura 6 - Representação de uma cava a céu aberto contendo três *pushbacks* com tamanhos muito diferentes, ocorrendo o *gap problem*.

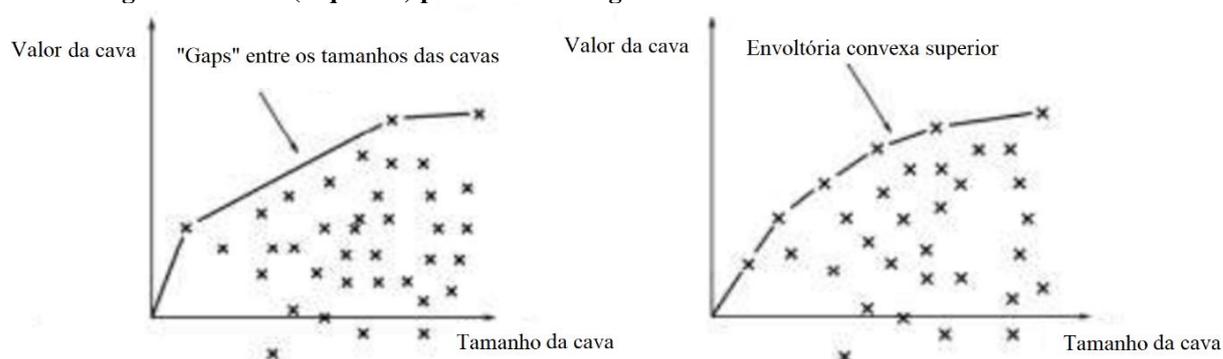


Fonte: MEAGHER; DIMITRAKOPOULOS; AVIS (2014).

Pode-se ainda citar Seymour (1995), que alterou o algoritmo de sequenciamento de Lerchs e Grossmann (1965) ao incorporar o tamanho da cava como variável de parametrização a fim de gerar cavas com valor econômico máximo relacionado a este parâmetro, retornando então as cavas obtidas na envoltória convexa superior (*upper convex hull*) de um conjunto de pontos definido em um gráfico contendo o valor econômico da cava vs. o valor do parâmetro. Quando o casco convexo superior é proporcional e existem cavas viáveis nele ou próximas a ele, pode-se selecionar *pushbacks* que não sofram com o *gap*

problem; entretanto, se as cavas do casco convexo superior estiverem distantes umas das outras em termos de tamanho, conseqüentemente ocorrerá este problema, já que o algoritmo não forneceu cavas com as dimensões desejadas. (LERCHS; GROSSMANN, 1965; SEYMOUR, 1995). A Figura 7 apresenta um casco convexo superior bem definido (direita) e um deficiente e com uma grande lacuna (esquerda), plotados em um gráfico de valor da cava vs. tamanho da cava.

Figura 7 – Representação de uma envoltória convexa superior bem definida (direita) e uma deficiente e com uma grande lacuna (esquerda) plotados em um gráfico de valor da cava vs. tamanho da cava.



Fonte: adaptado de SEYMOUR (1995); MEAGHER; DIMITRAKOPOULOS; AVIS (2014).

Para mais informações acerca dos métodos e algoritmos empregados na resolução do problema do sequenciamento de lavra em cavas a céu aberto, recomenda-se consultar a revisão efetuada por Mariz e Soofastaei (2022a).

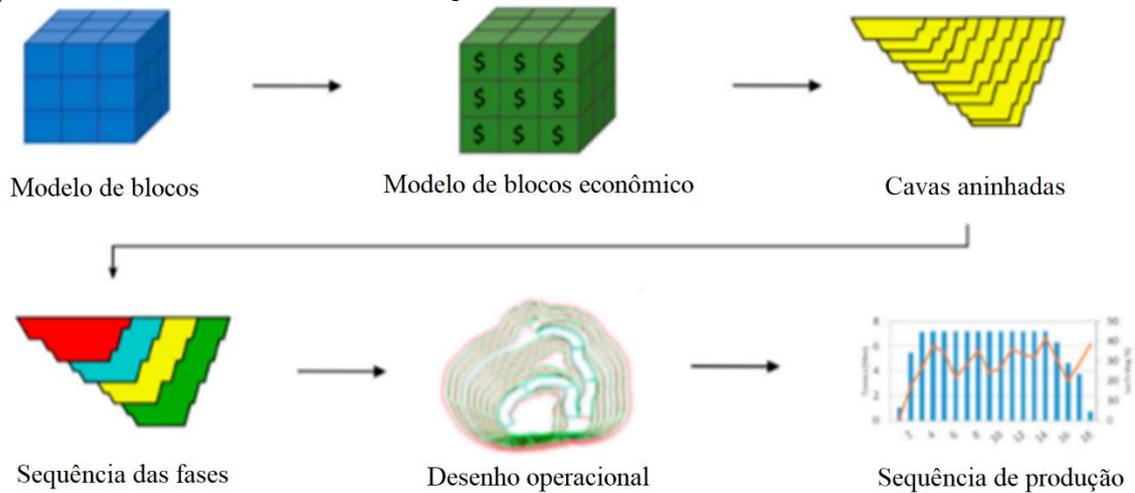
2.1.3 Horizontes temporais no sequenciamento de lavra: planejamento estratégico, tático e operacional

O sequenciamento da produção durante determinado período de tempo é conhecido como horizonte de programação, abrangendo geralmente três intervalos: o longo prazo (ou planejamento estratégico), o médio prazo (ou planejamento tático) e o curto prazo (ou planejamento operacional). O longo prazo pode variar de 20 a 30 anos, dependendo da situação, podendo mesmo contemplar toda a vida útil da mina, embora avaliações econômicas com horizontes temporais muito distantes sejam infrutíferas, devido ao mínimo retorno do VPL alcançado após algumas décadas e à improbabilidade de o plano se concretizar. Este período de 20 a 30 anos é então subdividido em vários períodos menores, de 1 a 5 anos, compondo então os horizontes de médio prazo, que buscam fornecer informações mais detalhadas que o plano estratégico e que permitam uma melhor operacionalização da cava,

definindo restrições de área e/ou informações que permitiriam a substituição da capacidade instalada, bem como possíveis períodos de *ramp-up* e consequente ampliação do maquinário. Daí em diante, o período de 1 a 5 anos da programação de médio prazo é subdividido em períodos mensais (tipicamente de 1 a 12 meses) para que uma programação ainda mais detalhada seja produzida, sendo este o horizonte de curto prazo, podendo ainda ser dividido em subperíodos semanais ou até diários, onde o objetivo principal é convergir a exequibilidade do médio prazo com as restrições operacionais de espaço, teor, equipamentos de lavra e de beneficiamento (OSANLOO; GHOLAMNEJAD; KARIMI, 2008).

A Figura 8 ilustra a metodologia tradicional aplicada ao planejamento de longo prazo, iniciando com a definição de um modelo de blocos determinístico para representar a distribuição espacial das litologias e teores dos elementos de interesse. A seguir é feita a conversão deste modelo de teores para um modelo de blocos econômico com base em uma função de transferência (LANE, 1964), a geração da cava final e das cavas aninhadas por meio de um algoritmo de parametrização (LERCHS; GROSSMANN, 1965), seguida da definição de *pushbacks*, incorporação de elementos operacionais e do sequenciamento dos blocos contidos nestas fases operacionalizadas. Deve-se salientar que mesmo que as etapas apresentadas na Figura 8 sejam eficientemente empreendidas, devido ao nível de detalhamento da informação empregada em um planejamento estratégico, adequações devem ser consideradas nos horizontes temporais de médio e curto prazo. Ao incluir informações da geologia outrora indisponíveis, é provável que seja necessária a revisão dos objetivos e das restrições empregadas nos modelos e, eventualmente, os próprios algoritmos empregados, assegurando assim que o plano de longo prazo seja exequível e busque a solução ótima.

Figura 8 - Metodologia tradicional para os planejamentos estratégico e tático em cavas a céu aberto, incluindo o uso de uma função de transferência para converter o modelo de teores em um modelo econômico, a geração de cavas aninhadas, a escolha de *pushbacks*, seguida da operacionalização e o sequenciamento dos blocos dentro de cada *pushback*.



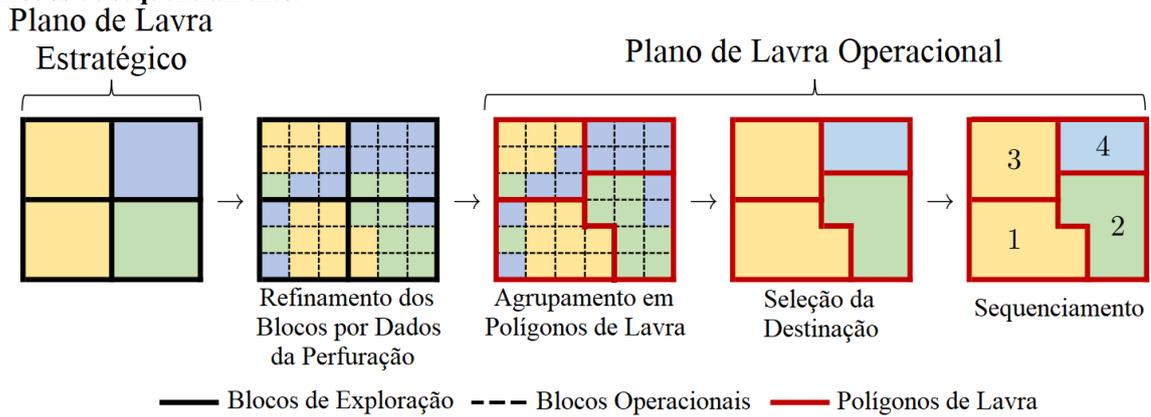
Fonte: adaptado de MORALES *et al.* (2019).

Durante a fase de operação da mina, é possível obter informações acerca da distribuição espacial das litologias e teores que não estavam disponíveis durante o planejamento de longo prazo, como mapeamento das faces dos taludes e análise do material oriundo da perfuração, fornecendo assim uma melhor estimativa do verdadeiro teor do depósito na frente de lavra trabalhada. Entretanto, deve-se considerar que se as dimensões dos blocos de curto prazo forem muito pequenas, não seria possível lavar estes blocos individualmente, razão pela qual é necessário introduzir o conceito de unidade seletiva de lavra (*Selective Mining Unit, SMU*).

As *SMUs* podem ser vistas como unidades de lavra cujas dimensões são as mínimas necessárias para propiciar a lavra seletiva pelo equipamento de lavra ora empregado em uma mina. Portanto, não deveria ser submetido ao sequenciamento de lavra uma unidade menor que uma *SMU*, sendo necessário o agrupamento dos blocos operacionais em polígonos de lavra (*mining cuts*) cujas dimensões mínimas sejam equivalentes às dimensões de uma *SMU*. Outro aspecto relevante é a distribuição espacial dos blocos sequenciados, visto que os equipamentos de esteira não devem movimentar-se frequentemente entre blocos distantes entre si, pois isto reduziria sobremaneira sua utilização física e, conseqüentemente, sua eficiência operacional. Portanto, além de gerar unidades de lavra com dimensões adequadas, a criação de *mining cuts* também resulta em planos de lavra consistentes e factíveis, produzindo frentes de lavra bem definidas que podem ser sequenciadas tanto no planejamento estratégico de lavra como no planejamento tático e operacional (NELIS; MORALES, 2022). A Figura 9

apresenta a conversão de blocos da exploração mineral em blocos de curto prazo, graças à adição de novas informações oriundas das frentes de lavra, além do agrupamento de blocos em polígonos de lavra.

Figura 9 – Esquema representando a conversão de blocos da exploração mineral em blocos de curto prazo através da adição de informação oriunda das amostras de perfuração, seguido do agrupamento destes blocos em polígonos de lavra, cujas as cores representam diferentes destinos e os números representam períodos de sequenciamento.



Fonte: adaptado de NELIS; MORALES (2022).

2.2 O PROBLEMA DO AGRUPAMENTO

De maneira geral, o problema do agrupamento (ou *clustering*) é uma das tarefas fundamentais à mineração de dados ou em muitos segmentos industriais, que tem como objetivo criar partições de dados em grupos com base na similaridade de objetos, ou seja, fazendo com que todos os objetos dentro de determinado *cluster* sejam semelhantes entre si enquanto os objetos agrupados nos demais *clusters* sejam diferentes destes. (BAGIROV; KARMITSA; TAHERI, 2020). Uma notação simplificada de um problema do agrupamento sem restrições é apresentada no Problema 3, contemplando as Equações (2.10) a (2.16):

$$A = \{a_1, \dots, a_m\}, a_i \in \mathbb{R}^n, i = 1, \dots, m \quad (2.10)$$

$$A_j \neq \mathbf{0}, \forall j = 1, \dots, k \quad (2.11)$$

$$A_j \cap A_q = \mathbf{0}, \forall j, q = 1, \dots, k, j \neq q \quad (2.12)$$

$$A = \bigcup_{j=1}^k A_j \quad (2.13)$$

$$m_j \geq 1 \quad (2.14)$$

$$x_j \in \mathbb{R}^n, j = 1, \dots, k \quad (2.15)$$

$$\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_k) \in \mathbb{R}^{nk} \quad (2.16)$$

onde:

A é um conjunto finito de pontos em um dado espaço n -dimensional (2.10);

$\mathbf{a}_i = \mathbf{1}, \dots, \mathbf{m}$ são instâncias (pontos), onde cada instância possui n atributos;

A_j representa as partições dos dados, ou seja, os *clusters*;

m_j representa o número de pontos em cada *clusters*;

\mathbf{x}_j está associado aos centros dos *clusters* A_j como seus representativos;

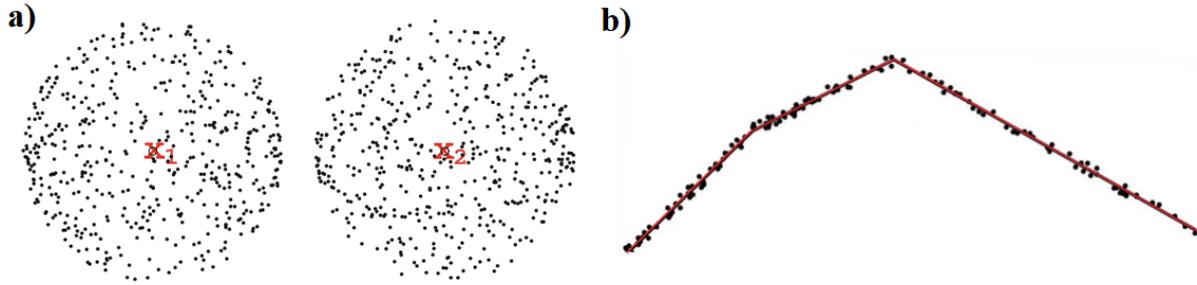
\mathbf{x} representa a coleção destes centros.

Com isso, nenhum *cluster* estará vazio (2.11) e a soma de todos os pontos distribuídos entre os *clusters* equivale ao próprio domínio A (2.13). O número de *clusters* (k) é um parâmetro fundamental aos algoritmos de agrupamento, e espera-se que este valor seja muito menor que a quantidade de instâncias (m). Deve-se ressaltar, entretanto, que esta é uma formulação de um *hard clustering*, ou seja, cada instância pode pertencer somente a um único *cluster* (2.12), como no caso do agrupamento de blocos em *mining cuts*. Em uma formulação *soft clustering*, como naquelas que emprega a lógica *Fuzzy*, uma instância pode pertencer ao mesmo tempo a mais de um *clusters*, mas este problema foge ao tema aqui proposto (BAGIROV; KARMITSA; TAHERI, 2020).

2.2.1 Medidas de similaridade entre agrupamentos

A similaridade entre as instâncias agrupadas em *clusters* pode ser medida por meio de uma função de correspondência ou similaridade, escolhida com base no representante do *cluster* (centros, por exemplo) e nos atributos contidos no conjunto de dados, sendo esta escolha um dos principais fatores na determinação da complexidade do problema do agrupamento. As medidas de similaridade para *clusters* cujos representantes são centros (pontos) ou hiperplanos são diferentes, bem como conjuntos de dados que contêm apenas instâncias com números são mensuradas diferentemente dos que contêm instâncias numéricas e categóricas ou apenas categóricas. No entanto, todos os algoritmos de agrupamento envolvem algum tipo de processo para medir a similaridade dos objetos, de modo que quaisquer dois pontos pertencentes ao mesmo *cluster* devem ser mais semelhantes entre si do que quaisquer dois pontos de dois *clusters* diferentes (BAGIROV; KARMITSA; TAHERI, 2020). A Figura 10 apresenta um exemplo em que a) os dados foram agrupados em dois *clusters* de formas circulares e b) em três hiperplanos.

Figura 10 - Agrupamento de dados em a) dois *clusters* circulares e b) em três hiperplanos.



Fonte: BAGIROV; KARMITSA; TAHERI (2020).

Considerando que $\mathbf{b}, \mathbf{c} \in \mathbb{R}^n$ são quaisquer pontos do domínio \mathbf{A} e que $\overline{\mathbb{R}_+} = \{\mathbf{r} \in \mathbb{R}: \mathbf{r} \geq \mathbf{0}\}$, uma função $\mathbf{d}: \mathbf{A} * \mathbf{A} \rightarrow \overline{\mathbb{R}_+}$ pode ser chamada de função ou medida de similaridade se satisfizer as condições em que $\mathbf{d}(\mathbf{b}, \mathbf{c}) = \mathbf{0}$ se e somente si $\mathbf{b} = \mathbf{c}$ e $\mathbf{d}(\mathbf{b}, \mathbf{c}) = \mathbf{d}(\mathbf{c}, \mathbf{b})$ para todo $\mathbf{b}, \mathbf{c} \in \mathbf{A}$. Isto posto, tem-se que um ponto \mathbf{b} é similar a si mesmo (não há dissimilaridade) e que a similaridade é simétrica. Tomando como referência atributos numéricos e centros de *clusters* como seus elementos representativos, Bagirov, Karmitsa e Taheri (2020) listam as seguintes variantes das normas gerais de Minkowski como possíveis medidas de similaridade, baseadas na equação $\mathbf{d}_p(\mathbf{b}, \mathbf{c}) = (\sum_{i=1}^n |\mathbf{b}_i - \mathbf{c}_i|^p)^{\frac{1}{p}}$, em que $p \in (0, \infty)$, mas somente quando $p \geq 1$ esta medida é propriamente uma função de distância:

1. $\mathbf{d}_1(\mathbf{b}, \mathbf{c}) = \sum_{i=1}^n |\mathbf{b}_i - \mathbf{c}_i|$, quando $p = 1$ e a medida de similaridade baseia-se na norma- L_1 , também conhecida como *city block* ou norma de Manhattan;
2. Baseada na norma- L_2 , quando $p = 2$, a medida de similaridade baseia-se na distância Euclidiana, sendo geralmente empregada em problemas de agrupamento o quadrado da distância Euclidiana, como $\mathbf{d}_2(\mathbf{b}, \mathbf{c}) = \sum_{i=1}^n (\mathbf{b}_i - \mathbf{c}_i)^2$; nestes casos, os centros dos *clusters* são chamados de centroides;
3. Por fim, a medida de similaridade baseada na norma- L_∞ , quando $p = \infty$, dá-se por $\mathbf{d}_\infty(\mathbf{b}, \mathbf{c}) = \max_{i=1, \dots, 1n} |\mathbf{b}_i - \mathbf{c}_i|$, também conhecida como norma de Chebyshev (BAGIROV; KARMITSA; TAHERI, 2020).

Existem diversas outras medidas de similaridade, embora Bagirov, Karmitsa e Taheri (2020) ressaltem que ao longo dos últimos 60 anos, os esforços concentraram-se em técnicas que empreguem a distância Euclidiana ao quadrado.

2.2.2 Medidas de similaridade em polígonos de lavra por Tabesh e Askari-Nasab (2011, 2013)

Visto que as normas gerais de Minkowski não contemplariam a similaridade entre variáveis categóricas, Tabesh e Askari-Nasab (2011) indicam que a codificação binária destas variáveis não permitiria levar em conta as semelhanças entre as categorias, bem como o uso do coeficiente de pareamento simples (*Simple Matching Coefficient*), que compara os atributos categóricos e considera uma resposta binária para valores iguais ou não, também falharia em considerar as semelhanças entre as categorias. Conseqüentemente, Askari-Nasab, Tabesh e Badiozamani (2010) empregaram a ideia da hierarquia de distâncias para poder calcular a semelhanças entre variáveis categóricas, técnica na qual a variável categórica é demonstrada em uma árvore ponderada e a distância entre dois valores é calculada a partir das ligações entre as duas ramificações. Destarte, através da implementação de uma calibração de pesos para cada variável categórica e da implementação de uma função de penalidade adotada para esses atributos, o valor de similaridade entre dois blocos $\mathbf{b}, \mathbf{c} \in \mathbf{A}_j$ poderia ser definido como $S_{b,c} = R_{b,c} C_{b,c} / \tilde{D}_{b,c}^{W_D} \tilde{G}_{b,c}^{W_G}$, onde $R_{b,c}$ é a penalidade atribuída aos blocos com diferentes litologias, $C_{b,c}$ é a penalidade atribuída a blocos não localizados acima do mesmo *cluster* (considerando uma otimização em múltiplas bancadas), $\tilde{D}_{b,c}$ representa a distância normalizada entre os blocos \mathbf{b} e \mathbf{c} ($D_{b,c}$ seria a distância Euclidiana entre eles) e $\tilde{G}_{b,c}$ representa a diferença normalizada de teores entre esses blocos (ASKARI-NASAB; TABESH; BADIOZAMANI, 2010; TABESH; ASKARI-NASAB, 2011).

Mesmo que a otimização se dê em múltiplas bancadas, como a determinação de *mining cuts* se dá em cada bancada individualmente, a coordenada Z de cada bloco deve ser desconsiderada, de modo que a matriz de distâncias para cada bancada se dá por $D_{b,c} = [(X_b - X_c)^2 + (Y_b - Y_c)^2]^{1/2}$, sendo estas distâncias normalizadas por meio da divisão de cada valor pelo valor máximo da matriz, $\tilde{D}_{b,c} = D_{b,c} / D_{max}$. O fator de distância normalizado então é elevado ao peso W_D , onde pesos mais elevados têm a capacidade de criar *clusters* mais circulares. Uma abordagem semelhante dá-se em relação à diferença entre os teores, que deve considerar o elemento de maior interesse caso seja um banco de dados multivariado, onde $G_{b,c} = (G_b - G_c)^2$ quando $G_b \neq G_c$ ou $G_{b,c} = \varepsilon$ caso contrário, sendo ε um valor muito pequeno para evitar números incomparáveis (infinitos) nos índices de similaridade. Esta “distância Euclidiana dos teores” pode então ser normalizada por meio da equação $\tilde{G}_{b,c} = G_{b,c} / G_{max}$, de modo que o fator de distância de teores normalizado então é elevado ao peso

W_G , onde pesos mais elevados têm a capacidade de criar *clusters* que aparentemente seguem a distribuição de teor no corpo de minério. O fator de similaridade $R_{b,c}$, referente às litologias da bancada, considera um valor de penalidade de $r \in [0, 1]$ se dois blocos tiverem litologias diferentes 1 caso contrário, e embora seja chamado de penalidade, valores mais baixos devem ser considerados para r se uma maior eficácia for desejada quanto à valorização da litologia no cálculo de similaridade. Finalmente, o fator $C_{b,c}$ é calculado ao se atribuir um valor $c \in [0, 1]$ para a dissimilaridade entre dois blocos que estão acima de *clusters* distintos (em baixo da bancada) ou 1 caso contrário (TABESH; ASKARI-NASAB, 2011).

Em Tabesh e Askari-Nasab (2013), por sua vez, algumas variantes desta medida de similaridade são apresentadas, como em $S_{b,c} = 1/\tilde{D}_{b,c}$, quando somente a distância entre os blocos é considerada, $S_{b,c} = T_{b,c}/\tilde{D}_{b,c}^{W_D} \tilde{G}_{b,c}^{W_G}$, sendo $T_{b,c}$ a penalidade aplicada aos blocos que não forem enviados para a mesma destinação onde, tal qual nas penalidades anteriores, $t \in [0, 1]$ e a atribuição de valores menores a este fator ressalta seu impacto no cálculo de similaridade. Os autores apresentaram ainda a equação $S_{b,c} = T_{b,c}/\tilde{D}_{b,c}^{W_D} \tilde{G}_{b,c}^{W_G} \tilde{M}_{b,c}^{W_M}$, onde $\tilde{M}_{b,c}$ representa uma medida de similaridade baseada na direção da lavra, implementada por meio da definição por parte do usuário de um ponto inicial e um final do vetor direção preferencial. Visto que o índice de similaridade de direção para um bloco b é calculado como $M_b = \text{sign} \left((M_b^1)^2 - (M_b^2)^2 \right) * \sqrt{|(M_b^1)^2 - (M_b^2)^2|}$, em que M_b^1 se refira à distância Euclidiana deste bloco b ao ponto inicial do vetor de direção e M_b^2 se refira à sua distância até o ponto final do vetor de direção, a distância de cada bloco a esses pontos é calculada e esse valor é então usado como medida de similaridade, o que pode resultar em *clusters* perpendiculares à direção de mineração. O sinal (*sign*) deve ser preservado para fins de distinção entre o ponto inicial e o ponto final do vetor de direção, de modo que o índice de similaridade de direção M_b pode ser usado para determinar a similaridade entre os blocos b e c , sendo combinado com outras medidas de similaridade e associado a um peso W_M especificado (TABESH; ASKARI-NASAB, 2013).

Tabesh e Askari-Nasab (2013) indicam também que se deve levar em consideração que a inserção de muitos parâmetros diminui a qualidade do agrupamento, já que seus efeitos podem se neutralizar, citando como exemplo que devem-se evitar empregar simultaneamente os fatores de similaridade relativos aos teores e aos destinos, embora os próprios autores tenham sugerido uma equação como esta, uma vez que geralmente estes estão correlacionados. Outra modificação proposta na medida de similaridade para agrupamento de

blocos em *mining cuts* visa seu emprego em depósitos estratificados, como areias betuminosas e carvão, cuja lavra se dá geralmente em tiras (*strips*) de uma mesma litologia e com largura pré-determinada, de modo que o cálculo do fator \mathbf{M}_b supracitado poderia ser substituído por $\mathbf{M}_b = \sqrt{|(\mathbf{M}_b^1)^2 + (\mathbf{M}_b^2)^2|}$, onde ao invés de especificar um vetor de direção, os dois pontos especificados pelo usuário representam o ponto inicial da operação de lavra (TABESH; ASKARI-NASAB, 2013).

2.2.3 Modelos de otimização na análise de agrupamentos

Continuando com a mesma notação empregada no Tópico 2.2.2, um modelo de otimização do problema do agrupamento pode ser formulado como um *MILP* ao considerar um peso de associação w_{ij} do ponto $\mathbf{a}_i \in \mathbf{A}$ com o *cluster* \mathbf{A}_j equivalente a 1 se o ponto \mathbf{a}_i for alocado ao *cluster* \mathbf{A}_j , ou equivalente a 0 caso contrário, conforme o Problema 4 a seguir, contemplando as Equações (2.17) a (2.20):

$$\text{Minimizar} \quad \zeta_k(\mathbf{x}, \mathbf{w}) \quad (2.17)$$

$$\text{Sujeito a} \quad w_{ij} \in \{0, 1\} \text{ para todo } i = 1, \dots, m, j = 1, \dots, k \quad (2.18)$$

$$\sum_{j=1}^k w_{ij} = 1 \text{ para todo } i = 1, \dots, m \quad (2.19)$$

$$\mathbf{x}_j \in \mathbb{R}^n, j = 1, \dots, k \quad (2.20)$$

onde $\zeta_k(\mathbf{x}, \mathbf{w}) = \frac{1}{m} \sum_{i=1}^m \sum_{j=1}^k w_{ij} \mathbf{d}_p(\mathbf{x}_j, \mathbf{a}_i)$ é chamada de k -ésima função de clusterização, em que $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_k) \in \mathbb{R}^{nk}$ e $\mathbf{w} = (\mathbf{w}_1, \dots, \mathbf{w}_k) \in \mathbb{R}^m$, função objetivo que converte o problema *MILP* em Programação Não Linear Inteira Mista (*Mixed Integer Nonlinear Programming, MINLP*). Esta formulação contém mk variáveis inteiras (binárias) w_{ij} , onde $i = 1, \dots, m$ e $j = 1, \dots, k$, e nk variáveis contínuas $\mathbf{x}_j \in \mathbb{R}^n$, onde $j = 1, \dots, k$. A função objetivo ζ_k é convexa para $k = 1$ e não convexa para $k > 1$, configurando em um *hard clustering* se forem usadas medidas de similaridade \mathbf{d}_1 ou \mathbf{d}_∞ . No caso da métrica \mathbf{d}_1 , os centros $\mathbf{x}_j, j = 1, \dots, k$ são computados como as medianas dos *clusters*, tornando a formulação do problema em um *IP* (BAGIROV; KARMITSA; TAHERI, 2020).

Já o modelo de otimização não suave (*Nonsmooth Optimization Model*) para o problema do agrupamento pode ser formulado conforme o Problema 5, contemplando as Equações (2.21) e (2.22):

$$\text{Minimizar} \quad \mathbf{f}_k(\mathbf{x}) \quad (2.21)$$

$$\text{Sujeito a} \quad \mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_k) \in \mathbb{R}^{nk} \quad (2.22)$$

onde $\mathbf{f}_k(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^m \min_{j=1, \dots, k} \mathbf{d}_p(\mathbf{x}_j, \mathbf{a}_i)$ é a k -ésima função de *cluster*. Este problema contém nk variáveis contínuas $\mathbf{x}_j \in \mathbb{R}^n, j = 1, \dots, k$, independentemente do número de instâncias. A função objetivo \mathbf{f}_k é convexa para $k = 1$ e não convexa para $k > 1$, também configurando em um *hard clustering* se forem usadas medidas de similaridade \mathbf{d}_1 ou \mathbf{d}_∞ , embora a combinação entre $k > 1$ e \mathbf{d}_2 também seja um *hard clustering* (BAGIROV; KARMITSA; TAHERI, 2020).

Pode-se ainda formular o problema do agrupamento como um modelo de otimização não suave de diferença de funções convexas (*Nonsmooth Difference of Convex Functions Optimization Model*) conforme o Problema 6, contemplando as Equações (2.23) e (2.24):

$$\text{Minimizar} \quad \mathbf{f}_k(\mathbf{x}) = \mathbf{f}_{k1}(\mathbf{x}) - \mathbf{f}_{k2}(\mathbf{x}) \quad (2.23)$$

$$\text{Sujeito a} \quad \mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_k) \in \mathbb{R}^{nk} \quad (2.24)$$

onde $\mathbf{f}_{k1}(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^m \sum_{j=1}^k \mathbf{d}_p(\mathbf{x}_j, \mathbf{a}_i)$ e $\mathbf{f}_{k2}(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^m \max_{j=1, \dots, k} \sum_{s=1, s \neq j}^k \mathbf{d}_p(\mathbf{x}_j, \mathbf{a}_i)$ (BAGIROV; KARMITSA; TAHERI, 2020).

Como mencionado anteriormente, a função \mathbf{d}_p é convexa em \mathbf{x} para $p = 1, p = 2$ e $p = \infty$. Portanto, a função \mathbf{f}_{k1} como soma de funções convexas é convexa. Além disso, como o máximo de um número finito de funções convexas é convexo, a função \mathbf{f}_{k2} também é convexa como a soma dos máximos da soma das funções convexas. Para a medida de similaridade \mathbf{d}_2 , a função \mathbf{f}_{k1} seria um *smooth clustering*, ao passo que a função \mathbf{f}_{k2} é, em geral, *hard clustering*. Se $p = 1$ ou $p = \infty$, então \mathbf{f}_{k1} e \mathbf{f}_{k2} são funções *hard clustering* (BAGIROV; KARMITSA; TAHERI, 2020).

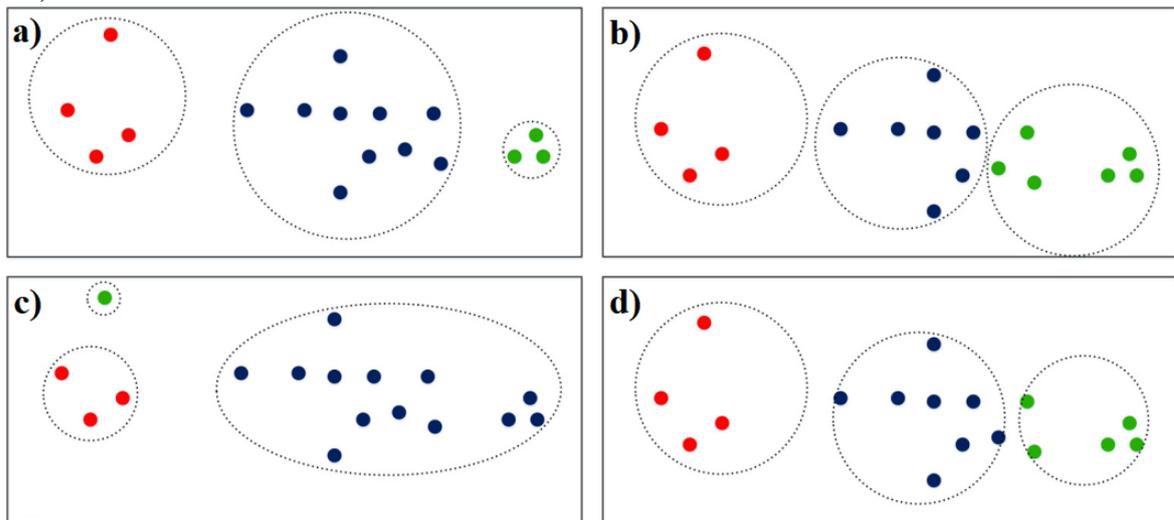
Ao considerar o problema do agrupamento como um problema de otimização, Dao, Duong e Vrain (2017) relatam ainda algumas estratégias comumente usadas considerando medidas de dissimilaridade, que seriam de certa forma o oposto das normas gerais de Minkowski apresentadas anteriormente, pois buscam quantificar as diferenças entre as instâncias. Considerando $\mathbf{d}_s(\mathbf{b}, \mathbf{c})$ uma medida de dissimilaridade entre dois pontos $\mathbf{b}, \mathbf{c} \in \mathbf{A}$, os autores citaram como estratégias:

1. A minimização do diâmetro máximo dos *clusters*, onde o diâmetro máximo de uma partição \mathbf{A}_j equivale à maior dissimilaridade entre dois objetos neste mesmo *cluster*, onde $D(\mathbf{A}) = \max_{j \in [1, k], \mathbf{b}, \mathbf{c} \in \mathbf{A}_j} (\mathbf{d}_s(\mathbf{b}, \mathbf{c}))$, critério também conhecido como

- agrupamento de conexão completa não hierárquica (*nonhierarchical complete-link clustering*);
2. A maximização da divisão mínima entre *clusters*, equivalendo à menor dissimilaridade entre dois objetos de *clusters* diferentes, onde $S(\mathbf{A}) = \min_{j < q \in [1, k], b \in A_j, c \in A_q} (d_s(\mathbf{b}, \mathbf{c}))$, critério também conhecido como agrupamento de conexão simples (*single-link clustering*);
 3. A minimização da soma de dissimilaridades dentro dos *clusters* (*Within-Cluster Sum of Dissimilarities, WCSD*), onde $WCSD(\mathbf{A}) = \sum_{j \in [1, k]} \frac{1}{2} \sum_{b, c \in A_j} d_s(\mathbf{b}, \mathbf{c})$, critério em que a dissimilaridade é geralmente medida pela distância Euclidiana ao quadrado;
 4. A minimização da soma dos quadrados dentro dos *clusters* (*Within-Cluster Sum of Squares, WCSS*), em que se computa a soma das distâncias Euclidianas quadradas entre cada objeto \mathbf{a}_i e o centróide \mathbf{m}_j do *cluster* contendo \mathbf{a}_i , onde $WCSS(\mathbf{A}) = \sum_{j \in [1, k]} \sum_{\mathbf{a}_i \in A_j} |\mathbf{a}_i - \mathbf{m}_j|^2$ (DAO; DUONG; VRRAIN, 2017).

A Figura 11 apresenta o resultado do agrupamento de um conjunto de dados por diferentes critérios, onde foi utilizado em a) um critério subjetivo, em b) um *complete-link clustering*, em c) um *single-link clustering* e em d) a métrica *WCSS*.

Figura 11 – Resultado do agrupamento de um conjunto de dados por diferentes critérios, onde foi utilizado em a) um critério subjetivo, em b) um *complete-link clustering*, em c) um *single-link clustering* e em d) a métrica *WCSS*.



Fonte: DAO; DUONG; VRRAIN (2017).

2.2.4 Métricas de avaliação da qualidade de agrupamentos

As métricas que avaliam a qualidade dos agrupamentos podem ser divididas em duas classes: aquelas que baseiam-se no conhecimento prévio de quais partições são verdadeiras, como o índice de Rand (1971), a pontuação de Fowlkes-Mallows (FOWKLES; MALLOWS, 1983), as pontuações baseadas em informações mútuas (STREHL; GOSH, 2002) e o critério V (*V-Measure*) (ROSENBERG; HIRSCHBERG, 2007), ou aquelas que não baseiam-se no conhecimento prévio das partições, como o índice Calinski-Harabasz (CALIŃSKI; HARABASZ, 1974), o índice Davies-Bouldin (DAVIES; BOULDIN, 1979) e o coeficiente da silhueta (ROUSSEEUW, 1987) (HUBERT; ARABIE, 1985; SCIKITLEARN, 2022). Visto que o problema do agrupamento de blocos em *mining cuts*, tema proposto neste estudo, não demanda o conhecimento prévio de quais partições seriam verdadeiras, somente a segunda classe foi aqui abordada.

O índice Calinski-Harabasz, também conhecido como critério da razão de variância, consiste na razão da soma da dispersão entre *clusters* e da dispersão intra-*cluster* para todos os agrupamentos, sendo a dispersão definida como a soma das distâncias ao quadrado, onde um índice mais elevado refere-se a um modelo com *clusters* mais bem definidos. Seja \mathbf{A} é um conjunto finito de pontos em um dado espaço n -dimensional subdividido em \mathbf{A}_j partições, onde $j = 1, \dots, k$, o índice Calinski-Harabasz s dá-se como $s = \text{tr}(\mathbf{B}_k) * (\mathbf{A} - k) / \text{tr}(\mathbf{W}_k) * (k - 1)$, onde $\text{tr}(\mathbf{B}_k)$ é o traço da matriz de dispersão entre grupos e $\text{tr}(\mathbf{W}_k)$ é o traço da matriz de dispersão dentro do *cluster*, ambos definidos por $\mathbf{B}_k = \sum_{j=1}^k \mathbf{m}_j (\mathbf{x}_j - \mathbf{x}_A) (\mathbf{x}_j - \mathbf{x}_A)^T$ e $\mathbf{W}_k = \sum_{j=1}^k \sum_{\mathbf{x} \in \mathbf{m}_j} (\mathbf{x} - \mathbf{x}_j) (\mathbf{x} - \mathbf{x}_j)^T$, com \mathbf{m}_j o conjunto de pontos no *cluster* \mathbf{A}_j , \mathbf{x}_j o centro do *cluster* \mathbf{A}_j , \mathbf{x}_A o centro de \mathbf{A} , e \mathbf{m}_j o número de pontos no *cluster* \mathbf{A}_j (CALINSKI; HARABASZ, 1974; SCIKITLEARN, 2022).

O índice Davies-Bouldin, por sua vez, identifica uma semelhança média entre os *clusters* \mathbf{A}_j , em que esta similaridade seria uma medida que compara a distância entre os *clusters* com seus tamanhos, onde índices baixos representam um modelo com melhor separação entre os *clusters* e sendo 0 a pontuação mais baixa possível. Ao definir o índice como a similaridade média entre cada *cluster* \mathbf{A}_j para seu *cluster* \mathbf{A}_q mais semelhante, a similaridade pode ser definida como $\mathbf{R}_{jq} = (\mathbf{s}_j + \mathbf{s}_q) / \mathbf{d}_{jq}$, em que \mathbf{s}_j representa a distância média entre cada ponto do *cluster* \mathbf{A}_j ao seu centroide \mathbf{x}_j , \mathbf{d}_{jq} equivale à distância entre os centróides \mathbf{x}_j e \mathbf{x}_q , tal que o índice Davies-Bouldin seria determinado como $\mathbf{DB} =$

$1/k \sum_{j=1}^k \max_{j \neq q} R_{jq}$. Esta é uma métrica simples, baseada apenas em recursos inerentes ao conjunto de dados, mas apresenta valores mais elevados para *clusters* convexos do que para outros conceitos de *clusters*, como os baseados em densidade (*Density-Based Spatial Clustering of Applications with Noise, DBSCAN*) (ESTER *et al.*, 1996), e o uso da distância entre centroides limita a métrica ao espaço euclidiano (DAVIES; BOULDIN, 1979; SCIKITLEARN, 2022).

O coeficiente da silhueta, finalmente, utiliza o próprio modelo na avaliação, composto por uma pontuação g , que calcula para cada amostra a distância média entre ela e todos os outros pontos da mesma classe, e outra h , que calcula sua distância média e todos os outros pontos no *cluster* mais próximo, tal que $s = (h - g)/\max(g, h)$, sendo o coeficiente de silhueta do conjunto de amostras a média do coeficiente para cada amostra. Uma pontuação mais alta se refere a um modelo com *clusters* mais bem definidos, podendo variar entre -1 para agrupamentos incorretos e 1 para agrupamentos altamente densos, onde pontuações próximas a 0 indicam *clusters* sobrepostos, mas também apresenta valores mais elevados para *clusters* convexos do que para outros conceitos de *clusters*, como os baseados em densidade (*DBSCAN*) (ROUSSEEUW, 1987; SCIKITLEARN, 2022).

2.3 PROGRAMAÇÃO POR RESTRIÇÕES E ALGORITMOS RELEVANTES

2.3.1 Heurísticas no problema do agrupamento

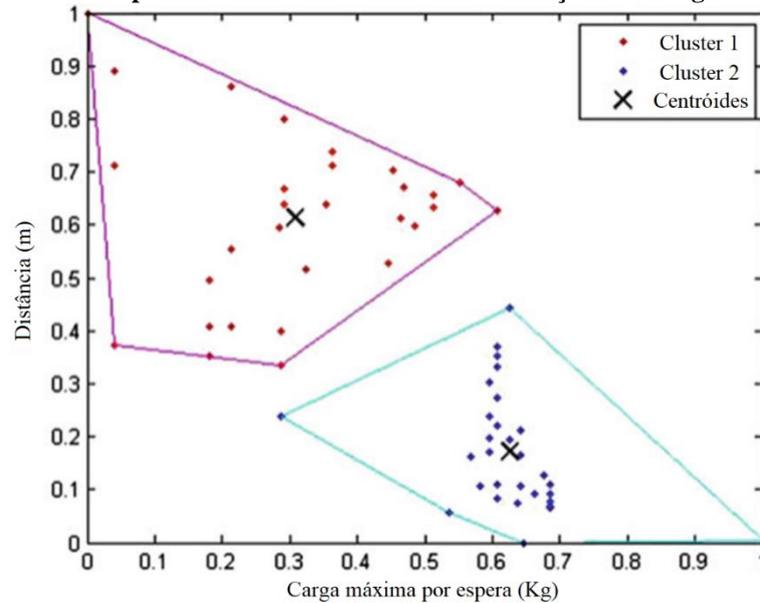
São diversas as técnicas e os algoritmos de agrupamento existentes, e independente da abordagem utilizada, um algoritmo de agrupamento deve produzir *clusters* estáveis às pequenas mudanças em seus objetos ou à adição destes, deve ser independente da ordenação inicial dos objetos, deve ser escalável e capaz de filtrar ruídos e *outliers*, deve ser capaz de definir *clusters* de diferentes formas e requerer o mínimo de conhecimento sobre os dados para determinar os parâmetros (HAN; KAMBER; PEI, 2011; AGGARWAL; REDDY, 2014; BAGIROV; KARMITSA; TAHERI, 2020).

A maioria dos algoritmos de agrupamento pode ser classificada como hierárquicos (*hierarchical*) ou particionais (*partitional*) (JAIN; MURTY; FLYNN, 1999), onde os primeiros produzem um dendrograma representando o agrupamento aninhado de padrões e níveis de similaridade, ao passo que os últimos encontram uma partição de objetos que

otimiza algum critério de agrupamento predefinido, podendo serem ainda divididos em *hard partitioning clustering*, onde cada objeto pertence a apenas um *cluster*, *soft partitioning clustering* (ou *Fuzzy*), onde cada objeto pode pertencer a mais de um *cluster*. Cada algoritmo contém suas particularidades, e não necessariamente conseguem atender a todos os critérios listados no parágrafo anterior; a forma dos *clusters* encontrados por algoritmos de agrupamento particional geralmente depende da medida de similaridade, enquanto os algoritmos *HC* não são bem dimensionados. Sobretudo, a maioria dos algoritmos de agrupamento demanda por parâmetros determinados empiricamente, sendo o número de *clusters* o mais importante, valor que não é conhecido *a priori* para muitos conjuntos de dados e deve ser fornecido pelo usuário (HAN; KAMBER; PEI, 2011; AGGARWAL; REDDY, 2014; BAGIROV; KARMITSA; TAHERI, 2020).

Os algoritmos baseados em *partitioning clustering* constroem partições de dados em que cada partição representa um *cluster*. Dentre os representantes mais populares dessa classe de algoritmos estão o algoritmo *k-means* (MACQUEEN, 1967) e suas variações: *kernel k-means* (SCHÖLKOPF; SMOLA; MÜLLER, 1998), *k-means* ponderado (HARTIGAN; WONG, 1979) e *k-means* genético (KRISHNA; MURTY, 1999; AGGARWAL; REDDY, 2014). Com base nos critérios de similaridade apresentados, um algoritmo *k-means* busca um ótimo local para o critério *WCSS*, bem como encontrar o ponto mais distante para o critério de diâmetro (GONZALEZ, 1985; DAO; DUONG; VRAIN, 2017; BAGIROV; KARMITSA; TAHERI, 2020). A Figura 12 apresenta uma aplicação do *k-means* ao desmonte de rochas com uso de explosivos, onde um conjunto de dados é subdividido em dois *clusters* a partir da distância ao local de detonação e da carga máxima por espera.

Figura 12 - Exemplo de uma aplicação do *k-means* ao desmonte de rochas, onde um conjunto de dados é subdividido em dois *clusters* a partir da distância ao local de detonação e da carga máxima por espera.

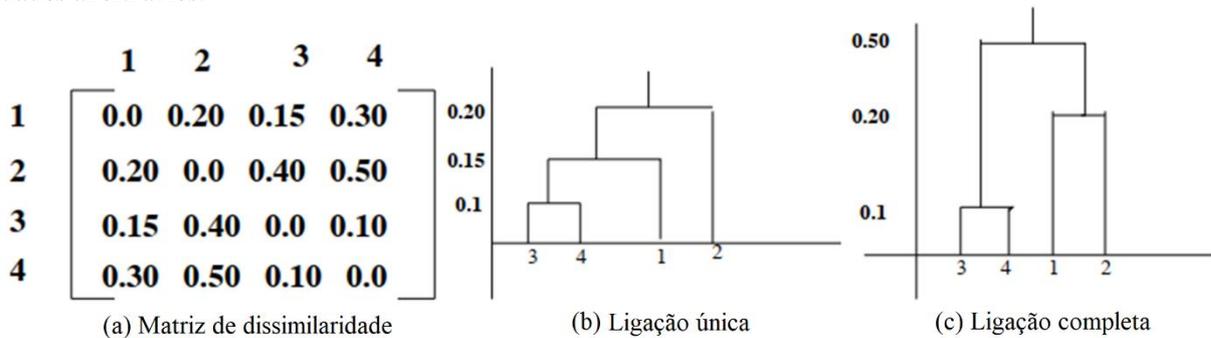


Fonte: adaptado de AMIRI *et al.* (2016).

Os algoritmos baseados em *HC* criam uma decomposição hierárquica de um conjunto de dados, gerando um dendrograma que representa o agrupamento aninhado dos dados e os níveis de similaridade nos quais os agrupamentos se subdividem, podendo ser classificados em algoritmos aglomerativos ou divisivos conforme o sentido da árvore de agrupamento. A abordagem aglomerativa dá-se de baixo para cima, ou seja, considera cada objeto como um *cluster* e começa a agrupá-los para formar novos *clusters*, em oposição à abordagem divisiva, por sua vez realizada de cima para baixo, onde todo o conjunto de dados é considerado como um único *cluster* e o algoritmo divide cada *cluster* em dois novos grupos iterativamente até que o número necessário de grupos seja alcançado. A maioria dos algoritmos *HC* baseiam-se em variantes das métricas de similaridade *single-link clustering* (SNEATH; SOKALL, 1973), *complete-link clustering* (KING, 1967) e pelo método da variância mínima de Ward (WARD JR, 1963), que pode ser interpretado como o quadrado da distância Euclidiana entre os centroides de dois *clusters* a serem fundidos, ponderados por um fator que é proporcional ao produto das cardinalidades dos *clusters* mesclados (TABESH; ASKARI-NASAB, 2011; AGGARWAL; REDDY, 2014; BAGIROV; KARMITSA; TAHERI, 2020). Podem-se citar ainda os algoritmos de agrupamento hierárquico *BIRCH* (*Balanced Iterative Reducing and Clustering using Hierarchies*) (ZHANG; RAMAKRISHNAN; LIVNY, 1996), *CURE* (*Clustering Using REpresentatives*) (GUHA; RASTOGI; SHIM, 2001) e *SOM* (*Self-Organizing Maps*) (KOHONEN, 1990). A Figura 13 apresenta dois exemplos de

dendrogramas obtidos por meio dos métodos (b) *single-link clustering* e (c) *complete-link clustering* a partir de uma (a) matriz de dissimilaridade calculada para quatro pontos de dados arbitrários.

Figura 13 – Exemplos de dendrogramas obtidos por meio dos métodos (b) *single-link clustering* e (c) *complete-link clustering* a partir de uma (a) matriz de dissimilaridade calculada para quatro pontos de dados arbitrários.



Fonte: adaptado de AGGARWAL; REDDY (2014).

Os algoritmos de agrupamento baseados na densidade (*density-based clustering*), por sua vez, consideram agrupamentos como regiões nas quais a densidade de pontos excede um limite predefinido (*threshold*), de modo que um dado *cluster* continua crescendo enquanto a densidade de sua vizinhança exceder este limiar, podendo até mesmo ser usado para criar agrupamentos de forma arbitrária. Dentre os representantes desta classe de algoritmos, podem-se citar o *DBSCAN* (ESTER *et al.*, 1996) e o *KNNCLUST* (*k-NN Density-Based Clustering*) (TRAN; WEHRENS; BUYDENS, 2006; BAGIROV; KARMITSA; TAHERI, 2020).

Os algoritmos de agrupamento baseados em uma grade (*grid-based clustering*) subdividem o espaço de dados em um número finito de células estruturadas em um *grid*, de modo que as operações de agrupamento são executadas na estrutura deste *grid*, como o algoritmo de agrupamento guiado por genética (*Genetic-Guided Clustering Algorithm*) (GARAI; CHAUDHURI, 2004).

Por fim, existem ainda os algoritmos probabilísticos ou baseados na lógica *Fuzzy*, onde ambas as classes permitem que um ponto de dados pertença a um ou mais agrupamentos simultaneamente. Dentre estes, podem-se citar o algoritmo *Fuzzy C-Means* (DUNN, 1973), o algoritmo de agrupamento de maximização da expectativa (*Expectation-Maximization Clustering*) (DEMPSTER; LAIRD; RUBIN, 1973) e os algoritmos baseados em modelos de Bernoulli e de mistura gaussiana (BAGIROV; KARMITSA; TAHERI, 2020).

2.3.2 Programação por restrições e seu emprego no problema do agrupamento

2.3.2.1 Fundamentos da programação por restrições

A programação por restrições (*CP*) é um poderoso paradigma capaz de solucionar problemas de busca combinatória por meio de técnicas de inteligência artificial, ciência da computação, pesquisa operacional e linguagens de programação, em que o usuário determina as restrições e um *solver* de propósito geral para resolvê-las. As restrições, portanto, determinam e relacionam as propriedades da solução a ser encontrada, de modo que um problema de satisfação de restrições (*Constraint Satisfaction Problem, CSP*) (MONTANARI, 1974; MACKWORTH, 1977) indica quais relações devem ser mantidas entre as variáveis de decisão dadas. Quanto à solução de problemas baseados em *CP*, os *solvers* geralmente baseiam-se em técnicas como propagação de restrição (*Constraint Propagation*) (KLEER; SUSSMAN, 1980), que reduz o domínio das variáveis ao remover valores inconsistentes, e em técnicas como o *Branch-and-Bound (BB)* (LAND; DOIG, 1960) e o retrocesso cronológico (*Chronological Backtracking*) (LEHMER, 1950; GOLOMB; BAUMERT, 1965; FILLMORE; WILLIAMSON, 1974), que exploram o espaço de soluções do problema e o dividem em subproblemas, tomando uma variável não atribuída e dividindo seu domínio em várias partes (ROSSI; VAN BEEK; WALSH, 2006).

Embora aplicações focadas na satisfação de restrições já existissem, esta metodologia que culminaria em *CP* começou a ser desenvolvida como um novo paradigma dentro da inteligência artificial e da ciência da computação a partir de Golomb e Baumert (1965), com seu trabalho seminal acerca da programação por retrocesso (*Backtracking Programming*). Na inteligência artificial, o interesse na satisfação de restrições se desenvolveu basicamente em dois fluxos, sendo eles o fluxo de linguagem, que produziu aplicativos em vários domínios e linguagens e sistemas de programação associados (WILKES, 1964; BURSTALL, 1969; FIKES, 1970; KOWALSKI, 1975; SUSSMAN; STEELE, 1980; ELCOCK, 1990), e o fluxo de algoritmo, que despertou o interesse em algoritmos de resolução de restrições na comunidade focada na visão das máquinas (MACKWORTH, 1973; MONTANARI, 1974; WALTZ, 1975; FREUDER, 1978; HARALICK; SHAPIRO, 1979, 1980). Segundo Freuder e Macworth (2006), os fluxos de linguagem e algoritmo divergiram e ambos se tornaram mais separados de domínios de aplicativos específicos. Enquanto os aplicativos e a exploração comercial proliferavam, as comunidades acadêmicas se concentravam mais em métodos

gerais, sendo ainda hoje um desafio a reconexão desses fluxos às suas raízes semânticas de resolução de problemas (FREUDER; MACWORTH, 2006).

O fluxo de linguagem foi fortemente influenciado pela programação lógica, na forma da programação lógica de restrição (*Constraint Logic Programming, CLP*), focado no desenvolvimento de linguagens de programação e bibliotecas, como as linguagens Planner (HEWITT, 1969), Prolog (COLMERAUER; ROUSSEL, 1993), dentre outras aplicações (SUSSMAN; WINOGRAD, 1970; HAYES, 1973; KOWALSKI, 1974; JAFFAR; LASSEZ, 1987). O fluxo de algoritmos, por outro lado, foi influenciado pelo paradigma da inteligência artificial como busca (NILSSON, 1971) e pelo desenvolvimento da ciência dos algoritmos (KNUTH, 1973), focada em algoritmos e heurísticas, esta segunda permanecendo em detrimento da primeira, concentrando-se cada vez mais na formulação simples, mas poderosa e geral, dos modelos *CSP* e suas variantes. Entretanto, a partir da década de 1990, quando Kanellakis, Lassez e Saraswat presidiram um *workshop* que logo tornou-se na *International Conference on Principles and Practice of Constraint Programming*, e quando Gene Freuder estabeleceu a *Constraints Journal*, essa tarefa de integração dos métodos e abordagens da satisfação de restrições em uma ciência concisa teve significativos avanços (FREUDER; MACWORTH, 2006).

Van Hoes e Katriel (2006) afirmam que em um problema baseado em *CP*, de forma genérica, o domínio $D(x)$ de uma variável x equivale a um conjunto finito de elementos que podem ser atribuídos a x , e para um conjunto de variáveis X , a união de seus domínios dá-se por $D(X) = \bigcup_{x \in X} D(x)$. Sendo então $X = \{x_1, \dots, x_k\}$ um conjunto de variáveis, uma restrição C em X é um subconjunto do produto cartesiano dos domínios das variáveis em X , ou seja, $C \subseteq D(x_1) * \dots * D(x_k)$, onde uma tupla $(d_1, \dots, d_k) \in C$ representa uma solução para C . Pode-se dizer ainda que, de modo equivalente, uma solução $(d_1, \dots, d_k) \in C$ é uma atribuição do valor d_i à variável x_i , para todo $1 \leq i \leq k$, e que esta atribuição satisfaz C . Se $C = \emptyset$, tem-se uma restrição inconsistente, ao passo que quando uma restrição C é definida em um conjunto X de k variáveis junto a um certo conjunto p de parâmetros, denota-se por $C(X, p)$, sendo este um conjunto de k -tuplas, e não $k + 1$ -tuplas (VAN HOEVE; KATRIEL, 2006).

Um problema de satisfação de restrições (*CSP*), portanto, pode ser descrito como um conjunto finito de variáveis X junto a um conjunto finito de restrições C , cada uma em um subconjunto de X , ao passo que uma solução para um *CSP* é uma atribuição de um valor $d \in D(x)$ a cada $x \in X$, tal que todas as restrições são satisfeitas simultaneamente. Portanto, dada

uma restrição \mathcal{C} definida nas variáveis $\{\mathbf{x}_1, \dots, \mathbf{x}_k\}$, a tarefa de filtragem é reduzir o domínio de cada variável de modo que ainda contenha todos os valores que essa variável pode assumir em uma solução para \mathcal{C} . Um algoritmo que atinge a filtragem completa calcula, para cada $1 \leq j \leq k$, $D(\mathbf{x}_j) \leftarrow D(\mathbf{x}_j) \cap \{v_i | D(\mathbf{x}_1) * \dots * D(\mathbf{x}_{j-1}) * \{v_i\} * D(\mathbf{x}_{j+1}) * \dots * D(\mathbf{x}_k) \cap \mathcal{C} \neq \emptyset\}$. Isto posto, como em muitas aplicações se deseja encontrar uma solução para um *CSP* que seja ótima em relação a determinados critérios, um problema de otimização de restrições (*Constraint Optimization Problem, COP*) equivale a um *CSP* cujo X definido nas variáveis $\mathbf{x}_1, \dots, \mathbf{x}_n$, juntamente com uma função objetivo $f: D(\mathbf{x}_1) * \dots * D(\mathbf{x}_k) \rightarrow \mathbb{Q}$ atribui um valor a cada atribuição de valores às variáveis. Uma solução ótima para um *COP* de minimização (maximização) seria uma solução \mathbf{d} para P que minimiza (maximiza) o valor de $f(\mathbf{d})$, sendo o valor dessa função objetivo frequentemente representado por uma variável \mathbf{z} , juntamente com a “restrição” de minimizar (maximizar) \mathbf{z} para um problema de minimização (maximização) (VAN HOEVE; KATRIEL, 2006).

2.3.2.2 Aplicação da programação por restrições ao problema do agrupamento

No âmbito da aplicação de um modelo baseado em *CP* para solucionar o problema do agrupamento, Dao, Duong e Vrain (2017) propuseram duas variantes contendo uma série de assunções que podem ser empregadas em qualquer circunstância. Seja um domínio com n pontos e uma medida de dissimilaridade entre pares de pontos i, j denotada por $\mathbf{d}(i, j)$; sem perda de generalidade, considerando que os pontos sejam indexados e nomeados pelo seu índice, em que 1 representa o primeiro ponto, o modelo *CP* proposto pelos autores visa encontrar uma partição do domínio em k *clusters* que satisfaça um conjunto de restrições e otimize um determinado critério, sendo então mais precisamente denotado por um modelo *COP*. Conforme os autores, as restrições poderiam ser classificadas como aquelas que expressam que o resultado deve ser uma partição, aquelas que expressam as restrições do usuário e aquelas que expressam o critério a ser otimizado, de modo que quando nenhum critério de otimização é fornecido, o modelo *CP* busca todas as partições que satisfaçam todas as restrições. Caso se determine um conjunto de variáveis para a atribuição de um representante a cada *cluster* e outro conjunto de variáveis para a atribuição de um representante a cada ponto, a escolha de tal representação requer que o número de k *clusters* seja fixado previamente, já que cada representante é modelado por uma variável no âmbito do *CP*; caso se opte por determinar apenas um conjunto de variáveis para a atribuição de um

índice de *cluster* a cada ponto, é possível determinar um número k *clusters* entre k_{min} e k_{max} , sendo estes limites inferiores e superiores (*upper bound* e *lower bound*, *UB* e *LB*) determinados pelo usuário (DAO; DUONG; VRRAIN, 2017).

Foi considerado o modelo em que somente um conjunto de variáveis para a atribuição de um representante a cada ponto é empregado, já que somente este modelo foi empregado neste estudo. Os *clusters* são identificados pelos seus índices, variando entre 1 e k para uma partição do domínio em k *clusters*, sendo usadas variáveis inteiras G_1, \dots, G_n , cujo domínio é o conjunto de inteiros $[1, k_{max}]$, para representar a atribuição dos pontos aos *clusters*, onde a atribuição $G_i = c$ significa que o ponto i recebe o número do *cluster* c , seja \mathcal{G} o array $[G_1, \dots, G_n]$. É ainda introduzida uma variável de valor flutuante para representar cada critério de otimização, sendo o critério de diâmetro denominado por D , o critério de divisão denominado por S e o critério *WCSD* denominado por V , sendo seus domínios representados por $Dom(D) = [\min_{i,j}(d(i,j)), +\infty)$, $Dom(S) = (-\infty, \max_{i,j}(d(i,j))]$ e $Dom(V) = [0, +\infty)$ (DAO; DUONG; VRRAIN, 2017).

Quanto às restrições de partição, os *clusters* são identificados pelo seu índice e cada variável G_i fornece o índice do *cluster* que contém o ponto i , onde uma atribuição completa das variáveis em \mathcal{G} representa uma partição. No entanto, uma partição pode ser representada por diferentes atribuições completas de \mathcal{G} , como por exemplo, dada uma atribuição completa de \mathcal{G} , se for feita uma permutação onde todas as variáveis G_i que têm o valor c_1 assumem o valor c_2 e, ao mesmo tempo, todas as variáveis G_j tendo o valor c_2 assumem o valor c_1 , obtêm-se uma nova atribuição para \mathcal{G} , que ainda representa a mesma partição em termos de *clusters*. Como segundo exemplo fornecido pelos autores, quando todas as variáveis G_i com valor c_i recebem um valor c_3 que ainda não é utilizado em uma atribuição das outras variáveis de \mathcal{G} , isso leva a uma nova atribuição representando uma solução simétrica. Tal situação surge ao definir os *clusters*, e um novo *cluster* pode receber qualquer valor entre os números de *cluster* restantes (DAO; DUONG; VRRAIN, 2017).

Para quebrar esse tipo de simetria, os *clusters* são numerados de tal forma que o número 1 é o índice do primeiro *cluster* criado e um novo número c , com $c > 1$, é usado se e somente se o número $c - 1$ já tiver sido usado. Uma maneira direta de expressar esta condição é usando uma restrição $G_1 = 1$ e as restrições $G_i \leq \max_{j \in [1, i-1]}(G_j) + 1$, para $i \in [2, n]$, condição que também pode ser expressa como uma restrição global em um bom algoritmo de filtragem para possibilitar melhores interações e propagações entre essas relações, onde $preceder(\mathcal{G}, [1, \dots, k_{max}])$. Esta restrição impõe que $G_1 = 1$ e, além disso,

se $G_i = c$ com $1 < c \leq k_{max}$, deve existir pelo menos um índice $j < i$ tal que $G_j = c - 1$. A exigência de ter pelo menos k_{min} clusters significa que todos os números entre 1 e k_{min} devem ser usados na atribuição das variáveis G_i . Ao usar a restrição preceder, basta exigir que pelo menos uma variável G_i seja igual a k_{min} . Isso é expresso pela relação $\#\{i \mid G_i = k_{min}\} \geq 1$, que pode ser representada pela restrição CP **pelo menos**($\mathbf{1}, \mathcal{G}, k_{min}$). Como o domínio de cada variável G_i equivale a $[1, k_{max}]$, haverá no máximo k_{max} clusters, e se o usuário necessitar exatamente de k clusters, deve-se definir $k_{min} = k_{max} = k$ (DAO; DUONG; VRAIN, 2017).

Quanto às restrições de usuário, é comum o emprego das seguintes restrições nos problemas de agrupamento:

1. Tamanho mínimo α de clusters, em que cada ponto deve estar em um cluster com pelo menos α pontos (incluindo ele próprio). Para cada $i \in [1, n]$, o valor atribuído da variável G_i deve então aparecer pelo menos α vezes no array \mathcal{G} , ou seja, $\#\{j \mid G_j = G_i\} \geq \alpha$. Portanto, para cada $i \in [1, n]$, emprega-se a restrição: **pelo menos**(α, \mathcal{G}, G_i), restrição que também ajuda a definir um limite para o número de clusters possíveis. De fato, o número de clusters não pode exceder $\lfloor n/\alpha \rfloor$, podendo ser expresso por $G_i \leq \lfloor n/\alpha \rfloor$, para todo $i \in [1, n]$;
2. Tamanho máximo β de clusters, em que cada número $c \in [1, k_{max}]$ deve aparecer no array \mathcal{G} no máximo β vezes (isso ainda é verdade para um valor não utilizado $c \in [k_{min} + 1, k_{max}]$, já que aparece 0 vezes), ou seja, $\#\{i \mid G_i = c\} \leq \beta$. Portanto, para cada $c \in [1, k_{max}]$, põe-se a restrição **no máximo**(β, \mathcal{G}, c);
3. Divisão mínima δ , que requer que a divisão entre dois clusters seja pelo menos δ . Consequentemente, para cada par $i < j \in [1, n]$ tal que $d(i, j) < \delta$, põe-se a restrição $G_j = G_i$, além da restrição $S \geq \delta$;
4. Diâmetro máximo γ , que requer que o diâmetro de cada cluster seja no máximo equivalente a γ , sendo para tal inserida a restrição $D \leq \gamma$ e, para cada par $i < j \in [1, n]$, tal que $d(i, j) > \gamma$, põe-se a restrição $G_i \neq G_j$;
5. Restrição de densidade, expressando que cada ponto deve ter em sua vizinhança de raio ϵ pelo menos m pontos pertencentes ao mesmo cluster que ele. Assim, para cada $i \in [1, n]$, o conjunto de variáveis correspondentes aos pontos em sua ϵ -vizinhança é calculado como $\mathcal{N}_{i\epsilon} = \{G_i \mid d(i, j) \leq \epsilon\}$, de modo que **pelo menos**($m, \mathcal{N}_{i\epsilon}, G_i$);
6. Restrição de vinculação obrigatória que, para dois pontos i, j , é expressa por $G_i = G_j$ e $D \geq d(i, j)$;

7. Restrição de não vinculação obrigatória que, para dois pontos i, j , é expressa por $G_i \neq G_j$ e $S \leq d(i, j)$ (DAO; DUONG; VRRAIN, 2017).

Quanto aos critérios de otimização empregados em seu modelo *CP*, Dao, Duong e Vrain (2017) sugerem o emprego de critérios que minimizem o diâmetro máximo dos *clusters* (D) tal que *diâmetro* (G, D, d), que maximizem a divisão mínima entre os *clusters* (S) tal que *divisão* (G, S, d), e que minimizem a soma das dissimilaridades dentro de cada *clusters* (*WCSD*) (V) tal que *wcsd* (G, V, d). Para tal, os autores elaboraram algoritmos de filtragem destas restrições globais, ligando as variáveis G , que representam partições, à variável que representa a função objetivo (D , S ou V). Destarte, quando uma solução é encontrada para estas restrições de otimização, seu valor objetivo correspondente é calculado e uma restrição expressando que novas soluções devem ter um valor melhor que este é adicionada, funcionando como um novo limite para estas funções objetivo, reduzindo os espaços de busca (DAO; DUONG; VRRAIN, 2017).

Dentre estas, a restrição *diâmetro* (G, D, d) garante que D seja o diâmetro máximo dos *clusters* formados pelas variáveis $[G_1, \dots, G_n]$, tal que $\forall i < j \in [1, n], D < d(i, j) \rightarrow G_i \neq G_j$. O domínio $Dom(D)$ é representado por $[D.lb, D.ub)$, onde $D.lb$ é o *LB*, que inicialmente pode ser a mínima dissimilaridade entre dois pontos, e $D.ub$ é o *UB*, que pode ser $+\infty$ ou o valor de D . O limite $D.ub$ é estrito, pois em uma busca com um algoritmo *BB*, a próxima solução deve ter um valor D estritamente menor que o anterior. Esta relação é útil quando, por exemplo, o limite superior $D.ub$ foi alterado devido à descoberta de uma nova solução ou graças a uma restrição de diâmetro, já que para cada par i, j , se $D.ub \leq d(i, j)$, pode-se concluir que $D < d(i, j)$ e $G_i \neq G_j$, filtrando o domínio G_i (ou G_j) contanto que a variável G_j (ou G_i , respectivamente) tenha sido instanciada. Esta relação também seria útil quando algumas variáveis G_i fossem instanciadas, já que para cada par i, j tal que G_i e G_j sejam instanciados e tenham o mesmo valor, pode-se inferir $D \geq d(i, j)$ e revisar $D.lb$ (DAO; DUONG; VRRAIN, 2017).

A restrição *divisão* (G, S, d), por sua vez, garante que S seja a divisão mínima entre os *clusters* formados pelas variáveis G_1, \dots, G_n , assegurando que $\forall i < j \in [1, n], S > d(i, j) \rightarrow G_i = G_j$. O domínio $Dom(S)$ é representado por $(S.lb, S.ub]$, onde $S.lb$ é o *LB*, que inicialmente pode ser que pode ser $-\infty$ ou o valor de S , já que este deve ser maximizado. Tal qual em *diâmetro* (G, D, d), esta restrição em um algoritmo seria invocada se o limite inferior $S.lb$ fosse alterado ou se algumas variáveis em G fossem instanciadas e, havendo uma alteração, para cada par i, j , se $S.lb \geq d(i, j)$, pode-se inferir que $G_i = G_j$, resultado

propagado quando se força $Dom(G_i) = Dom(G_j)$. Caso contrário, se algumas variáveis em \mathcal{G} fossem instanciadas e $G_i \neq G_j$, poder-se-ia inferir que $S \leq d(i, j)$, então o UB de S poderia ser alterado (DAO; DUONG; VRRAIN, 2017).

A restrição $wcsd(\mathcal{G}, V, d)$, finalmente, poderia ser descrita como $V = \sum_{1 \leq i < j \leq n} (G_i = G_j) d(i, j)$, onde $G_i = G_j$ equivale a 1 caso ambos tenham o mesmo valor, ou 0 caso contrário. Suponha-se que em uma atribuição parcial de variáveis em \mathcal{G} , $K = \{i \in [1, n] | G_i \text{ está atribuído}\}$ e $U = \{i \in [1, n] | G_i \text{ não está atribuído}\}$. Usando um cálculo do limite inferior que leve em consideração as variáveis não atribuídas, a soma que define V pode ser dividida em três partes, tal que $V = V_1 + V_2 + V_3$, onde $V_1 = \sum_{i, j \in K, i < j} (G_i = G_j) d(i, j)$ seria a soma das diferenças entre os pontos atribuídos, $V_2 = \sum_{i \in U, j \in K} (G_i = G_j) d(i, j)$ seria a soma das diferenças entre os pontos atribuídos e não atribuídos, e $V_3 = \sum_{i, j \in U, i < j} (G_i = G_j) d(i, j)$ seria a soma das diferenças entre os pontos não atribuídos. Sendo o conjunto K conhecido, o valor exato de V_1 pode ser calculado. Como os pontos de U não foram ainda atribuídos a um *cluster*, o valor de V_2 é desconhecido, mas um LB de V_2 denotado por $V_2 \cdot lb$ pode ser calculado pela soma da contribuição mínima de todos os pontos não atribuídos. Para cada ponto não atribuído $i \in U$, cada valor $c \in Dom(G_i)$ representa um índice de *cluster* ao qual o ponto i pode ser atribuído. Se o ponto i for atribuído ao *cluster* c , ele contribuirá para esse *cluster* com a soma das dissimilaridades entre o ponto i e todos os pontos atribuídos ao mesmo *cluster* c , ou seja, a soma das dissimilaridades $d(i, j)$ para todo $j \in K$ tal que $G_j = c$. A contribuição mínima v_{2i} do ponto i é o valor mínimo adicionado ao considerar todos os valores em $Dom(G_i)$, tal que $v_{2i} = \min_{c \in Dom(G_i)} (\sum_{j \in K, G_j = c} d(i, j))$, de modo que um limite inferior $V_2 \cdot lb$ pode ser calculado pela soma de v_{2i} para todo $i \in U$, como em $V_2 \cdot lb = \sum_{i \in U} v_{2i}$ (DAO; DUONG; VRRAIN, 2017).

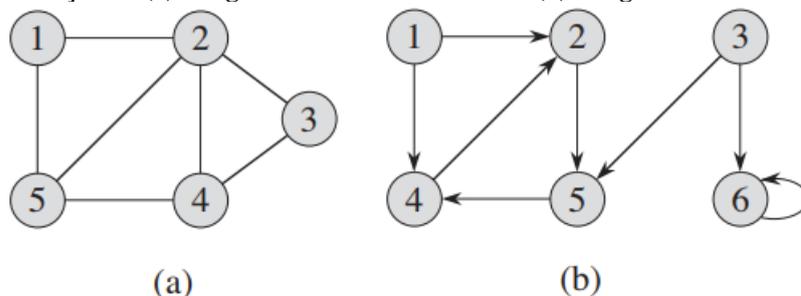
Sendo V_3 a soma de todos os $d(i, j)$ tais que $i, j \in U$ e estão no mesmo *cluster*, o valor exato de V_3 também é desconhecido, e pode-se usar uma heurística para calcular seu LB . Seja p a cardinalidade de U e seja k a cardinalidade da união $\bigcup_{i \in U} Dom(G_i)$, cada valor de $\bigcup_{i \in U} Dom(G_i)$ é o índice de um possível *cluster* ao qual os pontos em U podem ser atribuídos. O número k é então o número máximo de *clusters* aos quais os pontos em U podem ser atribuídos, tal que o número mínimo de termos $d(i, j)$ na soma V_3 é o número mínimo de pares dentro de um *cluster*, considerando todas as partições de p pontos em no máximo k *clusters*. Seja m o quociente da divisão de p por k e m' o resto, e seja $f(p, k) = (km^2 + 2mm' - km)/2$, o número total de pares dentro de um *cluster* para todos os

$clusters$ é maior ou igual a $f(p, k)$. A igualdade é alcançada quando m' $clusters$ possuem $m + 1$ elementos e $k - m'$ $clusters$ possuem m elementos. Portanto, para o conjunto U de pontos não atribuídos, se ordenarmos cada vez mais as constantes $d(i, j)$ para todo $i < j \in U$, um limite inferior V_3 , denotado por $V_3.lb$, é então calculado pela soma das $f(p, k)$ primeiras constantes nesta ordem (DAO; DUONG; VRRAIN, 2017).

2.3.3 Busca pelo menor caminho em um grafo

Embora não seja o escopo principal deste estudo, visto que a heurística *GPH* empregada na etapa de pós-processamento dos $clusters$ emprega uma etapa de busca em um grafo pelo menor caminho, é importante introduzir alguns conceitos sobre estes elementos e alguns algoritmos capazes de desempenhar esta tarefa. Um grafo é uma representação discreta capaz de relacionar objetos em um dado conjunto, podendo ser representado por $G = (V, E)$, onde V representa o conjunto dos vértices (*vertices* ou nós) e E representa o conjunto das arestas (*edges* ou pares de vértices). Grafos são considerados esparsos se $|E| \ll |V|^2$, mas caso $|E| \approx |V|^2$, um grafo é considerado denso. Ainda, os grafos podem ser classificados como direcionados (dígrafos) ou não direcionados. No primeiro caso, tem-se que $s, t: E \rightarrow V$, onde $s, t \in V$ são os nós inicial (*source*) e terminal (*target*) de uma aresta direcionada $e \in E$, e caso se deseje atribuir pesos às arestas (custos, por exemplo), tem-se uma função adicional $w: E \rightarrow \mathbb{R}$. Já um grafo não direcionado pode ser representado por uma função $f: E \rightarrow P(V)$, associando um subconjunto de dois ou mais vértices (pontos terminais) a cada aresta $e \in E$ (CORMEN *et al.*, 2009). A Figura 14 apresenta uma representação de (a) um grafo não direcionado e (b) um grafo direcionado.

Figura 14 - Representação de (a) um grafo não direcionado e de (b) um grafo direcionado.

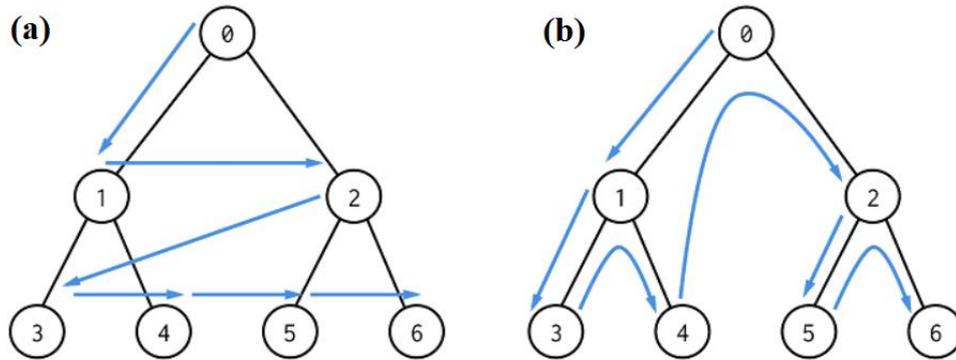


Fonte: adaptado de CORMEN *et al.* (2009).

Quando se deseja determinar um percurso em um grafo, é possível modelá-lo como uma árvore e visitar cada nó exatamente uma vez, onde um caminho seria a visita a todos os nós adjacentes entre uma origem e um destino, não necessariamente havendo uma ordem a ser seguida. Uma das estratégias mais comuns para resolver este problema é a busca em extensão ou largura (*Breadth-First Search, BFS*). Dado um gráfico $G = (V, E)$ direcionado ou não-direcionado, além de um vértice de origem s , um *BFS* explora sistematicamente as arestas para identificar cada vértice acessível a partir de s , calculando a distância (menor número de arestas) de s até cada vértice alcançável. Esta estratégia produz uma árvore em largura tendo s como raiz, contendo todos os vértices alcançáveis, em que qualquer caminho até um dado vértice v nesta árvore representa o caminho mais curto entre s e v . O nome busca em largura ocorre porque um algoritmo *BFS* expande a fronteira entre vértices descobertos e não descobertos uniformemente em todas as direções da fronteira, ou seja, todos os vértices à distância k de s são descobertos antes que qualquer vértice à distância $k + 1$ seja descoberto. Por ser um algoritmo que baseado em uma estrutura de fila (*queue*), seu método de progressão é o primeiro a entrar, primeiro a sair (*First-In, First-Out, FIFO*), de modo que se um novo nó entrar nesta fila, todos os nós anteriores deverão ser extraídos antes que este dado nó saia da fila (CORMEN *et al.*, 2009).

A busca em profundidade (*Depth-First Search, DFS*), por sua vez, explora primeiro arestas do vértice v descoberto mais recentemente que ainda possuir arestas inexploradas, se aprofundando na estrutura em árvore sempre que possível. Uma vez exploradas todas as arestas de v , a busca retrocede (*backtracking*) até que seja identificado um novo vértice u que possibilite a exploração de novas arestas partindo de u . Considerando s como nó de origem, este processo se repete até que todos os vértices acessíveis sejam descobertos, e de modo semelhante ao *BFS*, sempre que um *DFS* descobre um vértice v na lista de adjacências de um vértice u já descoberto, esse evento é registrado pela determinação de atributo predecessor $v: \pi$ para u . Entretanto, ao contrário da *BFS*, cujo subgrafo predecessor forma uma árvore, o subgrafo predecessor produzido pelo *DFS* pode ser composto por várias árvores, já que a busca pode ser repetida a partir de múltiplas fontes, resultando em uma *depth-first forest* compreendendo diversas *depth-first trees*. Ao invés de operar em fila, o *DFS* opera em pilha (*stack*), tal que seu método de progressão é o último a entrar, primeiro a sair (*Last-In, First-Out, LIFO*), onde se um novo nó entrar nesta fila, todos os nós anteriores só poderão ser extraídos após a extração deste dado nó (CORMEN *et al.*, 2009). A Figura 15 exemplifica as estratégias (a) *BFS* e (b) *DPF* para um mesmo grafo.

Figura 15 – Exemplificação das estratégias (a) busca em largura e (b) busca em profundidade.



Quando se deseja determinar o menor caminho entre um nó de origem s e um nó terminal t , tem-se o problema do menor caminho a partir de uma única origem (*single-source shortest paths*), onde nem sempre é necessário computar todos os nós em um grafo até que se obtenha o mais curto entre s e t . O algoritmo de Bellman-Ford (BELLMAN, 1958; FORD, 1956) ou Bellman-Ford-Moore (MOORE, 1959) baseia-se na estratégia *BFS* e, a partir de uma origem s , efetua uma etapa de pré-processamento para construir uma árvore de menores distâncias entre todos os nós de um grafo direcionado e ponderado $G = (V, E)$ com função de peso $w: E \rightarrow \mathbb{R}$. O algoritmo retorna um valor booleano indicando se existe ou não um ciclo de peso negativo neste grafo, determinando que não existe solução nestas circunstâncias ou prosseguindo caso contrário. Então, o algoritmo relaxa as arestas (u, v) , diminuindo progressivamente uma estimativa $v.d$ no peso de um caminho mais curto a partir da fonte s para cada vértice $v \in V$ até atingir o peso real do caminho mais curto $\delta(s, v)$. Outra maneira de exemplificar este procedimento é que o algoritmo de Bellman-Ford minimiza uma função de custo $f(x) = g(x)$, em que $g(x)$ representa o custo de mover-se do nó inicial s até um determinado estado x (CORMEN *et al.*, 2009).

O algoritmo de Dijkstra (DIJKSTRA, 1959), por sua vez, possui uma estratégia gulosa baseada em *BFS* para resolver o problema do menor caminho a partir de um único nó de origem. Considerando que um grafo direcionado e ponderado $G = (V, E)$ com origem s não possui pesos negativos, o algoritmo mantém um conjunto S de vértices cujos pesos finais do caminho mais curto da fonte s já foram determinados. Tem-se inicialmente que $S = \emptyset$ e uma fila de prioridade mínima Q contém todos os vértices. Então, enquanto $Q \neq \emptyset$, a cada iteração $Q = V - S$ é computado, sendo selecionado repetidamente o vértice $u \in V - S$ com a estimativa do mínimo caminho mais curto, adicionando u a S e relaxando todas as arestas (u, v) a partir de u . Com isso, sempre que for possível melhorar o caminho mais curto de u para v , são atualizados o atributo predecessor $v:\pi$ e a estimativa $v:d$. Mesmo que nem

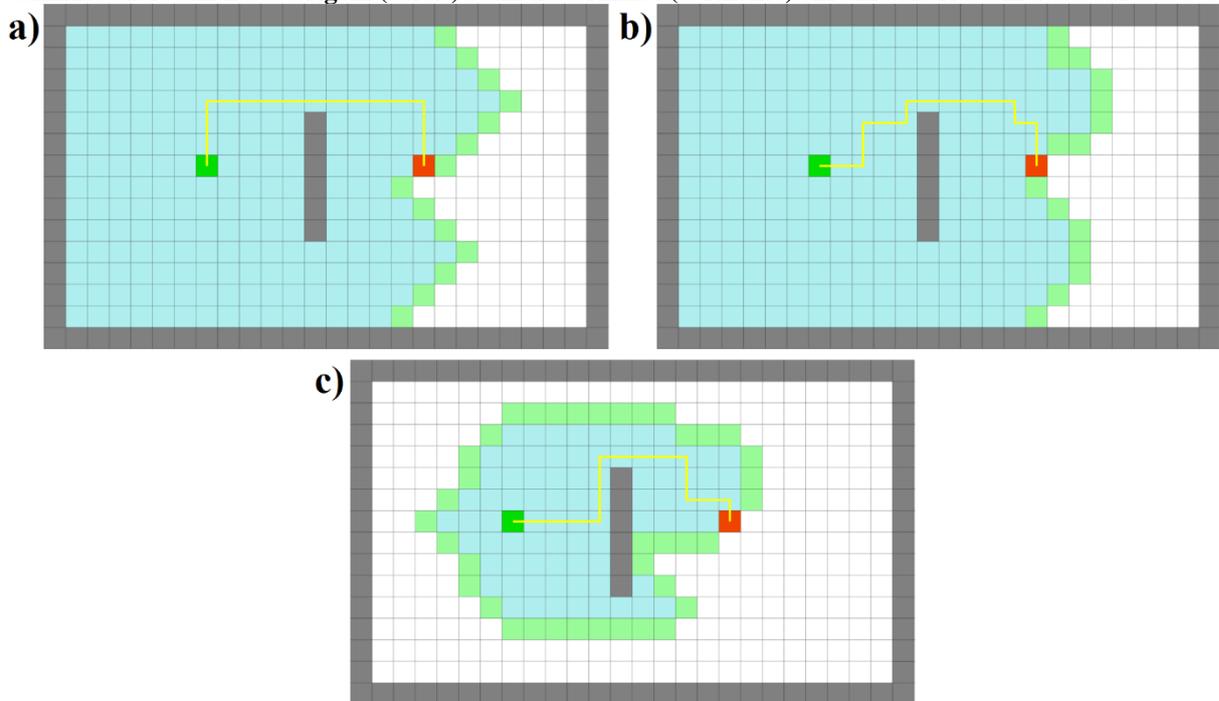
sempre estratégias gulosas resultem em uma solução ótima, é possível provar que o algoritmo de Dijkstra sempre retorna o caminho mais curto $u: d = \delta(s, u)$ a cada vez que um vértice u é adicionado ao conjunto S . Também é possível associar o procedimento deste algoritmo à minimização de uma função de custo $f(x) = g(x)$, onde $g(x)$ equivale ao custo de mover-se do nó inicial s até um determinado estado x (CORMEN *et al.*, 2009).

Finalmente, o algoritmo de busca A^* (*A* Search*) (HART; NILSSON; RAPHAEL, 1968) também é capaz de resolver o problema do menor caminho a partir de um único nó de origem em um grafo ponderado $G = (V, E)$ com origem s , podendo ser visto como uma extensão do algoritmo de Dijkstra. Entretanto, ao invés de fazer o pré-processamento de uma árvore de menores distâncias a partir de s e computar a todas as distâncias nesta árvore, o *A* Search* emprega uma heurística para guiar a busca diretamente até o nó terminal t , evitando assim visitar nós distantes e desnecessários. Este algoritmo cria uma árvore de caminhos a partir do nó inicial onde são estendidas uma aresta por vez até que seu critério de parada seja atingido. A cada iteração do *A* Search* ocorre a minimização de uma função de custo $f(x) = g(x) + h(x)$, em que $g(x)$ representa o custo de mover-se do nó inicial s até um determinado estado x , ao passo que $h(x)$ é o custo aproximado de mover-se deste estado x para o nó terminal t , sendo este o critério que determina qual caminho será expandido a cada iteração. Uma estrutura em fila é geralmente empregada, onde a cada iteração o nó com o menor valor de $f(x)$ é removido da fila, seus vizinhos são adicionados a ela e seus valores f e g são atualizados, até que por fim o nó removido seja o próprio nó terminal t , quando o valor de $f(t)$ vai equivaler ao caminho mais curto e $h(t)$ será igual a zero. A heurística $h(x)$ pode basear-se em diferentes métricas de distância, como as distâncias Euclidiana ou de Manhattan, contanto que represente corretamente o problema e sempre indique a menor distância entre um estado atual x e o nó terminal t . Ao invés de memorizar todas as distâncias presentes na árvore como ocorre com os algoritmos anteriores, que basicamente consideram todos os nós como possíveis destinos, o *A* Search* somente memoriza a distância entre s e t .

Naturalmente, existem ainda outros tipos de grafos, que são capazes de representar outras classes de problemas e, conseqüentemente, necessitam de diferentes algoritmos para resolvê-los. Todavia, visto que este assunto excede o escopo deste estudo, mais informações podem ser consultadas em Cormen *et al.* (2009). A Figura 16 apresenta ainda uma comparação entre os algoritmos (a) *BFS*, (b) Dijkstra e (c) *A* Search* na busca pelo menor caminho entre um nó de origem (verde) e um nó terminal (vermelho) em um cenário com

obstáculo, sendo necessários 486, 493 e 206 passos, respectivamente, até que um caminho mínimo com 16 passos fosse identificado por cada metodologia.

Figura 16 - Comparação entre os algoritmos (a) *BFS*, (b) *Dijkstra* e (c) *A* Search* na busca pelo menor caminho entre um nó de origem (verde) e um nó terminal (vermelho) em um cenário com obstáculo.



2.4 O PROBLEMA DO AGRUPAMENTO DE POLÍGONOS DE LAVRA

Conforme já mencionado, diversas técnicas de agrupamento de blocos em *clusters* já foram empregadas para reduzir o tamanho e o tempo de processamento de problemas de sequenciamento de lavra, visto que os arcos de precedência se dariam nos *clusters*, e não nos blocos originais, além do fundamental acréscimo de características operacionais ao sequenciamento de produção. Dentre as técnicas empregadas, este estudo direciona seu foco sobre a modelagem de camadas em depósitos estratigráficos e/ou sub-horizontais, técnica muito eficiente para modelar depósitos rasos com grandes extensões em área, como carvão, bauxita e fosfato, além do agrupamento de polígonos de lavra, que são *clusters* de blocos construídos em bancadas individualizadas com base em um critério de similaridade.

Uma das primeiras tentativas de representar um depósito estratificado em camadas para simplificar o modelo de otimização do sequenciamento de lavra foi proposta por Albach (1967), que implementou um modelo baseado em Programação Restrita ao Acaso (*Chance-Constrained Programming, CCP*) capaz de avaliar a incerteza do sequenciamento de longo prazo em relação aos limites entre as camadas de rejeitos e de carvão, bem como a quantidade

de rejeitos removidos antes do início da exploração dos estratos de carvão. Assim, os autores realizaram uma análise de risco na operação de uma empresa de carvão com duas cavas ao longo de 20 anos. Metz e Jain (1978), por sua vez, propuseram duas abordagens para o sequenciamento de longo prazo de depósitos de fosfato, primeiro empregando um modelo baseado em Programação Dinâmica (*Dynamic Programming, DP*) para lidar com restrições lineares e, em seguida, um modelo *MINLP* denominado como *deblending algorithm* para lidar com restrições não lineares, resolvido por uma heurística por eles proposta. Considerando os dados de cinco furos contendo 35 estratos, os autores compararam os resultados de três estudos de caso: um considerando todo o minério como economicamente lavrável, outro que avalia cada estrato independentemente por meio de fatores de corte e outro considerando o fluxo de trabalho do *deblending algorithm*.

Gershon (1983) sugeriu uma modificação em seu modelo baseado em colunas para que as camadas fossem geradas na direção de maior continuidade do minério em depósitos sub-horizontais antes de serem submetidas a um fluxo de trabalho de sequenciamento de mina baseado em *LP*, técnica denominada por ele como abordagem direcional. Busnach, Mehrez e Sinuany-Stern (1985) lidaram com um depósito de fosfato em Israel lavrado pelo método *strip mining*, onde além das três camadas já lavradas, havia uma significativamente mais profunda, cujo aproveitamento não ocorria em função das restrições do método de lavra. Portanto, os autores agruparam os blocos de uma mesma bancada em camadas horizontais por meio de um modelo *MINLP* para resolver o sequenciamento de longo prazo, tentando solucioná-lo por meio de uma heurística, mas não alcançaram um tempo computacional viável à época. Klingman e Phillips (1988), por sua vez, elaboraram um modelo *IP* capaz de determinar quais parcelas de quais camadas de fosfato de uma mineradora dos EUA seriam lavradas e quais seriam suas destinações com base nos teores de minérios e contaminantes, resolvendo o problema por meio de um algoritmo *BB*.

Gershon e Murphy (1989) apresentaram uma abordagem baseada em *DP* para otimizar *mining cuts* construídos com base na informação de um único furo de sondagem, a partir do qual as características do minério seriam avaliadas, seguido da agregação destes *mining cuts* em camadas a partir dos estratos, definidas a seguir como camadas de minério ou estéril, culminando no sequenciamento de produção das camadas visando a maximização do VPL. A metodologia foi programada em Lotus 123, e um estudo de caso de um depósito de xisto betuminoso foi apresentado. Chanda e Wilke (1992) propuseram um modelo baseado em Programação Linear de Metas (*Linear Goal Programming, LGP*) para resolver o planejamento operacional de lavra de depósitos estratiformes, visando maximizar o teor de

metal extraído e minimizar o desvio dos parâmetros de qualidade, sujeito a restrições relacionadas ao espaço de trabalho, ângulo de talude, acessibilidade e taxa de decapeamento, metodologia que conta ainda com uma fase heurística. O corpo mineral é então representado como regiões de escavação cuja extensão e tamanho dependem das zonas de teores em cada bancada, bem como dos requisitos operacionais, de modo que os três tipos de zonas de escavação considerados são aqueles localizados em zonas de minério, zonas de estéril e zonas limites entre minério e estéril. Enquanto a etapa heurística considera todas estas zonas, a etapa de otimização linear considera apenas as regiões da escavadeira nas zonas de minério, de modo que após a etapa baseada em *LGP* determinar quais zonas de escavação serão extraídas em cada período, a etapa heurística realiza a alocação das *shovels*, sendo apresentado por fim um estudo de caso de uma jazida de minério de cobre e cobre-cobalto na Zâmbia.

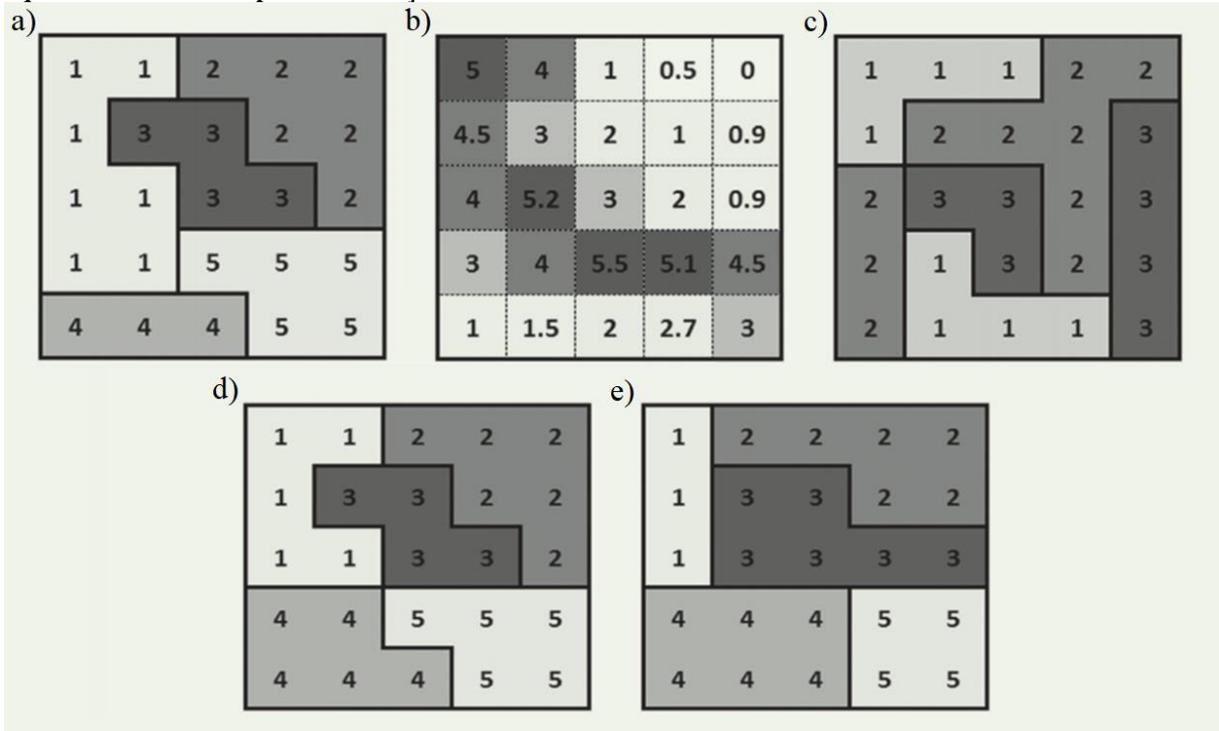
Chanda e Ricciardone (2002) desenvolveram o *software* de sequenciamento de lavra MineMaxTM, que é baseado em *MILP*, que utiliza um algoritmo *BB* para solucionar problemas de planejamento de lavra estratégico multivariados. Os autores propõem uma abordagem de reblocagem baseada em características como teores dos elementos, e como a otimização é realizada bancada a bancada, desconsiderando o ângulo de inclinação e as restrições verticais, as unidades reblocadas podem ser consideradas como polígonos de lavra. Assim, um estudo de caso com 55 cavas foi avaliado, considerando diferentes estratégias de reblocagem e diferentes opções de teores de corte. Samanta, Bhattacharjee e Ganguli (2005), por sua vez, propuseram a utilização de um algoritmo genético (*GA*) para solucionar o problema do sequenciamento estratégico de lavra de um depósito de bauxita ao longo de dois anos, visando minimizar o desvio mensal dos teores de sílica e óxido de alumínio sem violar as restrições operacionais. A jazida tem espessura média de 12 a 14 metros, e informações de 179 sondagens foram utilizadas para gerar 98 camadas horizontais, que podem ser lavradas em cortes únicos ou duplos, sendo a porção inferior do corpo de minério lavrada logo após a parte superior na segunda opção.

Askari-Nasab, Awuah-Offei e Eivazy (2010) apresentaram duas formulações *MILP* para o problema do sequenciamento de lavra no longo prazo, onde empregaram um algoritmo baseado na lógica de agrupamento difuso (*Fuzzy Logic Clustering, FLC*) para agregar blocos de uma mesma bancada em *mining cuts* com base em atributos de localização, litologia e distribuição de teor, reduzindo então o número de variáveis inteiras nas formulações do problema. Em um modelo, a otimização da lavra dá-se em *mining cuts*, enquanto no beneficiamento são considerados os blocos como *input*; já na outra formulação, tanto na lavra quanto no beneficiamento os *mining cuts* servem como *input*. Os modelos propostos foram

implementados em um depósito de minério de ferro contendo informações sobre três tipos de minérios, dois contaminantes e a recuperação estimada por peso magnético, e avaliados no *CPLEX*, cujos sequenciamento de lavra resultante apresentou maior lucratividade que o obtido pelo algoritmo comercial *Whittle Milawa Balanced*.

Askari-Nasab, Tabesh e Badiozamani (2010) introduziram um algoritmo em dois estágios capaz de agregar blocos a fim de diminuir o tamanho de formulações *MILP* para problemas de sequenciamento de lavra no longo prazo. Um algoritmo baseado em *HC* é aplicado na primeira fase, agregando blocos em *mining cuts* por meio de um índice de similaridade baseado nos teores dos blocos, nas litologias e nas distâncias entre blocos. A segunda fase consiste em uma etapa baseada em busca tabu (*Tabu Search, TS*), que tenta diminuir o número de arcos (precedência) entre os novos *clusters* e os das bancadas inferiores, reduzindo o número de variáveis binárias da formulação em uma metodologia guiada de forma ascendente, de modo que o algoritmo se desloca em direção à bancada mais elevada enquanto agregada individualmente cada bancada e a aprimora por meio do *TS*. O algoritmo proposto foi testado e aprovado em dados sintéticos. A Figura 17 apresenta um modelo sintético em que a) equivale aos *clusters* após a clusterização hierárquica, b) consiste na distribuição de teores, c) equivale nos tipos de rochas, d) representa os *clusters* após uma iteração *TS* e e) equivale aos *clusters* após oito iterações *TS*.

Figura 17 - Modelo sintético em que a) equivale aos *clusters* após a clusterização hierárquica, b) consiste na distribuição de teores, c) equivale às litologias, d) representa os *clusters* após uma iteração *TS* e e) equivale aos *clusters* após oito iterações *TS*.



Fonte: Adaptado de ASKARI-NASAB; TABESH; BADIOZAMANI (2010).

Askari-Nasab *et al.* (2010) propuseram um modelo *MILP* baseado na proposição de Askari-Nasab, Awuah-Offei e Eivazy (2010) para resolver grandes problemas de planejamento estratégico de lavra, mas ao invés de empregarem uma etapa *FLC* antes de solucionar o modelo para reduzir o tamanho do problema, os autores integraram uma etapa baseada em *HC* ao fluxo de trabalho de otimização. O algoritmo hierárquico aglomerativo conta com um índice de similaridade semelhante ao proposto por Askari-Nasab, Tabesh e Badiozamani (2010) e segue um fluxo *bottom-up*, no qual cada bancada é agregada individualmente e então, de baixo para cima, o algoritmo prossegue para a bancada seguinte. Os autores apresentaram um estudo de caso no mesmo conjunto de dados de minério de ferro usado por Askari-Nasab, Awuah-Offei e Eivazy (2010), em que os 19.492 blocos dentro da cava final foram agregados em 766 *mining cuts*.

Askari-Nasab *et al.* (2011a) propuseram uma formulação *MILP* para abordar o sequenciamento de lavra operacional com períodos mensais, considerando várias litologias, rotas de beneficiamento, depósitos de estéril, pilhas de homogeneização, acessos e rotas de transporte, tendo como objetivo minimizar os custos relacionados à mineração, ao beneficiamento, transporte, reabilitação e remanuseio, além de atender às restrições de lavra e beneficiamento. No entanto, uma etapa de agregação de blocos foi ainda empregada para

reduzir o número de variáveis binárias no modelo de otimização e gerar sequências de lavra operacionalmente viáveis, onde além de incorporar restrições relacionadas à precedência, lavra e beneficiamento, também foram incorporadas ao modelo as decisões de armazenamento e localização de rampas. Os autores apresentaram um estudo de caso em uma jazida de minério de ferro abordando quatro bancadas ao longo de 12 meses, agregando assim 3.089 blocos em 150 cortes de lavra e resolvendo o modelo no *CPLEX*.

Askari-Nasab *et al.* (2011b) apresentaram quatro formulações baseadas em *MILP* capazes de resolver grandes problemas de sequenciamento de lavra no longo prazo, tendo como objetivo maximizar o VPL sujeito a restrições de capacidade de lavra e beneficiamento, blendagem de teores e precedência. Os dois primeiros modelos consideram o nível de bloco tanto para as operações de lavra quanto para o beneficiamento, onde o primeiro considera apenas variáveis binárias e o segundo considera variáveis binárias e contínuas. No entanto, o terceiro e o quarto modelos consideram variáveis binárias e contínuas, o terceiro abordando a operação de lavra no nível de *mining cuts* e o beneficiamento no nível dos blocos, enquanto o quarto considerando ambas as operações no nível dos polígonos de lavra, que são gerados por meio de um algoritmo *FLC*. Os autores desenvolveram um estudo de caso em uma jazida de minério de ferro para validar os modelos, agrupando 19.492 blocos em 599 cortes de lavra e sequenciando-os ao longo de 12 anos, concluindo que o segundo modelo não é adequado para problemas de longo prazo, sendo mais adequado para problemas de curto prazo, juntamente com o terceiro modelo. Por outro lado, o quarto modelo apresentou bons resultados para cronogramas de vida útil da mina, mesmo em operações a céu aberto de larga escala.

Ben-Awuah e Askari-Nasab (2011) propuseram uma formulação baseada em programação inteira mista de metas (*Mixed Integer Goal Programming, MIGP*) para abordar o sequenciamento de produção de longo prazo de areias betuminosas a partir de uma cava final predeterminada, abrangendo tanto materiais de dique (onde as areias betuminosas se acumulam) quanto estéril. A formulação baseia-se em variáveis binárias inteiras para lidar com a precedência e variáveis contínuas para lidar com a exploração de minério e do material do dique, sendo necessário configurar as variáveis de desvio das metas, os custos de penalidade e as prioridades do modelo. Os diques têm o intuito de conter os rejeitos que são produzidos durante o beneficiamento das areias betuminosas, e são construídos com estéril e areia grossa de rejeitos, oriundos da própria operação, que devem atender às especificações granulométricas desejadas, caso contrário este material será enviado ao depósito de estéril. Antes de aplicar esta modelagem *MIGP*, entretanto, um algoritmo de agrupamento baseado em *FLC* foi empregado, onde blocos em uma mesma bancada são agrupados em *clusters* com

base em atributos de localização, de tipo de rocha e distribuição de teor. O modelo é flexível, uma vez que podem ser feitas compensações entre atingir uma determinada meta e maximizar o VPL, e o estudo de caso apresentado atingiu um cronograma consistente quanto aos requisitos regulatórios e à exploração do depósito de areias betuminosas.

Ben-Awuah, Askari-Nasab e Awuah-Offei (2012) elaboraram um modelo baseado em programação linear inteira mista de metas (*Mixed Integer Linear Goal Programming, MILGP*) para abordar o sequenciamento de lavra de areias betuminosas, sendo uma expansão do apresentado em Ben-Awuah, Askari-Nasab (2011). Dentre os objetivos do modelo, estão (1) determinar a ordem e o tempo de extração do minério, do material de dique e do estéril de modo a maximizar o VPL, 2) determinar a destinação do material do dique que minimize o custo de construção, e 3) minimizar os desvios das metas de produção. O modelo foi empregado após a determinação da cava final e dos *pushbacks* com dimensões adequadas à natureza do projeto, cujos *mining cuts* foram agrupados em cada *pushback* por meio de um *FLC* para reduzir o número de variáveis de decisão do modelo *MILGP*. Os autores resolveram o modelo no *CPLEX*, gerando um sequenciamento de lavra suave e uniforme, que ao mesmo tempo que maximiza o VPL, fornece uma estrutura robusta para o aproveitamento do minério e gerenciamento dos resíduos.

Tabesh e Askari-Nasab (2011) aplicaram a metodologia de agrupamento baseada em *HC* e *TS* proposta por Askari-Nasab, Tabesh e Badiozamani (2010) para em seguida solucionar um modelo baseado em *MILP* para o sequenciamento de lavra estratégico, semelhante ao modelo proposto por Askari-Nasab, Awuah-Offei e Eivazy (2010), que considera o nível de polígonos de lavra para ambas as decisões de lavra e de beneficiamento. Foi apresentado um estudo de caso de uma mina de minério de ferro, cujos resultados mostraram que o tamanho e a forma dos blocos agrupados em *clusters* são responsáveis por 10 a 15% da variação no VPL, também possibilitando melhorias no desempenho operacional e na viabilidade do cronograma de produção. Tabesh e Askari-Nasab (2013) expandiram o estudo anterior ao desenvolver um algoritmo baseado em *HC* guiado de baixo para cima que, além de aplicar um índice de similaridade para gerar *mining cuts*, respeita o tamanho máximo dos *clusters*. Em seguida, é realizada uma etapa de pós-processamento para refinar os *clusters* gerados, removendo arestas e mesclando blocos associados a pequenos *clusters* em outras partições, respeitando assim o tamanho mínimo determinado para todos os *clusters*. Diferentes índices de similaridade foram apresentados, além de uma medida de similaridade baseada na direção preferencial de lavra que também poderia ser adicionada caso o planejamento de produção operacional exija o controle da geometria da mina ou a

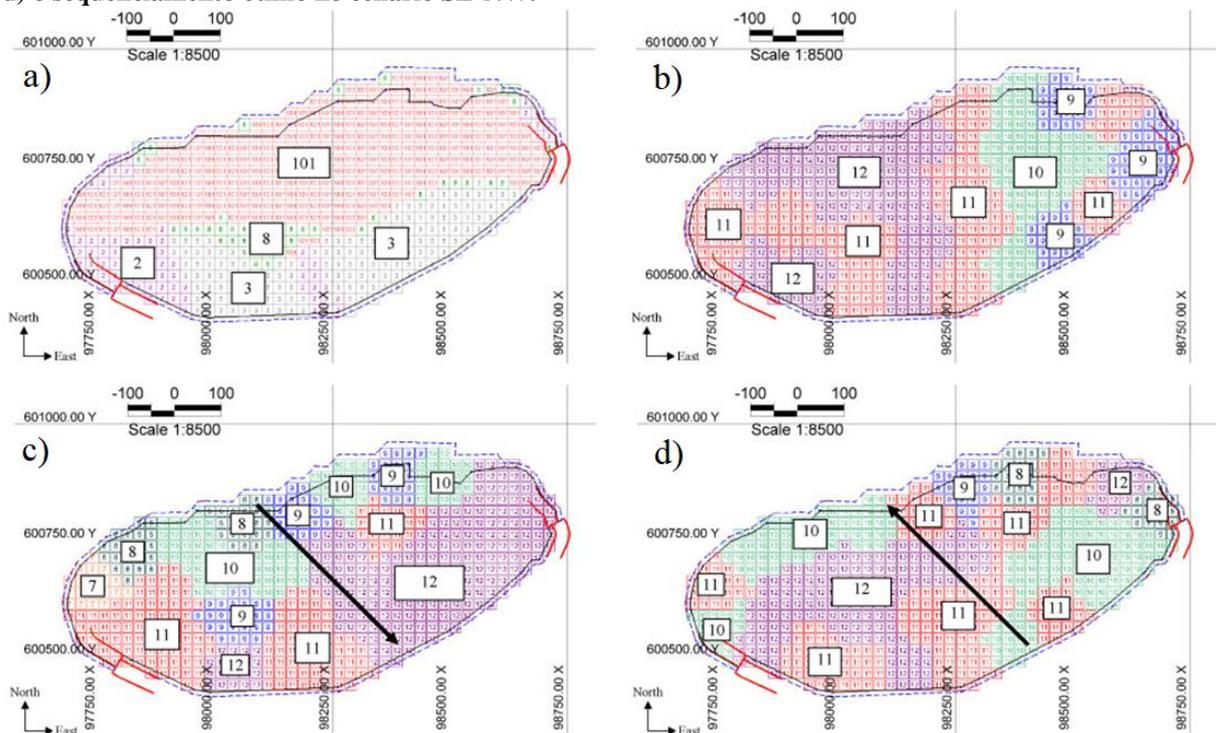
determinação de uma malha de perfuração. Os autores demonstraram a metodologia proposta em três estudos de caso: uma mina de ouro com 90.000 blocos para avaliar os recursos e o tempo de processamento necessários para otimizar um modelo de blocos em tamanho real, uma instância com 874 blocos de uma pequena região dessa mina para ilustrar diferentes características do algoritmo de agrupamento e um conjunto de dados de areias betuminosas com 2.792 blocos.

Badiozamani e Askari-Nasab (2012) propuseram uma formulação *MILP* capaz de integrar o sequenciamento de lavra de areias betuminosas no longo prazo ao gerenciamento dos rejeitos produzidos no processo de extração de betume, tendo como função objetivo a maximização do VPL do sequenciamento de lavra e a minimização dos custos de recuperação do rejeito. Foram consideradas restrições de capacidade de lavra e de beneficiamento, controle de teores e precedência, bem como restrições quanto aos materiais componentes dos rejeitos e à capacidade das instalações que os recuperam. Visando reduzir as variáveis binárias da formulação *MILP*, os blocos foram agrupados em *mining cuts*, embora o método utilizado não tenha sido explicitado. A metodologia proposta foi avaliada em dois *pushbacks* de um depósito real de areia betuminosa, e embora as restrições não tenham sido violadas durante a execução de um algoritmo *BB* em *CPLEX*, os autores indicaram que o tempo de execução até que a solução ótima fosse encontrada foi relativamente grande para o tamanho de problema. Badiozamani e Askari-Nasab (2014) estenderam o estudo anterior incorporando os custos com o manuseio de materiais de recuperação e o gerenciamento de rejeitos ao fluxo da programação da produção. Experimentos numéricos foram realizados no mesmo depósito de areias betuminosas, mas foram verificados quatro tamanhos diferentes de *mining cuts*, resultando em conjuntos de polígonos com 478, 535, 980 e 2.293 unidades, a partir dos quais os autores verificaram que a resolução do modelo aumenta à medida que o número de cortes de mineração aumenta, embora o tempo de solução necessário também seja maior.

Eivazy e Askari-Nasab (2012a) desenvolveram um modelo *MILP* multideestino para resolver o sequenciamento de minas de curto prazo com objetivo de minimizar os custos totais, como custos de lavra, beneficiamento, transporte, reabilitação e remanejamento. Resolvido por um algoritmo *BC*, o modelo incorpora como restrições operacionais pilhas de homogeneização e amortização, uma direção horizontal preferencial de lavra e decisões de localização de rampa. Para reduzir o número de variáveis binárias na formulação *MILP* e para gerar um sequenciamento prático e adequado às *SMUs*, os blocos foram agrupados em *mining cuts* por meio do método *Fuzzy C-means*. Um estudo de caso foi desenvolvido em uma mina de minério de ferro, onde foram avaliadas três direções horizontais de lavra distintas,

considerando o sequenciamento de lavra ao longo de 12 meses, sendo escolhido como melhor cenário aquele que apresentou maior viabilidade operacional, mesmo que outro tenha obtido custos menores. A Figura 18 apresenta a bancada estudada pelos autores, contendo uma diferenciação entre as zonas de estéril e mineralizada, bem como os *mining cuts* formados nos três cenários avaliados.

Figura 18 - Vista em planta de uma bancada mostrando a) a diferenciação entre as zonas de estéril e mineralizada, b) o sequenciamento ótimo no cenário base, c) o sequenciamento ótimo no cenário NW-SE e d) o sequenciamento ótimo no cenário SE-NW.



Fonte: Adaptado de EIVAZY; ASKARI-NASAB (2012).

Eivazy e Askari-Nasab (2012b) propuseram um fluxo de trabalho para solucionar o problema do planejamento operacional de lavra alinhado com os objetivos de longo prazo, onde primeiro os blocos são agregados em polígonos de lavra por um algoritmo *HC-TS*, então o sequenciamento estratégico é efetuado através de um modelo baseado em *MILP* inspirado em Askari-Nasab, Awuah-Offei e Eivazy (2010) e Askari-Nasab, Tabesh e Badiozamani (2010), culminando na resolução do problema operacional para períodos mensais. O modelo baseado em *MILP* para o curto prazo é uma extensão das proposições de Askari-Nasab *et al.* (2011a) e Eivazy e Askari-Nasab (2012a), e visa minimizar os custos totais sujeitos a restrições semelhantes às apresentadas nesses modelos, como blendagem, decisões de localização de rampas e restrições de longo prazo. Foi apresentado um estudo de caso de uma

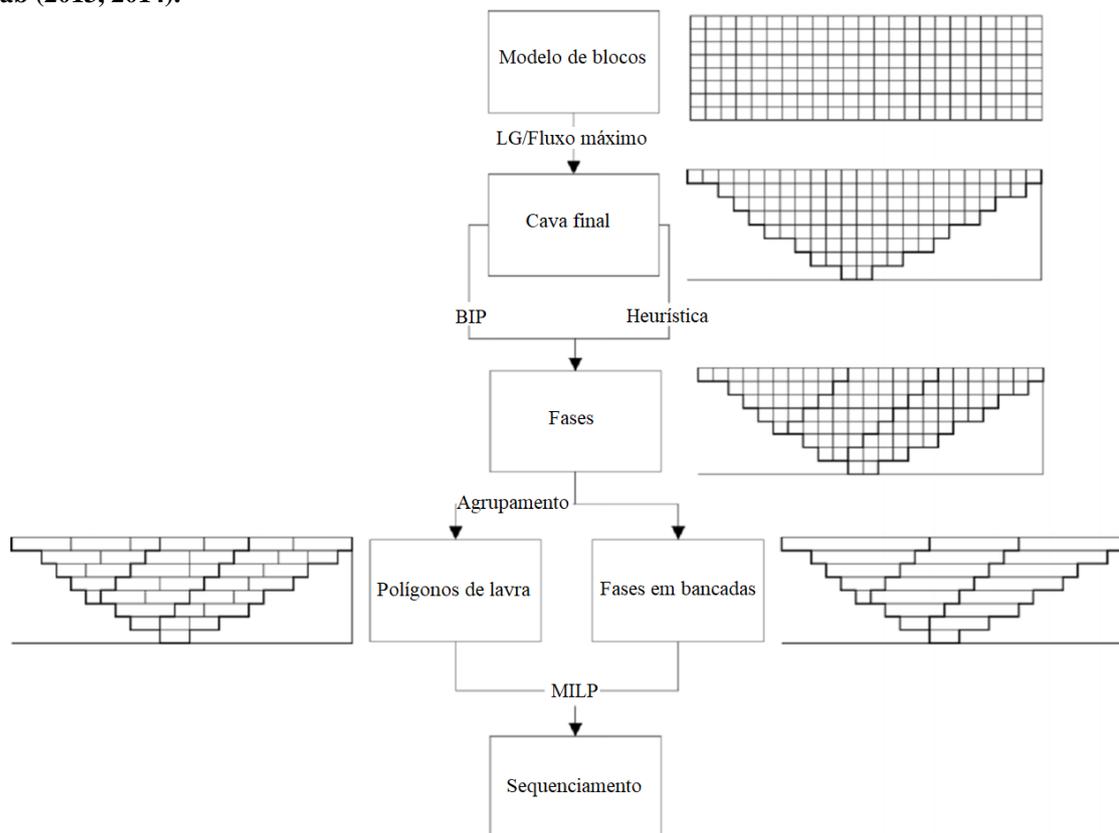
mina de minério de ferro no Irã, no qual 17 tamanhos de *clusters* foram avaliados para o problema no longo prazo, enquanto cenários com diferentes resoluções de polígonos e direções de lavra foram submetidos ao sequenciamento no curto prazo, cujos resultados foram em seguida discutidos.

Tabesh, Mieth e Askari-Nasab (2013) propuseram uma metodologia multiestágio para resolver problemas de planejamento de longo prazo considerando uma cava final predeterminada e modelos de sequenciamento baseados em *MILP*. Primeiro, um modelo baseado em *IP* determina os *pushbacks* dentre as cavas aninhadas geradas a partir da parametrização do problema da cava final, tendo como objetivo maximizar o retorno econômico dos *pushbacks* e considerando restrições relacionadas à precedência, ao ângulo de talude, à massa de rocha e de minério dentro de cada *pushback*. Portanto, técnicas como a relaxação da Programação Linear ou a relaxação Lagrangiana podem ser utilizadas para simplificar este modelo *IP*, que pode ser resolvido por um *solver MILP* ou por uma heurística. Em seguida, uma etapa de agregação baseada em *HC-TS* é usada para gerar polígonos de lavra dentro de cada *pushback*, de modo que o modelo de sequenciamento de mina depende dos níveis de *pushback*, *mining cuts* e painéis (fases em bancadas ou *bench-phases*). Este modelo de sequenciamento é capaz de determinar internamente o teor de corte, tendo como objetivo maximizar a receita de beneficiamento dos polígonos de lavra subtraída dos custos de extração de painéis, sujeito a restrições relacionadas às capacidades de lavra e beneficiamento, ao controle de teores e ângulo de talude. Foi apresentado um estudo de caso de uma jazida de minério de ferro com 19.561 blocos, enfatizando que a abordagem proposta é capaz de gerar melhores *pushbacks* que os *softwares* comerciais, além de aumentar em 2-4% o VPL quando comparada com a abordagem tradicional de sequenciamento de minas.

Askari-Nasab, Mieth e Tabesh (2013) discutem a mesma metodologia, apresentando mais detalhes sobre as heurísticas de busca gulosa e busca local desenvolvidas para resolver o problema de determinação de *pushbacks*. Foi apresentado um estudo de caso de uma jazida de minério de ferro com aproximadamente 177.000 blocos dentro da cava final predeterminada, onde diferentes configurações de *pushback* foram avaliadas ao longo dos 35 anos de vida útil da mina, resultados que foram comparados aos obtidos por um software comercial. Tabesh, Mieth e Askari-Nasab (2014) aprofundam a discussão adicionando mais detalhes sobre a etapa de redução do problema de geração de *pushback*, bem como as técnicas empregadas para reduzir o problema do sequenciamento de produção, como a remoção de variáveis desnecessárias e restrições redundantes. Os autores também apresentaram o estudo de caso anterior e outro depósito de minério de ferro, este contendo 277.000 blocos na cava final, no

qual foi apresentado o sequenciamento da mina considerando diferentes quantidades de *pushbacks*. Finalmente, Tabesh, Askari-Nasab e Peroni (2015) propuseram uma extensão deste problema de sequenciamento para incluir pilhas de estoque, transformando o modelo matemático baseado em *MILP* em *MINLP*, apresentando ainda uma versão linearizada deste modelo. A Figura 19 apresenta o fluxograma da metodologia de sequenciamento de lavra sugerida por Tabesh, Mieth e Askari-Nasab (2013, 2014).

Figura 19 - Fluxograma da metodologia de sequenciamento de lavra sugerida por Tabesh, Mieth e Askari-Nasab (2013, 2014).



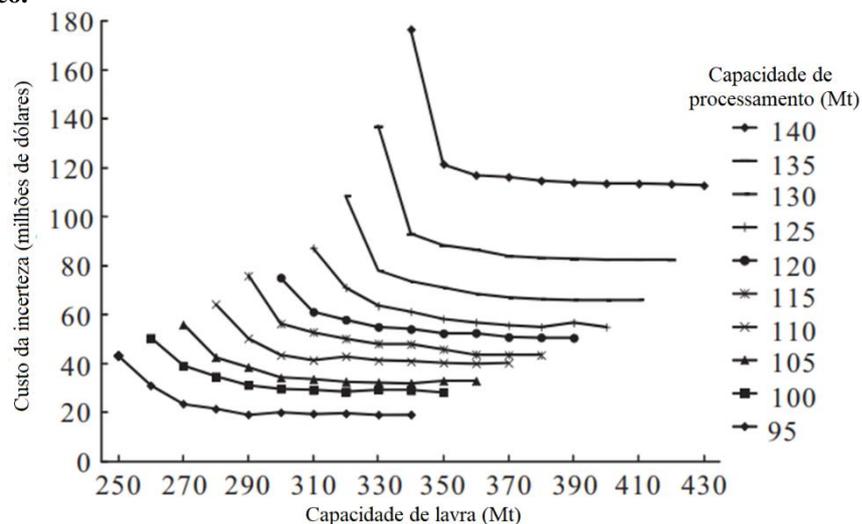
Fonte: adaptado de TABESH; MIETH; ASKARI-NASAB (2014).

**BIP*: Binary Integer Programming.

Koushavand, Askari-Nasab e Deutsch (2014) propuseram calcular o custo da incerteza no sequenciamento de lavra estocástico no longo prazo considerando os desvios da produção alvo. Um modelo *MILP* baseado em Askari-Nasab, Awuah-Offei e Eivazy (2010) e Askari-Nasab *et al.* (2011b) é formulado para encontrar a sequência de blocos a serem extraídos de uma cava final pré-definida e seus respectivos destinos com dois objetivos: maximizar o VPL de um modelo de estimativa (*E-type* das simulações) para minimizar o custo da incerteza das múltiplas realizações estocásticas. A função objetivo está sujeita a restrições relacionadas à blendagem de teores, capacidade de lavra e beneficiamento, reserva, precedência, restrições

de porção de lavrada e processada, limites superiores e inferiores de produção e de material na pilha de estoque, sendo o custo da incerteza modelado como uma função de penalidade assimétrica, onde o custo de produção abaixo do limite mínimo é maior que o custo de superprodução. Uma técnica de agrupamento baseada em *Fuzzy C-means* é empregada para reduzir o número de variáveis e tornar o problema tratável. Quando comparado a uma otimização formulada como *LP* em um depósito de areia betuminosa no Canadá, considerando 50 realizações do corpo de minério, o método proposto resultou em um VPL um pouco menor devido à necessidade de adicionar novas restrições à formulação do problema, ao passo que reduziu o desvio dos alvos de produção. A Figura 20 apresenta um exemplo de custos da incerteza *versus* diferentes capacidades de lavra e beneficiamento em um caso sintético.

Figura 20 - Exemplo de custos da incerteza versus diferentes capacidades de lavra e beneficiamento em um caso sintético.



Fonte: adaptado de KOUSHAVAND; ASKARI-NASAB; DEUTSCH (2014).

Upadhyay *et al.* (2015a, 2015b) propuseram um modelo baseado em *MILGP* para abordar o sequenciamento de lavra e alocação de escavadeira em problemas de planejamento de lavra no curto prazo, tendo como objetivo maximizar a utilização da escavadeira enquanto minimiza o custo operacional, desvios de teor e de tonelagem. As restrições são relacionadas à escavadeira, à frota de caminhões, à capacidade de beneficiamento, ao controle de teores, à atribuição da escavadeira e da frota às faces de lavra, à combinação de equipamentos, ao desvio de produção e à taxa de decapeamento. As faces de lavra correspondem a polígonos de lavra derivados do problema de sequenciamento de lavra conforme a metodologia proposta por Tabesh, Mieth e Askari-Nasab (2014), e o *output* do estágio de alocação da escavadeira

serve como *input* para um modelo baseado em simulação por eventos discretos (*Discrete Event Simulation, DES*) que visa avaliar o impacto de mudanças operacionais na produção de minério e estéril, bem como na utilização de recursos. Considerando um estudo de caso de uma mina de areias betuminosas de grande porte, as etapas de otimização foram realizadas separadamente, onde o modelo baseado em *MILGP* considerou turnos individualizados de produção e um sistema predefinido, enquanto o modelo baseado em *DES* foi validado contra dados históricos. Upadhyay e Askari-Nasab (2016), por sua vez, aplicaram o modelo baseado em *MILGP* em um estudo de caso de uma jazida de minério de ferro no Irã, utilizando frações de 2.904 blocos em 3 bancadas. O sistema foi solucionado para durações de meia hora até 12 horas por dia, onde as taxas de produção diárias foram armazenadas por um mês, alcançando-se assim uma utilização média da planta, utilização média de caminhões e utilização média da escavadeira de 99%, 92% e 95%, respectivamente.

Upadhyay e Askari-Nasab (2019) estenderam a metodologia proposta nos estudos anteriores ao propor uma ferramenta dinâmica para alocação de escavadeiras, capaz de atribuí-las a faces de lavra em um fluxo de trabalho integrado que engloba o modelo baseado em *MILGP* e estágios baseados em *DES*. Foi apresentado um estudo de caso de uma mina de minério de ferro, tal que um ano contendo 4.227 blocos agrupados em 174 faces de lavra distribuídas em quatro bancadas foi extraído do plano de longo prazo e submetido à metodologia proposta, considerando períodos de 12 horas e cenários cujas rotas de beneficiamento apresentavam prioridades iguais ou diferentes. Além disso, Upadhyay e Askari-Nasab (2018) ampliaram este escopo ao desenvolver uma estrutura de otimização de simulação que simula planos de curto prazo com base na alocação de caminhões e atribuição de escavadeiras para faces de lavra, respeitando também as restrições do plano estratégico. Os autores resolvem problemas de planejamento operacional multiobjetivo incorporando o modelo baseado em *MILGP* e os estágios *DES* em uma única ferramenta capaz de gerar planos de lavra que consideram incertezas, proporcionando assim uma tomada de decisão operacional dinâmica em tempo real, atendendo também as demandas estratégicas. Foi apresentado um estudo de caso na mesma mina de minério de ferro, abrangendo dois cenários com diferentes configurações de caminhões, cada um abarcando 10 realizações de operações durante 6 meses.

Tabesh e Askari-Nasab (2019) propuseram quatro algoritmos de agrupamento de blocos em polígonos de lavra contemplando os controles de tamanho e forma de *clusters* implementados por Tabesh *et al.* (2013) e possibilitando a geração de soluções sob incerteza geológica. O primeiro algoritmo foi baseado em um modelo de estimação com propriedades

determinísticas, e os outros três, incluindo um algoritmo *K-means-HC*, foram baseados em modelos e propriedades estocásticas do depósito. Nos casos probabilísticos, com base na abordagem dos mundos possíveis, realizações equiprováveis de um depósito são construídas por meio de simulação geoestatística, onde os algoritmos de agrupamento operam independentemente em cada realização e, finalmente, uma técnica de agregação é usada para extrair um único modelo de agrupamento das soluções individualizadas de todas as realizações. As metodologias foram implementadas em dados de uma jazida de cobre com 189.000 blocos e três tipos de litologias, gerando 20 realizações de litologias e teores, além de um banco de dados multivariado contendo 570.770 blocos distribuídos em 176 bancadas, cujos teores e litologias foram simulados em 100 realizações. Os resultados indicam que os algoritmos probabilísticos podem resultar em *clusters* menos suscetíveis às incertezas, enquanto as dimensões e formas dos *clusters* permaneceram dentro das restrições pré-determinadas.

Capponi (2019) propôs uma metodologia para abordar o problema dos limites de escavação em três etapas: primeiro, o problema *grade control* é abordado por meio de simulações geoestatísticas em furos de detonação; segundo, um algoritmo *Self-Organizing Map* baseado nas redes neurais de Kohonen é empregado na determinação dos *mining cuts*; por fim, um algoritmo *GA* com função objetivo de maximizar o lucro e minimizar os desvios das metas de teores é empregado na definição do limite de escavação (*dig limit*). Os mapas de Kohonen são redes neurais de aprendizado não-supervisionado dotadas de uma camada para a entrada dos dados (vetores) e uma camada de saída (*grade 2D*) totalmente conectadas, onde os neurônios de saída competem entre si para assumir os maiores pesos em determinada época por meio do treinamento de um vetor peso sináptico com mesma dimensão do espaço de entrada, onde o neurônio vencedor impacta no resultado de seus vizinhos. Ao longo das iterações, a vizinhança e a taxa de aprendizagem são reduzidas para forçar a convergência, progredindo de uma competição em que quase todo neurônio era atualizado para uma em que somente o vencedor é atualizado. Contemplando a extração simultânea de diversos *mining cuts* para atender às restrições de beneficiamento, a metodologia foi empregada no sequenciamento semanal de uma mina de fosfato, cujos resultados obtidos demonstraram o aumento do lucro e a redução da variabilidade do minério quando comparado ao método manual.

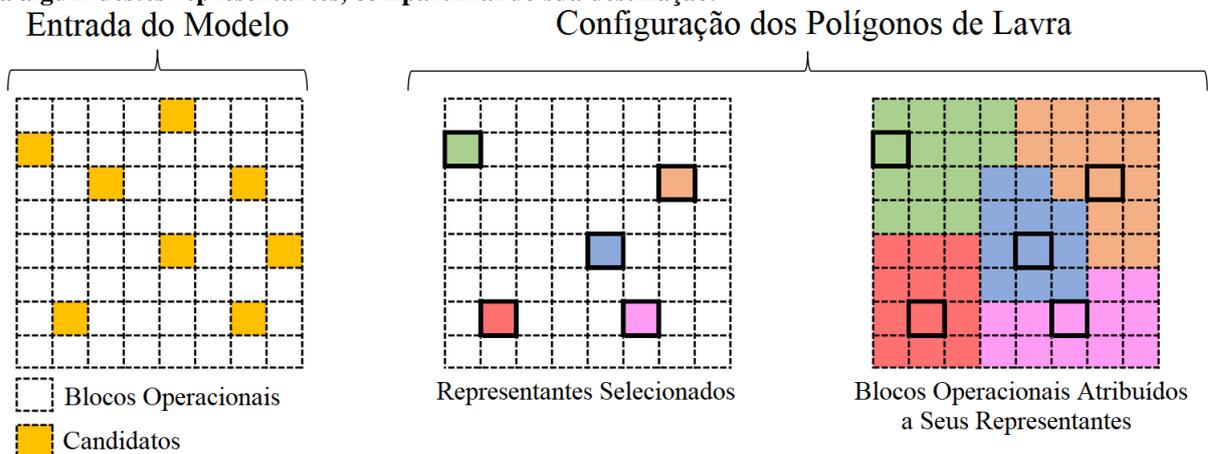
Salman *et al.* (2021) propuseram uma heurística multiestágio de agrupamento de blocos para determinação de *mining cuts*, cuja primeira etapa consiste em um algoritmo *K-means* para identificar classes geoquímicas, ao passo que uma segunda etapa agrupa blocos

adjacentes com base em um critério de similaridade. A etapa *K-means*, portanto, identifica os centróides dos *clusters* de classes de teores abrangendo blocos que excedam o teor de corte, sendo o número de classes geoquímicas em determinada bancada definido pela avaliação do coeficiente de silhueta. Em seguida, são empregadas *Fuzzy membership functions* para determinar os limites de transição entre as classes de teores, onde as *Triangular membership functions* representam a pertinência de um bloco pertencer cada uma das classes geoquímicas, sendo os blocos de estéril atribuídos a uma única classe de estéril. A similaridade entre blocos adjacentes considerando litologias, teores e destinos é então calculada, onde os blocos adjacentes com maior similaridade formarão serão agregados repetidamente em *clusters* até que não haja mais blocos a serem agrupados ou o tamanho máximo dos *clusters* tenha sido atingido. Finalmente, uma etapa de *post-processing* é empregada, onde blocos não agrupados e *clusters* muito pequenos são incorporados ao *cluster* mais adequado baseando-se na direção preferencial da lavra, bem como é aplicada uma etapa de correção da geometria dos *clusters*. A aplicação da metodologia proposta em dois *datasets* do *MineLib* (ESPINOZA *et al.*, 2013) demonstrou que foram obtidos polígonos de lavra com dimensões adequadas e que as restrições foram respeitadas.

Nelis e Morales (2022) propuseram um modelo de otimização capaz de resolver simultaneamente os problemas da determinação de polígonos de lavra viáveis, baseados em blocos de curto prazo criados a partir da informação de furos de detonação, e do sequenciamento de lavra operacional, visando maximizar seu retorno econômico. O modelo seleciona quais blocos operacionais devem constituir determinado *mining cut*, respeitando tamanhos mínimos e máximos, atribuindo uma destinação e um período de extração para cada um, ao passo que respeita restrições de capacidade de lavra e de beneficiamento, blendagem, ângulo de talude e precedência. Os autores propuseram uma abordagem que permite a segmentação do problema do agrupamento em duas partes: primeiro, um determinado conjunto de blocos operacionais é listado como candidatos, onde alguns destes são eleitos representantes de polígonos de lavra, e segundo, cada bloco operacional restante é atribuído a um dos representantes. Os blocos restantes serão agregados aos polígonos de lavra cujos representantes possuem arcos direcionados mais favoráveis a cada um, de acordo com a estratégia de precedência adotada, controlando assim a forma e a conectividade dos polígonos de lavra. Os blocos associados a cada polígono de lavra, então, irão compartilhar as mesmas destinações e períodos de extração que seus representantes, sendo possível determinar um número máximo de *mining cuts* ativos por período para que seja possível restringir as frentes de lavra operadas simultaneamente. Experimentos numéricos foram apresentados, em que

duas bancadas de uma mina de cobre no Chile resultaram em *mining cuts* e sequências de lavra operacionais, respeitando as restrições impostas, incluindo restrições de acesso, como a localização de rampas em cada bancada. A Figura 21 apresenta a definição dos polígonos de lavra conforme a metodologia proposta pelos autores, onde uma lista de blocos operacionais candidatos a representantes é determinada e alguns destes são eleitos como representantes dos polígonos de lavra, expressos na imagem em diferentes cores.

Figura 21 - Definição de polígonos de lavra com base na metodologia proposta por Nelis e Morales (2022), onde uma lista de blocos candidatos a representantes é determinada e alguns destes são eleitos como representantes dos *mining cuts*, expressos em diferentes cores. Em seguida, todos os blocos são associados a algum destes representantes, compartilhando sua destinação.



Fonte: adaptado de NELIS; MORALES (2022).

Nelis, Meunier e Morales (2022) propuseram um modelo baseado em *MILP* para resolver o problema de agrupamento de polígonos de lavra que visa maximizar o lucro total de atribuir uma destinação a cada *cluster* gerado, empregando para tal um algoritmo de geração de colunas (*Column Generation Algorithm, CGA*) e relaxação da programação linear para resolver o modelo. A técnica de geração de colunas resolve problemas lineares contínuos, começando a partir de um pequeno subconjunto restrito de variáveis (o problema mestre), então um relaxamento duplo do problema mestre (o subproblema de precificação) é resolvido iterativamente, usando os custos reduzidos das colunas para tomar decisões, encontrando assim novas colunas (variáveis) e adicionando-as ao problema mestre. Depois de resolver o problema relaxado até a otimalidade, uma etapa adicional deve ser empregada para convertê-la a uma solução inteira. Foi apresentado um estudo de caso em dados de furo de detonação em uma mina de cobre para validar a metodologia proposta, abrangendo quatro bancadas com tamanhos diferentes, entre 216 a 912 blocos.

Diversas outras soluções de agregação de blocos foram apresentadas, embora não focassem especificamente no agrupamento de blocos em *mining cuts*, como em Smith (1999), Ramazan (2001), Menabde *et al.* (2004), Stone *et al.* (2004), Ramazan, Dagdelen e Johnson (2005), Zhang (2006), Ramazan (2007), Boland *et al.* (2009), Yavarzadeh, Abodallheisharif e Neishabouri (2014), Smith e Wicks (2014), Del Castillo e Dimitrakopoulos (2016), Jélvez *et al.* (2016), Goodfellow e Dimitrakopoulos (2016), Mai, Topal e Erten (2018), Mai *et al.* (2019), Lotfian, Gholamnejad e Mirzaeian Lardkeyvan (2021). Podem-se citar ainda, no âmbito da agregação de blocos e criação de *clusters* para o sequenciamento de lavra, as contribuições de Weintraub, Pereira e Schultz (2008), Epstein *et al.* (2012), Pourrahimian, Askari-Nasab e Tannant (2012, 2013), Nezhadshahmohammad, Pourrahimian e Aghababaei (2018) e Nezhadshahmohammad e Pourrahimian (2019), embora estes últimos também não sejam aqui detalhados por lidarem com métodos de lavra subterrâneos, o que está além do escopo proposto.

A Tabela 1, finalmente, com o intuito de auxiliar uma eventual consulta, apresenta um resumo dos estudos apresentados neste tópico bem como seus principais aspectos relacionados aos problemas do agrupamento e do sequenciamento de lavra. Em alguns dos estudos citados há a proposição e a comparação de duas técnicas de resolução do problema do sequenciamento de lavra, por esta razão há duas colunas direcionadas aos algoritmos empregados no sequenciamento. Quando há a indicação de um algoritmo exato, este pode ser um algoritmo da família *BB* criado especificamente para o estudo, ou mesmo um *solver* comercial como o Gurobi e o *CPLEX*. Nos itens representados por ‘-’, ou o caso não se aplica ou a informação não pôde ser obtida no estudo.

Tabela 1 - Resumo dos estudos apresentados neste tópico, contemplando os principais aspectos relacionados aos problemas do agrupamento e do sequenciamento de lavra.

Autores (Ano de publicação)	Horizonte temporal	Banco de dados	Técnica de agrupamento	Algoritmo de agrupamento	Modelo de sequenciamento	Algoritmo de sequenciamento	Algoritmo de sequenciamento
						1	2
Albach (1967)	Estratégico	Carvão	Camadas	-	CCP	Exato	-
Metz e Jain (1978)	Estratégico	Fosfato	Camadas	-	DP/MINLP	DP	Heurística
Gershon (1983)	Estratégico	-	Camadas	-	LP	-	-
Busnach, Mehrez e Sinuany-Stern (1985)	Estratégico	Fosfato	Camadas	-	MINLP	Heurística	-
Klingman e Phillips (1988)	Estratégico	Fosfato	Camadas	-	IP	Exato	-
Gershon e Murphy (1989)	Estratégico	Xisto betuminoso	Camadas	-	DP	DP	-
Chanda e Wilke (1992)	Operacional	Cobre	Camadas	-	LGP	Exato	Heurística
Chanda e Ricciardone (2002)	Estratégico	Multivariado	Reblocagem	-	MILP	MineMax	-
Samanta, Bhattacharjee e Ganguli (2005)	Operacional	Bauxita	Camadas	-	-	GA	-
Askari-Nasab, Awuah-Offei e Eivazy (2010)	Estratégico	Minério de ferro	Otimização/blocos	FLC	MILP	Exato	-
Askari-Nasab, Tabesh e Badiozamani (2010)	Estratégico	Sintético	Otimização/blocos	HC-TS	MILP	Exato	-
Askari-Nasab <i>et al.</i> (2010)	Estratégico	Minério de ferro	Otimização/blocos	HC	MILP	Exato	-
Askari-Nasab <i>et al.</i> (2011a)	Operacional	Minério de ferro	Otimização/blocos	-	MILP	Exato	-
Askari-Nasab <i>et al.</i> (2011b)	Estratégico	Minério de ferro	Otimização/blocos	FLC	MILP	Exato	-
Ben-Awuah e Askari-Nasab (2011)	Estratégico	Areias betuminosas	Otimização/blocos	FLC	MILGP	Exato	-
Ben-Awuah, Askari-Nasab e Awuah-Offei (2012)	Estratégico	Areias betuminosas	Otimização/blocos	FLC	MILGP	Exato	-
Tabesh e Askari-Nasab (2011)	Estratégico	Minério de ferro	Otimização/blocos	HC-TS	MILP	Exato	-
Tabesh e Askari-Nasab (2013)	Ambos	Ouro/Areias betuminosas	Otimização/blocos	HC-TS	-	-	-
Badiozamani e Askari-Nasab (2012)	Estratégico	Areias betuminosas	Otimização/blocos	HC-TS	MILP	Exato	-
Badiozamani e Askari-Nasab (2014)	Estratégico	Areias betuminosas	Otimização/blocos	HC-TS	MILP	Exato	-
Eivazy e Askari-Nasab (2012a)	Operacional	Minério de ferro	Otimização/blocos	FCM	MILP	Exato	-
Eivazy e Askari-Nasab (2012b)	Operacional	Minério de ferro	Otimização/blocos	HC-TS	MILP	Exato	-
Tabesh, Mieth e Askari-Nasab (2013)	Estratégico	Minério de ferro	Otimização/painéis	HC-TS	MILP	Exato	-
Askari-Nasab, Mieth e Tabesh (2013)	Estratégico	Minério de ferro	Otimização/painéis	HC-TS	MILP	Exato	-
Tabesh, Mieth e Askari-Nasab (2014)	Estratégico	Minério de ferro	Otimização/painéis	HC-TS	MILP	Exato	-
Tabesh, Askari-Nasab e Peroni (2015)	Estratégico	-	Otimização/painéis	HC-TS	MILP/MINLP	-	-
Koushavand, Askari-Nasab e Deutsch (2014)	Estratégico	Areias betuminosas	Otimização/blocos	FCM	MILP	Exato	-
Upadhyay <i>et al.</i> (2015a)	Operacional	Areias betuminosas	Otimização/blocos	HC-TS	MILGP/DES	-	Simulação
Upadhyay <i>et al.</i> (2015b)	Operacional	Areias betuminosas	Otimização/blocos	HC-TS	MILGP/DES	-	Simulação

Autores (Ano de publicação)	Horizonte temporal	Banco de dados	Técnica de agrupamento	Algoritmo de agrupamento	Modelo de sequenciamento	Algoritmo de sequenciamento 1	Algoritmo de sequenciamento 2
Upadhyay e Askari-Nasab (2016)	Operacional	Minério de ferro	Otimização/blocos	<i>HC-TS</i>	<i>MILGP/DES</i>	-	Simulação
Upadhyay e Askari-Nasab (2019)	Operacional	Minério de ferro	Otimização/blocos	<i>HC-TS</i>	<i>MILGP/DES</i>	Exato	Simulação
Upadhyay e Askari-Nasab (2018)	Operacional	Minério de ferro	Otimização/blocos	<i>HC-TS</i>	<i>MILGP/DES</i>	Exato	Simulação
Tabesh e Askari-Nasab (2019)	Estratégico	Cobre/Multivariado	Otimização/blocos	<i>K-means-HC</i>	-	-	-
Capponi (2019)	Operacional	Fosfato	Otimização/ <i>SMUs</i>	<i>SOM</i>	-	<i>GA</i>	-
Salman <i>et al.</i> (2021)	Operacional	<i>MineLib</i>	Otimização/blocos	<i>K-means / FMF</i>	-	-	-
Nelis e Morales (2022)	Operacional	Cobre	Otimização/ <i>SMUs</i>	<i>MILP-Exato</i>	<i>MILP</i>	Exato	-
Nelis, Meunier e Morales (2022)	Operacional	Cobre	Otimização/ <i>SMUs</i>	<i>MILP-CGA</i>	-	-	-

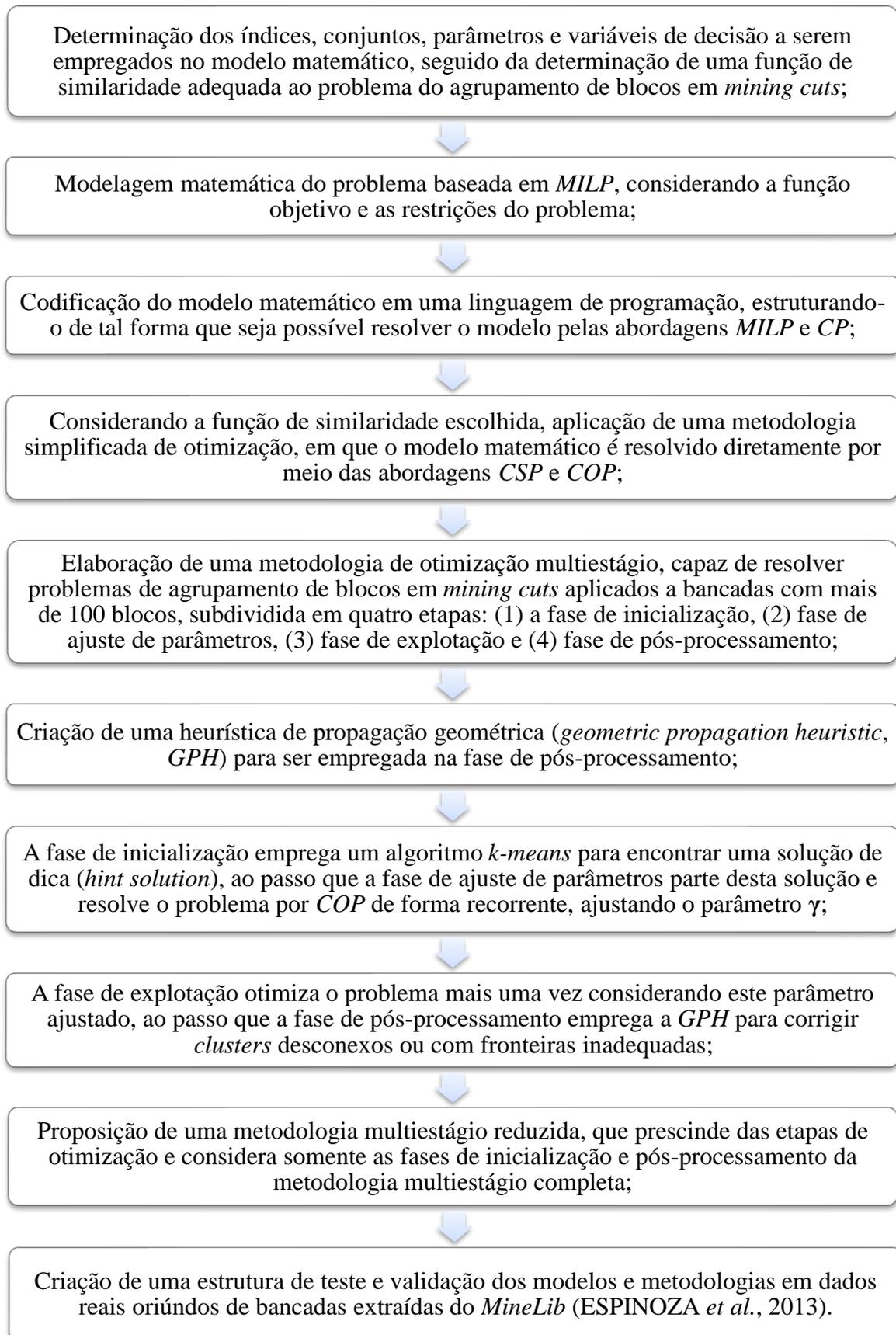
3 METODOLOGIA

Ao longo do desenvolvimento deste estudo foram elaborados modelos matemáticos do problema do agrupamento de polígonos de lavra condizentes com as abordagens *MILP* e *CP*, ambos resolvidos através das bibliotecas do *OR-Tools* na linguagem *Python* por meio dos solvers *SCIP* e *CpSolver*, respectivamente. Quando não foi possível obter uma solução exata do problema através da abordagem *MILP*, foi empregada a abordagem *CSP* para possibilitar a obtenção de uma amostra do espaço de busca do problema, servindo assim como parâmetro para avaliar a qualidade da solução obtida por *COP*.

Visto que a abordagem *CP* se baseia em otimização inteira, todas as variáveis empregadas foram inteiras, bem como o cálculo da função objetivo foi arredondado em números inteiros para que o *solver* pudesse resolver o modelo. O arquivo de saída da otimização consiste em um *array* representando os blocos das bancadas estudadas, contendo valores inteiros que correspondem à associação de um *cluster* a cada um destes blocos. Finalmente, foram computados o tempo de processamento e a função objetivo atingida por cada instância submetida à otimização, bem como parâmetros relativos à qualidade dos *clusters* (*e.g.* índice da silhueta) e indicadores-chave de desempenho no planejamento de lavra (*e.g.* valor presente) foram calculados.

Uma vez escolhida a instância de teste, é necessário efetuar um pré-processamento dos dados, contemplando a extração das bancadas de interesse, re-indexação e o ordenamento dos blocos, criação de novas variáveis de interesse, dentre outras alterações. Devem-se implementar parâmetros (*e.g.* tamanho máximo de um *cluster*), funções auxiliares (*e.g.* distância Euclidiana entre blocos), listas e dicionários (*e.g.* lista de blocos adjacentes a determinado bloco) para permitir a correta modelagem matemática do problema e sua solução. Blocos que não atingem um critério mínimo de vizinhança devem ser excluídos das instâncias, visto que a manutenção de blocos isolados impediria a convergência durante a etapa de otimização e comprometeria a operacionalidade da solução. A Figura 22 apresenta um fluxograma que sintetiza os procedimentos empregados nos experimentos deste estudo no âmbito da otimização do problema do agrupamento de polígonos de lavra.

Figura 22 - Fluxograma sintetizando os procedimentos empregados neste estudo no âmbito da otimização do problema do agrupamento para geração de polígonos de lavra.



3.1 MODELO MATEMÁTICO DO PROBLEMA DO AGRUPAMENTO DE POLÍGONOS DE LAVRA

Neste tópico, será apresentado o modelo matemático baseado em *MILP* elaborado para representar o problema do agrupamento de polígonos de lavra, incluindo seus índices, conjuntos, parâmetros, variáveis de decisão, função objetivo e restrições.

3.1.1 Índices e conjuntos

Para que seja possível compreender o modelo matemático do problema do agrupamento de polígonos de lavra que será apresentado mais adiante, deve-se primeiro apresentar os índices e conjuntos que compõem este modelo, podendo-se então listar os seguintes elementos:

I: Conjunto de blocos em uma bancada (o domínio do problema);

J: Conjunto de partições que, quando somadas, conforme a Equação (2.13), equivalem ao domínio do problema;

i: Índice de um bloco ($i = 1, \dots, m, \forall i \in I$);

j: Índice de um *cluster* ($j = 1, \dots, n, \forall j \in J$);

b, c: Índices representando dois blocos em um mesmo domínio *I* ($b, c = 1, \dots, m, \forall b, c \in I$);

adj4_b: Conjunto de blocos adjacentes ao bloco ***b*** nas direções Norte-Sul e Leste-Oeste ($b = 1, \dots, m, \forall b \in I$);

adj8_b: Conjunto de blocos adjacentes ao bloco ***b*** nas direções Norte-Sul, Leste-Oeste, Nordeste-Sudoeste e Noroeste-Sudeste ($b = 1, \dots, m, \forall b \in I$);

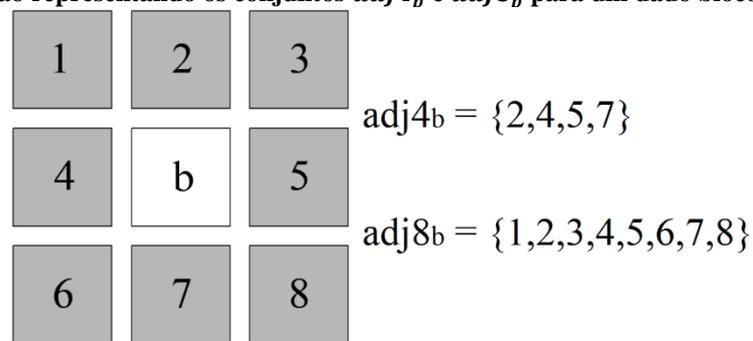
Dist_{b,c}: Matriz que computa a distância Euclidiana entre os blocos ***b*** and ***c*** ($b, c = 1, \dots, m, \forall b, c \in I$).

S_{b,c}: Conjunto ou matriz que computa a função de similaridade entre os blocos ***b*** and ***c*** ($b, c = 1, \dots, m, \forall b, c \in I$);

A matriz ***Dist_{b,c}*** computa a distância Euclidiana entre todos os pares de blocos do domínio, em que dois blocos ***b, c*** $\in I$, possibilitando então implementar uma restrição equivalente ao limite de diâmetro máximo de *cluster*. Por outro lado, a matriz ***S_{b,c}*** computa o valor do critério de similaridade entre os todos blocos ***b, c*** $\in I$, critério este que possibilita o cálculo da função objetivo do modelo matemático e que será aprofundado mais adiante. Já para formar os conjuntos ***adj4_b*** e ***adj8_b***, todos os blocos vizinhos de ***b*** e localizados nas

direções supracitadas, em que $b \in I$, foram armazenados em dicionários. O modelo matemático irá introduzir uma restrição para cada um destes critérios de vizinhança, tal que nenhum bloco b possa pertencer a um *cluster* caso não haja pelo menos um bloco em $adj4_b$ e dois em $adj8_b$ que pertençam ao mesmo *cluster*. Este é também o critério empregado na etapa inicial de exclusão dos blocos que impediriam a obtenção de uma solução viável. A Figura 23 apresenta uma ilustração dos conjuntos $adj4_b$ e $adj8_b$ para um dado bloco $b \in I$, ficando claro que $adj4_b \subset adj8_b$.

Figura 23 - Ilustração representando os conjuntos $adj4_b$ e $adj8_b$ para um dado bloco $b \in I$.



3.1.2 Parâmetros

Os parâmetros empregados na modelagem matemática do problema aqui abordado são os seguintes:

min_size_cluster: Número mínimo de blocos atribuídos a cada *cluster*;

max_size_cluster: Número máximo de blocos atribuídos a cada *cluster*;

min_n_cluster: Número mínimo de *clusters* permitidos por bancada;

max_n_cluster: Número máximo de *clusters* permitidos por bancada;

γ : Distância Euclidiana máxima (diâmetro máximo) entre dois blocos b, c atribuídos a um *cluster* j ($b, c = 1, \dots, m, \forall b, c \in I; j = 1, \dots, n, \forall j \in J$).

Uma vez definidos os valores para ***min_size_cluster*** e ***max_size_cluster***, permitindo assim que a quantidade ideal de *clusters* fosse determinada durante a otimização, divide-se o número de blocos em uma bancada (I) por eles, obtendo-se ***max_size_cluster*** e ***min_size_cluster***, respectivamente, onde eventuais resultados decimais são arredondados para cima. Entretanto, a determinação dos valores adequados para estes parâmetros e para o diâmetro máximo γ é arbitrária, configuração que deve ser escolhida em consonância com a instância sendo avaliada, buscando representar restrições reais como largura mínima

operacional, dimensão e capacidade do equipamento de lavra, dentre outros fatores que podem ser incluídos no problema.

3.1.3 Variáveis de decisão

As variáveis de decisão inteiras empregadas no modelo matemático do problema do agrupamento de polígonos de lavra são as seguintes:

$\mathbf{M}_{i,j} \in \{0, 1\}$: Variável binária de dimensão $\mathbb{R}^{i \times j}$ cujo valor equivale a 1 se um bloco i é atribuído ao *cluster* j , ou 0 caso contrário ($i = 1, \dots, m, \forall i \in I; j = 1, \dots, n, \forall j \in J$);

$\mathbf{x}_i \in \{1, J\}$: Variável inteira de dimensão \mathbb{R}^i que atribui a cada bloco i o valor correspondente ao índice de seu *cluster* j ($i = 1, \dots, m, \forall i \in I; j = 1, \dots, n, \forall j \in J$);

$\mathbf{y}_j \in \{0, I\}$: Variável inteira de dimensão \mathbb{R}^j que computa o número de blocos atribuídos a cada *cluster* j ($j = 1, \dots, n, \forall j \in J$);

$\mathbf{p}_{b,c} \in \{0, 1\}$: Variável binária empregada para linearizar a função objetivo ($\mathbf{b}, \mathbf{c} = 1, \dots, m, \forall \mathbf{b}, \mathbf{c} \in I$).

3.1.4 Critério de similaridade

Dentre as opções de critérios de similaridade apresentadas neste estudo, incluindo os métodos de Minkowski (\mathbf{d}_1 , \mathbf{d}_2 e \mathbf{d}_∞), alguns critérios de dissimilaridade $\mathbf{d}_s(\mathbf{b}, \mathbf{c})$ entre dois pontos $\mathbf{b}, \mathbf{c} \in I$ e os critérios de similaridade desenvolvidos Tabesh e Askari-Nasab (2011, 2013), optou-se pelo emprego de uma variante destes últimos. Destarte, ao se considerar dois blocos $\mathbf{b}, \mathbf{c} \in j$, o critério de similaridade $\mathbf{S}_{b,c}$ aqui empregado foi definido conforme a Equação (3.1):

$$\mathbf{S}_{b,c} = (\mathbf{T}_{b,c} \times \mathbf{R}_{b,c}) / (\tilde{\mathbf{D}}_{b,c} \times \tilde{\mathbf{G}}_{b,c}), \mathbf{b}, \mathbf{c} \in j, \forall j \in J \quad (3.1)$$

onde $\mathbf{T}_{b,c}$ é a penalidade aplicada aos blocos que não forem enviados para a mesma destinação, $\mathbf{R}_{b,c}$ é a penalidade atribuída aos blocos com diferentes litologias, $\tilde{\mathbf{D}}_{b,c}$ e $\tilde{\mathbf{G}}_{b,c}$ representam a distância Euclidiana e a distância de teores normalizadas entre estes blocos. Às penalidades $\mathbf{T}_{b,c}$ e $\mathbf{R}_{b,c}$ foi atribuído o valor 1 caso os blocos apresentassem a mesma destinação e litologia, respectivamente, ou o valor 0,2 caso contrário. Conforme explicitado em $\mathbf{b}, \mathbf{c} \in j$, ao se empregar $\mathbf{S}_{b,c}$ na função objetivo do modelo matemático do problema, deve-se assegurar o cálculo somente entre blocos pertencentes a um mesmo *cluster*. Outras funções apresentadas por Tabesh e Askari-Nasab (2011, 2013) atribuem diferentes pesos para

priorizar alguns parâmetros em detrimento de outros, estratégia que não foi investigada neste estudo. Não foram investigadas também diferentes combinações entre parâmetros e suas implicações nas soluções obtidas, tendo sido empregada somente a Equação (3.1) ao longo de todo o estudo.

3.1.5 Função objetivo

A função objetivo deve maximizar a soma das similaridades calculadas entre cada par de blocos $\mathbf{b}, \mathbf{c} \in \mathbf{j}$, cujo resultado é dividido por dois para evitar problemas devido à simetria no problema do agrupamento, quando seriam computados $\mathbf{S}_{\mathbf{b},\mathbf{c}}$ e $\mathbf{S}_{\mathbf{c},\mathbf{b}}$. Além disso, para que $\mathbf{S}_{\mathbf{b},\mathbf{c}}$ seja calculado apenas quando $\mathbf{b}, \mathbf{c} \in \mathbf{j}$, deve-se multiplicar este parâmetro pelas variáveis $\mathbf{M}_{\mathbf{b},\mathbf{j}}$ e $\mathbf{M}_{\mathbf{c},\mathbf{j}}$, embora isso fosse resultar em um modelo não linear. Tem-se, portanto, que o Problema 7 consiste na função objetivo e nas restrições apresentadas neste tópico e no Tópico 3.1.6. Na Equação (3.2), uma variável binária $\mathbf{p}_{\mathbf{b},\mathbf{c}} \in \{\mathbf{0}, \mathbf{1}\}$ é implementada como uma restrição de indicador (*indicator constraint*) para linearizar a função objetivo. Conseqüentemente, esta variável $\mathbf{p}_{\mathbf{b},\mathbf{c}}$ é igual a 1 somente quando $\mathbf{M}_{\mathbf{b},\mathbf{j}}$ e $\mathbf{M}_{\mathbf{c},\mathbf{j}}$ forem iguais a 1, sendo equivalente a 0 caso contrário:

$$\mathbf{M}_{\mathbf{b},\mathbf{j}} = \mathbf{1}, \mathbf{M}_{\mathbf{c},\mathbf{j}} = \mathbf{1} \Rightarrow \mathbf{p}_{\mathbf{b},\mathbf{c}} = \mathbf{1}, \forall \mathbf{b}, \mathbf{c} \in \mathbf{j}, \forall \mathbf{j} \in \mathbf{J} \quad (3.2)$$

Restrições de indicadores, portanto, são aquelas em que a ocorrência de determinado resultado de uma variável está associada à ocorrência de um outro resultado em outra variável, e.g. $\mathbf{y} = \mathbf{0} \Rightarrow \mathbf{x} = \mathbf{0}$. Assim, a satisfação de uma condição associada às variáveis à esquerda da Equação (3.2) ($\mathbf{M}_{\mathbf{b},\mathbf{j}} = \mathbf{1}, \mathbf{M}_{\mathbf{c},\mathbf{j}} = \mathbf{1}$) implicam no resultado da variável localizada à direita da equação ($\mathbf{p}_{\mathbf{b},\mathbf{c}}$). Visto que nem todo *solver* possui uma estrutura específica para a inserção de semelhantes restrições, muitas vezes é necessário empregar estratégias como o uso de um parâmetro *Big M*, que seria um valor grande o suficiente para ser maior que todos os valores presentes no modelo, de forma que sejam criadas relações lineares entre as variáveis capazes de representar a mesma restrição de indicadores (GUROBI, 2022). No Tópico 3.2 será apresentada uma versão linearizada deste modelo, em que todas as restrições de indicador são convertidas em restrições lineares através do uso de parâmetros *Big M*.

A função objetivo empregada neste estudo, finalmente, está descrita na Equação (3.3) tomando-se como base a formulação proposta por Dao, Duong e Vrain (2017), mas ao invés de empregar a função $\mathbf{WCSD}(\mathbf{b}, \mathbf{c})$, onde $\mathbf{WCSD}(\mathbf{b}, \mathbf{c}) = \sum_{\mathbf{j} \in [1, n]} \frac{1}{2} \sum_{\mathbf{b}, \mathbf{c} \in A_{\mathbf{j}}} \mathbf{d}_s(\mathbf{b}, \mathbf{c})$,

empregou-se um critério de similaridade $S_{b,c}$, baseado em Tabesh e Askari-Nasab (2011, 2013), conforme supracitado:

$$\text{Maximizar} \quad \sum_{j \in [1,n]} \sum_{b,c \in j} (S_{b,c} \times p_{b,c})/2 \quad (3.3)$$

3.1.6 Restrições do problema

Além da Equação (3.2), as demais restrições do problema estão representadas nas Equações (3.4) a (3.12):

$$y_j = \sum_{i \in [1,m]} M_{i,j}, \forall j \in J \quad (3.4)$$

$$x_i = \sum_{j \in [1,n]} M_{i,j} \times j, \forall i \in I \quad (3.5)$$

$$\sum_{j \in [1,n]} M_{i,j} = 1, \forall i \in I \quad (3.6)$$

$$j \neq \emptyset \Rightarrow y_j \geq \text{min_size_cluster}, \forall j \in J \quad (3.7)$$

$$y_j \leq \text{max_size_cluster}, \forall j \in J \quad (3.8)$$

$$j = \emptyset \Rightarrow y_j = 0, \forall j \in J \quad (3.9)$$

$$M_{b,j} = 1 \Rightarrow \sum_{c \in [1,n]} M_{c,j} \geq 1, \forall b \in j, c \in \text{adj}4_b \quad (3.10)$$

$$M_{b,j} = 1 \Rightarrow \sum_{c \in [1,n]} M_{c,j} \geq 2, \forall b \in j, c \in \text{adj}8_b \quad (3.11)$$

$$M_{b,j} = 1, M_{c,j} = 1 \Rightarrow \text{Dist}_{b,c} \leq \gamma, \forall b, c \in j, \forall j \in J \quad (3.12)$$

Dentre as restrições apresentadas, a Equação (3.4) calcula o número de blocos atribuídos a cada *cluster* y_j , enquanto a Equação (3.5) garante a atribuição de um *cluster* a cada bloco x_i no domínio I . A Equação (3.6), por sua vez, garante que somente um *cluster* seja atribuído a cada bloco. Para evitar a determinação antecipada do número de *clusters*, a utilização dos parâmetros *min_n_cluster* e *max_n_cluster* como o LB e UB , respectivamente, permite que a otimização escolha o número definitivo, desde que todos os blocos do domínio estejam atribuídos a algum *cluster*. Entretanto, deve-se ter em conta nem sempre este UB será atingido, de modo durante a otimização pode haver *clusters* vazios que devem ser desconsiderados da solução. Assim, as Equações (3.7) e (3.8) impõem que cada

cluster válido ($\mathbf{j} \neq \emptyset$) deve respeitar esses limites, ao passo que a Equação (3.9) impõem que *clusters* inválidos ($\mathbf{j} = \emptyset$) não devem ser atribuídos a nenhum bloco.

Considerando as restrições de vizinhança, as Equações (3.10) e (3.11) garantem que cada bloco $\mathbf{b} \in \mathbf{j}$ possui um ou mais vizinhos no conjunto $\mathbf{adj}\mathbf{4}_b$ e dois ou mais vizinhos no conjunto $\mathbf{adj}\mathbf{8}_b$ pertencentes ao *cluster* \mathbf{j} , respectivamente. Estas restrições garantem uma geometria suave e evitam a criação de “corredores” não operacionais nas fronteiras entre *clusters*. A Equação (3.12), finalmente, garante que todos os blocos \mathbf{b} e \mathbf{c} do domínio \mathbf{I} só pertencem ao mesmo *cluster* \mathbf{j} se sua distância Euclidiana for menor ou igual ao parâmetro γ . A definição deste parâmetro de diâmetro máximo é essencial para se alcançar boas soluções, pois garante a continuidade dos *clusters*. Enquanto valores grandes de γ podem considerar soluções ruins e desconexas como viáveis, valores pequenos podem impedir que soluções viáveis sejam encontradas.

3.2 METODOLOGIA SIMPLIFICADA: PARTICULARIDADES DA MODELAGEM MATEMÁTICA CONSIDERANDO DIFERENTES ABORDAGENS POR OTIMIZAÇÃO

O modelo matemático que consiste no Problema 7, como concebido e apresentado no Tópico 3.1, é um modelo geral capaz de abordar o problema do agrupamento de polígonos de lavra em uma ferramenta de otimização. Apresentando variáveis inteiras, o modelo foi projetado para ser solucionado no *Google OR-Tools* a partir da metodologia *COP*, sendo criado na plataforma um modelo `CpModel()` que, mediante os arquivos de saída indicados, é solucionado por um solver nativo `CpSolver()`. Entretanto, particularidades devem ser consideradas na implementação de um modelo matemático em diferentes implementações e *solvers*, já que a forma de inserir as restrições costuma variar. Os modelos matemáticos baseados em *CP* implementados no *OR-Tools* aceitam a inserção direta de restrições de indicadores, tal que restrições binárias \mathbf{z} são criadas para indicar se determinada condicionante será ou não atuante em determinada circunstância. Como exemplo aplicado nas Equações (3.7) e (3.9), a restrição $\mathbf{y}_j \geq \mathbf{min_size_cluster}$ se aplica a todo *cluster* válido ($\mathbf{j} \neq \emptyset$) por meio da lógica `OnlyEnforceIf(z)`, ao passo que a restrição $\mathbf{y}_j = \mathbf{0}$ se aplica a todo *cluster* não válido ($\mathbf{j} = \emptyset$) quando a lógica `OnlyEnforceIf(z.not())` for aplicada.

Por outro lado, o uso do *Google OR-Tools* para a solução de um modelo pela abordagem *CSP* demanda poucos ajustes, e visto que a abordagem propõe a obtenção de uma ou mais soluções viáveis ao invés de uma solução ótima, não é mais necessário empregar uma

função objetivo, de modo que esta pode ser omitida ou equivaler a 0. Com isso, a estrutura de otimização irá preocupar-se somente em encontrar soluções que respeitem as restrições, razão pela qual é importante determinar um critério de parada para o algoritmo, caso contrário haveria uma busca exaustiva por todas as soluções viáveis do problema. Destarte, ao invés de indicar ao *CpSolver*() que todas as soluções viáveis devem ser extraídas, pode-se determinar um tempo limite ou um número máximo de soluções até que o algoritmo seja interrompido. Mediante o exposto, o modelo baseado em *MILP* aqui proposto para solução por *CSP* pode ser descrito pelas mesmas restrições apresentadas no Problema 7 (Equações (3.2) a (3.12)), contanto que a função objetivo (Equação (3.3)) seja omitida ou formulada como a Equação (3.13):

$$\text{Maximizar} \quad \mathbf{0} \quad (3.13)$$

A resolução do problema do agrupamento de polígonos de lavra de forma exata, por sua vez, considerando a estrutura *OR-Tools* e o emprego do *solver* nativo *SCIP*, demanda uma série de adaptações ao modelo matemático apresentado no Problema 7, visto que todas as restrições de indicadores devem ser linearizadas para que o *SCIP* as compreenda. Para possibilitar a linearização do Problema 7, foi criada uma nova variável auxiliar binária $\mathbf{z}_j = \{0, 1\}, \forall j \in J$, além do uso de uma constante *Big M* e da já citada variável $\mathbf{p}_{b,c} = \{0, 1\}, \forall b, c \in j, \forall j \in J$ na função objetivo, de modo que o modelo linearizado pode ser descrito no Problema 8 a seguir, representado pelas Equações (3.14) a (3.27):

$$\text{Maximizar} \quad \sum_{j \in [1,n]} \sum_{b,c \in j} (\mathbf{s}_{b,c} \times \mathbf{p}_{b,c})/2 \quad (3.14)$$

$$\text{Sujeito a} \quad \mathbf{y}_j = \sum_{i \in [1,m]} \mathbf{M}_{i,j}, \forall j \in J \quad (3.15)$$

$$\mathbf{x}_i = \sum_{j \in [1,n]} \mathbf{M}_{i,j} \times \mathbf{j}, \forall i \in I \quad (3.16)$$

$$\sum_{j \in [1,n]} \mathbf{M}_{i,j} = \mathbf{1}, \forall i \in I \quad (3.17)$$

$$\mathbf{y}_j \geq -\mathbf{Big M} \times (\mathbf{1} - \mathbf{z}_j), \forall j \in J \quad (3.18)$$

$$\mathbf{y}_j \leq \mathbf{Big M} \times \mathbf{z}_j, \forall j \in J \quad (3.19)$$

$$\mathbf{y}_j \geq \mathbf{min_size_cluster} \times \mathbf{z}_j, \forall j \in J \quad (3.20)$$

$$\mathbf{y}_j \leq \mathbf{max_size_cluster} \times \mathbf{z}_j, \forall j \in J \quad (3.20)$$

$$\sum_{j \in [1,n]} \mathbf{M}_{c,j} \geq \mathbf{1} \times \mathbf{M}_{b,j}, \forall b \in j, c \in \mathbf{adj4}_b \quad (3.22)$$

$$\sum_{j \in [1n]} M_{c,j} \geq 2 \times M_{b,j}, \forall b \in j, c \in \text{adj}8_b \quad (3.23)$$

$$\text{Dist}_{b,c} \times M_{c,j} \leq \gamma + \text{Big } M \times (2 - M_{b,j} - M_{c,j}), \forall b, c \in j, \forall j \in J \quad (3.24)$$

$$p_{b,c} \leq M_{b,j}, \forall b, c \in j, \forall j \in J \quad (3.25)$$

$$p_{b,c} \leq M_{c,j}, \forall b, c \in j, \forall j \in J \quad (3.26)$$

$$p_{b,c} \geq M_{b,j} + M_{c,j} - 1, \forall b, c \in j, \forall j \in J \quad (3.27)$$

À primeira vista, pode causar certa estranheza o emprego das variáveis **Big M** e **z_j** nas restrições que antes não as continham, bem como a inserção de novas restrições de modo que estas variáveis interajam da forma desejada, além da adição de restrições relacionando diretamente **p_{b,c}** a **M_{b,j}** e **M_{c,j}**. Entretanto, o resultado alcançado a partir da solução do Problema 8 é o mesmo alcançado pelo Problema 7 caso seja concedido tempo suficiente para que ambas as técnicas obtenham uma solução ótima, de modo que estas variáveis não têm nenhum papel além de garantir que as restrições de indicadores sejam efetivamente implementadas em um *solver* voltado a problemas lineares.

Finalmente, as Equações lineares (3.25) a (3.27) induzem a variável **p** a ser igual a **1** somente quando **M_{b,j}** e **M_{c,j}** também o forem, possibilitando o cálculo correto dos blocos pertencentes a cada partição na função objetivo, lógica que não foi apresentada no Problema 7 por estar implícita a partir do uso de funções específicas ao **CpSolver()**, as funções **AddBoolOr()** e **AddImplication()**. Os APÊNDICES I e II apresentam, respectivamente, a implementação em *python* dos modelos aqui apresentados pelas lógicas **COP** e **MILP**, tal que uma consulta diretamente aos códigos pode dirimir quaisquer dúvidas que o leitor venha a ter. Conforme supracitado, a resolução direta dos modelos matemáticos propostos foi chamada neste estudo de metodologia simplificada de otimização.

De qualquer maneira, independente do *framework* e do *solver* empregados em otimização de problemas em *python*, linguagem escolhida neste estudo graças à sua versatilidade, popularidade e vasta documentação disponível, há geralmente uma sequência a ser seguida que é comum à maioria das bibliotecas. Inicialmente o usuário faz todas as instalações necessárias, em seguida deve ser criado um modelo vazio, que é preenchido com as variáveis de decisão, as restrições e a função objetivo do problema, para somente então o *solver* ser empregado, quando os resultados e as variáveis de saída podem ser extraídos. O usuário deve estar atento ainda que a formatação de modelos matemáticos, interação com os *solvers* e extração de resultados e variáveis dá-se de maneira diferente em cada biblioteca, tal

que um mesmo problema seria reescrito de maneiras distintas caso se empregue os frameworks e bibliotecas *OR-Tools*, *Pyomo*, *Gurobi*, *CVXOPT*, *SCIP*, *PuLP*, *SciPy*, *MealPy*, dentre outros. Da mesma maneira, o usuário deve compreender as particularidades de seu problema para optar pelo *solver* correto, já que cada um baseia-se em algoritmos diferentes que não são capazes de abordar todas as classes de problemas, *e.g.* problemas baseados em *LP*, *IP*, *MIP* e *MILP* podem ser solucionados por meio dos *solvers* *Gurobi*, *CPLEX*, *CBC*, *GLPK* e *SCIP*, ao passo que problemas não lineares podem ser resolvidos pelos *solvers* *IPOPT*, *SCIP* e *Baron*.

3.3 METODOLOGIA MULTIESTÁGIO BASEADA EM PROGRAMAÇÃO POR RESTRIÇÕES

Embora seja possível representar o problema do agrupamento de polígonos de lavra em um modelo baseado em *MILP* e resolvê-lo diretamente através da abordagem *COP*, conforme a metodologia apresentada até aqui, instâncias com mais de 100 blocos podem apresentar dificuldades adicionais. A primeira dificuldade está relacionada ao tamanho do espaço de soluções em problemas NP-difíceis, que tende a crescer de forma exponencial ou fatorial à medida que o tamanho da instância analisada aumenta linearmente, resultando em um processo de otimização muito custoso do ponto de vista computacional. Outro aspecto que merece atenção em instâncias maiores é a dificuldade na determinação do parâmetro γ , pois caso seja dada liberdade em demasia as *clusters*, a tendência é que eles se misturem, gerando soluções inviáveis. Com o objetivo de mitigar estes problemas, foi proposta uma abordagem multiestágio baseada em *COP*, capaz de otimizar o valor do parâmetro γ e acelerar a convergência da otimização na direção de um ótimo local. Por fim, foi proposta uma etapa de refinamento dos *clusters*, capaz de corrigir geometrias inadequadas e assegurar que a dimensão do equipamento de lavra seja respeitada na solução. A heurística de propagação geométrica (*GPH*), que será apresentada neste tópico, foi desenvolvida especificamente para esta finalidade.

3.3.1 Fase de inicialização

Considerando a incapacidade de explorar todo o espaço de soluções, pode ser útil fornecer ao algoritmo empregado na otimização (o *COP* neste caso) uma solução de dica (*hint solution*) viável ou quase viável como ponto de partida. Devido à maldição da

dimensionalidade e ao fato de estarmos lidando com um modelo baseado em *MILP*, o espaço de soluções é muito grande e descontínuo, com soluções viáveis dispersas. Acima de tudo, o espaço de solução não é convexo, de modo que inicializar o algoritmo com uma boa solução de dica pode impedir que nenhuma solução viável seja encontrada.

Portanto, é razoável usar um algoritmo eficiente como o *k-means++* para gerar centroides de *cluster* distantes entre si. Então, pode-se usar o algoritmo *k-means* para rapidamente atribuir cada bloco do domínio estudado a um *cluster*, gerando assim uma boa solução de dica, mesmo que possivelmente ainda não seja viável em relação ao modelo matemático. Consequentemente, a seguinte estratégia foi determinada para gerar uma *hint solution* para inicializar a abordagem *COP*:

- (1) Determinação do número de *clusters* antes de inicializar o *k-means++*;
- (2) Geração de múltiplas soluções através do *k-means*, seguida da comparação dos principais parâmetros de cada uma;
- (3) Seleção e validação de uma das soluções contendo um valor pequeno de γ .

Para a etapa (1), o usuário pode determinar um *LB* e um *UB* para o tamanho dos *clusters* (***min_size_cluster*** e ***max_size_cluster***) considerados viáveis para a instância analisada. Ao se dividir o número de blocos por estes parâmetros e arredondar o resultado, têm-se valores equivalentes aos parâmetros ***min_n_cluster*** e ***max_n_cluster***, e como é necessário determinar a priori o número de *clusters* para utilizar o *k-means++*, a média aritmética desses valores pode ser empregada.

Embora o algoritmo *k-means* seja capaz de gerar soluções muito rapidamente mesmo para grandes instâncias, sua versão padrão não permite o uso de restrições como as implementadas no modelo matemático proposto. Portanto, uma vez escolhido o número de *clusters*, eles podem assumir tamanhos, diâmetro máximo (γ) e relações de vizinhança que não seriam desejáveis do ponto de vista prático do problema de agrupamento de blocos em *mining cuts*. Conforme mencionado acima, a garantia de que haverá conectividade nos *clusters* gerados pelo modelo baseado em *MILP* proposto está na definição do parâmetro γ , uma vez que valores restritivos tendem a inviabilizar o problema e grandes diâmetros geram *clusters* desconectados.

Para a etapa (2), sugere-se que múltiplas soluções sejam geradas e um critério seja estabelecido para determinar qual seria a mais adequada para ser usada como *hint solution*. Ao gerar 20 diferentes candidatos, pode-se optar por aquele que apresente o menor valor de γ , tendo a menor diferença entre os valores de ***min_n_cluster*** e ***max_n_cluster*** como critério secundário. Logicamente, uma solução com um diâmetro menor será um melhor

ponto de partida em termos de conectividade, enquanto um número menor de *clusters* possíveis reduziria o tamanho do espaço de soluções, e conseqüentemente a complexidade do modelo matemático. Diversos conjuntos de 20 candidatos podem ser gerados, para que o usuário tenha uma ideia de qual seria um valor plausível para o parâmetro γ .

Independentemente do algoritmo empregado na otimização, muitas instâncias possuem restrições de espaço inerentes devido a motivações operacionais, portanto já existe uma descontinuidade entre os blocos que tornará a solução do problema mais desafiadora. Além disso, para o passo (3) o usuário deve visualizar a solução escolhida e ponderar se ela é satisfatória como uma solução de dica a ser empregada na abordagem baseada em *CP*. Por fim, todos os parâmetros devem ser atualizados de acordo com a solução escolhida, incluindo *min_size_cluster*, *max_size_cluster*, *min_n_cluster*, *max_n_cluster* e γ , pois assim o usuário terá certeza que a combinação destes valores é capaz de gerar uma solução viável. A Figura 24 apresenta o pseudocódigo da fase de inicialização.

Figura 24 - Pseudocódigo da fase de inicialização da otimização multiestágio.

Algorithm 1: Fase de Inicialização

Data: Todos os blocos i no Domínio I

Result: *Hint Solution*

$\gamma \leftarrow$ O *Diâmetro Máximo* dos clusters em uma dada solução;

n clusters = (max n clusters + min n clusters)/2;

Iterações = [1, 20];

Initialization:

while *Usuário não está Satisfeito* **do**

for k em *Iterações* **do**

Kmeans(*Data*, n clusters, Inicialização = *Kmeans* + +);

Parâmetros[k] = (γ [k], min n clusters[k], max n clusters[k])

end

Sort(*Parâmetros*);

 Seleção da solução k^* ;

 Validação da Solução k^* pelo *Usuário*;

Hint Solution = k^*

end

3.3.2 Fase de ajuste de parâmetros

Uma vez determinada a *hint solution* e atualizados os parâmetros, é possível otimizar o modelo baseado em *MILP* por *COP*, embora os parâmetros provavelmente ainda não sejam definitivos. Mesmo que exista um valor inicial para o parâmetro γ , para evitar a geração de *clusters* desconexos, seria uma boa prática minimizar este parâmetro até que seja obtido o

menor valor possível capaz de satisfazer as restrições do problema para os outros parâmetros considerados. Portanto, esta etapa visa solucionar recorrentemente o modelo matemático por *COP*, onde sempre que for possível encontrar uma solução viável, o valor de γ é atualizado.

A partir do γ determinado na fase de inicialização, e considerando uma única iteração completa do algoritmo *COP* e um determinado tempo de execução como critérios de parada simultâneos, o parâmetro γ será reduzido até que não seja possível obter uma nova solução viável no tempo determinado. Por fim, uma vez identificado este novo valor de γ , a solução da dica deve ser atualizada, aumentando assim a probabilidade de obtenção de melhores soluções por meio da abordagem *COP* para resolver o modelo matemático proposto. A Figura 25 mostra o pseudocódigo da fase de ajuste dos parâmetros.

Figura 25 - Pseudocódigo da fase de ajuste de parâmetros da otimização multiestágio.

Algorithm 2: Fase de Ajuste de Parâmetros

Data: Todos os blocos i no Domínio I

Result: *Hint Solution 2*, γ

$\gamma \leftarrow$ O *Diâmetro Máximo* dos clusters em uma dada solução;

Hint Solution \leftarrow Solução obtida durante a Fase de Inicialização;

Tempo de Parada \leftarrow Parâmetro determinado pelo *Usuário*;

Iterações = [1];

Parâmetros = (γ , min size cluster, max size cluster, min n clusters, max n clusters);

$\gamma = \gamma + 1$;

Initialization:

while *Solução* $\neq \emptyset$ **do**

$\gamma = \gamma - 1$;

COP(*Data*, *Hint Solution*, *Parâmetros*, *Tempo de Parada*, *Iterações*);

Output: *COP Solução Ótima Local*

Hint Solution 2 = *COP Solução Ótima Local*

end

Porém, caso não seja dado tempo suficiente para o algoritmo convergir na direção de um ótimo local, é possível que mesmo após a minimização o menor parâmetro γ obtido ainda permita certa desconexão nos *clusters* sem que haja violação das restrições. Conforme o modelo proposto, o algoritmo sempre buscará soluções que maximizem a função objetivo, e caso as restrições não sejam capazes de direcionar a busca, é possível que sejam encontradas soluções viáveis, mas não operacionais.

Uma forma de mitigar parcialmente problemas com soluções mal formuladas é implementar uma janela correspondente ao tamanho do equipamento de lavra no centroide de cada *cluster*. Portanto, o usuário deve definir a dimensão adequada desta janela em termos do

número de blocos nos eixos x e y de acordo com o tamanho dos blocos da instância avaliada. Em seguida, são identificados os centroides dos *clusters* na solução de dica obtida através da abordagem *COP*, uma janela é colocada em cada centroides e todos os blocos pertencentes a essas janelas são atribuídos aos *clusters* correspondentes. Esta é uma operação opcional, e caso os *clusters* da nova *hint solution* já estejam bem definidos, seria redundante aplicá-la. A Figura 26 mostra o pseudocódigo do procedimento de janela mínima.

Figura 26 - Pseudocódigo do procedimento de janela mínima empregado na otimização multiestágio.

Algorithm 3: Procedimento da Janela Mínima

Data: Todos os blocos i no Domínio I atribuídos a cada cluster j de acordo com $Solution, \forall j \in J$
Result: *Improved Solution*
Input: *Solution*
Initialization:
for j in J do
 | *Centróides* são computados para cada j ;
 | Uma *Janela* é posicionada em cada *Centróide* e todos os blocos dentre desta *Janela* são atribuídos ao cluster j correspondente;
end

3.3.3 Fase de exploração

Esta fase consiste em uma etapa de otimização convencional considerando os parâmetros determinados nas fases anteriores. Critérios de parada relacionados ao número de iterações e tempo de execução devem ser determinados para o algoritmo *COP*, e espera-se seja obtida uma solução ótima local que atenda a todas as restrições do problema. A Figura 27 apresenta o pseudocódigo da fase de exploração.

Figura 27 - Pseudocódigo da fase de exploração da otimização multiestágio.

Algorithm 4: Fase de Exploração

Data: Todos os blocos i no Domínio I
Result: *COP Solução Ótima Local*
 $\gamma \leftarrow$ O *Diâmetro Máximo* dos clusters atualizado durante a Fase de Ajuste de Parâmetros;
Hint Solution 2 \leftarrow Solução obtida durante a Fase de Ajuste de Parâmetros;
Tempo de Parada \leftarrow Parâmetro determinado pelo *Usuário*;
Iterações = [1, 30];
Parâmetros = (γ , min size cluster, max size cluster, min n clusters, max n clusters);
Initialization:
COP(Data, Hint Solution 2, Parâmetros, Tempo de Parada, Iterações);
Output: *COP Solução Ótima Local*

3.3.4 Fase de pós-processamento

Mesmo tomando todos esses cuidados, principalmente quando o diâmetro máximo encontrado é ainda grande, o usuário não consegue evitar *clusters* desconexos na solução encontrada através da abordagem *COP*. Outro aspecto que não pôde ser plenamente considerado nas etapas anteriores é a garantia de que as dimensões do equipamento de escavação estejam sendo respeitadas. Mesmo em *clusters* conectados e com fronteiras bem definidas, é ilusório imaginar que grandes equipamentos serão capazes de escavar cada bloco individualmente nas regiões de fronteira. Portanto, para evitar a necessidade de o usuário empregar alguma etapa manual para realizar esses ajustes, foi proposta uma heurística de propagação geométrica (*GPH*) para realizar um procedimento de pós-processamento, composto por três etapas. Tendo a solução obtida por *COP* como *input*, esta fase cria um novo modelo de blocos que atua como um espelho da solução já obtida, só que os blocos não estão ainda atribuídos a nenhum *cluster*, então a heurística *GPH* fará avaliações na solução base e decidirá quais blocos continuarão com a mesma atribuição ou quais serão reajustados.

3.3.4.1 Etapa de inicialização heurística

Nesta etapa, o usuário inicia implementando nos centroides de cada *cluster* determinado por *COP* o mesmo procedimento de inserção de uma janela representando as dimensões do equipamento, para que os blocos dentro dessas janelas possam ser atribuídos a cada *cluster* correspondente. Como o novo modelo de blocos não possuía nenhuma atribuição, os blocos abarcados por estas janelas representam a primeira decisão efetuada pela *GPH*, sendo ainda adicionados a uma lista de blocos já visitados, assim a heurística evitará retornar a estes blocos. Os blocos pertencentes às bordas dessas janelas nas direções Norte, Leste, Sul e Oeste são identificados e armazenados, e todos os demais blocos que não foram atribuídos a nenhum *cluster* são considerados como não visitados.

3.3.4.2 Etapa de reconstrução da solução

Uma vez implementadas as janelas iniciais e, conseqüentemente, uma vez que os primeiros blocos foram atribuídos a seus respectivos *clusters*, a *GPH* considerará expandir estes limites de modo que seja respeitada a viabilidade operacional da solução. A cada iteração, a heurística considera a propagação dos *clusters* nas direções Norte, Leste, Sul e

Oeste, armazenando uma lista de movimentos permitidos para cada *clusters*. Por exemplo, se um *clusters* cuja fronteira Norte contém cinco blocos for considerado para expansão nesta direção, os cinco blocos vizinhos na direção Norte serão verificados e, se escolhidos, serão atribuídos ao *clusters* em expansão. Assim, enquanto os critérios de parada não forem atendidos, ou seja, visitar todos os blocos da instância ou eliminar todas as possibilidades de propagação do *clusters*, a *GPH* buscará opções de expansão.

Portanto, a cada iteração a heurística procura (1) escolher um *cluster*, (2) escolher uma direção de propagação, (3) atribuir os novos blocos ao *clusters* em expansão, (4) atualizar suas fronteiras, (5) atualizar seus movimentos permitidos e (6) atualizar a lista de blocos visitados. Visando dar oportunidade para todos os *clusters* se expandirem, desde que haja movimentações permitidas, a heurística prioriza o *cluster* que até o momento recebeu menos blocos, e caso haja empate, o menor *cluster* na *hint solution* obtida pela abordagem *COP* será selecionado.

Uma vez selecionado um *cluster* para expansão, são analisadas suas possíveis direções de propagação e é escolhida aquela com maior potencial. O critério considera a quantidade de blocos que seriam adicionados ao *cluster*, desde que mais de 50% desses blocos ainda não tenham sido visitados. Além disso, uma classificação calcula quantos desses blocos candidatos pertencem ao *cluster* em expansão na *hint solution* a, de modo que será dada prioridade às expansões que sejam mais aderentes aos *clusters* determinados pela abordagem *COP*. Caso os blocos candidatos sejam incorporados ao *cluster* em expansão, suas fronteiras e movimentos permitidos são atualizados, bem como a lista de blocos já visitados. De outra forma, caso não seja possível expandir o *cluster* na direção selecionada, esta direção será excluída da lista de movimentos permitidos para este *cluster*, e um novo será selecionado na iteração seguinte. A Figura 28 apresenta um pseudocódigo abordando as etapas de inicialização e reconstrução da solução por *GPH*.

Figura 28 - Pseudocódigo das etapas de inicialização e reconstrução da solução por GPH durante a otimização multiestágio.

Algorithm 5: Heurística de Propagação Geométrica - Inicialização e Reconstrução da Solução

Data: Todos os blocos i no Domínio I atribuídos a cada $COP Cluster_j$ de acordo com $Solution$, $\forall j \in J$

Result: *GPH Solução Parcial*

$Solution = COP Solução Ótima Local$;

$GPH Cluster_j \leftarrow$ Novas atribuições a clusters feitas durante as iterações da heurística, $\forall j \in J$;

$Fronteiras_j \leftarrow$ Listas contendo blocos que limitam cada $GPH Cluster_j$ nas direções N, E, S e W, $\forall j \in J$;

$Blocos Visitados \leftarrow$ Lista que irá incorporar os blocos visitados durante as iterações da heurística;

Inicialização:

O *Procedimento da Janela Mínima* é implementado;

for j in J do

$Fronteiras_j$ são atualizadas;

$Blocos Visitados$ é atualizada;

end

Reconstrução da Solução:

$Movimentos Permitidos_j \leftarrow$ Listas indicando se um $GPH Cluster_j$ pode ser propagado nas direções N, E, S e W, $\forall j \in J$;

while $Blocos Visitados < Domínio I$ e $Movimentos Permitidos_j \neq \emptyset$, $\forall j \in J$ do

$Seleção de Cluster \leftarrow$ Os $GPH Clusters_j$ com $Movimentos Permitidos_j > 0$ são identificados;

$Candidato a Expansão_j =$ Menor $GPH Cluster_j$;

 if Tie then

$Candidato a Expansão_j =$ Menor $COP Cluster_j$;

 end

$Possibilidades de Expansão_j \leftarrow$ Blocos vizinhos ao $Candidato a Expansão_j$ nas direções N, E, S e W são identificados;

$Seleção de Ranking \leftarrow$ A direção com maior potencial de propagação é identificada considerando o número de blocos em cada fronteira atribuído ao $Candidato a Expansão_j$ em $COP Cluster_j$, contanto que menos de 50% deles estejam em $Blocos Visitados$;

 if $Candidato a Expansão_j$ for expandido then

$Candidato a Expansão_j$ é atualizado;

$Fronteiras_j$ são atualizadas;

$Blocos Visitados$ é atualizada;

$Movimentos Permitidos_j$ são atualizados;

 else

$Movimentos Permitidos_j = \emptyset$;

 end

end

Output: *GPH Solução Parcial*

Este mecanismo visa propagar as janelas originais para que sejam acrescentadas linhas ou colunas inteiras de blocos, evitando assim reentrâncias nas fronteiras entre *clusters* e mantendo a aderência ao tamanho do equipamento de lavra. Porém, devem existir regiões que esta etapa não conseguiu visitar, como na expansão de um *cluster* em dada direção que se depara com um conjunto de blocos onde muitos deles já foram visitados. Nessas circunstâncias, uma expansão seria indesejável, pois acabaria destruindo parte dos *clusters* já formados, gerando eventualmente regiões não operacionais. Por esta razão, em casos semelhantes a *GPH* não expande os *clusters*, sendo então proposta uma etapa de refinamento de formas para lidar especificamente com esses blocos não visitados durante a etapa de reconstrução da solução.

3.3.4.3 Etapa de refinamento de formas

Finalmente, a etapa de refinamento das formas dá-se nos blocos ainda não visitados e implementa um procedimento baseado em HC para agregá-los em *clusters* provisórios, empregando o método de ligação única (*single-link*) e a métrica Euclidiana. Considerou-se que os *clusters* provisórios com menos blocos que a largura da janela seriam problemas marginais, mais associados às fronteiras do modelo de blocos, onde foi implementado um procedimento simples de refinamento de formas para estes grupos. Para cada bloco ***b*** em cada *cluster* provisório, os conjuntos ***adj4_b*** e ***adj8_b*** são verificados e os blocos já visitados nestes conjuntos são computados. Portanto, todos os blocos em um *cluster* provisório pequeno serão atribuídos ao *cluster* mais frequente em todos os conjuntos ***adj4*** relacionados a esses blocos. Em caso de empate, esses blocos serão atribuídos ao *cluster* mais frequente em todos os conjuntos ***adj8*** a eles relacionados. Caso haja outro empate, será considerado o *cluster* mais frequente na *hint solution* obtida pela abordagem *COP* dentre os associados aos blocos deste *cluster* provisório. Caso ainda não tenha sido obtida uma solução, um desses candidatos será escolhido aleatoriamente. A Figura 29 apresenta o pseudocódigo do procedimento simples de refinamento de formas empregado pela *GPH*.

Figura 29 - Pseudocódigo do procedimento simples de refinamento de formas empregado pela GPH.**Algorithm 6: Procedimento Simples de Refinamento de Formas**

Data: Todos os blocos i em um *ClusterProvisório*
for i *in ClusterProvisório* **do**
 | Computar os *Clusters* associados aos blocos visitados em $Adj4_i$ e $Adj8_i$;
end
Escolher o *Cluster* mais frequente em todos os $Adj4_i$;
Em caso de empate, escolher o mais frequente destes *Clusters* em todos os $Adj8_i$;
Em caso de novo empate, escolher o mais frequente destes *Clusters* associado aos blocos i na solução obtida pela abordagem COP;
Havendo outro empate, escolher aleatoriamente dentre estes *Clusters*;
Blocos Visitados é atualizada;

Por outro lado, se o número de blocos em um *cluster* provisório for maior ou igual à largura da janela, um procedimento completo de refinamento das formas deverá ser realizado. Inicialmente, para cada bloco no *cluster* provisório é calculado o número de vizinhos também pertencente ao *cluster* provisório, de modo que aqueles com menos vizinhos são considerados como extremidades do *cluster*. Então, duas dessas extremidades são obtidas aleatoriamente e o algoritmo *A* Search* (Hart, Nilsson e Raphael 1968) é empregado para gerar um caminho dentro desse *cluster* provisório, considerando os nós inicial e terminal selecionados. Se a lista de extremidades contém apenas um bloco, este é considerado o nó inicial do caminho e essa restrição é relaxada, para que se possa determinado um nó terminal.

Partindo do nó inicial e considerando a ordem de blocos determinada pelo caminho escolhido, a heurística tenta inserir no caminho subgrupos com o número de blocos equivalente à largura da janela, desde que estejam todos na mesma direção. Se tal subgrupo for implementado, um procedimento simples de refinamento de formas é realizado neste subgrupo, conforme apresentado anteriormente, as listas de blocos visitados e não visitados são atualizadas, a direção de propagação do caminho é mantida e um novo subgrupo contendo um número de blocos equivalente à largura da janela é proposto. Se não for possível formar tal subgrupo, seja por mudança de direção no caminho ou porque o nó terminal foi alcançado, é proposto um subgrupo contendo apenas um bloco. Se este bloco estiver na mesma direção que o caminho já vinha percorrendo, para manter a continuidade da solução, ele é atribuído ao mesmo *cluster* atribuído ao subgrupo anterior. Caso contrário, caso haja uma mudança iminente de direção no caminho, um procedimento simples de refinamento de formas é realizado neste bloco e a direção do caminho é atualizada, bem como as listas de blocos visitados e não visitados. Este procedimento geral será realizado até que o nó terminal seja alcançado, e caso ainda existam blocos no *cluster* provisório não visitados pelo caminho,

novos caminhos serão criados até que todos os blocos sejam visitados. A Figura 30 apresenta o pseudocódigo do procedimento completo de refinamento de forma empregado pela *GPH*.

Figura 30 - Pseudocódigo do procedimento completo de refinamento de formas empregado pela *GPH*.

Algorithm 7: Procedimento Completo de Refinamento de Formas

```

Data: Todos os blocos  $i$  em um grande Cluster Provisório
while Cluster Provisório  $\neq \emptyset$  do
  for  $i$  in Cluster Provisório do
    | Computar os vizinhos dentro do Cluster Provisório;
  end
  Extremidades  $\leftarrow$  Conjunto que contém blocos com o mínimo número de vizinhos;
  Nó Inicial, Nó Terminal  $\leftarrow$  Blocos amostrados no conjunto Extremidades;
  Algoritmo A* Search (Nó Inicial, Nó Terminal) = Caminho;
  while Caminho  $\neq \emptyset$  do
    | if Número de blocos  $i$  igual à largura da Janela em uma mesma direção
    |   considerando o ordenamento em Caminho;
    |   then
    |     | Subgrupo = Próximos blocos  $i$  equivalentes à largura da Janela;
    |     | Refinamento de Formas Simples (Subgrupo);
    |     | Cluster Provisório é atualizado;
    |     | Blocos Visitados é atualizada;
    |     | Direção é mantida;
    |     | Caminho é atualizado;
    |     | Cluster obtido pelo Refinamento de Formas Simples é armazenado;
    |   else
    |     | Subgrupo = Próximo bloco  $i$ ;
    |     | if Direção do Subgrupo = Direção;
    |     |   then
    |     |     | Subgrupo adicionado ao Cluster armazenado;
    |     |     | Cluster Provisório é atualizado;
    |     |     | Blocos Visitados é atualizada;
    |     |     | Direção é mantida;
    |     |     | Caminho é atualizado;
    |     |   else
    |     |     | Refinamento de Formas Simples (Subgrupo);
    |     |     | Cluster Provisório é atualizado;
    |     |     | Blocos Visitados é atualizada;
    |     |     | Direção é atualizada;
    |     |     | Caminho é atualizado;
    |     |     | Cluster obtido pelo Refinamento de Formas Simples é armazenado;
    |     |   end
    |   end
  end
end
end

```

Uma vez que todos os blocos em todos os *clusters* provisórios sejam atribuídos a *clusters*, seja por meio do procedimento simples ou completo de refinamento de formas, não haverá mais blocos não visitados, portanto, a *GPH* terá atingido sua solução. Os procedimentos de pós-processamento propostos visam alterar o mínimo possível a solução obtida através da metodologia *COP*, aumentando, contudo, a sua aderência às restrições operacionais. A Figura 31 apresenta o pseudocódigo completo da heurística *GPH*.

Figura 31 - Pseudocódigo da heurística *GPH*.

Algorithm 8: Heurística de Propagação Geométrica (*GPH*)

Data: Todos os blocos i no Domínio I atribuídos a cada cluster j de acordo com $Solution, \forall j \in J$
Result: *GPH Solution*
Solution = COP LocalOptimumSolution;
Clusters \leftarrow Novos clusters atribuídos durante as iterações da heurística;
Inicialização
Reconstrução da Solução
Clusterização Hierárquica (Método = single-linkage, Métrica = Euclidiana) = *Cluster Provisório*;
for j *in Cluster Provisório* **do**
 if j possui um número de blocos menor que ou igual à largura da Janela;
 then
 | Procedimento Simples de Refinamento de Formas;
 else
 | Procedimento Completo de Refinamento de Formas;
 end
end

Porém, a aplicação desta etapa ainda não garante que os blocos localizados nas bordas do domínio sejam adequados para o tamanho mínimo do equipamento. Na verdade, o problema mais adequado para identificar se um bloco apresenta uma configuração de vizinhança tal que possa suportar adequadamente a operação de uma escavadeira é o problema do limite de escavação, ou *dig limit problem* (SARI; KUMRAL, 2018), e não o problema de agrupamento de polígonos de lavra, que está além do escopo deste estudo.

3.4 METODOLOGIA MULTISTÁGIO REDUZIDA

Visto que a metodologia multiestágio completa apresenta duas etapas baseadas em otimização, mesmo que sejam empregadas soluções de dicas (*hint solutions*) para que os algoritmos iniciem em determinados locais do espaço de soluções há ainda um custo

computacional significativo. Portanto, visando reduzir a complexidade da abordagem do problema do agrupamento de blocos em *mining cuts* em instâncias maiores, foi proposta uma metodologia multiestágio reduzida que prescindiria das fases de otimização apresentadas na metodologia multiestágio completa. Portanto, a metodologia multiestágio reduzida seria composta somente pelas fases de inicialização e de pós-processamento apresentadas nos Tópicos 3.3.1 e 3.3.4. Conseqüentemente, primeiro uma solução para o problema é gerada rapidamente por *k-means*, sendo em seguida aplicada a heurística *GPH* nesta solução inicial para corrigir as geometrias entre os *clusters*. Com isso, se espera verificar se é viável dispensar as etapas baseadas em otimização e empregar somente heurísticas com o intuito de acelerar a obtenção de uma solução satisfatória.

3.5 VALIDAÇÃO E COMPARAÇÃO ENTRE METODOLOGIAS DE AGRUPAMENTO DE POLÍGONOS DE LAVRA

Conforme apresentado no Tópico 2.2.4, as instâncias estudadas foram avaliadas pelas métricas de qualidade que não dependem de um prévio conhecimento de quais partições são verdadeiras, como o índice Calinski-Harabasz, o índice Davies-Bouldin e o coeficiente da silhueta. Ainda conforme citado, o tempo total de execução (s) para obtenção de cada solução também foi considerado, bem como o valor da função objetivo atingida por cada método.

Informações relacionadas ao planejamento de lavra também foram verificadas, como números de *clusters* e número de blocos enviados à rota de beneficiamento em cada solução, bem como o teor médio (g/t). Visto que há uma determinação prévia da destinação de cada bloco, aplicada pela combinação de fatores como litologia e teor de Au, é provável que haja diferenças entre estas destinações, atribuídas nos suportes de blocos, e as destinações atribuídas no suporte de *clusters*. A diluição (t) foi calculada com base na massa dos blocos que a priori foram considerados como estéril, mas cujos *clusters* foram enviados à rota de beneficiamento, ao passo que a perda de minério (t) foi calculada a partir da massa dos blocos que a priori foram considerados como minério, mas cujos *clusters* foram enviados ao depósito de estéril. Foram computadas também as massas totais de minério (t) e de estéril (t) que foram processadas ou descartadas após a atribuição aos *clusters*. Por fim, o retorno econômico foi obtido pela análise do valor presente de cada solução, considerando que toda a bancada seria extraída e que todos os blocos atribuídos a determinado *cluster* seriam enviados à mesma destinação deste *cluster*, independente de sua destinação a priori.

Visto que diante de diversas métricas pode ser difícil comparar duas soluções distintas, é importante ressaltar que, do ponto de vista da otimização, o valor da função objetivo e o tempo de processamento seriam os fatores mais importantes a serem comparados. Já do ponto de vista da mineração, deve-se atribuir maior importância a fatores práticos como diluição, perda de minério e valor presente. Entretanto, uma estrutura de otimização busca encontrar uma solução que é um ótimo local ou global baseado em uma determinada função objetivo, que por sua vez baseia-se em uma determinada função de similaridade. Destarte, não necessariamente buscou-se neste estudo minimizar a diluição ou maximizar o valor presente, sendo estas métricas mais uma consequência da função objetivo empregada do que um fator causal.

Todos os experimentos apresentados neste estudo foram executados em um servidor pertencente ao Laboratório de Pesquisa Mineral e Planejamento Mineiro (LPM), do Departamento de Engenharia de Minas (DEMIN) da Universidade Federal do Rio Grande do Sul (UFRGS). Este servidor tem como sistema operacional o *Windows Server 2022*, possuindo 2 Processadores Xeon 4214R 2.40 GHz de 12 núcleos, 256 GB de memória RAM, 4 SSDs de 1 TB e 1 HD externo de 4 TB.

4 ANÁLISE DE RESULTADOS E DISCUSSÃO

Neste estudo, ambos bancos de dados utilizados nos experimentos numéricos foram extraídos do *MineLib* (ESPINOZA *et al.*, 2013). O primeiro, denominado “*Newman 1 dataset*”, foi submetido a um estudo de caso solucionado por uma metodologia simplificada baseada em *COP*, ao passo que o segundo, chamado de “*W23 dataset*”, foi submetido a um estudo de caso solucionado pela otimização multiestágio baseada em *COP*. Estes bancos de dados são reais e possuem informações sobre a posição de cada bloco, a identificação dos teores do elemento de interesse, as diferentes litologias presentes nos depósitos, informações sobre custos e retorno econômico ao se processar um bloco, destinação de cada bloco, dentre outras informações. A Tabela 2 apresenta uma descrição dos modelos de blocos e das bancadas selecionadas para este estudo.

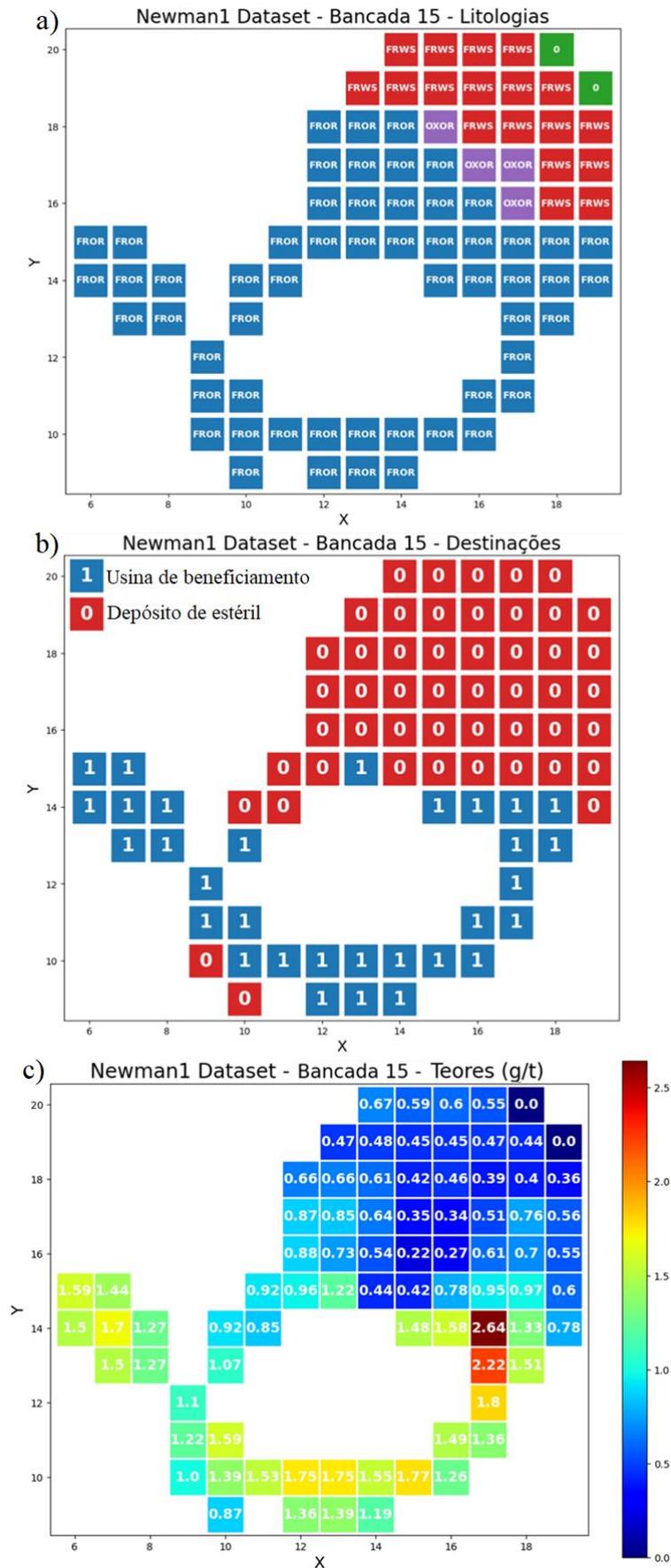
Tabela 2 - Descrição dos modelos de blocos e das respectivas bancadas utilizadas neste estudo.

Instância	Bancadas	Total de blocos	Litologias	Destinos	Cota da bancada (Z)	Blocos na bancada
<i>Newman 1 dataset</i>	24	1.060	4	2	15	83
<i>W23 dataset</i>	63	74.260	12	4	4	197

4.1 METODOLOGIA SIMPLIFICADA: BANCADA 15 DO *NEWMAN 1 DATASET*

O *Newman1 Dataset* (ESPINOZA *et al.*, 2013) possui 1.060 blocos distribuídos em 24 bancadas, contendo 4 litologias e 2 destinos *a priori*, e apesar da bancada 15 possuir 83 blocos, 3 deles foram excluídos da análise por não possuírem blocos vizinhos suficientes para atender às restrições de vizinhança do modelo matemático, restando, portanto, 80 blocos na análise. Os parâmetros *min_size_cluster* e *max_size_cluster* foram determinados arbitrariamente neste estudo como equivalentes a 5 e 16 blocos, respectivamente. O *Newman1 Dataset* não possui um sistema de coordenadas nem um *Datum* específico, de modo que a distância Euclidiana entre dois blocos nas direções Norte-Sul e Leste-Oeste equivale a uma unidade de distância. Partindo deste pressuposto, o parâmetro γ de distância máxima entre dois blocos pertencentes ao mesmo *cluster* foi estipulado como 5. A Figura 32 contém as distribuições das litologias (a), das destinações *a priori* (b) e dos teores (c) dos blocos da bancada 15 do *Newman1 Dataset*.

Figura 32 – Distribuição espacial a) das litologias, b) das destinações a priori e c) dos teores dos blocos da Bancada 15 do Newman1 Dataset.

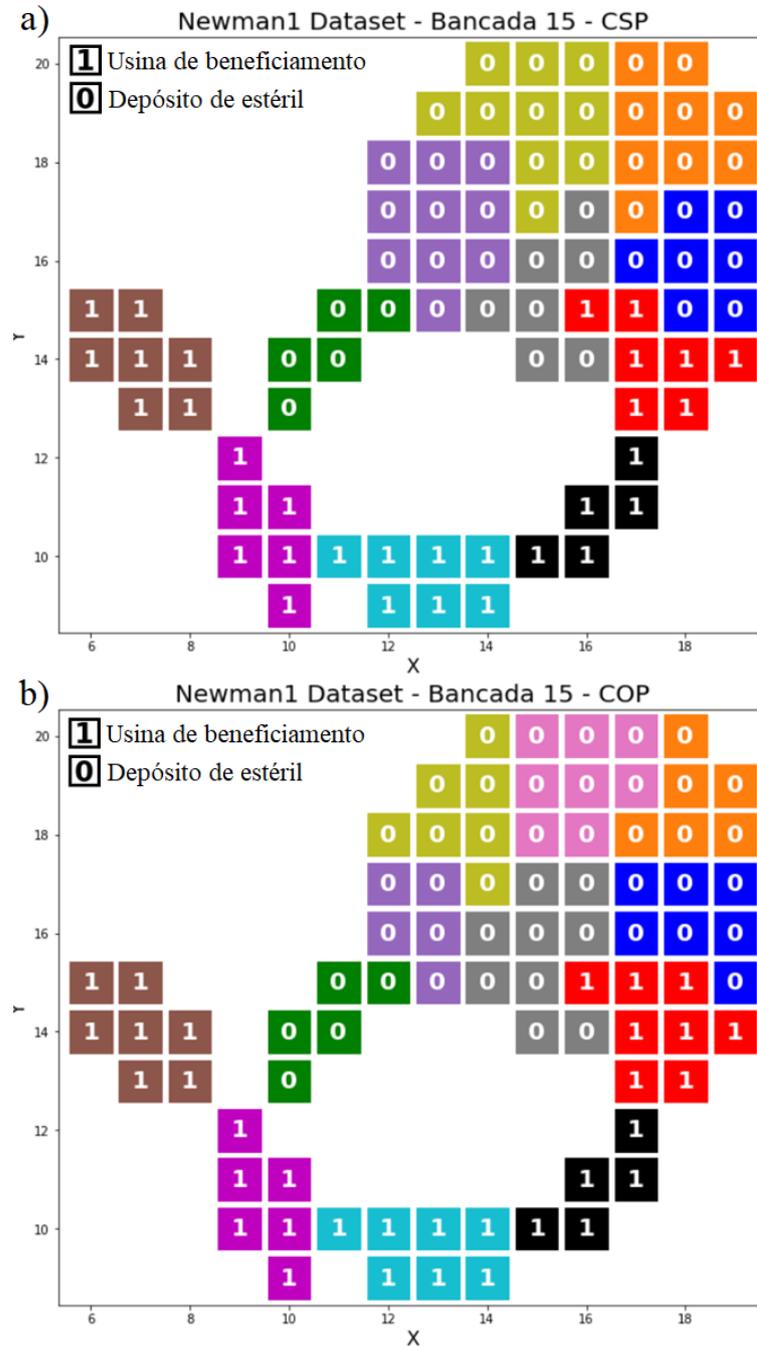


Inicialmente, tentou-se obter uma solução exata do problema *MILP* por meio de um algoritmo *Branch-and-Bound* para que pudesse ser comparada com as soluções obtidas por *CP*, mas após rodar por 1.180.110,13 segundos (13,66 dias) considerando um *gap* de 10% como critério de parada, nenhuma solução ótima foi encontrada. Para melhor ilustrar a dificuldade de resolução de problemas NP-difíceis por algoritmos exatos à medida que o tamanho do problema aumenta, foi possível obter uma solução exata para a Bancada 3 do *Newman1 Dataset*, contendo 15 blocos, em 11 segundos. Por outro lado, uma solução com um *gap* de 1% foi obtida para a Bancada 9, com 49 blocos, em 2.472 segundos, demorando 221 vezes mais para resolver um problema aproximadamente 3 vezes maior.

Portanto, diante da impossibilidade de obtenção de uma solução exata em um tempo aceitável para servir de referência, optou-se por explorar o espaço de soluções viáveis da Bancada 15 por meio da abordagem *CSP*. Considerando um critério de parada de 7 dias (604.839 segundos), o algoritmo *CSP* identificou 74.581 soluções viáveis, não havendo, entretanto, garantia de que todo o espaço de soluções tenha sido explorado ou de que a solução ótima tenha sido identificada. Foi possível identificar ainda uma grande descontinuidade no espaço de soluções, pois 27.308 soluções viáveis (36,62%) apresentaram valores de função objetivo entre $2,46 \times 10^8$ e $4,05 \times 10^8$, enquanto 47.273 soluções (63,38%) atingiram valores entre $4,73 \times 10^9$ e $4,89 \times 10^9$, sendo este último o maior valor de função objetivo encontrado.

Em seguida, o modelo proposto foi resolvido pela abordagem *COP* considerando um critério de parada de 60 segundos e, incluindo o tempo necessário para gerar o modelo matemático no *solver*, em 117 segundos o algoritmo alcançou uma solução com função objetivo de $4,89 \times 10^9$. Esta solução apresentou uma melhora pouco significativa, da ordem de $1,89 \times 10^6$ em relação à melhor solução obtida pela abordagem *CSP*, razão pela qual seus valores podem ser considerados equivalente, mas foi obtida 5.163 vezes mais rápido que a outra abordagem, que investiga exaustivamente o espaço de soluções. A Figura 33 apresenta os *clusters* gerados pelas melhores soluções da Bancada 15 do *Newman1 Dataset* obtidas pelas abordagens *CSP* (a) e *COP* (b), respectivamente. As diferentes cores representam os *mining cuts* gerados pelos algoritmos onde valores 0 e 1 representam as destinações dos blocos ao depósito de estéril ou à rota de beneficiamento, respectivamente.

Figura 33 - Clusters gerados na Bancada 15 do Newman1 Dataset a partir das abordagens a) CSP e b) COP, respectivamente, representados em diferentes cores.



As diferentes abordagens baseadas em *CP* resultaram em soluções com diferentes números de *clusters*, e os resultados atestam a superioridade da abordagem *COP* em termos de velocidade de processamento e capacidade de convergência desta técnica na direção de um ótimo local. No entanto, o *cluster* vermelho localizado no lado direito em ambas as soluções chama a atenção para a inclusão de um bloco a mais a ser enviado à planta de beneficiamento de acordo com a abordagem *COP*. Para avaliar o impacto que esta decisão traria no retorno econômico de um projeto, foi calculado o valor presente de cada uma das soluções,

considerando que toda a Bancada 15 seria escavada e que a destinação de todos os blocos i atribuídos a um dado *cluster* j seria a mesma destinação de j . A Equação (4.1) apresenta o cálculo do valor econômico empregado neste estudo, tal que:

$$VP_i = (CE_i + RB_i) \times \alpha_i + (CE_i) \times (1 - \alpha_i), \forall i \in I \quad (4.1)$$

onde VP_i representa o valor presente de um bloco i , CE_i equivale ao custo de escavação deste bloco e RB_i representa o retorno econômico ao beneficiar um bloco, incluindo os custos de processamento. O parâmetro α_i , por sua vez, equivale a 1 caso o bloco seja enviado para o beneficiamento ou a 0 caso contrário. Destarte, $VP = \sum_{i=1}^I VP_i$, tal que o valor presente da solução obtida por *CSP* foi equivalente a 217.379,77 unidades, enquanto a solução obtida por *COP* resultou em um valor presente de 202.523,99 unidades. A Tabela 3 apresenta um resumo dos melhores resultados obtidos pelos dois modelos de otimização, onde é possível verificar um aumento de 55,02 t de minério enviadas ao beneficiamento na solução obtida por *COP*. Em função desse acréscimo, 5.608,98 t de estéril foram também enviadas ao beneficiamento nesta solução, aumentando a diluição e diminuindo ligeiramente o teor médio quando comparado com a solução por *CSP*.

Tabela 3 - Parâmetros das melhores soluções alcançadas pelas técnicas *CSP* e *COP*, respectivamente, ao resolver o problema de agrupamento de corte de mineração para a Bancada 15 do *Newman1 Dataset*.

Parâmetros ¹	Melhor <i>CSP</i>	<i>COP</i>
Função objetivo	$4,89 \times 10^9$	$4,89 \times 10^9$
Tempo de processamento (s)	604.839	117
Número de <i>clusters</i>	11	12
Blocos enviados ao beneficiamento	32	33
Índice Calinski-Harabasz	11,39	10,92
Índice Davies-Bouldin	1,96	1,93
Índice da Silhueta	0,05	0,00
Teor médio (g/t)	1,38	1,37
Minério processado (t)	2.109,18	2.164,20
Minério descartado (t)	1.533,60	1.478,58
Estéril processado (t)	150.313,82	155.922,80
Estéril descartado (t)	255.982,72	250.373,74
Diluição (t)	27.504,90	33.168,90
Perda de minério (t)	13.145,73	13.145,73
Valor presente (unidades)	217.379,77	202.523,99

¹ Os parâmetros hachurados apresentam os melhores resultados; aos que não possuem hachura, esta lógica não se aplica.

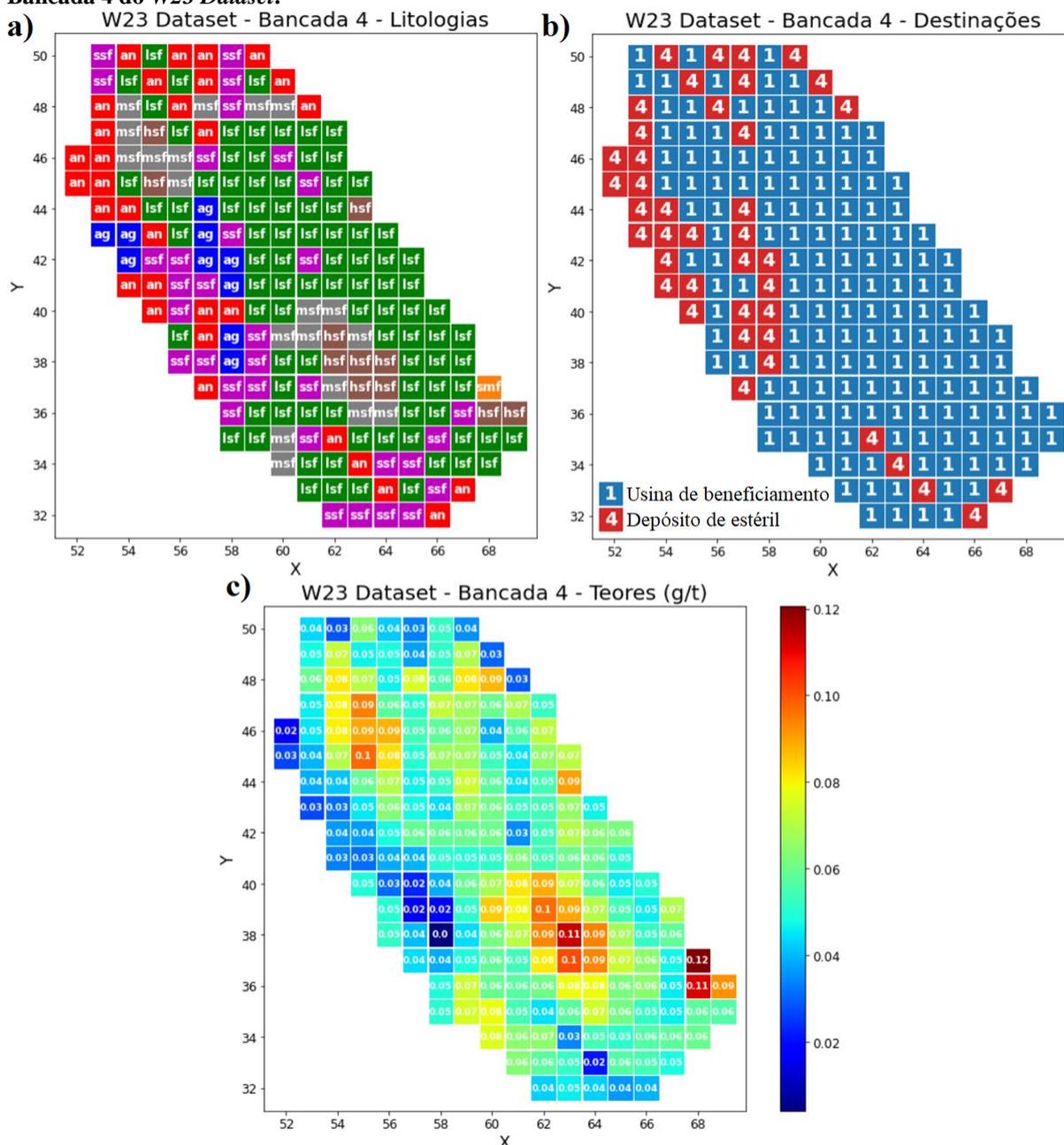
Provavelmente esta diferença entre as técnicas deve ser atribuída à função de similaridade escolhida, pois a abordagem *COP* foi mais eficiente em otimizar o modelo para o exemplo proposto que a abordagem *CSP*, visto que sua função objetivo foi ligeiramente

maior, embora essa diferença seja pequena, menor que a ordem de grandeza apresentada na Tabela 3. Entretanto, como havia vários fatores compondo a função de similaridade, o *trade-off* entre eles deve ter afetado o retorno econômico da solução, representado pela penalidade $T_{b,c}$ aplicada a blocos que não são enviados à mesma destinação. Não houve diferença quanto à perda de minério. Quanto às métricas relacionadas à qualidade das partições, a solução obtida por *COP* apresentou melhores índices da Silhueta e Davies-Bouldin, enquanto a solução obtida por *CSP* apresentou um melhor índice Calinski-Harabasz. Entretanto, estas diferenças são tão sutis que seria mais indicado empregar os parâmetros relacionados à mineração no processo de comparação e validação das soluções.

4.2 METODOLOGIA MULTISTÁGIO COMPLETA: BANCADA 4 DO W23 DATASET

O *W23 dataset* (ESPINOZA *et al.*, 2013) é um depósito de minério de ouro contendo 74.260 blocos com tamanho de 25 x 25 x 20 *ft* (7.62 x 7.62 x 6.09 m), 12 litologias e 4 destinos. As destinações disponíveis são moagem de minério sulfetado, moagem de minério oxidado, lixiviação de minério oxidado e depósito de estéril, e as diferentes rochas compreendem 5 litologias de minério sulfetado (HSF, LSF, MSF, SMF e SSF), 3 litologias de minério oxidado (OL, OM e OW) e 4 tipos de estéril (AG, AN, BF e COV). Somente blocos sulfetados devem ir para a moagem de minério sulfetado. Os blocos de óxido, por sua vez, podem seguir tanto para a moagem de minério oxidado ou para a lixiviação de minério oxidado, enquanto as litologias restantes não devem ser processadas. A metodologia multiestágio proposta neste estudo é aplicada na bancada 4 do *W23 dataset*, composta por 197 blocos, 7 litologias (HSF, LSF, MSF, SMF, SSF, AG e AN) e 2 destinações (moagem de minério sulfetado e depósito de estéril). A Figura 34 contém as distribuições espaciais (a) das litologias, (b) das destinações a priori e (c) dos teores de Au nos blocos da bancada 4 do *W23 Dataset*. As destinações consideradas na Figura 34b são (1) moagem de minério sulfetado e (4) depósito de estéril.

Figura 34 - Distribuição espacial a) das litologias, b) destinações a priori e c) teores de Au nos blocos da Bancada 4 do W23 Dataset.

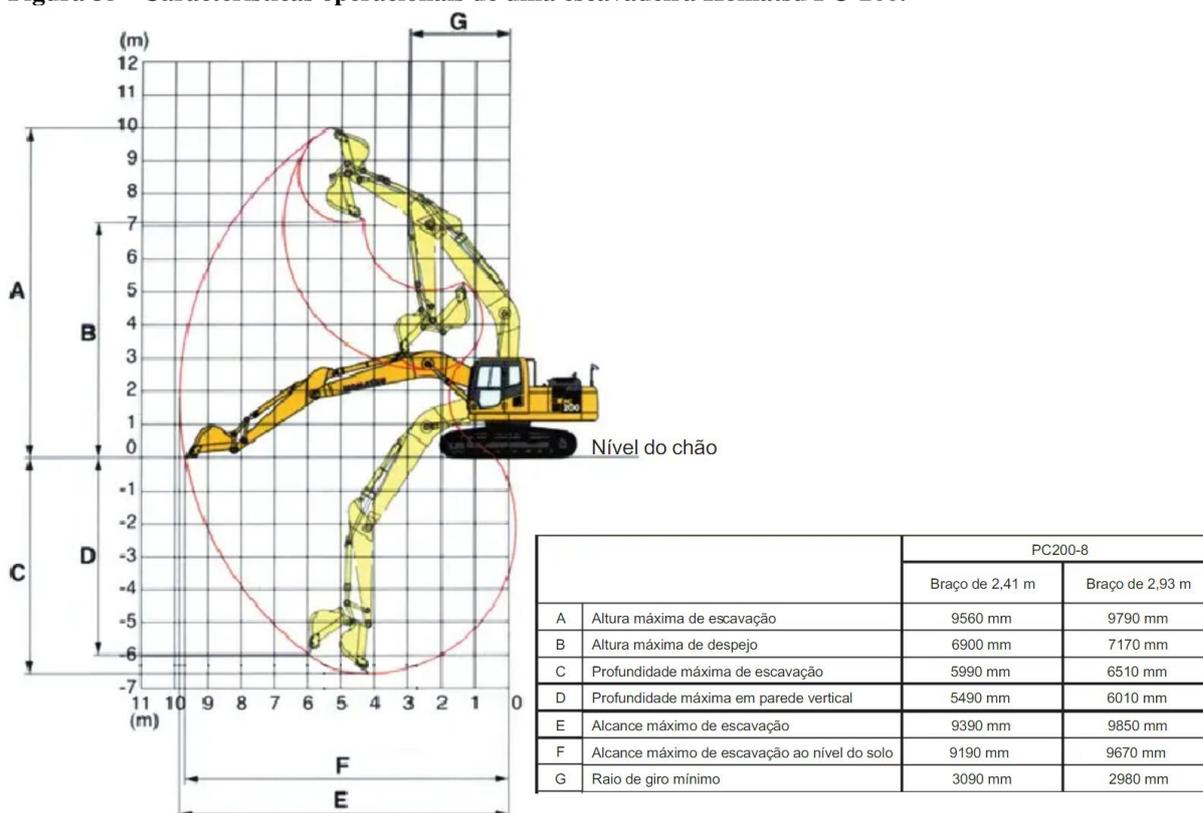


Enquanto foi possível aplicar a metodologia simplificada na bancada 15 do *Newman1 dataset*, por ser uma bancada de tamanho reduzido e com diversas restrições de espaço, não foi possível obter uma solução satisfatória para a bancada 4 do *W23 dataset* por meio desta metodologia. A determinação do parâmetro γ , conforme supracitado, é preponderante na qualidade da solução obtida por *COP*, onde valores elevados permitem a mistura entre *clusters*, ao passo que valores reduzidos inviabilizam a solução. Pôde-se verificar, entretanto, que mesmo que este parâmetro fosse reduzido ao máximo, não havia garantia de que os *clusters* desta instância não iriam se misturar, de modo que aplicar a metodologia multiestágio

e a heurística *GPH* propostas neste estudo foi crucial para a obtenção de soluções satisfatórias.

Para validar a robustez da metodologia proposta, foram desenvolvidos três estudos de caso na referida instância, considerando diferentes números de *clusters*. Como os blocos do *W23 dataset* têm dimensões de 25 x 25 x 20 ft (7,62 x 7,62 x 6,09 m), considerou-se que o equipamento de lavra adequado seria uma escavadeira de 20 t, como a Komatsu PC-200, que apresenta um diâmetro operacional de aproximadamente 20 m. Esta informação foi obtida a partir do raio operacional do equipamento, indicado na Figura 35 (parâmetro E), sendo aproximadamente 10 m, tal que o diâmetro ideal seria equivalente a, no mínimo, três blocos do *W23 dataset*. Portanto, a janela operacional considerada em todos estes cenários equivale a uma matriz de 3 x 3 blocos, de modo que a largura desta janela seja equivalente à largura de 3 blocos (correspondendo a 22,86 m), adequada ao porte do equipamento selecionado.

Figura 35 - Características operacionais de uma escavadeira Komatsu PC-200.



Fonte: adaptado de KOMATSU (2009).

É importante ressaltar que o *W23 dataset* não apresenta coordenadas verdadeiras, tal que a distância entre um bloco e seu vizinho nas direções N-S ou E-W equivale a 1 unidade de distância, de modo que o parâmetro de diâmetro máximo γ deve ser determinado nessa

escala. Além disso, semelhante ao estudo de caso apresentado anteriormente, os valores de função objetivo apresentados para cada solução são apenas comparáveis entre si, pois representam uma composição das variáveis utilizadas na função de similaridade, e não uma variável real. Finalmente, os números relativos aos experimentos numéricos mostrarão a divisão entre *clusters* através de cores diferentes, enquanto cada bloco estará associado a uma destinação, onde 1 representa a moagem de minério sulfetado e 2 representa o depósito de estéril.

4.2.1 Experimento numérico 1

O experimento numérico 1 começa pela obtenção de uma solução de dica com 6 *clusters* por *k-means*, cujo parâmetro γ empregado foi equivalente a 29. Ao utilizar os valores $\mathit{min_n_cluster} = 6$ e $\mathit{max_n_cluster} = 7$ na otimização pela abordagem *COP* durante a fase de ajuste dos parâmetros, o valor γ foi reduzido para 26, após um tempo de execução de 7.157 s, resultando na solução de dica 2. Esta solução de dica passou por um processo de melhoria, onde foi inserida uma janela no centroide de cada *cluster*, melhorando assim suas geometrias, resultando em uma solução parcial com valor de função objetivo equivalente a $7,89 \times 10^{10}$. Partindo deste parâmetro γ otimizado e da solução de dica 2, a abordagem *COP* foi novamente empregada executando 21 iterações durante um tempo de 2.011 s, resultando em uma solução otimizada com valor objetivo de $1,50 \times 10^{11}$, mas longe de ser operacional. Finalmente, a heurística *GPH* foi aplicada na solução obtida durante a fase de exploração durante 15 s, resultando em uma solução final viável do ponto de vista operacional contendo 6 *clusters*. O valor da função objetivo da solução final foi $7,56 \times 10^{10}$, e o tempo total de execução considerando as quatro etapas da otimização multiestágio foi equivalente a 9.183 s. Considerando a implementação da metodologia proposta no experimento numérico 1, a Figura 36 apresenta a evolução da solução durante as quatro fases da otimização multiestágio baseada em *COP*, enquanto a Tabela 4 apresenta a evolução dos parâmetros durante estas fases.

Figura 36 - Experimento numérico 1: a) Solução dica 1, obtida por *k-means*; b) Solução dica 2, obtida por COP; c) Solução exploratória, obtida por COP; d) e Solução final com 6 *clusters*, obtida por GPH.

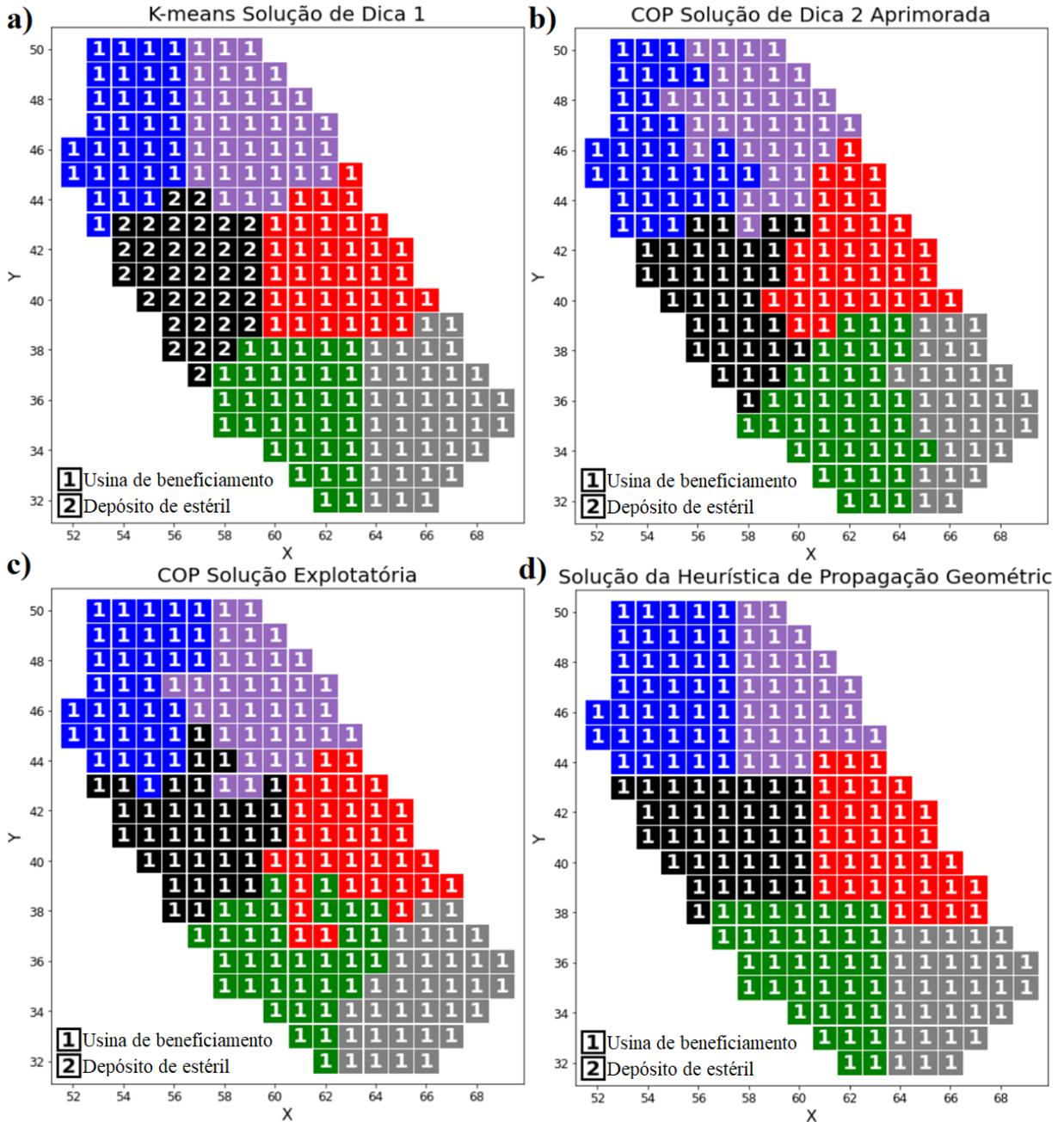


Tabela 4 - Evolução dos parâmetros durante a otimização no experimento numérico 1.

Parâmetros*	Solução de dica 1 (<i>k-means</i>)	Solução de dica 2 aprimorada (COP)	Solução exploratória (COP)	Solução final (GPH)
Número de <i>clusters</i>	6	6	6	6
Diâmetro máximo (γ)	29	26	26	36
<i>min_size_cluster</i>	30	30	30	28
<i>max_size_cluster</i>	36	36	36	37
Função objetivo	$8,30 \times 10^{10}$	$1,26 \times 10^{11}$	$1,50 \times 10^{11}$	$7,56 \times 10^{10}$

* Os parâmetros hachurados apresentam os melhores resultados; aos que não possuem hachura, esta lógica não se aplica.

As diferenças apresentadas pelo parâmetro γ durante as fases da otimização multiestágio corroboram com a dificuldade de resolver o modelo matemático de uma só vez e obter uma solução com melhor aderência operacional. Restringir γ tende a formar *clusters* com bordas diagonais, enquanto flexibilizar esse parâmetro acaba permitindo a mistura entre *clusters*, justificando assim o uso da metodologia proposta e destacando o valor do algoritmo *GPH*. Este mesmo comportamento poderá ser notado nos demais experimentos numéricos.

4.2.2 Experimento numérico 2

O experimento numérico 2, por sua vez, tem início pela obtenção de uma solução de dica com 7 *clusters* através da heurística *k-means*, com parâmetro γ equivalente a 22. Ao utilizar os valores *min_n_cluster* = 7 e *max_n_cluster* = 8 na otimização pela abordagem *COP* durante a fase de ajuste dos parâmetros, o valor γ foi reduzido para 20 após um tempo de execução de 4.257 s, resultando na solução de dica 2. Novamente, esta solução de dica passou por um procedimento para melhorar a forma dos *clusters*, e após a inserção das janelas em seus centroides foi obtido um valor de função objetivo de $1,15 \times 10^{11}$. Partindo deste parâmetro γ otimizado e da solução de dica 2, a abordagem *COP* foi outra vez empregada durante 12 iterações ao longo de 2.056 s, resultando em uma solução otimizada com valor de função objetivo de $1,25 \times 10^{11}$, embora ainda não fosse considerada operacional. Finalmente, após aplicar a heurística *GPH* na solução obtida durante a fase de exploração, uma solução final com 7 *clusters*, viável do ponto de vista operacional, foi alcançada em 14 s. O valor da função objetivo da solução final foi $8,05 \times 10^{10}$, e o tempo total de execução considerando as quatro etapas da otimização multiestágio foi equivalente a 6.327 s. Considerando a implementação da metodologia proposta no experimento numérico 2, a Figura 37 apresenta a evolução da solução durante as quatro fases da otimização, enquanto a Tabela 5 apresenta a evolução dos parâmetros durante estas fases.

Figura 37 - Experimento numérico 2: a) Solução dica 1, obtida por *k-means*; b) Solução dica 2, obtida por *COP*; c) Solução exploratória, obtida por *COP*; e d) Solução final com 7 *clusters*, obtida por *GPH*.

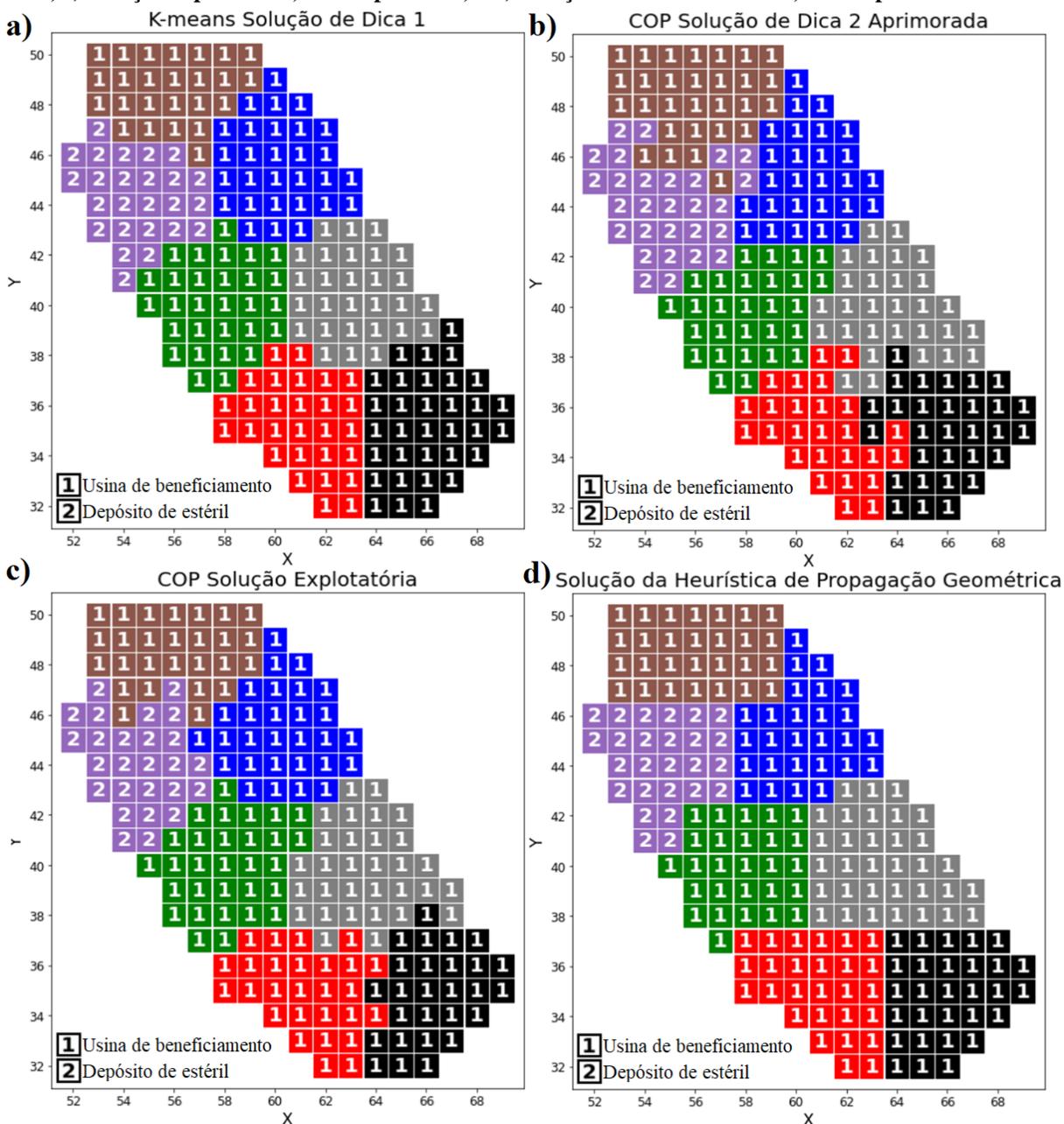


Tabela 5 - Evolução dos parâmetros durante a otimização no experimento numérico 2.

Parâmetros*	Solução de dica 1 (<i>k-means</i>)	Solução de dica 2 aprimorada (<i>COP</i>)	Solução exploratória (<i>COP</i>)	Solução final (<i>GPH</i>)
Número de <i>clusters</i>	7	7	7	7
Diâmetro máximo (γ)	22	20	20	26
<i>min_size_cluster</i>	25	25	25	26
<i>max_size_cluster</i>	30	30	30	33
Função objetivo	$8,14 \times 10^{10}$	$1,15 \times 10^{11}$	$1,24 \times 10^{11}$	$8,05 \times 10^{10}$

* Os parâmetros hachurados apresentam os melhores resultados; aos que não possuem hachura, esta lógica não se aplica.

4.2.3 Experimento numérico 3

Finalmente, o experimento numérico 3 começa pela obtenção de uma solução de dica com 9 *clusters* por meio de um *k-means*, cujo parâmetro γ foi equivalente a 17. Foram empregados os valores ***min_n_cluster* = 8** e ***max_n_cluster* = 11** na otimização pela abordagem *COP* durante a fase de ajuste dos parâmetros, tal que o valor de γ foi reduzido para 14 após um tempo de execução de 2.088 s, resultando na solução de dica 2. A solução de dica foi submetida a um procedimento de melhoria da forma dos *clusters* e, após a inserção das janelas em seus centroides, foi obtido um valor de função objetivo de $7,32 \times 10^{10}$. Ao empregar o parâmetro γ otimizado e a solução de dica 2 na abordagem *COP* durante a fase exploratória, foi obtida uma solução otimizada com valor de função objetivo de $1,31 \times 10^{11}$ ao longo de 14 iterações durante 2.193 s, solução parcial que mais uma vez ainda não era operacional. Finalmente, após submeter esta solução parcial à heurística *GPH*, uma solução final com 10 *clusters* foi obtida em 13 s, sendo mais uma vez viável do ponto de vista operacional. O valor da função objetivo da solução final foi equivalente a $1,28 \times 10^{11}$, e o tempo total de execução considerando as quatro etapas da otimização multiestágio foi de 4.294 s. Considerando a implementação da metodologia proposta no experimento numérico 3, a Figura 38 apresenta a evolução da solução durante as quatro fases da otimização multiestágio baseada em *COP*, e enquanto a Tabela 6 apresenta a evolução dos parâmetros durante estas fases.

Figura 38 - Experimento numérico 3: a) Solução dica 1, obtida por *k-means*; b) Solução dica 2, obtida por *COP*; c) Solução exploratória, obtida por *COP*; e d) Solução final com 10 *clusters*, obtida por *GPH*.

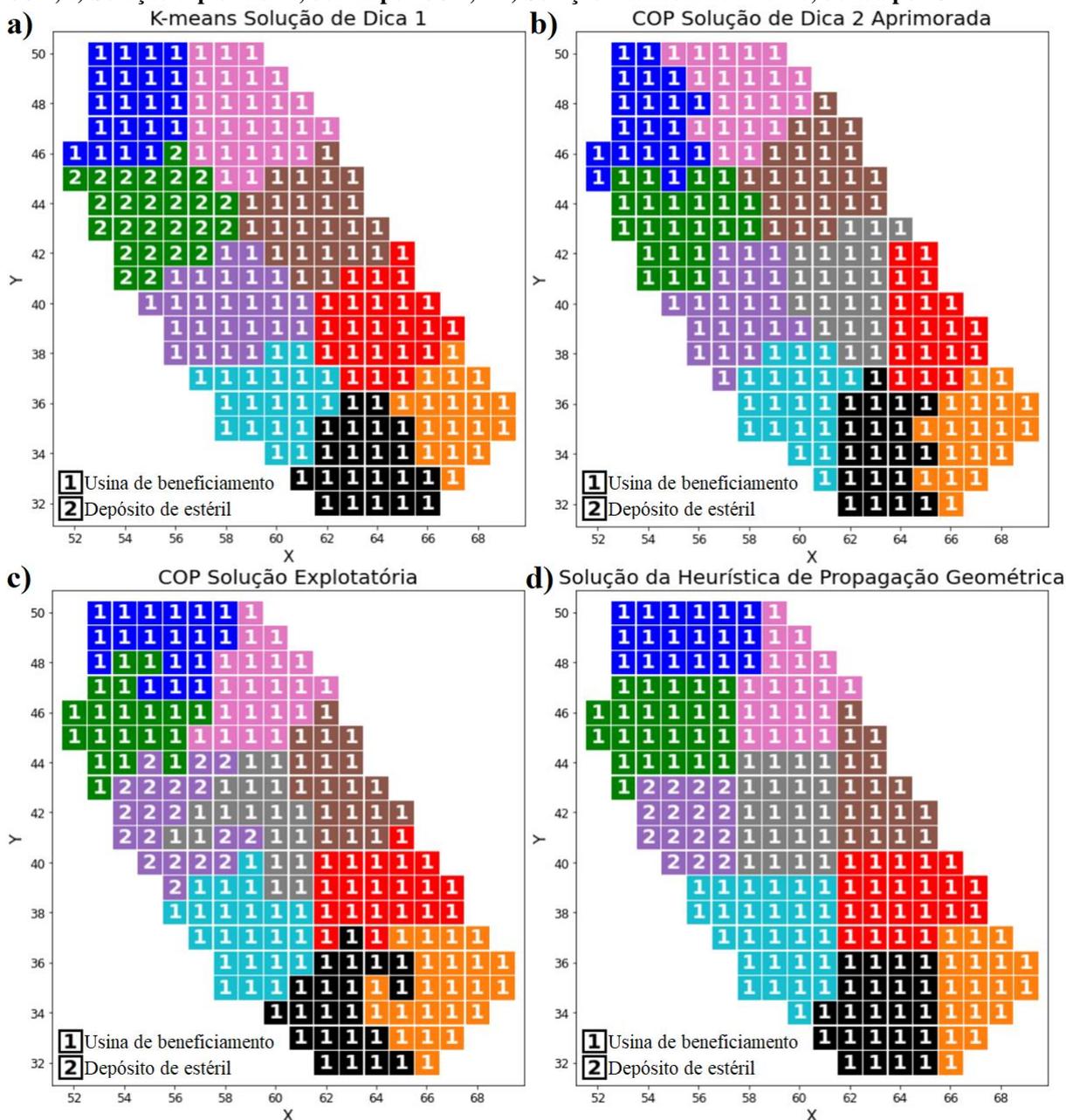


Tabela 6 - Evolução dos parâmetros durante a otimização no experimento numérico 3.

Parâmetros*	Solução de dica 1 (<i>k-means</i>)	Solução de dica 2 aprimorada (<i>COP</i>)	Solução exploratória (<i>COP</i>)	Solução final (<i>GPH</i>)
Número de <i>clusters</i>	9	10	10	10
Diâmetro máximo (γ)	17	14	14	20
<i>min_size_cluster</i>	18	18	18	15
<i>max_size_cluster</i>	25	22	22	26
Função objetivo	$1,27 \times 10^{11}$	$7,32 \times 10^{10}$	$1,31 \times 10^{11}$	$1,28 \times 10^{11}$

* Os parâmetros hachurados apresentam os melhores resultados; aos que não possuem hachura, esta lógica não se aplica.

4.2.4 Comparação entre os experimentos numéricos

Para possibilitar a validação dos resultados por meio de um critério econômico, foi calculado o valor presente de cada experimento numérico, considerando que toda a Bancada 4 seria escavada e que a destinação de todos os blocos i atribuídos a um dado *cluster* j seria a mesma destinação de j . A Equação (4.2) apresenta o cálculo do valor presente empregado nesta comparação, tal que:

$$VP_i = (Teor_i \times Recuperação_i \times \$900 - CE_i - CB_i - CT_i) \times Massa_i, \forall i \in I \quad (4.2)$$

conforme proposto no domínio *MineLib* (ESPINOZA *et al.*, 2013) para esta instância, onde $VP = \sum_{i=1}^I VP_i$, ao passo que CE_i representa o custo de escavação, CB_i o custo de beneficiamento e CT_i o custo de transporte associado ao bloco i . A Tabela 7, por sua vez, apresenta os resultados obtidos durante os experimentos numéricos, considerando tanto a qualidade dos *clusters* como os principais indicadores de desempenho no planejamento de lavra.

Tabela 7 - Resultados obtidos durante os experimentos numéricos relativos à qualidade dos *clusters* e indicadores-chave de desempenho no planejamento de lavra.

Parâmetros*	Experimento numérico 1	Experimento numérico 2	Experimento numérico 3
Função objetivo	$7,56 \times 10^{10}$	$8,05 \times 10^{10}$	$1,28 \times 10^{11}$
Tempo de processamento (s)	9.183	6.327	4.294
Índice da Silhueta	-0,02	-0,04	-0,11
Índice Calinski-Harabasz	11,31	9,96	7,65
Índice Davies-Bouldin	3,31	3,69	3,76
Teor médio (g/t)	0,06	0,06	0,06
Minério processado (t)	273,80	241,35	259,07
Minério descartado (t)	0,00	32,45	14,73
Estéril processado (t)	473.711,14	411.187,20	437.635,64
Estéril descartado (t)	0,00	62.523,94	36.075,50
Diluição (t)	98.646,61	64.962,40	76.992,48
Perda de minério (t)	0,00	28.872,18	14.436,09
Valor presente (\$)	$1,38 \times 10^6$	$2,05 \times 10^6$	$2,13 \times 10^6$

* Os parâmetros hachurados apresentam os melhores resultados; aos que não possuem hachura, esta lógica não se aplica.

Conforme evidenciado na Tabela 7, o valor da função objetivo aumenta à medida que o número de *clusters* na otimização aumenta, de modo que a função de similaridade empregada a blocos atribuídos a um mesmo *cluster* pôde extrair melhores resultados quando houve menos unidades nesses *clusters*. Por outro lado, o tempo total de processamento

diminuiu significativamente à medida que o número de *clusters* aumentou, provavelmente porque havia menos liberdade (valores menores de γ) e um menor espaço de decisão nessas circunstâncias. Do ponto de vista dos critérios de qualidade dos *clusters*, entretanto, à medida que mais grupos foram adicionados à solução, piores foram as métricas, embora as diferenças tenham sido sutis, onde somente o índice Calinski-Harabasz apresentou uma diferença mais significativa.

No experimento numérico 1, todos os blocos foram enviados para o beneficiamento, mesmo aqueles localizados em regiões menos promissoras, o que resultou em uma maior diluição, mas não houve perda de minério a ser considerada. Entretanto, como há um custo associado ao beneficiamento, esta opção não foi vantajosa, resultando em um menor valor presente, justamente porque é preferível enviar blocos indesejados diretamente à pilha de estéril do que submetê-los ao processamento. Ao se comparar os experimentos numéricos 2 e 3, por sua vez, embora ambos destinem um *cluster* à pilha de estéril, há mais blocos descartados no experimento 2, gerando menos diluição e mais perda de minério que o experimento 3. Entretanto, uma melhor determinação dos *clusters* neste experimento 3 resultou no maior de todos os valores presentes, assegurando mais uma vez que melhores *clusters* resultam não somente em melhores valores de função objetivo, mas também em maior retorno econômico.

Ressalta-se que o número de *clusters* escolhido é um parâmetro determinado pelo usuário alinhado às características de sua operação, tanto relacionado aos tamanhos do modelo de bloco e do equipamento de mineração, quanto do número desejado de frentes de lavra. Portanto, a comparação entre estes três experimentos numéricos tem um propósito mais voltado à validação da metodologia sob diferentes características operacionais do que à determinação de quais seriam os melhores parâmetros. Portanto, é possível afirmar que a metodologia proposta foi capaz de resolver o modelo matemático para todas as configurações de parâmetros utilizadas. Já a *GPH* se mostrou muito eficiente em transformar soluções que apresentavam diversas falhas e misturas entre *clusters* em soluções aderentes do ponto de vista operacional, consumindo aproximadamente 10 segundos quando solicitada, mesmo para configurações de *cluster* tão diferentes.

4.3 UMA METODOLOGIA MULTISTÁGIO REDUZIDA SERIA SUFICIENTE?

Ao fazer uma validação visual das soluções parciais obtidas em cada etapa durante a aplicação da metodologia multistágio, paira uma sensação de forte semelhança entre o

resultado da fase de inicialização, onde uma *hint solution* é obtida por *k-means*, e a solução final obtida após a fase de pós-processamento. Conforme indicado ao longo dos experimentos apresentados no Tópico 4.2, efetuados na bancada 4 do *W23 dataset*, as fases de ajuste de parâmetros e de exploração são significativamente mais custosas em termos de tempo de processamento que as fases de inicialização e pós-processamento. Portanto, é natural questionar-se quanto à necessidade de empregar todas estas etapas ao invés de simplesmente gerar uma solução inicial por *k-means* e refiná-la por meio da heurística *GPH*, já que esta seria uma metodologia muito mais ágil. Este tópico, portanto, visa comparar as soluções obtidas nos experimentos numéricos do Tópico 4.2 com uma versão reduzida da metodologia multiestágio, onde somente as fases de inicialização e pós-processamento são empregadas.

Considerando os parâmetros empregados no experimento numérico 1, a Figura 39 apresenta a distribuição espacial dos *clusters* gerados (a) por uma metodologia multiestágio reduzida, incluindo o uso do *k-means* e da *GPH*, e (b) dos *clusters* gerados pela metodologia multiestágio completa. A Tabela 8, por sua vez, compara os resultados de ambas.

Figura 39 - Considerando as configurações do experimento numérico 1, comparação entre os agrupamentos gerados pela a) aplicação do *k-means* combinado com a heurística *GPH* e pela b) otimização multiestágio baseada em *COP*.

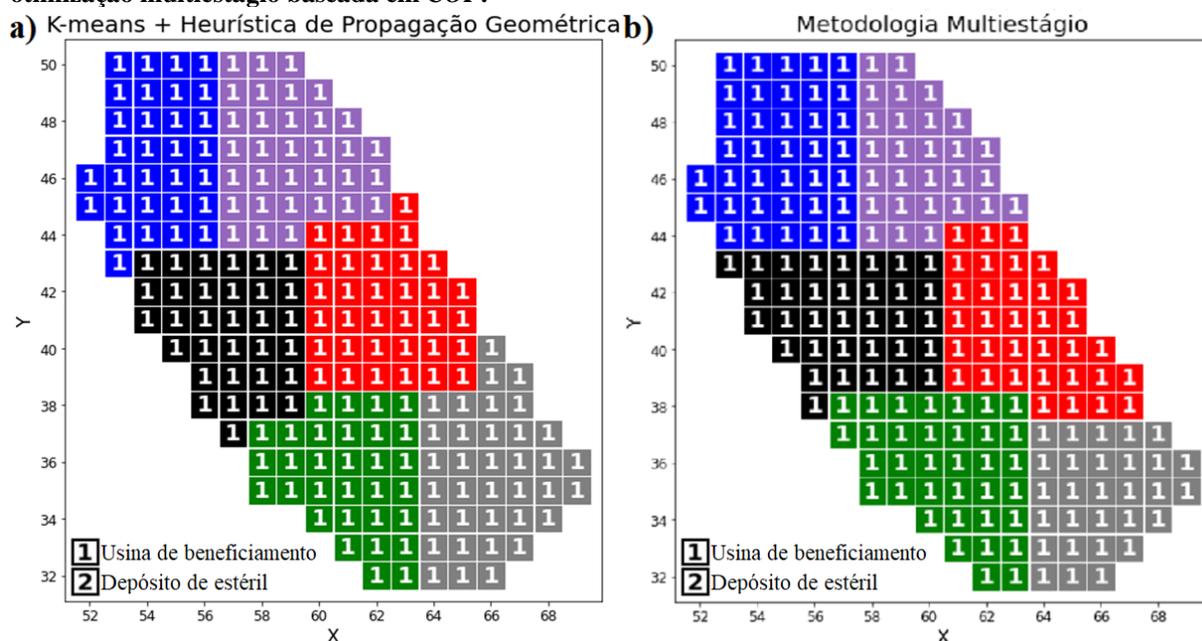


Tabela 8 – Comparação entre os resultados obtidos durante o experimento numérico 1 e uma versão reduzida da metodologia multiestágio.

Parâmetros*	<i>K-Means</i> + <i>GPH 1</i>	Experimento numérico 1
Número de <i>clusters</i>	6	6
Diâmetro máximo (γ)	34	36
<i>min_size_cluster</i>	31	28
<i>max_size_cluster</i>	36	37
Função objetivo	$8,27 \times 10^{10}$	$7,56 \times 10^{10}$
Tempo de processamento (s)	13	9.183
Índice da Silhueta	-0,02	-0,02
Índice Calinski-Harabasz	12,16	11,31
Índice Davies-Bouldin	3,44	3,31
Teor médio (g/t)	0,06	0,06
Minério processado (t)	273,80	273,80
Minério descartado (t)	0,00	0,00
Estéril processado (t)	473.711,14	473.711,14
Estéril descartado (t)	0,00	0,00
Diluição (t)	98.646,61	98.646,61
Perda de minério (t)	0,00	0,00
Valor presente (\$)	$1,38 \times 10^6$	$1,38 \times 10^6$

* Os parâmetros hachurados apresentam os melhores resultados; aos que não possuem hachura, esta lógica não se aplica.

Neste experimento, a diferença entre as configurações dos *clusters* parecem ser pequenas, e em ambas soluções todos os blocos são destinados ao beneficiamento, sendo pequenas as diferenças entre os parâmetros γ , *min_size_cluster* e *max_size_cluster*. Considerando os *cluster* com menos e mais blocos em ambas as soluções, a diferença entre estes na solução obtida pela metodologia reduzida é de 5 blocos, ao passo que a metodologia multiestágio completa apresenta uma diferença de 9 blocos. Visto que todos os blocos são destinados ao beneficiamento em ambos, todos os parâmetros relacionados à mineração foram idênticos, inclusive o valor presente das soluções. Tampouco a comparação entre as métricas de qualidade dos *clusters* apresenta diferenças relevantes. Entretanto, ao se desconsiderar o tempo de execução da fase de inicialização da metodologia multiestágio, visto que o *k-means* é uma heurística muito ágil, o tempo de execução da metodologia reduzida foi de somente 13 segundos, ao passo que a metodologia multiestágio completa demandou 9.183 segundos para encontrar uma solução. Portanto, a diferença entre o tempo de execução é brutal, e não somente a metodologia multiestágio reduzida encontrou uma solução viável em 0,14% do tempo empregado pela metodologia multiestágio completa, como ainda obteve uma solução com valor de função objetivo 9,59% maior. No experimento numérico 1, portanto, embora a metodologia multiestágio tenha obtido sucesso ao encontrar uma solução que é um ótimo

local em um tempo aceitável, uma versão reduzida desta abordagem traria um resultado equivalente do ponto de vista do retorno econômico, mas melhor quanto ao tempo de execução e valor da função objetivo.

Considerando os parâmetros empregados no experimento numérico 2, a Figura 40 apresenta a distribuição espacial dos *clusters* gerados (a) pela metodologia multiestágio reduzida e (b) daqueles gerados pela metodologia multiestágio completa. A Tabela 9, por sua vez, compara os resultados de ambas.

Figura 40 - Considerando as configurações do experimento numérico 2, comparação entre os agrupamentos gerados a) pela aplicação do *k-means* combinado com a heurística *GPH* e b) pela otimização multiestágio baseada em *COP*. Considerando ainda c) o *cluster* destinado ao depósito de estéril pela metodologia multiestágio completa (em preto), é possível identificar quais destes blocos foram mal classificados.

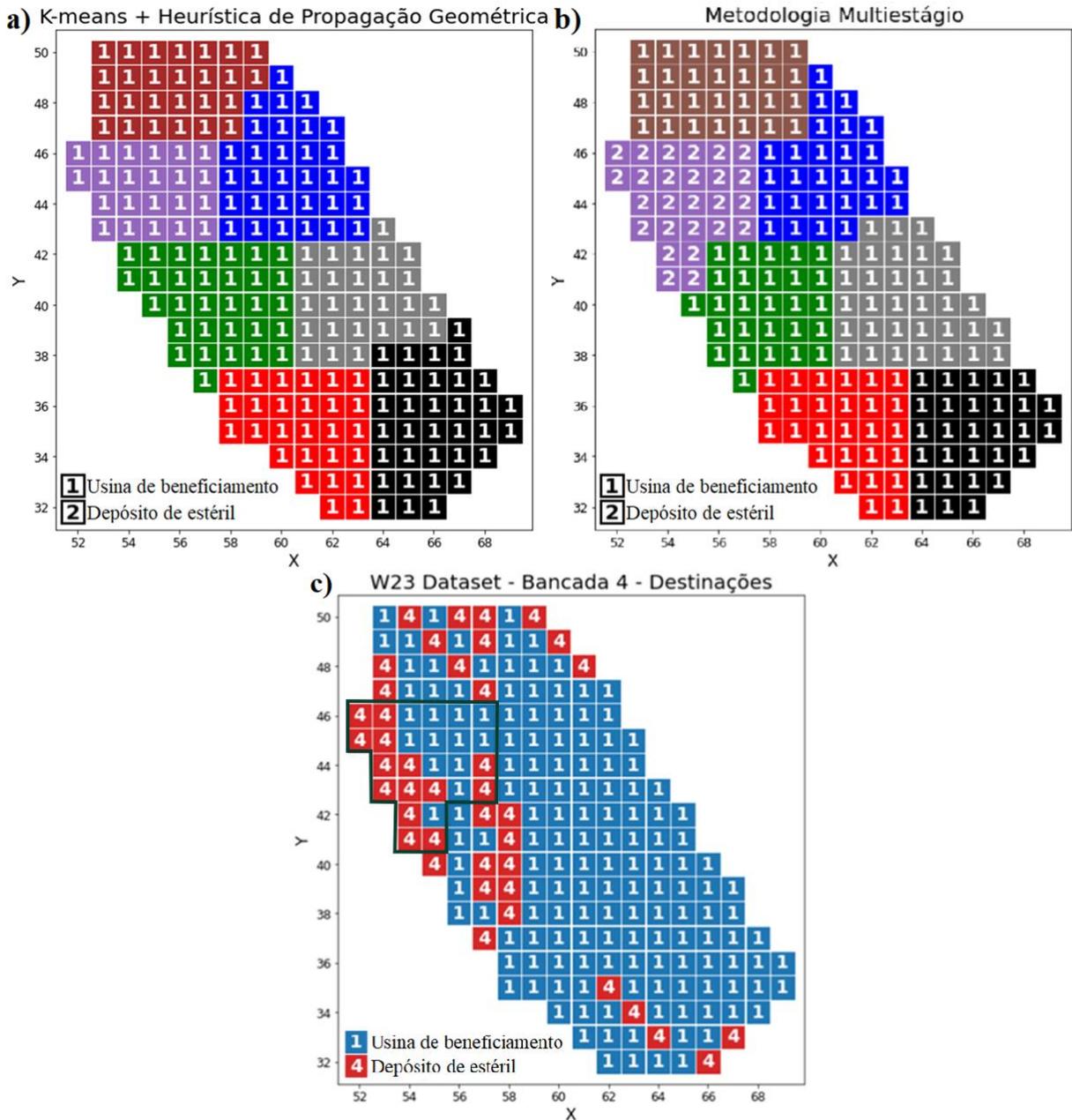


Tabela 9 – Comparação entre os resultados obtidos durante o experimento numérico 2 e uma versão reduzida da metodologia multiestágio.

Parâmetros*	<i>K-Means + GPH 2</i>	Experimento numérico 2
Número de <i>clusters</i>	7	7
Diâmetro máximo (γ)	29	26
<i>min_size_cluster</i>	22	26
<i>max_size_cluster</i>	34	33
Função objetivo	$7,35 \times 10^{10}$	$8,05 \times 10^{10}$
Tempo de processamento (s)	13	6.327
Índice da Silhueta	-0,04	-0,04
Índice Calinski-Harabasz	9,94	9,96
Índice Davies-Bouldin	3,89	3,69
Teor médio (g/t)	0,06	0,06
Minério processado (t)	273,80	241,35
Minério descartado (t)	0,00	32,45
Estéril processado (t)	473.711,14	411.187,20
Estéril descartado (t)	0,00	62.523,94
Diluição (t)	98.646,61	64.962,40
Perda de minério (t)	0,00	28.872,18
Valor presente (\$)	$1,38 \times 10^6$	$2,05 \times 10^6$

* Os parâmetros hachurados apresentam os melhores resultados; aos que não possuem hachura, esta lógica não se aplica.

Neste experimento, a diferença entre as configurações dos *clusters* parecem mais uma vez ser pequenas, mas enquanto a metodologia reduzida obteve uma solução em que todos os blocos seriam enviados ao beneficiamento, a solução obtida pela metodologia multiestágio completa contém um *cluster* que será enviado à pilha de estéril. Desta vez, as diferenças entre os parâmetros γ , *min_size_cluster* e *max_size_cluster* são um pouco maiores. Considerando os *cluster* com menos e mais blocos em ambas as soluções, a diferença entre estes na solução obtida pela metodologia reduzida é de 12 blocos, ao passo que a metodologia multiestágio completa apresenta uma diferença de 7 blocos. Por outro lado, as métricas de qualidade dos *clusters* são quase idênticas em ambas soluções. Visto que uma fração dos blocos foi descartada pela metodologia multiestágio completa, houve uma redução na diluição, já que menos estéril foi enviado à planta. Entretanto, houve uma perda de minério durante este descarte, quando a solução reduzida processava todo o minério. O valor presente obtido pela metodologia completa foi significativamente maior que o obtido pela metodologia reduzida, havendo um incremento de 32,68% no retorno econômico quando comparado a esta última. Mais uma vez o tempo de execução da metodologia reduzida foi significativamente menor, demandando apenas 13 segundos enquanto a metodologia multiestágio completa empregou 6.327 segundos na busca por uma solução, tendo a primeira encontrado uma

solução viável em 0,21% do tempo empregado pela última. Desta vez, a metodologia completa atingiu também um maior valor de função objetivo, havendo um incremento de 8,69% em relação à metodologia reduzida. No experimento numérico 2, portanto, embora a metodologia multiestágio reduzida continue sendo muito mais ágil que a metodologia completa, a primeira não foi capaz de produzir uma solução com boa seletividade, enviando todos os blocos para o beneficiamento e reduzindo, assim, a função objetivo e o valor presente.

Finalmente, considerando os parâmetros empregados no experimento numérico 3, a Figura 41 apresenta a distribuição espacial dos *clusters* gerados (a) pela metodologia multiestágio reduzida e (b) daqueles gerados pela metodologia multiestágio completa. Já a Tabela 10 compara os resultados de ambas.

Figura 41 - Considerando as configurações do experimento numérico 3, comparação entre os agrupamentos gerados a) pela aplicação do *k-means* combinado com a heurística *GPH* e b) pela otimização multiestágio baseada em *COP*. Considerando ainda c) os *clusters* destinados ao depósito de estéril pelas metodologias multiestágio reduzida (em amarelo) e completa (em preto), é possível identificar quais destes blocos foram mal classificados.

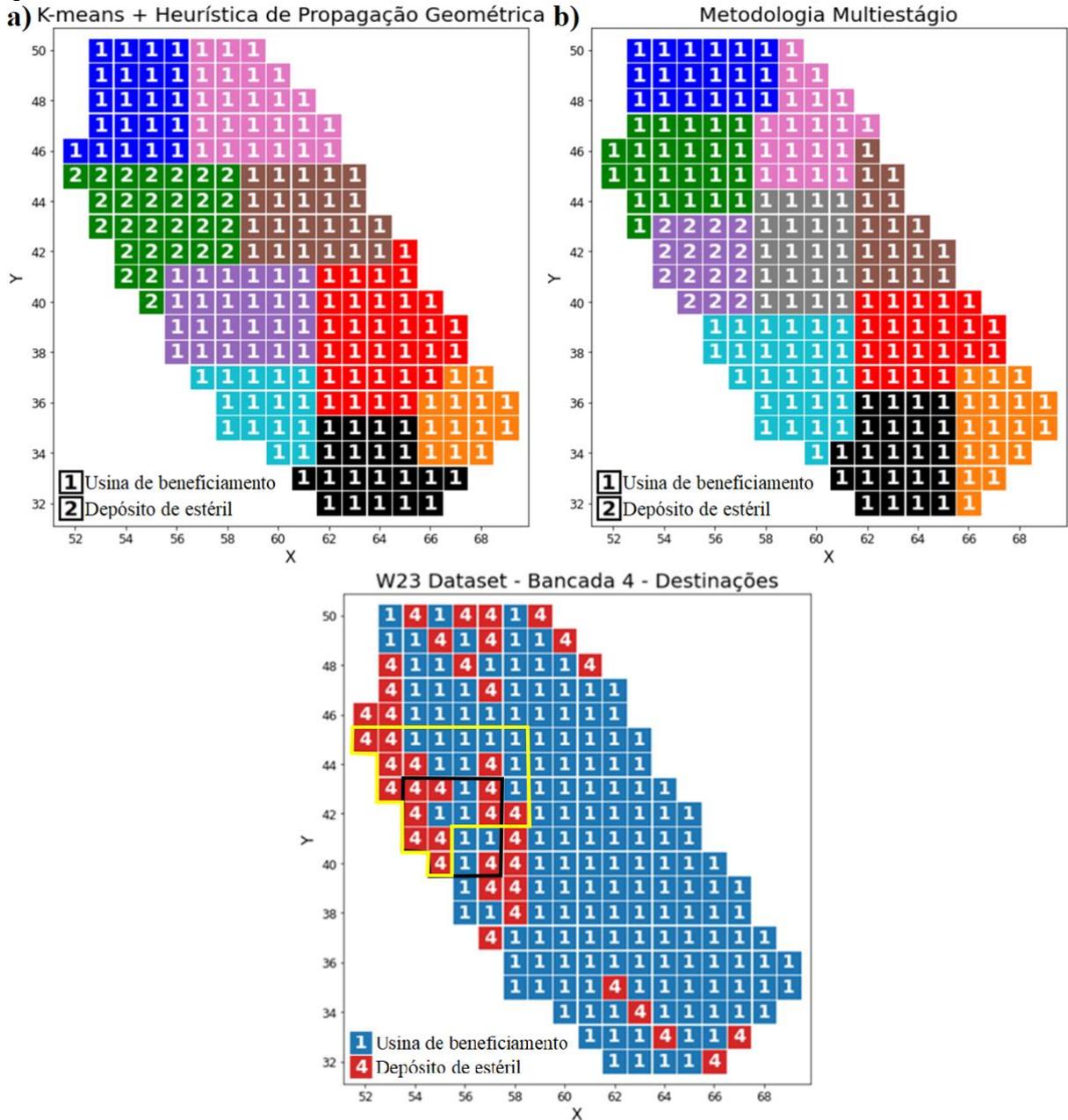


Tabela 10 – Comparação entre os resultados obtidos durante o experimento numérico 3 e uma versão reduzida da metodologia multiestágio.

Parâmetros*	<i>K-Means</i> + <i>GPH 3</i>	Experimento numérico 3
Número de <i>clusters</i>	9	10
Diâmetro máximo (γ)	22	20
<i>min_size_cluster</i>	13	15
<i>max_size_cluster</i>	31	26
Função objetivo	$1,20 \times 10^{11}$	$1,28 \times 10^{11}$
Tempo de processamento (s)	12	4.294
Índice da Silhueta	-0,08	-0,11
Índice Calinski-Harabasz	8,58	7,65
Índice Davies-Bouldin	3,51	3,76
Teor médio (g/t)	0,06	0,06
Minério processado (t)	241,29	259,07
Minério descartado (t)	32,51	14,73
Estéril processado (t)	408.781,24	437.635,64
Estéril descartado (t)	64.929,90	36.075,50
Diluição (t)	62.556,39	76.992,48
Perda de minério (t)	28.872,18	14.436,09
Valor presente (\$)	$2,28 \times 10^6$	$2,13 \times 10^6$

* Os parâmetros hachurados apresentam os melhores resultados; aos que não possuem hachura, esta lógica não se aplica.

Nesta última comparação, ao contrário dos experimentos anteriores, a diferença entre as configurações dos *clusters* são relevantes e seus formatos são bastante distintos. Não somente a metodologia multiestágio completa encontrou uma solução com 1 *cluster* a mais que a metodologia reduzida, mas é conspícuo que a primeira destina uma quantidade muito menor de blocos à pilha de estéril (15 blocos no *cluster* roxo) que a última (27 blocos no *cluster* verde). Embora as diferenças quanto ao parâmetro γ continuem pequenas, os parâmetros *min_size_cluster* e *max_size_cluster* diferem ainda mais. Considerando os *cluster* com menos e mais blocos em ambas as soluções, a diferença entre estes na solução obtida pela metodologia reduzida é de 18 blocos, ao passo que a metodologia multiestágio completa apresenta uma diferença de 11 blocos. Novamente as métricas de qualidade dos *clusters* são quase idênticas em ambas soluções. Como a metodologia completa descartou menos blocos, sua perda de minério foi menor que a da metodologia reduzida, todavia, esta apresentou menor diluição ao enviar menos estéril ao beneficiamento. O valor presente obtido pela metodologia reduzida foi maior, incrementando 6,58% no retorno econômico quando comparado com a metodologia completa. Novamente o tempo de execução foi muito discrepante, sendo empregados 12 segundos pela metodologia reduzida e 4.294 segundos pela metodologia multiestágio completa, tendo a primeira encontrado uma solução viável em

0,28% do tempo empregado pela última. O valor da função objetivo da metodologia completa, entretanto, foi 6,67% mais elevado que o da metodologia reduzida, de modo que no experimento numérico 3 ambas metodologias foram satisfatórias quanto à obtenção de uma solução ótima local, embora o tempo de processamento seja o fator mais relevante em favor da metodologia reduzida.

Percebe-se que o aumento do número de *clusters* durante os experimentos numéricos gerou soluções com maiores diferenças quanto aos parâmetros *min_size_cluster* e *max_size_cluster*, e em geral as soluções obtidas pela metodologia reduzida apresentaram uma diferença maior que as soluções obtidas pela metodologia multiestágio completa. As métricas de qualidade dos *clusters*, por outro lado, mostraram-se pouco relevantes quanto à análise da qualidade das soluções. Considerando as três configurações empregadas, a diferença no valor de função objetivo entre as metodologias completa e reduzida variou entre 6% e 10%, onde a última foi superior no experimento numérico 1 e a primeira foi melhor nos experimentos numéricos 2 e 3. Quanto ao retorno econômico, houve um empate e uma vitória para cada metodologia, mas no experimento 2 a metodologia multiestágio completa obteve uma solução cujo valor presente foi 32,68% mais elevado que a obtida pela metodologia reduzida.

Por meio destes dados é possível afirmar que a metodologia completa obtém soluções melhores que a metodologia reduzida, principalmente por esta última ter sido incapaz de gerar uma solução que permitisse a seletividade operacional quando comparada com os resultados do experimento numérico 2, reduzindo significativamente seu retorno econômico.

Entretanto, a diferença entre os tempos de processamento das duas metodologias é brutal, visto que a metodologia reduzida consumiu entre 0,14% e 0,28% do tempo empregado pela metodologia multiestágio completa. Esta comparação indica que, em trabalhos futuros, esforços devem ser envidados com o propósito de empregar uma heurística ou metaheurística mais eficiente na geração de uma solução ótima local que a metodologia multiestágio baseada em *COP* proposta neste estudo. Ao se empregar um algoritmo mais eficiente que o *k-means*, mas menos custoso que todo o *framework* de otimização baseada em *MILP* empregado, deve haver ganhos significativos quanto ao tempo de execução sem que haja perda na qualidade das soluções obtidas.

A heurística *GPH*, finalmente, provou ser uma valiosa ferramenta no pós-processamento de soluções que apresentam falhas nas fronteiras e na continuidade dos *clusters*, corrigindo soluções produzidas por ambas metodologias. Entretanto, por não conter nenhum mecanismo capaz de destruir e reconstruir partes das soluções, eventualmente ela fica

presa e não consegue atingir seu objetivo, sendo necessário reiniciá-la ou mesmo gerar uma nova solução por meio de otimização. Um trabalho futuro, portanto, deveria focar no emprego de um algoritmo mais robusto que o *k-means* para gerar uma solução ótima local e em uma variante da heurística *GPH* que seja mais robusta que a atual. Entretanto, deve-se reiterar que as soluções propostas neste estudo atingiram satisfatoriamente o objetivo proposto, que era resolver o problema do agrupamento de blocos em polígonos de lavra para minas a céu aberto, visando melhorar o desempenho e a operacionalidade do sequenciamento de lavra determinístico no nível do planejamento operacional.

5 CONCLUSÕES

Considerando a demanda crescente da indústria mineral por um planejamento de lavra robusto e aderente à realidade, além da dificuldade de resolver problemas de sequenciamento de lavra por meio da otimização, encontrar formas mais eficientes de solucionar esta tarefa continua sendo um desafio relevante. Neste contexto, o agrupamento de blocos em polígonos de lavra pode, sem dúvida, simplificar as tarefas relacionadas ao planejamento de minas céu aberto, reduzindo o tamanho do problema de sequenciamento e acrescentando restrições operacionais a este que possibilitarão a geração de planos mais aderentes à realidade.

Este estudo apresentou uma metodologia consistente e capaz de ser aplicada a diversos níveis de problemas. Dentre suas contribuições inovadoras estão a utilização de uma nova função de similaridade para ser empregada no problema do agrupamento, a elaboração de um modelo matemático baseado em *MILP* para representar este problema e o emprego da programação por restrições para resolver o modelo. Como esta metodologia provou-se insatisfatória ao tentar resolver instâncias com mais de 100 blocos, foi proposta uma otimização multiestágio baseada em programação por restrições, que dentre seus estágios contém a heurística de propagação geométrica (*GPH*), um novo algoritmo capaz de corrigir fronteiras de *clusters* mal formuladas. A metodologia multiestágio foi comparada ainda com uma metodologia reduzida, quando somente o *k-means* e a *GPH* eram empregados, sendo esta também uma opção para gerar soluções ótimas locais.

5.1 CONSIDERAÇÕES FINAIS EM RELAÇÃO AOS EXPERIMENTOS

Primeiramente, ao aplicar a metodologia simplificada, onde o modelo matemático baseado em *MILP* era diretamente resolvido por meio de uma abordagem *COP*, foi possível perceber que instâncias com menos de 100 blocos podiam ser rapidamente solucionadas mesmo quando algoritmos exatos não eram capazes de resolver o problema em tempo computacional aceitável. Entretanto, ao comparar com a melhor solução obtida por *CSP*, percebeu-se que o valor presente da solução obtida por *COP* não era tão elevado quanto ao resultado inicial, indicando que outras funções de similaridade podem ser empregadas caso maximizar o retorno econômico seja a finalidade principal do usuário.

Por outro lado, ao empregar a metodologia simplificada em instâncias maiores, foi constatado que o tempo de execução para resolver o modelo matemático aumentou muito e que um mau ajuste do parâmetro γ (diâmetro máximo) muitas vezes resultava em soluções

com *clusters* descontínuos ou com fronteiras misturadas. Portanto, ao fornecer uma solução de dica (*hint solution*) por meio do *k-means*, depois ajustar o parâmetro γ e efetuar a exploração do espaço de busca usando o *COP*, e ainda aplicar uma fase de pós-processamento para corrigir a solução, a metodologia multiestágio baseada em *COP* foi capaz de convergir para soluções que eram ótimos locais em um tempo aceitável. Entretanto, o tempo computacional cresceu substancialmente ao longo deste processo. Ao concluir os três experimentos numéricos propostos, pairou também uma dúvida quanto a real necessidade de se empregar todas estas fases da metodologia multiestágio, visto que os *clusters* de algumas soluções apresentavam certa semelhança com os gerados por *k-means*.

Foi proposta então uma metodologia multiestágio reduzida, onde somente o *k-means* e a *GPH* foram empregados na resolução do problema, cujos resultados foram comparados com os obtidos durante os três experimentos numéricos em que a metodologia multiestágio completa foi empregada. O que mais chamou a atenção nesta comparação foi o tempo de execução da metodologia reduzida, demandando menos de 1% do tempo necessário à metodologia completa ao longo dos três experimentos numéricos. Quanto aos outros parâmetros, ora o valor presente e o valor da função objetivo foram melhores através da metodologia completa, ora foram melhores através da metodologia simplificada, mas durante o experimento numérico 2 esta não foi capaz de gerar uma solução com boa seletividade operacional, reduzindo sobremaneira o retorno econômico. Portanto, a metodologia reduzida demonstrou ser menos confiável quanto à sua capacidade de chegar a um ótimo local, mas sem dúvida o tempo de execução indica que ela poderia ser uma ferramenta valiosa, indicando até mesmo que a metodologia multiestágio completa pode ser aprimorada.

Finalmente, é notável a eficiência demonstrada pela *GPH* no reparo de soluções com fronteiras mal formuladas ou *clusters* desconexos, corrigindo em pouco mais de 10 segundos as soluções obtidas pelos diferentes conjuntos de parâmetros e metodologias empregados neste estudo. Seu potencial vai além das metodologias de otimização propostas neste estudo, pois ela é capaz de ser empregada como uma fase de pós-processamento em soluções obtidas por quaisquer métodos de otimização. Entretanto, por carecer de um mecanismo capaz de destruir e reconstruir partes de uma solução, eventualmente esta heurística pode ficar presa em um *looping* e não consegue concluir sua tarefa, situação que deve ser corrigida em trabalhos futuros.

5.2 TRABALHOS FUTUROS

Conforme demonstrado na comparação entre os resultados obtidos pelas metodologias multiestágio completa (baseada em *COP*) e reduzida, embora as etapas de otimização baseadas na solução do modelo matemático tenham sido capazes de gerar soluções mais robustas que aquelas geradas apenas por heurísticas, o custo computacional foi considerável. Estudos futuros poderão abordar um modelo matemático mais simples que o proposto neste estudo, devendo-se investigar a possibilidade de representar o problema do agrupamento de polígonos de lavra como um modelo matemático convexo similar ou equivalente ao problema original. Desde a concepção dos métodos dos pontos interiores para resolução de problemas de Programação Linear (*LP*) por Karmarkar (1984), uma nova classe de algoritmos foi criada e generalizada para resolver problemas baseados em Programação Quadrática (*Quadratic Programming, QP*), Programação Quadrática Quadraticamente Restringida (*Quadratically Constrained Quadratic Program, QCQP*), Programação Cônica de Segunda Ordem (*Second-Order Cone Programming, SOCP*), dentre outras (BOYD; VANDENBERGHE, 2009).

Um domínio $\mathcal{S} \subset \mathbb{R}^n$ pode ser considerado convexo se $\lambda \mathbf{x} + (\mathbf{1} - \lambda) \mathbf{y} \in \mathcal{S}$ sempre que \mathbf{x} e \mathbf{y} estão em \mathcal{S} e $\lambda \in [0, 1]$, ou seja, o domínio é convexo se o segmento de reta fechado $[\mathbf{x}, \mathbf{y}]$ estiver inteiramente contido em \mathcal{S} sempre que seus extremos \mathbf{x} e \mathbf{y} estiverem em \mathcal{S} . Seja $I = \{i: i = 1, \dots, m\}$. Se os domínios $\mathcal{S}_i \subset \mathbb{R}^n$ são convexos para $i \in I$, então sua interseção $\bigcap_{i \in I} \mathcal{S}_i$ é convexa. Se, além disso, cada \mathcal{S}_i é não vazio e $\mu_i \in \mathbb{R}$, então o domínio $\sum_{i \in I} \mu_i \mathcal{S}_i$ é convexo. Uma combinação linear $\sum_{i \in I} \lambda_i \mathcal{S}_i$ é considerada uma combinação convexa de elementos $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathbb{R}^n$ se cada $\lambda_i \geq 0$ e $\sum_{i \in I} \lambda_i = 1$. Para qualquer domínio $\mathcal{S} \subset \mathbb{R}^n$, pode-se definir uma envoltória convexa (*convex hull*) de \mathcal{S} como um domínio de todas as combinações convexas possíveis de seus elementos, onde $\mathbf{conv} \mathcal{S} = \{\mathbf{x} \in \mathbb{R}^n: \mathbf{x} = \sum_{i \in I} \lambda_i \mathbf{x}_i, \sum_{i \in I} \lambda_i = 1, \mathbf{x}_i \in \mathcal{S}, \lambda_i \geq 0\}$. O $\mathbf{conv} \mathcal{S}$ é o menor domínio convexo contendo \mathcal{S} , e \mathcal{S} é convexo se e somente se $\mathcal{S} = \mathbf{conv} \mathcal{S}$ (BOYD; VANDENBERGHE, 2009; BAGIROV; KARMITSA; TAHERI, 2020).

Ao consultar a literatura disponível sobre problemas de agrupamento, é notável que poucas técnicas de agrupamento não supervisionado tenham sido utilizadas até então para resolver o problema de agrupamento de polígonos de lavra. Mesmo em relação ao agrupamento não supervisionado, basicamente versões de um algoritmo baseado em clusterização hierárquica aglomerativa têm sido utilizadas nos últimos anos. Porém, apenas no âmbito das metaheurísticas, Bagirov, Karmita e Taheri (2020) citam o uso do *Tabu Search*

(TS), *Simulated Annealing (SA)*, *Genetic Algorithm (GA)*, *Artificial Bee Colony (ABC)*, *Particle Swarm Optimization (PSO)* e *Ant Colony Optimization (ACO)* na resolução de problemas de agrupamento, técnicas utilizadas desde a década de 1980. Embora estas sejam as metaheurísticas mais conhecidas pelo público em geral, existem dezenas de técnicas e variantes disponíveis na literatura que podem trazer resultados satisfatórios na abordagem do problema do agrupamento de blocos, tais quais o *Firefly Algorithm (FFA)*, *Greedy Randomized Adaptive Search Procedures (GRASP)* e *Grey Wolf Optimization (GWO)*. Outra maneira de reduzir o custo computacional da metodologia multiestágio completa seria substituir uma ou mais de suas etapas por algoritmos mais eficientes, tal que seja obtida mais rapidamente uma configuração de *clusters* por otimização, possibilitando a aplicação da *GPH* apenas para corrigir falhas nesta solução. Uma vez que seja proposta uma forma de resolver o *mining cut clustering problem* com menos esforço computacional, é possível abordar diferentes escopos de planejamento de lavra por meio de variantes das soluções propostas neste estudo, já que os experimentos executados se voltaram ao planejamento de lavra operacional.

O problema do limite de escavação (SARI; KUMRAL, 2018), por sua vez, determina zonas de minério e estéril em uma bancada considerando a dimensão do equipamento de lavra, mas existem poucas abordagens para o problema de agrupamento de polígonos de lavra que sejam capazes de gerar *clusters* que satisfaçam plenamente restrições relacionadas a estes equipamentos. Portanto, ambos problemas estão relacionados e podem ser abordados em conjunto, o que possibilitaria a melhoria das soluções tanto no que diz respeito às fronteiras dos depósitos quanto às fronteiras entre *clusters*. Desta forma, seria assegurado que a *GPH* operaria em instâncias com configuração 100% operacional, sem arestas e reentrâncias nas fronteiras do depósito que sejam inacessíveis aos equipamentos de lavra.

É possível ainda abordar diferentes funções de similaridade, incluindo sua subdivisão em diferentes funções, o que resultaria em um problema de otimização multiobjetivo. Os estudos publicados até o momento empregam diferentes parâmetros em uma única função de similaridade e atribuem pesos diferentes a eles, mas este procedimento resulta em uma única função objetivo, sendo, portanto, uma abordagem de otimização mono-objetivo. Entretanto, dependendo dos pesos atribuídos a cada parâmetro componente da função de similaridade, principalmente quando estes representam objetivos mutuamente conflitantes, alguns parâmetros serão ofuscados pelos demais. Por outro lado, ao subdividir diferentes parâmetros em funções de similaridade distintas representando diferentes objetivos, é possível encontrar

soluções que satisfaçam simultaneamente objetivos conflitantes, soluções que provavelmente não seriam encontradas por meio de uma abordagem mono-objetivo.

Considerando o escopo da otimização multiobjetivo, ao invés de gerar uma única solução ótima, múltiplas soluções viáveis são geradas por técnicas como a abordagem da soma ponderada, a abordagem da restrição ϵ e os algoritmos evolutivos. Idealmente, essas técnicas devem gerar soluções distribuídas ao longo de todo o espaço de soluções multiobjetivo, devendo ser capazes também de gerar soluções que sejam ótimos locais em relação a todas as funções objetivo simultaneamente. Uma vez geradas as soluções, deve-se utilizar o critério de não dominância para determinar uma frente de Pareto, indicando quais dessas soluções são as melhores considerando os objetivos escolhidos (MIETTINEN; RUIZ; WIERZBICKI, 2008). Então, métodos de decisão multicritério podem ser usados para selecionar a mais adequada entre as soluções não dominadas para o problema de agrupamento de polígonos de lavra, como os métodos *TOPSIS*, *LINMAP*, *VIKOR*, dentre outros (WANG; RANGAIAH, 2017). Por exemplo, seria possível determinar uma solução cuja configuração de *clusters* maximizasse o critério de similaridade em relação às litologias dos blocos, ao mesmo tempo que minimizasse os custos na planta de processamento, configuração que dificilmente seria identificada pela abordagem tradicional.

Em relação às soluções propostas para agregar blocos em *mining cuts* considerando seu emprego em uma estrutura voltada ao planejamento de lavra estocástico, poucos estudos foram publicados. Contudo, há uma tendência de popularização das técnicas probabilísticas, dada a sua capacidade de gerar soluções robustas e avessas ao risco. Ainda hoje, o potencial de incorporar a incerteza geológica através do sequenciamento de lavra estocástico é subutilizado, assim como a incerteza econômica, sendo poucas as soluções industriais capazes de desenvolver estas metodologias. O uso da simulação por eventos discretos (*DES*) para gerar cenários considerando a incerteza operacional, incluindo o desempenho de frotas de caminhões e usinas de beneficiamento, é mais difundido do que as outras abordagens probabilísticas mencionadas. No entanto, existe pouca integração entre as etapas de otimização e simulação aplicadas ao planejamento de lavra operacional, sendo esta outra lacuna. Portanto, ainda há um enorme potencial a ser explorado no que diz respeito à agregação de blocos considerando a incerteza. Se muitas simplificações já são necessárias para viabilizar a soluções de grandes problemas de sequenciamento de lavra determinístico, a abordagem estocástica exigiria ainda mais esforço computacional, tal que agregar blocos em *mining cuts* pode reduzir significativamente o tamanho destes problemas de planejamento.

REFERÊNCIAS

- AGGARWAL, Charu C.; REDDY, Chandan K.. **Data Clustering: Algorithms and Applications**. Boca Raton: CRC Press, 2014.
- ALBACH, Horst. Long Range Planning in Open-Pit Mining. **Management Science**, [S.L.], v. 13, n. 10, p. B549-B568, jun. 1967. Institute for Operations Research and the Management Sciences (INFORMS). <http://dx.doi.org/10.1287/mnsc.13.10.b549>.
- ALIPOUR, Aref; KHODAIARI, Ali Asghar; JAFARI, Ahmad; TAVAKKOLI-MOGHADDAM, Reza. Production scheduling of open-pit mines using genetic algorithm: a case study. **International Journal Of Management Science And Engineering Management**, [S.L.], v. 15, n. 3, p. 176-183, 1 nov. 2020. Informa UK Limited. <http://dx.doi.org/10.1080/17509653.2019.1683090>.
- AMIRI, Maryam; AMNIEH, Hassan Bakhshandeh; HASANIPANAH, Mahdi; KHANLI, Leyli Mohammad. A new combination of artificial neural network and K-nearest neighbors models to predict blast-induced ground vibration and air-overpressure. **Engineering With Computers**, [S.L.], v. 32, n. 4, p. 631-644, 29 fev. 2016. Springer Science and Business Media LLC. <http://dx.doi.org/10.1007/s00366-016-0442-5>.
- ASKARI-NASAB, Hooman; AWUAH-OFFEI, Kwame; EIVAZY, Hesameddin. Large-scale open pit production scheduling using Mixed Integer Linear Programming. **International Journal of Mining and Mineral Engineering**, [S.L.], v. 2, n. 3, p. 185-214, dez. 2010. Inderscience Enterprises Ltd.
- ASKARI-NASAB, Hooman; TABESH, Mohammad; BADIOZAMANI, Mohammad. Creating Mining Cuts Using Hierarchical Clustering and Tabu Search Algorithms. In: International Conference On Mining Innovation, 4., 2010, Santiago. **Proceedings [...]**. Santiago: Gecamin, 2010. p. 159-171.
- ASKARI-NASAB, Hooman; TABESH, Mohammad; BADIOZAMANI, Mohammad. M., EIVAZY, Hesam. Hierarchical clustering algorithm for block aggregation in open pit mines. In: TOPAL, Erkan; KURUPPU, Mahinda (Eds.) International Symposium on Mine Planning and Equipment Selection - MPES 2010, 19., 2010, Fremantle. **Proceedings [...]**. Carlton: The Australasian Institute of Mining and Metallurgy, 2010. p. 469-479.
- ASKARI-NASAB, Hooman; EIVAZY, Hesam; TABESH, Mohammad; BADIOZAMANI, Mohammad Mahdi. A mathematical programming model for open pit short-term production scheduling. In: Society for Mining, Metallurgy & Exploration Annual Meeting, 2011, Denver. **Proceedings [...]**. Littleton: Society Of Mining, Metallurgy And Exploration, 2011a.

ASKARI-NASAB, Hooman; POURRAHIMIAN, Yashar; BEN-AWUAH, Eugene; KALANTARI, Samira. Mixed integer linear programming formulations for open pit production scheduling. **Journal Of Mining Science**, [S.L.], v. 47, n. 3, p. 338-359, maio 2011b. Pleiades Publishing Ltd. <http://dx.doi.org/10.1134/s1062739147030117.2011b>.

ASKARI-NASAB, Hooman; MIETH, Clemens; TABESH, Mohammad. A hybrid local search/mixed integer programming approach to open pit controlled phase-design. In: COSTA, João Felipe; KOPPE, Jair; PERONI, Rodrigo (Eds.) *Application of Computers and Operations Research in the Mineral Industry*, 36., 2013, Porto Alegre. **Proceedings [...]**. Porto Alegre: Fundação Luiz Englert, 2013. p. 1-14.

BADIOZAMANI, Mohammad Mahdi; ASKARI-NASAB, Hooman. Towards integration of oil sands mine planning with tailings and reclamation plans. In: International Conference on Tailings and Mine Waste, 16., 2012, Keystone. **Proceedings [...]**. Vancouver: University of British Columbia, 2012.

BADIOZAMANI, Mohammad Mahdi; ASKARI-NASAB, Hooman. Integration of reclamation and tailings management in oil sands surface mine planning. **Environmental Modelling & Software**, [S.L.], v. 51, p. 45-58, jan. 2014. Elsevier BV. <http://dx.doi.org/10.1016/j.envsoft.2013.09.026>.

BAGIROV, Adil M.; KARMITSA, Napsu; TAHERI, Sona. **Partitional Clustering via Nonsmooth Optimization: Unsupervised And Semi-Supervised Learning**, [S.L.], v. 0, n. 0, p. 0-10, 2020. Springer International Publishing. <http://dx.doi.org/10.1007/978-3-030-37826-4>.

BELLMAN, Richard. On a routing problem. **Quarterly Of Applied Mathematics**, [S.L.], v. 16, n. 1, p. 87-90, 1958. American Mathematical Society (AMS). <http://dx.doi.org/10.1090/qam/102435>.

BEN-AWUAH, Eugene; ASKARI-NASAB, Hooman. Oil sands mine planning and waste management using mixed integer goal programming. **International Journal Of Mining, Reclamation And Environment**, [S.L.], v. 25, n. 3, p. 226-247, set. 2011. Informa UK Limited. <http://dx.doi.org/10.1080/17480930.2010.549656>.

BEN-AWUAH, Eugene; ASKARI-NASAB, Hooman; AWUAH-OFFEI, Kwame. Production Scheduling and Waste Disposal Planning for Oil Sands Mining Using Goal Programming. **Journal Of Environmental Informatics**, [S.L.], v. 20, n. 1, p. 20-33, 1 set. 2012. International Society for Environmental Information Science (ISEIS). <http://dx.doi.org/10.3808/jei.201200217>.

- BLOM, Michelle; PEARCE, Adrian R.; STUCKEY, Peter J.. Short-term planning for open pit mines: a review. **International Journal of Mining, Reclamation and Environment**, v. 33, n.5, p. 318-339, 2019. <http://dx.doi.org/10.1080/17480930.2018.1448248>.
- BOLAND, Natashia; DUMITRESCU, Irina; FROYLAND, Gary; GLEIXNER, Ambros M.. LP-based disaggregation approaches to solving the open pit mining production scheduling problem with block processing selectivity. **Computers & Operations Research**, [S.L.], v. 36, n. 4, p. 1064-1089, abr. 2009. Elsevier BV. <http://dx.doi.org/10.1016/j.cor.2007.12.006>.
- BOND, Gary D.. **A mathematical analysis of the Lerchs and Grossmann algorithm and the nested Lerchs and Grossmann algorithm**. 1995. 250 f. PhD Thesis (Doctorate) - Mining Engineering Course, Colorado School of Mines, Golden, 1995.
- BOYD, Stephen; VANDENBERGHE, Lieven. **Convex Optimization**. Cambridge: Cambridge University Press; 2009.
- BURSTALL, R. M.. A Program for Solving Word Sum Puzzles. **The Computer Journal**, [S.L.], v. 12, n. 1, p. 48-51, 1 fev. 1969. Oxford University Press (OUP). <http://dx.doi.org/10.1093/comjnl/12.1.48>.
- BUSNACH, E.; MEHREZ, A.; SINUANY-STERN, Z.. A Production Problem in Phosphate Mining. **The Journal of the Operational Research Society**, [S.L.], v. 36, n. 4, p. 285-288, 1985. Palgrave Macmillan Journals. <https://doi.org/10.2307/2582414>.
- CACCETTA, Louis; HILL, Stephen P.. An application of branch and cut to open pit mine scheduling. **Journal Of Global Optimization**, [S.L.], v. 27, n. 2/3, p. 349-365, 2003. Springer Science and Business Media LLC. <http://dx.doi.org/10.1023/a:1024835022186>.
- CALINSKI, Tadeusz; HARABASZ, J.. A dendrite method for cluster analysis. **Communications In Statistics - Theory And Methods**, [S.L.], v. 3, n. 1, p. 1-27, 1974. Informa UK Limited. <http://dx.doi.org/10.1080/03610927408827101>.
- CANESSA, Gianpiero; MORENO, Eduardo; PAGNONCELLI, Bernardo K.. The risk-averse ultimate pit problem. **Optimization And Engineering**, [S.L.], 8 ago. 2020. Springer Science and Business Media LLC. <http://dx.doi.org/10.1007/s11081-020-09545-4>.
- CAPPONI, Luciano Nunes. **Planejamento estocástico de curto prazo incorporando a incerteza da estimativa no controle de teores**. 2019. 184 f. Tese (Doutorado) – Programa de Pós-Graduação em Engenharia de Minas, Metalúrgica e de Materiais, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2019.
- CHANDA, Emmanuel C. Knox. An application of goal programming to production planning in the crushed stone industry. **International Journal of Surface Mining, Reclamation and Environment**, v. 4, n. 3, p. 125-129, 1990. <https://doi.org/10.1080/09208119008944179>.

CHANDA, Emmanuel C. Knox; WILKE, F.. An EPD-model of open pit short term production scheduling optimization for stratiform orebodies. In: International APCOM Symposium, 23., 1992, Tucson. **Proceedings [...]**. Littleton: Society for Mining, Metallurgy, and Exploration, 1992. p. 759-768.

CHANDA, Emmanuel C. Knox; RICCIARDONE, Jacob. Long Term Production Scheduling Optimisation for a Surface Mining Operation: an application of MineMax™ scheduling software. **International Journal Of Surface Mining, Reclamation And Environment**, [S.L.], v. 16, n. 2, p. 144-158, jun. 2002. Informa UK Limited. <http://dx.doi.org/10.1076/ijsm.16.2.144.3400>.

CHERCHENEVSKI, Pablo Koury; COSTA, João Felipe Coimbra Leite; RUBIO, Ricardo Hundelshausen. Grade uncertainty embedded in long term scheduling: stochastic mine planning. **Rem - International Engineering Journal**, [s.l.], v. 72, n. 2, p.275-284, jun. 2019. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/0370-44672018720119>.

COLMERAUER, Alain; ROUSSEL, Philippe. The birth of Prolog. **Acm Sigplan Notices**, [S.L.], v. 28, n. 3, p. 37-52, mar. 1993. Association for Computing Machinery (ACM). <http://dx.doi.org/10.1145/155360.155362>.

CORMEN, Thomas H.; LEISERSON, Charles E.; RIVEST, Ronald L.; STEIN, Clifford. **Introduction to Algorithms**. 3. ed. Cambridge: The Mit Press, 2009. 1313 p.

COWLING, Peter I.; GREENBERG, Harvey J.; DE JONG, Kenneth. Artificial Intelligence. In: GASS, Saul I.; FU, Michael C. (Eds.) **Encyclopedia of Operations Research and Management Science**. 3. ed. New York: Springer, p.78-83, 2013.

DAGDELEN, Kadri; FRANÇOIS-BONGARÇON, Dominique Marie. Towards the complete double parameterization of recovered reserves in open pit mining. In: International APCOM Symposium, 17., 1982, Golden. **Proceedings [...]**. Littleton: Society for Mining, Metallurgy, and Exploration, 1982. p. 288-296.

DANTZIG, George B.; WOLFE, Philip. Decomposition Principle for Linear Programs. **Operations Research**, v. 8, n. 1, p.101-101, jan. - fev. 1960.

DAO, Thi-Bich-Hanh; DUONG, Khanh-Chuong; VRAIN, Christel. Constrained clustering by constraint programming. **Artificial Intelligence**, [S.L.], v. 244, p. 70-94, mar. 2017. Elsevier BV. <http://dx.doi.org/10.1016/j.artint.2015.05.006>.

DAVIES, David L.; BOULDIN, Donald W.. A Cluster Separation Measure. **Ieee Transactions On Pattern Analysis And Machine Intelligence**, [S.L.], v. -1, n. 2, p. 224-227, abr. 1979. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/tpami.1979.4766909>.

DEL CASTILLO, Maria Fernanda; DIMITRAKOPOULOS, Roussos. A multivariate destination policy for geometallurgical variables in mineral value chains using coalition-formation clustering. **Resources Policy**, [S.L.], v. 50, p. 322-332, dez. 2016. Elsevier BV. <http://dx.doi.org/10.1016/j.resourpol.2016.10.003>.

DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B.. Maximum Likelihood from Incomplete Data Via the EM Algorithm. **Journal Of The Royal Statistical Society: Series B (Methodological)**, [S.L.], v. 39, n. 1, p. 1-22, set. 1977. Wiley. <http://dx.doi.org/10.1111/j.2517-6161.1977.tb01600.x>.

DIMITRAKOPOULOS, Roussos G.; FARRELLY, Christopher Terence; GODOY, Marcelo. Moving forward from traditional optimization: grade uncertainty and risk effects in open-pit design. **Mining Technology**, [s.l.], v. 111, n. 1, p.82-88, abr. 2002. Informa UK Limited. <http://dx.doi.org/10.1179/mnt.2002.111.1.82>.

DIMITRAKOPOULOS, Roussos G.; MARTINEZ, Luis; RAMAZAN, Salih. A maximum upside / minimum downside approach to the traditional optimization of open pit mine design. **Journal of Mining Science**, [s.l.], v. 43, n. 1, p.73-82, jan. 2007. Springer Science and Business Media LLC. <http://dx.doi.org/10.1007/s10913-007-0009-3>.

DIJKSTRA, Edsger W.. A note on two problems in connexion with graphs. **Numerische Mathematik**, [S.L.], v. 1, n. 1, p. 269-271, dez. 1959. Springer Science and Business Media LLC. <http://dx.doi.org/10.1007/bf01386390>.

DUNN, J. C.. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. **Journal Of Cybernetics**, [S.L.], v. 3, n. 3, p. 32-57, jan. 1973. Informa UK Limited. <http://dx.doi.org/10.1080/01969727308546046>.

EIVAZY, Hesam; ASKARI-NASAB, Hooman. A mixed integer linear programming model for short-term open pit mine production scheduling. **Mining Technology**, [S.L.], v. 121, n. 2, p. 97-108, jun. 2012a. Informa UK Limited. <http://dx.doi.org/10.1179/1743286312y.0000000006>.

EIVAZY, Hesameddin; ASKARI-NASAB, Hooman. A hierarchical open-pit mine production scheduling optimisation model. **International Journal Of Mining And Mineral Engineering**, [S.L.], v. 4, n. 2, p. 89-115, 2012b. Inderscience Publishers. <http://dx.doi.org/10.1504/ijmme.2012.052436>.

ELCOCK, E.W.. Absys: the first logic programming language :a retrospective and a commentary. **The Journal Of Logic Programming**, [S.L.], v. 9, n. 1, p. 1-17, jul. 1990. Elsevier BV. [http://dx.doi.org/10.1016/0743-1066\(90\)90030-9](http://dx.doi.org/10.1016/0743-1066(90)90030-9).

EPSTEIN, Rafael; GOIC, Marcel; WEINTRAUB, Andrés; CATALÁN, Jaime; SANTIBÁÑEZ, Pablo; URRUTIA, Rodolfo; CANCINO, Raúl; GAETE, Sergio; AGUAYO, Augusto; CARO, Felipe. Optimizing Long-Term Production Plans in Underground and Open-Pit Copper Mines. **Operations Research**, [S.L.], v. 60, n. 1, p. 4-17, fev. 2012. Institute for Operations Research and the Management Sciences (INFORMS). <http://dx.doi.org/10.1287/opre.1110.1003>.

ESPINOZA, Daniel; GOYCOOLEA, Marcos; MORENO, Eduardo; NEWMAN, Alexandra. MineLib: a library of open pit mining problems. **Annals Of Operations Research**, [S.L.], v. 206, n. 1, p. 93-114, jul. 2013. Springer Science and Business Media LLC. <http://dx.doi.org/10.1007/s10479-012-1258-3>.

ESTER, Martin; KRIEGEL, Hans-Peter; SANDER, Jörg; XU, Xiaowei. A density-based algorithm for discovering clusters in large spatial databases with noise. In: SIMOUDIS, Evangelos; HAN, Jiawei; FAYYAD, Usama M. (Eds.) **KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining**. Palo Alto: AAAI Press, p. 226-231, 1996.

FIKES, Richard E.. REF-ARF: a system for solving problems stated as procedures. **Artificial Intelligence**, [S.L.], v. 1, n. 1-2, p. 27-120, 1970. Elsevier BV. [http://dx.doi.org/10.1016/0004-3702\(70\)90003-2](http://dx.doi.org/10.1016/0004-3702(70)90003-2).

FILLMORE, Jay P.; WILLIAMSON, S. G.. On Backtracking: a combinatorial description of the algorithm. **Siam Journal On Computing**, [S.L.], v. 3, n. 1, p. 41-55, mar. 1974. Society for Industrial & Applied Mathematics (SIAM). <http://dx.doi.org/10.1137/0203004>.

FORD JR., Lester Randolph. **Network Flow Theory**. Santa Monica: Rand Corporation, 1956. Paper P-923.

FOWLKES, E. B.; MALLOWS, C. L.. A Method for Comparing Two Hierarchical Clusterings. **Journal Of The American Statistical Association**, [S.L.], v. 78, n. 383, p. 553-569, set. 1983. Informa UK Limited. <http://dx.doi.org/10.1080/01621459.1983.10478008>.

FREUDER, Eugene C.. Synthesizing constraint expressions. **Communications Of The Acm**, [S.L.], v. 21, n. 11, p. 958-966, nov. 1978. Association for Computing Machinery (ACM). <http://dx.doi.org/10.1145/359642.359654>.

FREUDER, Eugene C.; MACKWORTH, Alan K.. Constraint Satisfaction: An Emerging Paradigm. In: ROSSI, Francesca; VAN BEEK, Peter; WALSH, Toby (Eds.) **Handbook of Constraint Programming**. Amsterdam: Elsevier, p. 13-27, 2006.

- GARAI, Gautam; CHAUDHURI, Bidyut Baran. A novel genetic algorithm for automatic clustering. **Pattern Recognition Letters**, [S.L.], v. 25, n. 2, p. 173-187, jan. 2004. Elsevier BV. <http://dx.doi.org/1>
- GERSHON, Mark E.. Optimal mine production scheduling evaluation of large scale mathematical programming approaches. **International Journal of Mining Engineering**, v. 1, n. 4, p.315-329, dez. 1983.
- GERSHON, Mark E.. Heuristic approaches for mine planning and production scheduling. **International Journal Of Mining And Geological Engineering**, [S.L.], v. 5, n. 1, p. 1-13, mar. 1987. Springer Science and Business Media LLC. <http://dx.doi.org/10.1007/bf01553529>.
- GERSHON, Mark E.; MURPHY, Frederic H.. Optimizing single hole mine cuts by dynamic programming. **European Journal of Operational Research**, [S.L.], v. 38, n. 1, p. 56-62, 1989. Elsevier BV. [https://doi.org/10.1016/0377-2217\(89\)90468-2](https://doi.org/10.1016/0377-2217(89)90468-2).
- GOLDBERG, D. E.. **Genetic Algorithms in Search, Optimization & Machine Learning**. Boston: Addison Wesley, 1989.
- GOLOMB, Solomon W.; BAUMERT, Leonard D.. Backtrack Programming. **Journal Of The Acm**, [S.L.], v. 12, n. 4, p. 516-524, out. 1965. Association for Computing Machinery (ACM). <http://dx.doi.org/10.1145/321296.321300>.
- GONZALEZ, Teofilo F.. Clustering to minimize the maximum intercluster distance. **Theoretical Computer Science**, [S.L.], v. 38, p. 293-306, 1985. Elsevier BV. [http://dx.doi.org/10.1016/0304-3975\(85\)90224-5](http://dx.doi.org/10.1016/0304-3975(85)90224-5).
- GOODFELLOW, Ryan C.; DIMITRAKOPOULOS, Roussos. Global optimization of open pit mining complexes with uncertainty. **Applied Soft Computing**, [S.L.], v. 40, p. 292-304, mar. 2016. Elsevier BV. <https://doi.org/10.1016/j.asoc.2015.11.038>.
- GUHA, Sudipto; RASTOGI, Rajeev; SHIM, Kyuseok. Cure: an efficient clustering algorithm for large databases. **Information Systems**, [S.L.], v. 26, n. 1, p. 35-58, mar. 2001. Elsevier BV. [http://dx.doi.org/10.1016/s0306-4379\(01\)00008-4](http://dx.doi.org/10.1016/s0306-4379(01)00008-4).
- GUROBI. **Dealing with big-M constraints**, 2022. Disponível em: https://www.gurobi.com/documentation/9.5/refman/dealing_with_big_m_constra.html. Acesso em: 31 out. 2022.
- HARALICK, Robert M.; SHAPIRO, Linda G.. The Consistent Labeling Problem: Part I. **Ieee Transactions On Pattern Analysis And Machine Intelligence**, [S.L.], v. -1, n. 2, p. 173-184, abr. 1979. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/tpami.1979.4766903>.

HARALICK, Robert M.; SHAPIRO, Linda G.. The Consistent Labeling Problem: Part II. **Ieee Transactions On Pattern Analysis And Machine Intelligence**, [S.L.], v. -2, n. 3, p. 193-203, maio 1980. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/tpami.1980.4767007>.

HART, Peter; NILSSON, Nils; RAPHAEL, Bertram. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. **IEEE Transactions On Systems Science And Cybernetics**, [S.L.], v. 4, n. 2, p. 100-107, 1968. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/tssc.1968.300136>.

HARTIGAN, J. A.; WONG, M. A.. Algorithm AS 136: a k-means clustering algorithm. **Applied Statistics**, [S.L.], v. 28, n. 1, p. 100-108, 1979. JSTOR. <http://dx.doi.org/10.2307/2346830>.

HAYES, P. J.. Computation and deduction. In: **International Symposium on Mathematical Foundations of Computer Science**, 2., 1973, Strbské Pleso. **Proceedings [...]**. 1973, p. 105-118.

HEWITT, Carl. PLANNER: A language for proving theorems in robots. In: **International Joint Conference on Artificial Intelligence**, 1., 1969, Washington, D. C.. **Proceedings [...]**. 1969, p. 295-301.

HUBERT, Lawrence; ARABIE, Phipps. Comparing partitions. **Journal Of Classification**, [S.L.], v. 2, n. 1, p. 193-218, dez. 1985. Springer Science and Business Media LLC. <http://dx.doi.org/10.1007/bf01908075>.

JAFFAR, Joxan; LASSEZ, Jean-Louis. Constraint logic programming. **Proceedings Of The 14Th ACM Sigact-Sigplan Symposium On Principles Of Programming Languages - POPL '87**, [S.L.], p. 111-119, out. 1987. ACM Press. <http://dx.doi.org/10.1145/41625.41635>.

JAIN, A. K.; MURTY, M. Narasimha; FLYNN, P. J.. Data clustering: a review. **ACM Computing Surveys**, [S.L.], v. 31, n. 3, p. 264-323, set. 1999. Association for Computing Machinery (ACM). <http://dx.doi.org/10.1145/331499.331504>.

JÉLVEZ, Enrique; MORALES, Nelson; NANCEL-PENARD, Pierre; PEYPOUQUET, Juan; REYES, Patricio. Aggregation heuristic for the open-pit block scheduling problem. **European Journal Of Operational Research**, [S.L.], v. 249, n. 3, p. 1169-1177, mar. 2016. Elsevier BV. <http://dx.doi.org/10.1016/j.ejor.2015.10.044>.

JENSEN, Robert E.. A Dynamic Programming Algorithm for Cluster Analysis. **Operations Research**, [S.L.], v. 17, n. 6, p. 1034-1057, dez. 1969. Institute for Operations Research and the Management Sciences (INFORMS). <http://dx.doi.org/10.1287/opre.17.6.1034>.

- JOHNSON, Thys B.. Optimum production scheduling. In: International APCOM Symposium, 8., 1969, Salt Lake City. **Proceedings [...]**.
- KARMAKAR, Narendra. A new polynomial-time algorithm for linear programming. **Combinatorica**, [S.L.], v. 4, n. 4, p. 373-395, dez. 1984. Springer Science and Business Media LLC. <http://dx.doi.org/10.1007/bf02579150>.
- KING, Benjamin. Step-wise clustering procedures. **Journal of the American Statistical Association**, [S.L.], v. 62 n. 317, p. 86-101, 1967. Taylor & Francis. <http://dx.doi.org/10.1080/01621459.1967.10482890>.
- KLEER, Johan de; SUSSMAN, Gerald Jay. Propagation of constraints applied to circuit synthesis. **International Journal Of Circuit Theory And Applications**, [S.L.], v. 8, n. 2, p. 127-144, abr. 1980. Wiley. <http://dx.doi.org/10.1002/cta.4490080206>.
- KLINGMAN, Darwin; PHILLIPS, Nancy. Integer Programming for Optimal Phosphate-Mining Strategies. **The Journal of the Operational Research Society**, [S.L.], v. 39, n. 9, p. 805-810, 1988. Palgrave Macmillan Journals. <https://doi.org/10.2307/2583523>.
- KNUTH, Donald E.. **The Art of Computer Programming: Volume 1 - Fundamental Algorithms**. California: Addison-Wesley, 1973.
- KOHONEN, T.. The self-organizing map. **Proceedings Of The Ieee**, [S.L.], v. 78, n. 9, p. 1464-1480, 1990. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/5.58325>.
- KOMATSU. **PC200-8**. 2009. Disponível em: <https://appl.komatsu.com.br/pt/catalogo.php?categoria=219%7CEscavadeiras&catalogo=PC200-8&id=2537>. Acesso em: 14 nov. 2023.
- KOUSHAVAND, Behrang; ASKARI-NASAB, Hooman; DEUTSCH, Clayton 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**, [S.L.], v. 24, n. 4, p. 451-459, jul. 2014. Elsevier BV. <http://dx.doi.org/10.1016/j.ijmst.2014.05.006>.
- KOWALSKI, Robert A.. Predicate logic as a programming language. In: IFIP Congress, 6., 1974, Stockholm. **Proceedings [...]**. 1974, p. 569-574.
- KOWALSKI, Robert. A Proof Procedure Using Connection Graphs. **Journal Of The Acm**, [S.L.], v. 22, n. 4, p. 572-595, out. 1975. Association for Computing Machinery (ACM). <http://dx.doi.org/10.1145/321906.321919>.
- KRISHNA, K.; MURTY, M. Narasimha. Genetic K-means algorithm. **Ieee Transactions On Systems, Man And Cybernetics, Part B (Cybernetics)**, [S.L.], v. 29, n. 3, p. 433-439, jun.

1999. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/3477.764879>.

LAND, A. H.; DOIG, A. G.. An Automatic Method of Solving Discrete Programming Problems. **Econometrica**, [S.L.], v. 28, n. 3, p. 497, jul. 1960. JSTOR. <http://dx.doi.org/10.2307/1910129>.

LEHMER, Derrick Henry. Combinatorial problems with digital computers. In: Canadian Mathematical Congress, 4., 1957, Banff. **Proceedings [...]**. Toronto: University of Toronto Press, 1959. p. 160-173.

LERCHS, Helmut; GROSSMANN, Ingo F.. Optimum design of open pit mines. **CIM Bulletin**, v. 58, p. 17–24, 1965.

LOTFIAN, Reza; GHOLAMNEJAD, Javad; MIRZAEIAN LARDKEYVAN, Yousef. Effective solution of the long-term open pit production planning problem using block clustering. **Engineering Optimization**, [S.L.], v. 53, n. 7, p. 1119-1134, 2021. Informa UK Limited. <http://dx.doi.org/10.1080/0305215x.2020.1771703>.

MACQUEEN, J.. Some methods for classification and analysis of multivariate observations. In: Berkeley Symposium on Mathematical Statistics and Probability, 5., 1967, Berkeley. **Proceedings [...]**. 1967, p. 281-297.

MACKWORTH, Alan K.. Interpreting pictures of polyhedral scenes. **Artificial Intelligence**, [S.L.], v. 4, n. 2, p. 121-137, 1973. Elsevier BV. [http://dx.doi.org/10.1016/0004-3702\(73\)90003-9](http://dx.doi.org/10.1016/0004-3702(73)90003-9).

MACKWORTH, Alan K.. Consistency in networks of relations. **Artificial Intelligence**, [S.L.], v. 8, n. 1, p. 99-118, fev. 1977. Elsevier BV. [http://dx.doi.org/10.1016/0004-3702\(77\)90007-8](http://dx.doi.org/10.1016/0004-3702(77)90007-8).

MAI, Ngoc Luan; TOPAL, Erkan; ERTEN, Oktay. A new open-pit mine planning optimization method using block aggregation and integer programming. **Journal Of The Southern African Institute Of Mining And Metallurgy**, [S.L.], v. 118, n. 7, p. 705-714, 2018. Academy of Science of South Africa. <http://dx.doi.org/10.17159/2411-9717/2018/v118n7a4>.

MARIZ, Jorge Luiz Valença; SOOFASTAEI, Ali. Advanced Analytics for Surface Mine Planning. **Advanced Analytics In Mining Engineering**, [S.L.], p. 205-306, 2022. Springer International Publishing. http://dx.doi.org/10.1007/978-3-030-91589-6_9.

MEAGHER, C.; DIMITRAKOPOULOS, Roussos G.; AVIS, David. Optimized open pit mine design, pushbacks and the gap problem—a review. **Journal Of Mining Science**, [S.L.],

v. 50, n. 3, p. 508-526, maio 2014. Pleiades Publishing Ltd. <http://dx.doi.org/10.1134/s1062739114030132>.

MENABDE, Merab; FROYLAND, Gary; STONE, Peter; YEATES, Gavin. Mining schedule optimization for conditionally simulated orebodies. In: International Symposium on Orebody Modelling and Strategic Mine Planning, 2004, Perth. **Proceedings [...]** Melbourne: Australasian Institute of Mining and Metallurgy, 2004.

METZ, Horst J.; JAIN, Suresh K.. Optimal Mining and Processing Decisions in Stratiform Phosphate Deposits. **Interfaces**, [S.L.], v. 9, n. 1, p. 1-12, nov. 1978. Institute for Operations Research and the Management Sciences (INFORMS). <http://dx.doi.org/10.1287/inte.9.1.1>.

MIETTINEN, Kaisa; RUIZ, Francisco; WIERZBICKI, Andrzej P.. Introduction to Multiobjective Optimization: interactive approaches. **Multiobjective Optimization**, [S.L.], p. 27-57, 2008. Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-540-88908-3_2.

MONTANARI, Ugo. Networks of constraints: fundamental properties and applications to picture processing. **Information Sciences**, [S.L.], v. 7, p. 95-132, jan. 1974. Elsevier BV. [http://dx.doi.org/10.1016/0020-0255\(74\)90008-5](http://dx.doi.org/10.1016/0020-0255(74)90008-5).

MOORE, Edward F.. The Shortest Path through a Maze. In: International Symposium on the Theory of Switching, 1959, Cambridge. **Proceedings [...]**. Cambridge: Harvard University Press, 1959. p. 285-292.

MORALES, Nelson; JÉLVEZ, Enrique; NANCEL-PENARD, Pierre; MARINHO, Alexandre; GUIMARÃES, Octávio. A Comparison of Conventional and Direct Block Scheduling Methods for Open Pit Mine Production Scheduling. In: Application of Computers and Operations Research in the Mineral Industry, 37., 2015, Fairbanks. **Proceedings [...]** Littleton: Society for Mining, Metallurgy, and Exploration, 2015. p. 1040-1051.

MORALES, Nelson; SEGUEL, Sebastián; CÁCERES, Alejandro; JÉLVEZ, Enrique; ALARCÓN, Maximiliano. Incorporation of Geometallurgical Attributes and Geological Uncertainty into Long-Term Open-Pit Mine Planning. **Minerals**, [S.L.], v. 9, n. 2, p. 108, 13 fev. 2019. MDPI AG. <http://dx.doi.org/10.3390/min9020108>.

NELIS, Gonzalo; MORALES, Nelson. A mathematical model for the scheduling and definition of mining cuts in short-term mine planning. **Optimization And Engineering**, [S.L.], n. 23, p. 233-257, mar. 2022. Springer Science and Business Media LLC. <http://dx.doi.org/10.1007/s11081-020-09580-1>.

NELIS, Gonzalo; MEUNIER, Frédéric; MORALES, Nelson. Column Generation for Mining Cut Definition with Geometallurgical Interactions. **Natural Resources Research**, [S.L.], v.

31, n. 1, p. 131-148, 25 nov. 2021. Springer Science and Business Media LLC. <http://dx.doi.org/10.1007/s11053-021-09976-5>.

NEWMAN, Alexandra M.; KUCHTA, Mark. Using aggregation to optimize long-term production planning at an underground mine. **European Journal Of Operational Research**, [S.L.], v. 176, n. 2, p. 1205-1218, jan. 2007. Elsevier BV. <http://dx.doi.org/10.1016/j.ejor.2005.09.008>.

NEWMAN, Alexandra M.; RUBIO, Enrique; CARO, Rodrigo; WEINTRAUB, Andrés; EUREK, Kelly. A Review of Operations Research in Mine Planning. **Interfaces**, [S.L.], v. 40, n. 3, p. 222-245, jun. 2010. Institute for Operations Research and the Management Sciences (INFORMS). <http://dx.doi.org/10.1287/inte.1090.0492>.

NEZHADSHAHMOHAMMAD, Farshad; POURRAHIMIAN, Yashar; AGHABABAEI, Hamid. Presentation of a multi-index clustering technique for the mathematical programming of block-cave scheduling. **International Journal Of Mining Science And Technology**, [S.L.], v. 28, n. 6, p. 941-950, nov. 2018. Elsevier BV. <http://dx.doi.org/10.1016/j.ijmst.2017.11.005>.

NEZHADSHAHMOHAMMAD, Farshad; POURRAHIMIAN, Yashar. A Clustering Algorithm for Block-Cave Production Scheduling. **Global Journal Of Earth Science And Engineering**, [S.L.], v. 5, p. 45-53, 6 mar. 2019. Avanti Publishers. <http://dx.doi.org/10.15377/2409-5710.2018.05.4>.

NILSSON, Nils J.. **Problem-Solving Methods in Artificial Intelligence**. New York: McGraw-Hill, 1971.

OSANLOO, Morteza; GHOLAMNEJAD, Javad; KARIMI, Behrooz. Long-term open pit mine production planning: a review of models and algorithms. **International Journal Of Mining, Reclamation And Environment**, [S.L.], v. 22, n. 1, p. 3-35, mar. 2008. Informa UK Limited. <http://dx.doi.org/10.1080/17480930601118947>.

PERONI, Rodrigo de Lemos. **Análise da sensibilidade do sequenciamento de lavra em função da incerteza do modelo geológico**. 2002. 143 f. Tese (Doutorado) - Curso de Engenharia de Minas, Programa de Pós Graduação em Engenharia de Minas, Metalúrgica e de Materiais, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2002.

POURRAHIMIAN, Yashar; ASKARI-NASAB, Hooman; TANNANT, Dwayne. Mixed-Integer Linear Programming formulation for block-cave sequence optimisation. **International Journal Of Mining And Mineral Engineering**, [S.L.], v. 4, n. 1, p. 26, 2012. Inderscience Publishers. <http://dx.doi.org/10.1504/ijmme.2012.047998>.

- POURRAHIMIAN, Yashar; ASKARI-NASAB, Hooman; TANNANT, Dwayne D.. A multi-step approach for block-cave production scheduling optimization. **International Journal Of Mining Science And Technology**, [S.L.], v. 23, n. 5, p. 739-750, set. 2013. Elsevier BV. <http://dx.doi.org/10.1016/j.ijmst.2013.08.019>.
- RAMAZAN, Salih; DAGDELEN, Kadri. A new push back design algorithm in open pit mining. In: International Symposium on Mine Planning and Equipment Selection, 17., 1998, Calgary. **Proceedings [...]**. Boca Raton: CRC Press, 1998. p. 119-124.
- RAMAZAN, Salih. **Open Pit Mine Scheduling Based on Fundamental Tree Algorithm**. 2001. 181 f. Tese (Doutorado) – Mining And Earth System Engineering, Colorado School Of Mines, Golden, 2001.
- RAMAZAN, Salih; DAGDELEN, Kadri; JOHNSON, Thys B.. Fundamental tree algorithm in optimising production scheduling for open pit mine design. **Mining Technology**, [S.L.], v. 114, n. 1, p. 45-54, mar. 2005. Informa UK Limited. <http://dx.doi.org/10.1179/037178405x44511>.
- RAMAZAN, Salih. The new Fundamental Tree Algorithm for production scheduling of open pit mines. **European Journal Of Operational Research**, [S.L.], v. 177, n. 2, p. 1153-1166, mar. 2007. Elsevier BV. <http://dx.doi.org/10.1016/j.ejor.2005.12.035>.
- RAND, William M.. Objective Criteria for the Evaluation of Clustering Methods. **Journal Of The American Statistical Association**, [S.L.], v. 66, n. 336, p. 846-850, dez. 1971. Informa UK Limited. <http://dx.doi.org/10.1080/01621459.1971.10482356>.
- ROSENBERG, Andrew; HIRSCHBERG, Julia. V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. In: Joint Conference On Empirical Methods In Natural Language Processing And Computational Natural Language Learning (EMNLP-CONLL), 2007. 2007, Prague. **Proceedings [...]**. Pennsylvania: Association For Computational Linguistics, 2007. p. 410-420.
- ROSSI, Francesca; VAN BEEK, Peter; WALSH, Toby. Introduction. In: ROSSI, Francesca; VAN BEEK, Peter; WALSH, Toby (Eds.) **Handbook of Constraint Programming**. Amsterdam: Elsevier, p. 3-12, 2006.
- ROUSSEEUW, Peter J.. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. **Journal Of Computational And Applied Mathematics**, [S.L.], v. 20, p. 53-65, nov. 1987. Elsevier BV. [http://dx.doi.org/10.1016/0377-0427\(87\)90125-7](http://dx.doi.org/10.1016/0377-0427(87)90125-7).
- SALMAN, Saad; MUHAMMAD, Khan; KHAN, Asif; GLASS, Hylke J.. A Block Aggregation Method for Short-Term Planning of Open Pit Mining with Multiple Processing

Destinations. **Minerals**, [S.L.], v. 11, n. 3, p. 288, 10 mar. 2021. MDPI AG. <http://dx.doi.org/10.3390/min11030288>.

SAMANTA, Biswajit; BHATTACHERJEE, Ashis; GANGULI, Rajive. A genetic algorithms approach for grade control planning in a bauxite deposit. **Application Of Computers And Operations Research In The Mineral Industry**, [S.L.], p. 337-342, 15 mar. 2005. Taylor & Francis. <http://dx.doi.org/10.1201/9781439833407.ch44>.

SARI, Yuksel Asli; KUMRAL, Mustafa. Dig-limits optimization through mixed-integer linear programming in open-pit mines. **Journal Of The Operational Research Society**, [S.L.], v. 69, n. 2, p. 171-182, 2018. Informa UK Limited. <http://dx.doi.org/10.1057/s41274-017-0201-z>

SCHÖLKOPF, Bernhard; SMOLA, Alexander; MÜLLER, Klaus-Robert. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. **Neural Computation**, [S.L.], v. 10, n. 5, p. 1299-1319, 1 jul. 1998. MIT Press - Journals. <http://dx.doi.org/10.1162/089976698300017467>.

SCIKITLEARN. **2.3. Clustering**. 2022. Disponível em: <https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation>. Acesso em: 18 jun. 2022.

SELIM, Shokri Z.: A global algorithm for the clustering problem. In: 1982 ORSA/TIMS Joint Meeting, 1982, San Diego. **Proceedings [...]**.

SEYMOUR, F.. **Pit limit parametrization from modified 3D Lerchs-Grossmann algorithm**. Society of Mining, Metallurgy and Exploration. Manuscript, 1994.

SMITH, Martin L.. Optimizing inventory stockpiles and mine production: An application of separable and goal programming to phosphate mining using AMPL/CPLEX. **CIM Bulletin**, v. 92, n. 1030, p. 61-64, 1999.

SMITH, Martin L.; WICKS, Stewart J.. Medium-Term Production Scheduling of the Lumwana Mining Complex. **Interfaces**, [S.L.], v. 44, n. 2, p. 176-194, abr. 2014. Institute for Operations Research and the Management Sciences (INFORMS). <http://dx.doi.org/10.1287/inte.2014.0737>.

SNEATH, P. H. A.; SOKAL, R. R.. **Numerical taxonomy: the principles and practice of numerical classification**. San Francisco: WF Freeman & Co., 1973.

STONE, P.; FROYLAND, Gary; MENABDE, Merab; LAW, B.; PASYAR, R.; MONKHOUSE, Peter. Blaser-blended iron-ore mine planning optimization at Yandi, Western Australia. In: International Symposium on Orebody Modelling and Strategic Mine

Planning, 2004, Perth. **Proceedings [...]** Melbourne: Australasian Institute of Mining and Metallurgy, 2004.

STREHL, Alexander; GOSH, Joydeep. Cluster Ensembles – A Knowledge Reuse Framework for Combining Multiple Partitions. **Journal Of Machine Learning Research**, [S.L.], v. 1, n. 3, p. 583-617, dez. 2002. Test accounts. <http://dx.doi.org/10.1162/153244303321897735>.

SUSSMAN, Gerald; WINOGRAD, Terry. **Micro-Planner Reference Manual**. Cambridge: Massachusetts Institute of Technology, 1970.

SUSSMAN, Gerald Jay; STEELE, Guy Lewis. Constraints—A language for expressing almost-hierarchical descriptions. *Artificial Intelligence*, [S.L.], v. 14, n. 1, p. 1-39, ago. 1980. Elsevier BV. [http://dx.doi.org/10.1016/0004-3702\(80\)90032-6](http://dx.doi.org/10.1016/0004-3702(80)90032-6).

TABESH, Mohammad; ASKARI-NASAB, Hooman. Two-stage clustering algorithm for block aggregation in open pit mines. **Mining Technology**, [S.L.], v. 120, n. 3, p. 158-169, set. 2011. Informa UK Limited. <http://dx.doi.org/10.1179/1743286311y.0000000009>.

TABESH, Mohammad; ASKARI-NASAB, Hooman. Automatic creation of mining polygons using hierarchical clustering techniques. **Journal Of Mining Science**, [S.L.], v. 49, n. 3, p. 426-440, maio 2013. Pleiades Publishing Ltd. <http://dx.doi.org/10.1134/s1062739149030106>.

TABESH, Mohammad; MIETH, Clemens; ASKARI-NASAB, Hooman. 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, Mohammad; MIETH, Clemens; NASAB, Hooman Askari. A multi-step approach to long-term open-pit production planning. **International Journal Of Mining And Mineral Engineering**, [S.L.], v. 5, n. 4, p. 273-298, 2014. Inderscience Publishers. <http://dx.doi.org/10.1504/ijmme.2014.066577>.

TABESH, Mohammad; ASKARI-NASAB, Hooman; PERONI, Rodrigo. A comprehensive approach to strategic open pit mine planning with stockpile consideration. In: Application of Computers and Operations Research in the Mineral Industry, 37., 2015, Fairbanks. **Proceedings [...]** Littleton: Society for Mining, Metallurgy, and Exploration, 2015. p. 326-332.

TABESH, Mohammad; ASKARI-NASAB, Hooman. Clustering mining blocks in presence of geological uncertainty. **Mining Technology**, [S.L.], v. 128, n. 3, p. 162-176, 27 mar. 2019. Informa UK Limited. <http://dx.doi.org/10.1080/25726668.2019.1596425>.

TRAN, Thanh N.; WEHRENS, Ron; BUYDENS, Lutgarde M. C.. KNN-kernel density-based clustering for high-dimensional multivariate data. **Computational Statistics & Data**

Analysis, [S.L.], v. 51, n. 2, p. 513-525, nov. 2006. Elsevier BV. <http://dx.doi.org/10.1016/j.csda.2005.10.001>.

UPADHYAY, Shiv Prakash; ASKARI-NASAB, Hooman; TABESH, Mohammad; BADIOZAMANI, Mohammad Mahdi. Simulation and optimization in open pit mining. In: 2015 SME Annual Conference, 2015, Denver. **Proceedings [...]**. Denver: Society for Mining, Metallurgy and Exploration Inc., 2015a. Preprint 15-051, p. 1-3.

UPADHYAY, Shiv Prakash; ASKARI-NASAB, Hooman; TABESH, Mohammad; BADIOZAMANI, Mohammad Mahdi. Simulation and optimization in open pit mining. In: BANDOPADHYAY, Sukumar (Eds.) *Application of Computers and Operations Research in the Mineral Industry*, 37., 2015, Fairbanks. **Proceedings [...]**. Englewood: Society for Mining, Metallurgy and Exploration Inc., 2015b. p. 532-543.

UPADHYAY, Shiv Prakash; ASKARI-NASAB, Hooman. Truck-shovel allocation optimisation: a goal programming approach. **Mining Technology**, v. 125, n. 2, p. 82-92, 2016. <https://doi.org/10.1179/1743286315Y.0000000024>.

UPADHYAY, Shiv Prakash; ASKARI-NASAB, Hooman. Simulation and optimization approach for uncertainty-based short-term planning in open pit mines. **International Journal Of Mining Science And Technology**, [S.L.], v. 28, n. 2, p. 153-166, mar. 2018. Elsevier BV. <http://dx.doi.org/10.1016/j.ijmst.2017.12.003>.

UPADHYAY, Shiv Prakash; ASKARI-NASAB, Hooman. Dynamic shovel allocation approach to short-term production planning in open-pit mines. **International Journal of Mining, Reclamation and Environment**, v. 33, n. 1, p. 1–20, 2019. doi:10.1080/17480930.2017.1315524.

VAN HOEVE, Willem-Jan; KATRIEL, Irit. Global Constraints. In: ROSSI, Francesca; VAN BEEK, Peter; WALSH, Toby (Eds.) *Handbook of Constraint Programming*. Amsterdam: Elsevier, p. 169-208, 2006.

WALTZ, D.. Understanding line drawings of scenes with shadows. In: Winston, P. H. (Eds.) **The Psychology of Computer Vision**. New York: McGraw-Hill, p. 19-91, 1975.

WANG, Zhiyuan; RANGAIAH, Gade Pandu. Application and Analysis of Methods for Selecting an Optimal Solution from the Pareto-Optimal Front obtained by Multiobjective Optimization. **Industrial & Engineering Chemistry Research**, [S.L.], v. 56, n. 2, p. 560-574, 5 jan. 2017. American Chemical Society (ACS). <http://dx.doi.org/10.1021/acs.iecr.6b03453>.

- WARD JR, Joe H.. Hierarchical Grouping to Optimize an Objective Function. **Journal of the American Statistical Association**, [S.L.], v. 58 n. 301, p. 236-244, 1963. Taylor & Francis. <http://dx.doi.org/10.1080/01621459.1963.10500845>.
- WANG, Q.; SEVIM, H.. (1995) Alternative to parameterization in finding a series of maximum metal pits for production planning. **Mining Engineering**, p. 178-182, 1995.
- WEINTRAUB, Andres; PEREIRA, Marianela; SCHULTZ, Ximena. A Priori and A Posteriori Aggregation Procedures to Reduce Model Size in MIP Mine Planning Models. **Electronic Notes In Discrete Mathematics**, [S.L.], v. 30, p. 297-302, fev. 2008. Elsevier BV. <http://dx.doi.org/10.1016/j.endm.2008.01.051>.
- WILKES, Maurice V.. Constraint-type statements in programming languages. **Communications Of The Acm**, [S.L.], v. 7, n. 10, p. 587-588, out. 1964. Association for Computing Machinery (ACM). <http://dx.doi.org/10.1145/364888.364967>.
- WHITTLE, Jeff. Beyond optimization in open pit design. Whittle Programming. In: FYTAS, Kosta; COLLINS, J. L.; SINGHAL, R. K. (Eds.) **Computer Applications in the Mineral Industry: First Canadian Conference**. Boca Raton: CRC Press, p. 331-337, 1988.
- YAVARZADEH, Saeid; ABODALLHEISHARIF, Jafar; NEISHABOURI, Arash. Modeling of Short Term Production Scheduling with the Objective Grade Control. **Mine Planning And Equipment Selection**, [S.L.], p. 379-387, 2014. Springer International Publishing. http://dx.doi.org/10.1007/978-3-319-02678-7_37.
- ZHANG, Tian; RAMAKRISHNAN, Raghu; LIVNY, Miron. BIRCH: an efficient data clustering method for very large databases. **Acm Sigmod Record**, [S.L.], v. 25, n. 2, p. 103-114, jun. 1996. Association for Computing Machinery (ACM). <http://dx.doi.org/10.1145/235968.233324>.
- ZHANG, M.. Combining genetic algorithms and topological sort to optimize open-pit mine plans. In: In: International Conference on Mine Planning and Equipment Selection (MPES), 15., 2006, Torino. **Proceedings [...]**. Torino: Fiordo Srl, 2006. p. 1234-1239.

APÊNDICE I – CÓDIGO EM *PYTHON* DO MODELO DO AGRUPAMENTO DE POLÍGONOS DE LAVRA SOLUCIONADO POR PROGRAMAÇÃO POR RESTRIÇÕES NO *OR-TOOLS*

```

#Dataset#
df = pd.DataFrame(Dataset)
#Parâmetros do modelo matemático determinados pelo usuário#
min_size_cluster, max_size_cluster, min_n_clusters, max_n_clusters,
Max_Diameter
#Concluntos e matrizes calculados para cada bloco do dataset#
adj_4, adj_8, Dist, S

tic = time.time()
model = cp_model.CpModel()
clusters = []
for i in range(max_n_clusters):
    clusters.append(i)

#Criação de uma Função Objetivo vazia, que irá maximizar a similaridade
dos blocos dentro de cada cluster#
objective_terms = []

#Criação de uma matriz para atribuir cada bloco a um cluster#
M = []
for i in range(len(df)):
    t = []
    for j in clusters:
        t.append(model.NewIntVar(0, 1, f'M[{i},{j}]'))
    M.append(t)

#Criação das variáveis y[j], representando os o número de blocos em cad
a cluster#
y = np.zeros(len(clusters)).tolist()
#Atribuo os resultados da variável y[j] à soma de cada linha da matriz
M(i,j)#
for j in clusters:
    y[j] = model.NewIntVar(0, max_size_cluster, 'y[{}]'.format(j))
    model.Add(y[j] == sum([M[i][j] for i in range(len(df))]))

#Criação das variáveis x[i], representando os clusters atribuídos a cad
a bloco#
x = np.zeros(len(df)).tolist()
for i in range(len(df)):
    x[i] = model.NewIntVar(0, max_size_cluster-1, 'x[{}]'.format(j))
    model.Add(x[i] == sum([M[i][j]*j for j in clusters]))

#Constraints 1 - Cada bloco pode e deve receber somente um cluster#

```

```

for i in range(len(df)):
    model.Add(sum([M[i][j] for j in clusters]) == 1)

#Constraints 2 -
    Determino o tamanho mínimo dos clusters que são utilizados, e forço os
    desnecessários a serem 0#
for i in clusters:
    z = model.NewBoolVar(f'z_{j}')
    model.Add(y[j] >= min_size_cluster).OnlyEnforceIf(z)
    model.Add(y[j] == 0).OnlyEnforceIf(z.Not())
    model.Add(y[j] <= max_size_cluster).OnlyEnforceIf(z)

#Constraints 3: Determino que um bloco só pode pertencer a um cluster c
    aso ao menos dois blocos numa adjacência de 8 blocos também pertençam a
    este mesmo cluster#
#Constraints 4: Determino que um bloco só pode pertencer a um cluster c
    aso ao menos um bloco numa adjacência de 4 blocos também pertença a est
    e mesmo cluster
#Constraints 5: Inserção da restrição que limita o diâmetro máximo de c
    ada cluster a 3#
for j in clusters:
    for i in range(len(df)):
        adj_bool = model.NewBoolVar(f'constr_{i}_{j}')
        model.Add(M[i][j] == 1).OnlyEnforceIf(adj_bool)
        model.Add(M[i][j] == 0).OnlyEnforceIf(adj_bool.Not())
        model.Add(sum([M[k][j] for k in range(len(df)) if df.iloc[k]['id']
in adj_4[df.loc[i][0]]]) >= 1).OnlyEnforceIf(adj_bool)
        model.Add(sum([M[k][j] for k in range(len(df)) if df.iloc[k]['id']
in adj_8[df.loc[i][0]]]) >= 2).OnlyEnforceIf(adj_bool)
#Abordando um loop em k como uma list comprehension, determino o diâmet
ro máximo e adiciono recorrentemente a lista objective terms#
        [model.Add(Dist[i][k]*M[k][j] <= Max_Diameter).OnlyEnforceIf(adj_bo
ol) for k in range(len(df)) if i != k
        for k in range(len(df)):
            adj_bool2 = model.NewBoolVar(f'constr_{k}_{j}')
            model.Add(M[k][j] == 1).OnlyEnforceIf(adj_bool2)
            model.Add(M[k][j] == 0).OnlyEnforceIf(adj_bool2.Not())
            if i != k:
                p = model.NewBoolVar(f'p_{i}_{k}_{j}')
                model.AddBoolOr(adj_bool.Not(), adj_bool2.Not(), p)
                model.AddImplication(p, adj_bool)
                model.AddImplication(p, adj_bool2)
                objective_terms.append(S[i][k]/2 *p)
model.Maximize(sum(objective_terms))

solver = cp_model.CpSolver()
solution_collector = VarArraySolutionCollector([x[i] for i in range(len
(df))])
status = solver.Solve(model, solution_collector)

```

```
solution_collector.solution_list
print('Status =', solver.StatusName(status))
print('FO =', solver.ObjectiveValue())
toc = time.time()
secl = toc - tic
print(secl)
```

APÊNDICE II – CÓDIGO EM *PYTHON* DO MODELO DO AGRUPAMENTO DE POLÍGONOS DE LAVRA SOLUCIONADO POR PROGRAMAÇÃO INTEIRA MISTA NO *OR-TOOLS*

```

#Dataset#
df = pd.DataFrame(Dataset)
#Parâmetros do modelo matemático determinados pelo usuário#
min_size_cluster, max_size_cluster, min_n_clusters, max_n_clusters,
Max_Diameter
#Concluntos e matrizes calculados para cada bloco do dataset#
adj_4, adj_8, Dist, S

tic = time.time()
model = cp_model.CpModel()
clusters = []
for i in range(max_n_clusters):
    clusters.append(i)
solver = pywraplp.Solver.CreateSolver('SCIP')

#Criação de uma Função Objetivo vazia, que irá maximizar a similaridade
dos blocos dentro de cada cluster#
objective_terms = []

#Criação de uma matriz para atribuir cada bloco a um cluster#
M = []
for i in range(len(df)):
    t = []
    for j in clusters:
        t.append(solver.IntVar(0, 1, f'M[{i},{j}]'))
    M.append(t)

#Criação das variáveis y[j], representando os o número de blocos em cad
a cluster#
y,z = np.zeros(len(clusters)).tolist(), np.zeros(len(clusters)).tolist(
)

#Atribuo os resultados da variável y[j] à soma de cada linha da matriz
M(i,j)#
for j in clusters:
    y[j] = solver.IntVar(0, max_size_cluster, 'y[{}]'.format(j))
    z[j] = solver.BoolVar('z[{}]'.format(j))
    solver.Add(solver.Sum([M[i][j] for i in range(len(ll_df))]) == y[j] )

#Criação das variáveis x[i], representando os clusters atribuídos a cad
a bloco#
x = np.zeros(len(ll_df)).tolist()

```

```

for i in range(len(ll_df)):
    x[i] = solver.IntVar(0, max_size_cluster-1, 'x[{}]'
                        .format(j))
    solver.Add(solver.Sum([M[i][j] * j for j in clusters]) == x[i])

#Constraints 1 - Cada bloco pode e deve receber somente um cluster#
for i in range(len(df)):
    solver.Add(solver.Sum([M[i][j] for j in clusters]) == 1)

#Constraints 2 -
    Determino o tamanho mínimo dos clusters que são utilizados, e forço os
    desnecessários a serem 0#
BigM = 1e5
for j in clusters:
    solver.Add(y[j] >= 1 - (BigM * (1 - z[j])))
    solver.Add(y[j] <= BigM * z[j])
    solver.Add(y[j] >= min_size_cluster * z[j])
    solver.Add(y[j] <= max_size_cluster * z[j])

#Constraints 3: Determino que um bloco só pode pertencer a um cluster c
    aso ao menos dois blocos numa adjacência de 8 blocos também pertençam a
    este mesmo cluster#
#Constraints 4: Determino que um bloco só pode pertencer a um cluster c
    aso ao menos um bloco numa adjacência de 4 blocos também pertença a est
    e mesmo cluster
#Constraints 5: Inserção da restrição que limita o diâmetro máximo de c
    ada cluster a 3#
for j in clusters:
    for i in range(len(df)):
        solver.Add(solver.Sum([M[k][j] for k in range(len(df)) if df.iloc[k]
                                ['id'] in adj_4[df.loc[i][0]]]) >= 1 * M[i][j])
        solver.Add(solver.Sum([M[k][j] for k in range(len(df)) if df.iloc[k]
                                ['id'] in adj_8[df.loc[i][0]]]) >= 2 * M[i][j])
#Abordando um loop em k como uma list comprehension, determino o diâmet
    ro máximo e adiciono recorrentemente a lista objective terms#
    [solver.Add(Dist[i][k]*M[k][j] <= Max_Diameter + (BigM * (2 -
    M[i][j] - M[k][j])) for k in range(len(ll_df)) if i != k]
    for k in range(len(df)):
        if i != k:
            p = solver.IntVar(0, 1, f'p_{i}_{k}_{j}')
            solver.Add(p <= M[i][j])
            solver.Add(p <= M[k][j])
            solver.Add(p >= M[i][j] + M[k][j] - 1)
            objective_terms.append(S[i][k]/2 * p)
solver.Maximize(solver.Sum(objective_terms))
status = solver.Solve()

if status == pywraplp.Solver.OPTIMAL or status == pywraplp.Solver.FEASIBLE:
    print('FO =', solver.Objective().Value())

```

```
#print('X =', [x[i].solution_value() for i in range(len(ll_df))])
print('Problem solved in ', solver.wall_time(), ' milliseconds')
print('Problem solved in ', solver.iterations(), ' iterations')
else:
    print('The problem does not have an optimal solution.')

toc = time.time()
sec1 = toc - tic
print('Time =',sec1)
```